



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Doctorado en Ingeniería

"Desarrollo de un sistema *Hardware In-the-Loop* para aplicación de técnicas de control en un Robot"

Tesis

Que como parte de los requisitos para obtener el grado de Doctor en ingeniería

Presenta:

**M. en C. Edson Eduardo Cruz Miguel**

Dirigido por:

**Dr. Juvenal Rodríguez Resendíz**

SINODALES

Dr. Juvenal Rodríguez Resendíz

Presidente

Firma

Dr. Roberto Valentín Carrillo Serrano

Secretario

Firma

Dr. Gerardo Israel Pérez Soto

Vocal

Firma

Dra. Karla Anhel Camarillo Gómez

Suplente

Firma

Dr. Fortino Mendoza Mondragón

Suplente

Firma

Centro Universitario Querétaro, QRO

Octubre 2020

México.

## *AGRADECIMIENTOS*

A Dios, por permitirme vivir plenamente cada día y realizar mis metas.

A mi esposa Leslie y mi hijo Edson, por su paciencia, por su comprensión, por apoyarme en todo momento y ser mi inspiración.

A mis padres, Ma. Luisa y Candelario, por su apoyo, por enseñarme a seguir mis sueños, por su amor incondicional y por ser un ejemplo a seguir.

A mis hermanas y sobrinos, por apoyarme en todas las decisiones, por estar siempre a mi lado.

A mi director de tesis, Dr. Juvenal, por sus consejos, gran apoyo y darme la oportunidad de formar parte de su equipo de trabajo.

A mi comité doctoral, Dr. Roberto Valentín, Dr. Gerardo Israel, Dra. Karla Anhel y Dr. Fortino, por sus comentarios acertados, por guiarme con sus conocimientos en este camino y sobre todo por la oportunidad de trabajar con ustedes.

A mis compañeros y amigos, Wilfrido Paredes y José Manuel, por compartir su experiencia, conocimiento y amistad.

# Reconocimientos

A la UAQ por brindarme la oportunidad de realizar mis estudios de doctorado en esta institución.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por proveer los fondos para realizar esta investigación.

Dirección General de Bibliotecas UAQ

# Resumen

El sistema HIL (Hardware In-the-Loop, por sus siglas en ingles) es una simulación en tiempo real que emula el comportamiento de un sistema físico. Éste sistema representa una herramienta de bajo costo, con un entorno seguro e ideal para la validación de técnicas de control en el desarrollo de prototipos y con fines de investigación científica. En la presente investigación se desarrolló un sistema HIL para la simulación en tiempo real de un robot manipulador de dos grados de libertad. En el modelo dinámico del robot manipulador se consideró la dinámica del actuador. Para aplicar diferentes técnicas de control para el seguimiento de trayectorias, se desarrolló un controlador de arquitectura abierta. De acuerdo a los requisitos del sistema de control, el sistema principal de procesamiento se implementó en un FPGA y una interfaz gráfica de usuario con base en python. Se utilizó herramientas libres desde la implementación del controlador (FPGA open-source) hasta el desarrollo de la interfaz gráfica de usuario. Para la sintonización del lazo de control principal se desarrolló un algoritmo genético que tiene como objetivo minimizar los esfuerzos de control y un error en estado estacionario cero en un tiempo acotado.

(**Palabras Clave:** HIL, FPGA, Robot manipulador, Algoritmo Genético, Control de movimiento, Open-source.)

# Abstract

The Hardware In-the-Loop system is a real-time simulation that emulates the behavior of a physical system. This system represents a low cost tool, with a safe and ideal environment for the validation of control techniques in the development of prototypes and for scientific research purposes. In the present investigation, a HIL system was developed for the real-time simulation of a manipulator robot with two degrees of freedom. In the dynamic model of the manipulator robot, the dynamics of the actuator was considered. In order to apply different control techniques for tracing traces, an open architecture controller was developed. According to the requirements of the control system, the main processing system was implemented in an FPGA and a python-based graphical user interface. Free tools were used from the implementation of the controller (open-source FPGA) to the development of the graphical user interface. A Genetic Algorithm for the tuning of the main control loop, was developed that aims to minimize control efforts and a zero steady-state error in a limited time.

# Índice general

Reconocimientos

Resumen

Abstract

Índice

Índice de Figuras III

Índice de Tablas v

Índice de Algoritmos v

**1. INTRODUCCIÓN 1**

1.1. Descripción del problema . . . . . 5

1.2. Justificación . . . . . 6

1.3. Estructura del documento . . . . . 7

**2. ANTECEDENTES 8**

2.1. Revisión literaria . . . . . 8

2.2. Modelo matemático del PMSM . . . . . 11

2.3. Robot Industrial . . . . . 12

2.3.1. Modelo cinemático . . . . . 13

2.3.2. Cinemática diferencial y singularidades . . . . . 15

2.3.3. Modelo dinámico . . . . . 16

2.4. Perfiles de movimiento . . . . . 19

Dirección General de Bibliotecas UAQ

2.5. CONTROL PID . . . . .	22
2.6. Lazo de control interno de corriente PI . . . . .	23
2.7. Sintonización . . . . .	23
<b>3. HIPÓTESIS</b>	<b>27</b>
<b>4. OBJETIVOS</b>	<b>28</b>
4.1. Objetivo General. . . . .	28
4.2. Objetivos Específicos. . . . .	28
<b>5. METODOLOGÍA</b>	<b>29</b>
5.1. Modelo del robot con actuadores . . . . .	29
5.2. Descripción Hardware . . . . .	30
<b>6. RESULTADOS</b>	<b>38</b>
<b>7. CONCLUSIONES</b>	<b>51</b>
<b>Referencias</b>	<b>58</b>
<b>8. ANEXOS PRODUCTOS DEL TRABAJO DE INVESTIGACIÓN</b>	<b>59</b>
A. Artículo indizado . . . . .	60
B. Artículo indizado . . . . .	62
C. Artículo indizado, Co-Autor . . . . .	64
D. Artículo indizado, Co-Autor . . . . .	66
E. Artículo revista arbitrada . . . . .	68
F. Constancia ponencia . . . . .	70
G. Artículo revista arbitrada . . . . .	71
H. Constancia poster . . . . .	73
I. Artículo revista arbitrada . . . . .	74

J.	Constancia seminario doctoral . . . . .	75
K.	Formación de recurso humano . . . . .	76
L.	Formación de recurso humano . . . . .	78
M.	Formación de recurso humano . . . . .	80
N.	Formación académica . . . . .	82
Ñ.	Formación académica . . . . .	83
O.	Formación académica . . . . .	84

Dirección General de Bibliotecas UAG



# Índice de Figuras

2.1. Esquema cinemático del robot. . . . .	14
2.2. Perfil de velocidad trapezoidal. . . . .	21
2.3. Diagrama de flujo del AG. . . . .	26
5.1. Flujo de diseño en IceStorm. . . . .	32
5.2. Módulos implementados en FPGA. . . . .	35
5.3. Controlador PID implementado en FPGA con base en módulos DSP de ICE40 . . . . .	36
5.4. Metodología para control y sintonía del robot. . . . .	37
6.1. Respuesta de cada eslabón de HIL aplicando el AG para la sintonización y perfil de velocidad (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente de la fase q eslabón 1, (d) corriente de la fase q eslabón 2, (e) corriente de la fase d eslabón 1 (f) corriente de la fase d eslabón 2. . . . .	40
6.2. Respuesta del HIL aplicando el AG para la sintonización y perfil de velocidad (a) par aplicado a cada eslabón, (b) velocidad medida en cada eslabón. . . . .	41
6.3. Respuesta de función objetivo para cada eslabón de HIL, (a) IAE, (b) ISE, (c) TV, (d) Función objetivo. . . . .	41

6.4. Análisis estadístico del AG propuestola distribución (a) distribución del número de iteraciones, (b) análisis de solución no factible (NF), factible (F) y solución factible y adecuada (F&S), (c) desviación estándar estimada del objetivo alcanzado, (d) valores obtenidos de la función objetivo, (e) cambio de la función objetivo, (f) cambio de la restricción de Error Absoluto Integral (IAE) y (g) cambio de la restricción de TV. . . . .	43
6.5. Banco de pruebas utilizado para aplicación de técnicas de control.	46
6.6. Respuesta del robot sin aplicar el AG para la sintonización (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente del actuador 1 (d) corriente del actuador 2 (e) trayectoria realizada. . . . .	47
6.7. Respuesta del robot aplicando el AG para la sintonización (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente del actuador 1 (d) corriente del actuador 2 (e) trayectoria realizada. . . . .	48
6.8. Vista principal de la GUI desarrollada en PYTHON. . . . .	49
6.9. Tarjeta electrónica desarrollada para acondicionamiento de señales.	50

# Índice de Tablas

2.1. Estado del arte de simulación HIL. . . . .	11
2.2. Antecedentes de controladores de arquitectura abierta . . . . .	11
5.1. Parámetros mecánicos del robot y eléctricos de los actuadores. . .	31
6.1. Revisión literaria de recursos lógicos y dispositivos FPGA emplean- dos en controladores similares al propuesto. . . . .	49

Dirección General de Bibliotecas UAO

# Índice de Algoritmos

1. Pseudocódigo del AG. . . . .	38
---------------------------------	----

Dirección General de Bibliotecas UAO

# INTRODUCCIÓN

Un robot manipulador es un sistema mecánico complejo cuya descripción analítica requiere ecuaciones diferenciales para explicar todos los fenómenos físicos que se encuentran en su estructura mecánica, tales como efectos inerciales, fuerzas centrípetas y de coriolis, par gravitacional y fricción. La utilidad del modelo dinámico de robots manipuladores es fundamental para propósitos de simulación, diseño y construcción del sistema mecánico, así como análisis y diseño de algoritmos de control (Reyes, 2011).

Un robot manipulador, de acuerdo con Sciavicco and Siciliano (2000), está formado por:

1. Una estructura mecánica, que generalmente consiste en una serie de cuerpos rígidos (eslabones) conectados entre sí por medio de articulaciones; el manipulador se divide en tres partes: un brazo que permite movilidad, una muñeca que le proporciona destreza y un órgano terminal que ejecuta la tarea encomendada al robot.
2. Actuadores, que permiten el movimiento del manipulador a través de sus articulaciones; usualmente, los actuadores empleados son motores eléctricos de corriente directa o motores síncronos de imán permanente (PMSM's, por sus siglas en inglés), hidráulicos y neumáticos.

3. Sensores, que miden el estado del robot.
4. Un sistema de control que permite el control y supervisión del movimiento del manipulador.

Por otro lado, uno de los aspectos más importantes en la robótica es el diseño de trayectorias o perfiles (movimiento, velocidad y aceleración), cuyo objetivo es incrementar la precisión de la trayectoria y disminuir el desgaste en las articulaciones de los robots, con lo cual se puede obtener una mejor operación que se ve reflejada en el consumo energético y a su vez en la calidad del producto o proceso de operación (Martínez-Prado et al., 2018; Shu et al., 2018).

Debido al aumento de los costos laborales en los países en desarrollo, existe una demanda creciente de automatización en los campos industriales. Basado en este hecho, unos récords de 248 mil robots se enviaron en todo el mundo en 2015 y el número total de robots en operación en el mundo fue de 1.63 millones de unidades de acuerdo a la Federación Internacional de Robótica, 2017. Por lo tanto, la demanda de controladores de movimiento ha aumentado en aplicaciones industriales, tales como manipuladores de robots, máquinas de control numérico computarizado (CNC) y convertidores de frecuencia, (Martínez-Prado et al., 2018; Morales-Velazquez et al., 2010). En este contexto, los investigadores se han esforzado por mejorar el rendimiento de estos sistemas de control. Sin embargo, la mayoría de los sistemas de control presentan limitaciones para modificar o reemplazar algoritmos de control (Jokić et al., 2020). Estas limitaciones son presentadas por la arquitectura del sistema de control que se puede clasificar en: (1) arquitectura cerrada, donde no es posible acceder al algoritmo de control y protocolos de comunicación, (2) arquitectura híbrida, donde las estrategias de control no se pueden modificar, pero es posible agregar dispositivos

al sistema, y (3) la arquitectura abierta, donde tanto el hardware como el software y las estrategias de control se pueden modificar (Pritschow et al., 2001; Morales-Velazquez et al., 2010). A diferencia del sistema mecánico, las etapas eléctricas y de software se vuelven obsoletas en menos tiempo, lo que representa un área de oportunidad en los servosistemas (Jokić et al., 2020). Los Field Programmable Gate Arrays (FPGA) son dispositivos con una gran cantidad de elementos lógicos programables, como compuertas, flip-flops y cables, y utilizan lenguaje de descripción de hardware (HDL). Tanto la portabilidad de los núcleos desarrollados entre familias y empresas FPGA como la disponibilidad de estos en el mercado han hecho que el uso de FPGAs proliferen en diferentes áreas de la industria, aeronáutica, robótica y automatización (Ordóñez Cerezo et al., 2019; Brant and Lemieux, 2012; Romanov et al., 2017; Monmasson and Cirstea, 2007). Sin embargo, cada fabricante proporciona su software para generar el archivo bitstream y programar la FPGA, es decir, tanto la operación como la distribución de recursos lógicos son arquitecturas cerradas y solo las grandes empresas saben cómo funciona su interior. Afortunadamente, se desarrolló un proyecto para sintetizar y generar el archivo bitstream a partir del lenguaje verilog con herramientas gratuitas, con lo que surge el concepto de hardware abierto (Wolf and Lasser, 2015).

El concepto de sistema de simulación hardware in the loop (HIL) es un sistema que combina elementos de un sistema real con su contraparte simulada y realizar el proceso de simulación en tiempo real (Rodríguez et al., 2020). Los sistemas de simulación HIL permiten probar sistemas embebidos al proporcionar la posibilidad de validar la ejecución del software en el hardware original sin los peligros que involucra un mal funcionamiento del controlador en combinación con el sistema mecánico real (Paiz et al., 2009). Las ventajas de utilizar HIL

son: reducción del tiempo de desarrollo, requisitos de pruebas exhaustivos para aplicaciones críticas de seguridad, costos reducidos del hardware necesario para ejecutar la simulación, por mencionar algunas (Kumar et al., 2017; Ghorbani et al., 2018). Por lo anterior, HIL es una herramienta útil en el campo de control y la robótica aplicable en etapas tempranas de diseño.

A continuación, se describen algunas aplicaciones donde los sistemas de simulación HIL toman un papel importante. En el campo de calidad de la energía, se propone un sistema de simulación en tiempo real como lo es HIL que describe el diseño, modelado y estudio de un compensador estático de derivación para redes de distribución de baja tensión. En particular, el Compensador Universal presentado está diseñado para compensar corrientes armónicas, potencia reactiva, desequilibrios de corriente y componentes oscilantes de potencia activa de cargas perturbadoras. Se presentan simulaciones en tiempo real como una técnica de simulación avanzada y efectiva para probar los resultados del circuito en diferentes condiciones de funcionamiento (Belloni et al., 2017).

La simulación HIL se utiliza con el objetivo de evaluar el rendimiento del sistema de control de vuelo de la aeronave y el sistema de propulsión con la influencia de las características de acoplamiento entre los dos sistemas, tales como ordenador de control de vuelo, controlador del motor, servoaccionadores y ordenador de simulación de tierra. Con el fin de cumplir con el requisito de tiempo real, el software de simulación se desarrolla sobre la base del sistema operativo en tiempo real (Zhao and Mai, 2013).

Por lo antes mencionado se propone utilizar un controlador de arquitectura abierta basado en FPGA y desarrollar un sistema HIL de un robot manipulador tomaod en cuanta la dinámica eléctrica de los actuadores del robot, para precisar trayectorias y disminución de desgaste mecánico del robot industrial.



## 1.1. Descripción del problema

En los últimos años, el uso de robots para aplicaciones avanzadas, desde la exploración espacial hasta los servicios y el entretenimiento, requiere mejores resultados, lo que se puede lograr utilizando soluciones avanzadas de control basadas en modelos, especialmente cuando los fenómenos parasitarios (principalmente fricción y desgaste en las articulaciones) deben ser compensados (Zhang et al., 2019).

Cuando un proceso está automático utilizando un robot, el resultado debe ser que la manufactura y la calidad del producto del robot satisfagan los requisitos para el proceso de fabricación, por lo que se invierte mucho tiempo asegurando esta característica (Yasuda, 2017).

En un entorno académico y de investigación, los robots de arquitectura abierta son cruciales para evaluar nuevas leyes de control e integración de sensores múltiples. Desafortunadamente, la mayoría de los fabricantes de robots comerciales no ofrecen arquitecturas de control abierto. Las nuevas propuestas de tecnología como la implementación de sistemas de visión o inteligencia artificial amplían la variedad de posibles aplicaciones de robótica industrial, por lo tanto, el desarrollo de algoritmos de control para este propósito lleva a la necesidad de desarrollar controladores de arquitectura abierta, en los que uno pueda tener acceso libre a las señales de retroalimentación y la capacidad de diseñar y probar nuevos algoritmos de control (Garcia-Valdovinos et al., 2016).

Uno de los problemas más comunes a la hora de realizar simulaciones y poder medir el desempeño de un controlador, es que simplemente se realizan simulaciones sobre modelos lineales y por lo tanto el desempeño del controlador

no funciona de la misma forma que en el sistema físico. Por lo tanto, es necesario obtener un modelo confiable. Otro problema es la necesidad de experimentar con el sistema real, lo que ocasiona, en el caso de robots, un gasto al energizarlo, posibles riesgos como pérdida de hardware o en peores casos daños físicos a personal, debido a una mala implementación o una falla del sistema de control (Alvarez-Gonzalez et al., 2017).

Para probar un nuevo controlador en un robot puesto en marcha provoca el paro del robot para comprobar el funcionamiento y desempeño del controlador lo que genera un tiempo de desperdicio, ya que por tenerlo en pruebas no se tiene en la línea de producción, lo que implica pérdidas en la inversión del propio robot ya que son equipos costosos.

## **1.2. Justificación**

La motivación de realizar este trabajo de investigación radica en la necesidad de tener un sistema que emule la dinámica de un robot manipulador. La contribución científica es el desarrollo de modelos que permitan validar modelos reportados por la literatura y propuestos en este trabajo, probar técnicas de control experimentalmente y evaluar el rendimiento de éstas. El desarrollo tecnológico e implementación de un sistema HIL se justifica principalmente como un entorno seguro y de bajo costo, cuando se consideran condiciones extremas de operación tales como fallas y en etapas tempranas de diseño de prototipos. Por otro lado, los sistemas HIL permiten el desarrollo de pruebas y validación de técnicas de control en menor tiempo y con menores costos, debido a que, durante la pruebas, no se requiere de hardware específico que sufra desgaste o requiera mantenimiento y más aún cuando existe la necesidad de evaluar

resultados estadísticamente con gran número de corridas.

### **1.3. Estructura del documento**

El capítulo 1, presenta la introducción general del trabajo de investigación y la estructura del documento.

En capítulo 2, presenta el estado del arte de las aplicaciones desarrolladas con sistemas HIL y controladores de arquitectura abierta.

El capítulo 3, muestra la hipótesis de la investigación desarrollada.

El capítulo 4, muestran el objetivo general y los objetivos específicos del trabajo de investigación.

El capítulo 5, muestra la metodología propuesta del desarrollo tecnológico e implementación del sistema HIL, así como la implementación del algoritmo de sintonía del controlador, diseño del algoritmo de generación de trayectorias y descripción e implementación del controlador con herramientas libres.

En el capítulo 6, se discuten los resultados obtenidos y rendimiento de los controladores implementados.

En el capítulo 7, se presenta las principales conclusiones de esta investigación y se sugieren los trabajos futuros.

# ANTECEDENTES

## 2.1. Revisión literaria

La simulación es la utilización de un modelo de sistemas, que trata de acercarse más a las características de la realidad, a fin de reproducir la esencia de las operaciones reales. Asimismo, es una representación de un proceso real, mediante el empleo de un modelo o sistema que reaccione de la manera similar a la que reaccionaría uno real, en un conjunto de condiciones dadas (Borangiu et al., 2010). Una definición más formal formulada por Robert E. Shannon es: "La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias dentro de los límites impuestos por un cierto criterio o un conjunto de ellos para el funcionamiento del sistema".

Una simulación tiene como finalidad conocer, analizar y evaluar el funcionamiento de un sistema sin necesidad de la construcción o de la experimentación con el sistema real, ya que esto podría resultar extremadamente caro o peligroso (Husain, 2011).

Hoy en día, la simulación se realiza mediante computadoras y software específicos, siendo una poderosa técnica de resolución de problemas reales, se

experimenta con un modelo numérico, de tal forma que con los resultados se puede obtener una estimación de las características del sistema (Borangiu et al., 2010).

Thomas H. Naylor, (2009) ha sugerido que un estudio de simulación es importante para la ingeniería de sistemas porque presenta las siguientes ventajas en el diseño de estos:

- A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema.
- Una observación detallada del sistema que se está simulando puede conducir a un mejor entendimiento del sistema y por consiguiente a sugerir estrategias que mejoren la operación y eficiencia del sistema.
- La simulación de sistemas complejos puede ayudar a entender mejor la operación del sistema, a detectar las variables más importantes que interactúan en el sistema y a entender mejor las interrelaciones entre estas variables.
- La técnica de simulación puede ser utilizada para probar con nuevas situaciones, sobre las cuales tiene poca o ninguna información. A través de esta simulación se puede anticipar mejor a posibles resultados no previstos.

Teniendo en cuenta lo expuesto anteriormente es fácil concluir que la simulación también es una herramienta útil en robótica.

A diferencia de las ventajas mencionadas, la técnica de simulación presenta el problema de requerir equipo computacional y recursos humanos costosos.

Además, generalmente se requiere bastante tiempo para que un modelo de simulación sea desarrollado y perfeccionado. Finalmente, en algunas ocasiones la simulación puede llegar a ser imprecisa, sin poder medir el grado de error, aunque esto dependerá de las simplificaciones realizadas a la hora de diseñar el modelo del sistema (Kain et al., 2011).

La construcción de un robot es cara y complicada, en ocasiones son difíciles de programar y su fase de desarrollo muy probablemente puede dañar considerablemente el robot, por esto se utiliza uno virtual que pueda interactuar con elementos físicos del sistema, con la finalidad que la respuesta de la simulación sea en tiempo real y lo más parecida al sistema real, para esto se utilizan técnicas de simulación HIL (Kawabata et al., 2017).

Un simulador HIL consiste en tres componentes principales. Los requerimientos para cada componente dependen en las características de las aplicaciones.

- GUI: Normalmente es una computadora personal que tiene función de interfaz gráfica de usuario donde se pueden ajustar los parámetros del simulador para visualizar los resultados.
- Sistema Digital: Puede ser una computadora personal, una computadora industrial, un procesador digital de señal (DSP, microcontrolador) dependiendo de los requerimientos de cada aplicación.
- Interfaz I/O: Emula las señales eléctricas (digital, analógica o compuesta) del simulador, se pueden usar tarjetas de adquisición de datos o una placa FPGA.

En la Tabla 2.1 se muestra el estado del arte referente a aplicaciones

realizadas en simulación HIL.

Tabla 2.1: Estado del arte de simulación HIL.

Autor(es)	Año	Resumen	Software o hardware
Chenkun Qi	2017	Realiza una una compensación para un controlador de vuelo por medio de la simulación HIL. Evalúa el rendimiento del sistema de control de vuelo	LabVIEW, DSpace
Yao Zhao	2017	de la aeronave y el sistema de propulsión con la influencia de las características de acoplamiento entre los dos sistemas por medio de simulación HIL.	DSpace
Alvarez-Gonzalez	2017	Evalúa el rendimiento de técnicas de control y compensación de armónicos por medio de simulación HIL.	LabVIEW
Sebastian Kain	2011	Realiza una metodología para aplicar técnicas de simulación HIL en tiempo real para la optimización de procesos en la industria.	LabVIEW

En la Tabla 2.2 se muestran los antecedentes de controladores de arquitectura abierta para aplicaciones de control con sistemas similares. Se observa que debido a los requerimientos del sistema de control, tales como procesamiento, respuesta del sistema, la tendencia es utilizar dispositivos FPGA, sin embargo, el trabajo que se propone es emplear herramientas libres que no dependen de un fabricante.

Tabla 2.2: Antecedentes de controladores de arquitectura abierta

Trabajo	Controlador	Fabricante	Herramienta-síntesis	Sintonización	Tiempo de muestreo
Propuesto	FPGA ICE40 UltraPlus 5k	Lattice	Open tool	Algoritmo genético	0.1 ms
(Martínez-Prado et al., 2018)	FPGA Spartan-3	Xilinx	ISE-Xilinx	Empírico	1 ms
(Morales-Velazquez et al., 2010)	FPGA Spartan-3	Xilinx	ISE-Xilinx	-	-
(Jokić et al., 2020)	FPGA	Altera	MATLAB/Simulink	-	-
(Ordóñez Cerezo et al., 2019)	FPGA ICE40hx4k	Lattice	Open tool	Empírico	-
(García-Martínez et al., 2017)	Raspberry Pi-FPGA ZYNQ 7	Xilinx	Vivado-Xilinx	-	-
(Ponce et al., 2015)	FPGA Spartan-3	Xilinx	LabVIEW NI	Empírico	0.1 ms
(Concha Sánchez et al., 2020)	PC-Arduino	Arduino	MATLAB/Simulink	asignación de polos	60 ms
(Correa et al., 2017)	PC-Launchpad-Arduino	Texas Instruments	-	-	50 ms
(Ricci and Meacci, 2018)	FPGA Max10	Altera	NiosIIsoft	-	0.2 ms
(García-Martínez et al., 2019)	Raspberry Pi-FPGA ZYNQ 7	Xilinx	Vivado-Xilinx	Fuzzy	5 ms

## 2.2. Modelo matemático del PMSM

El modelo matemático de un PMSM puede ser representado utilizando un cambio de coordenadas que permiten expresar el modelo de ecuaciones simples, transformación DQ en la que se definen 3 fases ficticias. La fase  $d$  produce un

campo magnético que gira junto al rotor y es paralelo al polo norte del rotor. La fase  $q$  también gira junto con el rotor y forma  $90^\circ$  con la fase  $d$ . Por último, se tiene la fase de secuencia cero, que no se considera al no tener efecto en el motor. El modelo del motor PMSM se muestra en las siguientes ecuaciones de acuerdo con Nicola and Nicola (2020). Considerando  $L_q \neq L_d$ .

$$L_q \frac{di_q}{dt} + R_q i_q + N_p L_d i_D \dot{q} + K_{T2} \dot{q} = v_q \quad (2.1)$$

$$L_d \frac{di_d}{dt} + R_d i_d - N_p L_q i_Q \dot{q} = v_d \quad (2.2)$$

$$\tau_l = [K_{T1} i_D + K_{T2}] i_q \quad (2.3)$$

Donde  $v_q$ ,  $v_d$ ,  $i_q$  e  $i_d$  son voltajes y corrientes eléctricas de la transformada de fase DQ, respectivamente.  $L_q$ ,  $L_d$ ,  $R_q$  y  $R_d$  inductancias y resistencias de DQ.  $N_p$  números de pares de polos.  $K_{T1} = N_p(L_d - L_q)$  y  $K_{T2} = \sqrt{\frac{3}{2}} N_p K_B$  son las constantes de pares.  $K_B$  el coeficiente de la fuerza electromotriz.  $\tau_l$  es el par de carga.

## 2.3. Robot Industrial

El modelado de un robot manipulador es una premisa necesaria para encontrar las estrategias de control que permiten ejecutar el movimiento deseado. El análisis cinemático de la estructura del robot manipulador se realiza con el estudio analítico de la geometría del movimiento del robot con respecto a un



marco de referencia fijo sin considerar las fuerzas o momentos que originan dicho movimiento. La cinemática inversa busca obtener una descripción analítica del comportamiento de las variables articulares a partir de las coordenadas de postura del efector final del robot (Kelly and Santibáñez, 2003).

Por otra parte, la dinámica del robot define la matemática de las ecuaciones de movimiento del robot que incluyen los pares y fuerzas aplicadas al mismo. El modelo dinámico real de un robot se puede obtener mediante las leyes físicas conocidas como las leyes de Newton-Euler y la mecánica lagrangiana (Fu et al., 1987).

Las ecuaciones dinámicas de movimiento del robot son útiles para la simulación del movimiento del robot en computadora, el diseño de ecuaciones de control apropiadas para el robot. Una vez conocido el modelo cinemático y dinámico del robot manipulador, es interesante planear una tarea deseada controlando al manipulador para que siga un camino previamente establecido (Craig, 1977).

### **2.3.1. Modelo cinemático**

La cinemática del robot estudia los aspectos geométricos del movimiento respecto a un sistema de referencia, sin considerar las causas que lo producen. Un robot manipulador se puede modelar, como una cadena cinemática de cuerpos rígidos (eslabones) conectados entre sí por medio de articulaciones prismáticas o rotacionales. En cualquier caso, cada articulación posee un eje articular sobre el cual se realiza el movimiento de traslación o rotación. En una cadena cinemática serial abierta uno de los extremos está unido a una base fija y el otro extremo se monta en el efector, cuya finalidad es manipular objetos o

realizar tareas específicas. El movimiento de la estructura mecánica se constituye de movimientos elementales de cada articulación con respecto a la previa. Entonces, con la finalidad de manipular un objeto en el espacio, es necesario describir la posición y orientación del órgano terminal (Craig, 1977).

El extremo se mueve en un único plano, su posición está determinada por las coordenadas cartesianas  $(x, y)$  en metros. Las coordenadas articulares son representadas por las variables  $q_1$  para el eslabón 1 y  $q_2$  para el eslabón 2, en radianes.

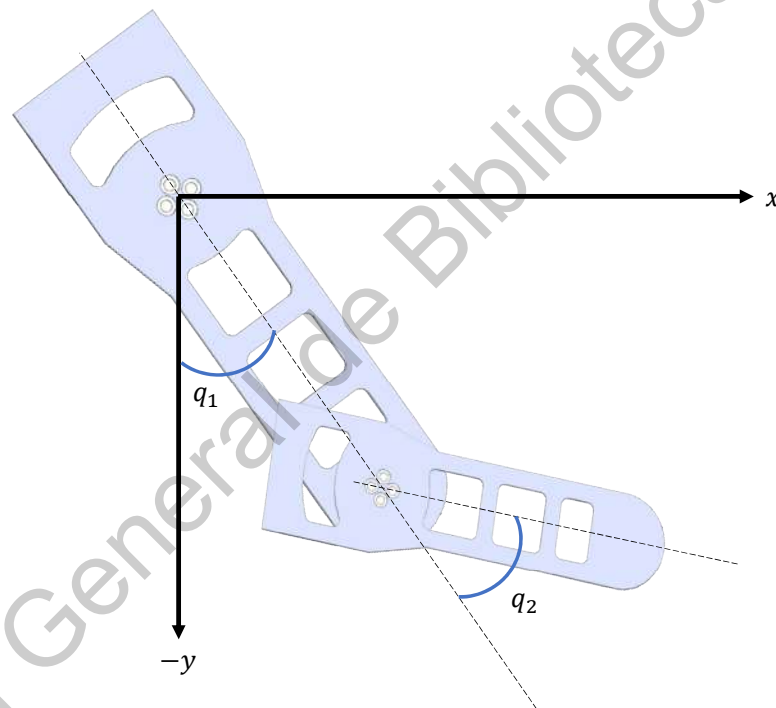


Figura 2.1: Esquema cinemático del robot.

El modelo cinemático del robot de 2 gdl se obtiene a través del método geométrico, debido a que el robot tiene un grado reducido de libertad y no posee herramienta. La posición *home* o posición de equilibrio estable del robot manipu-

lador está ubicada sobre el eje  $-y$ .

$$x = l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 + q_2) \quad (2.4)$$

$$y = -l_1 \text{sen}(q_1) - l_2 \text{sen}(q_1 + q_2) \quad (2.5)$$

### 2.3.2. Cinemática diferencial y singularidades

Es la relación entre las velocidades articulares  $\dot{q}$  con las velocidades lineales  $v$ . El modelo diferencial queda concretado en la matriz Jacobiana geométrica directa e inversa.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \iff \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.6)$$

Como el valor numérico de la matriz depende de los valores de las coordenadas articulares  $q_i$ , su valor será distinto para cada valor del espacio articular.

$$J = J(q) = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \frac{\partial f_x}{\partial q_2} \\ \frac{\partial f_y}{\partial q_1} & \frac{\partial f_y}{\partial q_2} \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 + q_2) & l_2 \text{sen}(q_1 + q_2) \end{bmatrix} \quad (2.7)$$

El Jacobiano es de especial interés para generar trayectorias, obtener las velocidades cartesianas a velocidades articulares y viceversa.

Cuando el determinante de la Jacobiana se anula, el brazo presenta una configuración singular. En estos puntos, se generan grandes velocidades en los

motores, inalcanzable e indeseable en la práctica.

La configuración singular se encuentra en el interior del espacio de trabajo y en el límite de la zona de trabajo, producida por la alineación de los dos eslabones coincidiendo sus direcciones, es decir, cuando la articulación del codo toma un ángulo de cero o 180 grados para  $q_1$  cualquier valor.

$$|J| = 0 \text{ para } q_2 = 0, \pm n\pi \quad (2.8)$$

### 2.3.3. Modelo dinámico

Para analizar matemáticamente el comportamiento de un sistema físico se deben considerar dos vertientes, la primera es la cinemática que se encarga de describir el comportamiento a través de una trayectoria sin considerar las causas que lo producen, y la segunda es la dinámica que relaciona las fuerzas y el movimiento para conocer la evolución en el tiempo del mismo.

La energía potencial representa a las fuerzas conservativas como la fuerza gravitacional y la fuerza de los resortes. La energía cinética representa el trabajo necesario para acelerar un cuerpo de una masa determinada desde el reposo hasta la velocidad indicada.

Al considerar un robot manipulador de cadena cinemática abierta con  $n$  grados de libertad, su energía total  $E$  está dada por la suma de la energía cinética  $K$  y la energía potencial  $U$  (Kelly et al., 2005). El  $L$  Lagrangiano es diferencia entre la energía cinética y energía potencial, para calcular las denominadas

ecuaciones de movimiento de Euler-Lagrange para  $\tau_i$ .

$$L = K - U \quad (2.9)$$

$$\tau_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad i = 1, 2 \quad (2.10)$$

$$\tau_1 = [\theta_1 + 2\theta_2 \cos(q_2)] \cdot \ddot{q}_1 + [\theta_3 + \theta_2 \cos(q_2)] \cdot \ddot{q}_2 - \theta_2 \sin(q_2) \cdot \dot{q}_2 \cdot (2\dot{q}_1 + \dot{q}_2) + \theta_4 \sin(q_1) + \theta_5 \sin(q_1 + q_2) \quad (2.11)$$

$$\tau_2 = [\theta_3 + \theta_2 \cos(q_2)] \cdot \ddot{q}_1 + \theta_3 \cdot \ddot{q}_2 - \theta_2 \sin(q_2) \cdot \dot{q}_1^2 + \theta_5 \sin(q_1 + q_2) \quad (2.12)$$

donde:

$$\theta_1 = m_1 L_{c1}^2 + m_2 L_{c2}^2 + m m_2 L_1^2 + I_1 + R_1^2 I_{m1} + I_2 + R_2^2 I_{m2} \quad (2.13)$$

$$\theta_2 = m_2 L_1 L_{c2} \quad (2.14)$$

$$\theta_3 = m_2 L_{c2}^2 + I_2 + R_2^2 I_{m2} \quad (2.15)$$

$$\theta_4 = (m_1 L_{c1} + m_2 L_1 + m m_2 L_1) g \quad (2.16)$$

$$\theta_5 = m_2 L_{c2} g \quad (2.17)$$

EL modelo dinámico del robot manipulador de 2 gdl en su forma matricial más compacta, se presenta en las siguientes ecuaciones:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (2.18)$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} \quad (2.19)$$

$q \in \mathbb{R}^2$  es el vector de posiciones articulares.

$M(q) \in \mathbb{R}^{2 \times 2}$  es la matriz de inercia, la cual es simétrica y semidefinida positiva.

Representa el cambio de estado de movimiento del sistema.

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} = \begin{bmatrix} \theta_1 + 2\theta_2 \cos(q_2) & \theta_3 + \theta_2 \cos(q_2) \\ \theta_3 + \theta_2 \cos(q_2) & \theta_3 \end{bmatrix} \quad (2.20)$$

$C(q, \dot{q}) \in \mathbb{R}^{2 \times 2}$  es la matriz de fuerzas centrípetas y coriolis.

Las fuerzas centrípetas son fuerzas radiales con signo contrario a las centrífugas. La fuerza de Coriolis simboliza la desviación del movimiento de-

bido a su componente de rotación.

$$\begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} -\theta_2 \dot{q}_2 \text{sen}(q_2) & -\theta_2 (\dot{q}_1 + \dot{q}_2) \text{sen}(q_2) \\ \theta_2 \dot{q}_1 \text{sen}(q_2) & 0 \end{bmatrix} \quad (2.21)$$

$g(q) \in \mathbb{R}^2$  es el vector de fuerzas de gravedad.

$$\begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \begin{bmatrix} \theta_4 \text{sen}(q_1) + \theta_5 \text{sen}(q_1 + q_2) \\ \theta_5 \text{sen}(q_1 + q_2) \end{bmatrix} \quad (2.22)$$

De esta manera, el movimiento deseado del efector final demanda un análisis preciso de las características de la estructura mecánica, actuadores y sensores del robot.

El objetivo de la planificación de trayectorias es generar leyes temporales para las variables relevantes de la tarea, ya sean articulares o de postura, basada en una descripción precisa del movimiento deseado. Las trayectorias generadas, constituyen las entradas de referencia al control de movimiento.

## 2.4. Perfiles de movimiento

En un robot manipulador, la velocidad debe cambiar siguiendo una forma característica para lograr una posición deseada, donde el movimiento debe ser menos agresivo en comparación con una entrada escalonada para evitar una alta tensión en el eje del motor. Los perfiles de velocidad consisten en diferentes formas, que dependen del movimiento; pueden ser funciones triangulares,

trapezoidales, curva S o parabólicas (García-Martínez et al., 2019). El perfil de velocidad de la curva S es una trayectoria suave compuesta por un polinomio de tercer grado. Permite delimitar el tirón en un intervalo propuesto por el diseñador para asegurar la durabilidad del actuador (Fang et al., 2019). Una desventaja que se presenta en este tipo de perfiles de movimiento es el costo computacional, ya que requiere diferentes técnicas de derivadas e integrales (García-Martínez et al., 2017). Por otro lado, el perfil de velocidad trapezoidal es un polinomio de segundo grado que consta de una fase de aceleración y desaceleración y una etapa de velocidad constante que está presente en el medio de las fases de aceleración-desaceleración. El costo computacional es adecuado para una estructura simple y una respuesta rápida (Heo et al., 2019). Por ésta razón es uno de los perfiles de velocidad más utilizados en los controladores comerciales (García-Martínez et al., 2017). Sus movimientos consisten en un período de aceleración  $t_0 \leq t \leq t_0 + T_a$ , seguido de un período de velocidad constante  $t_0 + T_a \leq t \leq T - T_a$  y terminan con un período de desaceleración  $T - T_a \leq t \leq T$ . Mientras que el perfil de velocidad tiene una forma trapezoidal, como se muestra en la figura 2.2, la evolución de la posición  $\theta_d(t)$  se representa usando un polinomio de segundo orden, ver ecuación (2.23), que representa la posición general.



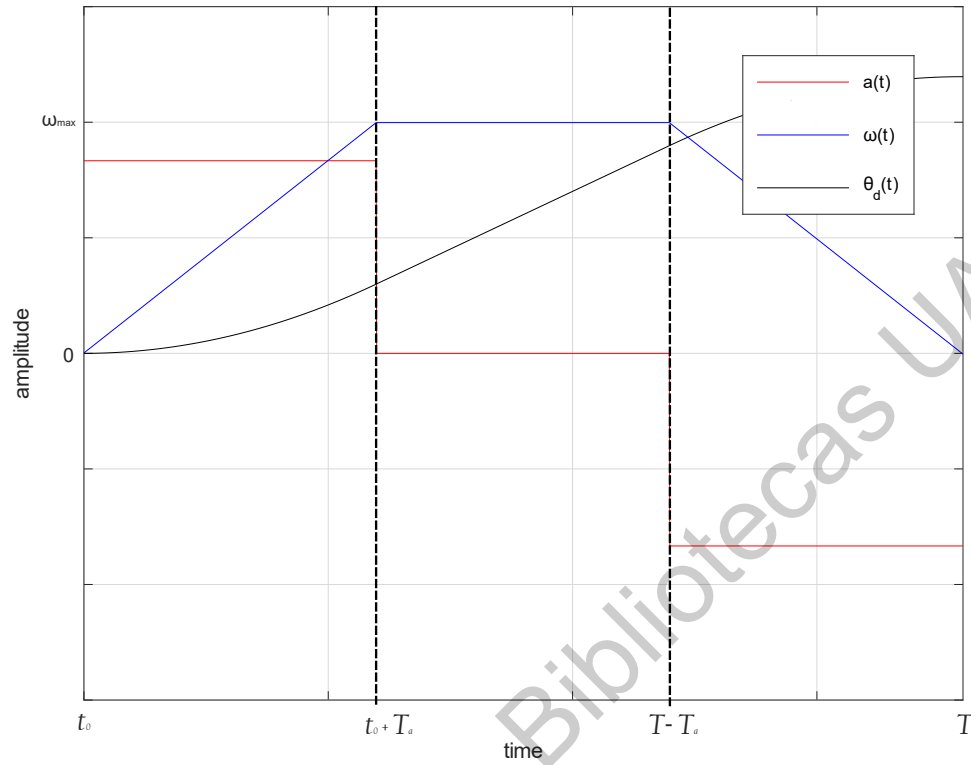


Figura 2.2: Perfil de velocidad trapezoidal.

$$\theta_d(t) = \begin{cases} \frac{a}{2}t^2 & t_0 \leq t \leq t_0 + T_a \\ \frac{aT}{3}t - \frac{5aT^2}{18} & t_0 + T_a \leq t \leq T - T_a \\ \frac{-a}{2}t^2 + aTt - \frac{5aT^2}{18} & T - T_a \leq t \leq T \end{cases} \quad (2.23)$$

Se proponen la velocidad angular máxima  $\omega_{max}$  y la posición angular deseada  $\theta_d$  del espolón motor y, con esto, se calcula la duración total  $T$  del movimiento y la aceleración  $a$ , como sigue (Heo et al., 2019):

$$T = \frac{3\theta_d}{2\omega_{max}} \quad (2.24)$$

$$a = \frac{3\omega_{max}}{T} \quad (2.25)$$

## 2.5. CONTROL PID

El controlador PID es la estrategia de control más utilizada en procesos industriales, debido a su notable efectividad, simplicidad de implementación y amplia aplicabilidad (Emre et al., 2015; Yang et al., 2018; Mu et al., 2019; Jimenez-Fernandez et al., 2012). El controlador PID juega un papel vital para el rendimiento del sistema al minimizar el error y proporcionar una mejor estabilidad funcional del sistema. A continuación se presenta la ecuación de un controlador PID (2.26) en tiempo continuo (Hernández-Guzmán et al., 2013).

$$i_c(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \quad (2.26)$$

donde  $k_p$  es la ganancia proporcional,  $k_i$  es la ganancia integral,  $k_d$  es la ganancia derivada,  $e(t)$  es el error y  $i_c(t)$  es la señal de control.

Usando el método de Euler hacia atrás, la discretización del controlador PID (2.27) se realizó con un tiempo de muestreo  $T_s$ . En la ecuación (2.27)  $e[k]$ ,  $e[k-1]$  y  $e[k-2]$  representan el error actual y los errores anteriores en tiempo discreto;  $i_c[k]$  y  $i_c[k-1]$  denotan la salida actual y retrasada del controlador (Yang et al., 2018), respectivamente.

$$i_c[k] = a_0 e[k] + a_1 e[k-1] + a_2 e[k-2] + i_c[k-1] \quad (2.27)$$

donde:

$$a_0 = k_p + T_s k_i + \frac{k_d}{T_s}, \quad a_1 = -(k_p + 2\frac{k_d}{T_s}), \quad a_2 = \frac{k_d}{T_s}$$

## 2.6. Lazo de control interno de corriente PI

A continuación se establecen las ecuaciones del controlador interno, tomando en cuenta el subsistema eléctrico del motor PMSM esta dado por la ecuación 2.1 y 2.2 y el subsistema mecánico esta dado por la ecuación 2.3. El controladores PI de corriente, dados por las ecuaciones (2.28) y (2.29).

$$v_q = -k_{qp}(i_q - i_{qc}) - k_{qi} \int_0^t (i_q - i_{qc}) d\tau \quad (2.28)$$

$$v_d = -k_{dp}(i_d - i_{dc}) - k_{di} \int_0^t (i_d - i_{dc}) d\tau \quad (2.29)$$

Donde  $v_d$  y  $v_q$  representan los voltajes de las fases DQ. Las corrientes de fase DQ  $i_d$  e  $i_q$ .  $i_{dc}$  representa la corriente deseada para la fase  $d$  y es igual a cero.  $i_{qc}$  representa la corriente deseada para la fase  $q$  y es la salida del controlador maestro de posición.  $k_{qp}$  y  $k_{dp}$  representan las ganancias proporcionales del controlador.  $k_{qi}$  y  $k_{di}$  representan las ganancias integrales del controlador.

## 2.7. Sintonización

El proceso de ajuste es fundamental en el diseño de controladores. Si se conoce el modelo del sistema, entonces es posible aplicar técnicas de diseño como el lugar de las raíces o la respuesta de frecuencia para identificar

las especificaciones requeridas del sistema. Sin embargo, también existen técnicas experimentales para ajustar el controlador PID; uno de ellos es el caso de Ziegler-Nichols, donde no se requiere el modelo del sistema. De acuerdo con (Reynoso-Meza et al., 2013), la tendencia de los controladores PID es resolver múltiples requisitos, especificaciones y objetivos de diseño que entran en conflicto entre sí. Un controlador PID robusto también puede resolverse mediante un procedimiento de optimización.

Un Algoritmo Genético (GA) es un algoritmo heurístico cuya función es evolucionar un conjunto de individuos (población) y hacer que estos individuos se acerquen a la solución óptima con el paso de cada generación (iteración del algoritmo). Los principales operadores de GA son el cruce y la mutación. Los individuos de las poblaciones están mejorando a través de las generaciones con estos operadores y procesos de selección. Una característica que diferencia a GA es la forma de representar el problema, ya que plantea la modificación de cada individuo a través de sus cromosomas. En este caso, se representa mediante números binarios. La figura 2.3 muestra los procesos iterativos que están involucrados en GA. Se propone un algoritmo monoobjetivo, ya que requiere, además de un error de estado estable cero, que el tiempo de establecimiento del sistema sea corto y que la respuesta del controlador sea suave, de modo que el sistema no esté saturado. .

A diferencia de las obras (Xin et al., 2013; Bandyopadhyay et al., 2001; Flores-Morán et al., 2018), nuestro algoritmo no se ejecuta en simulación. Sin embargo, es un proceso que se ejecuta en línea desde la interfaz gráfica de usuario (GUI) desarrollada. La función objetivo se compone de dos funciones de costo convexas: la ecuación (2.30) de la Integral del Error Absoluto (ISE) y la Integral del Cuadrado del Error 2.31 (por sus siglas en error), como una medida

indirecta del sobrepaso y la ecuación (2.32) de Variación Total (TV) para evaluar cambios en la señal de control  $i_d(t)$ . Estas ecuaciones se minimizan mediante el algoritmo GA. Como resultado, el sobreimpulso máximo se limita y los tiempos de establecimiento y error se minimizan mediante una señal de control suave. No obstante, para minimizar la complejidad y el tiempo de convergencia del algoritmo,  $J_m$  ecuación (2.33) se describe como una función objetivo ponderada de las ecuaciones descritas. Además,  $\alpha$ ,  $\beta$  y  $\gamma$  son los parámetros de penalización y su función es controlar la región de búsqueda penalizando la función de restricción. Estos valores se modifican a través de las restricciones, por ejemplo, cuando una restricción sale de la región de búsqueda, la función objetivo se ve fuertemente penalizada y, cuando se cumplen estas restricciones, los valores de  $\alpha$  y  $\beta$  permanecen 1, mientras que  $\gamma$  son 0.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (2.30)$$

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (2.31)$$

$$TV = \int_0^{\infty} |i_c(t+1) - i_c(t)| dt \quad (2.32)$$

$$\min(J_m = \alpha IAE + \beta ISE + \gamma TV) \quad (2.33)$$

con restricciones:

$$0 < k_p \leq 100 ; 0 < k_i \leq 200 ; 0 < k_d \leq 10$$

$$\mu_p \leq 15\%$$

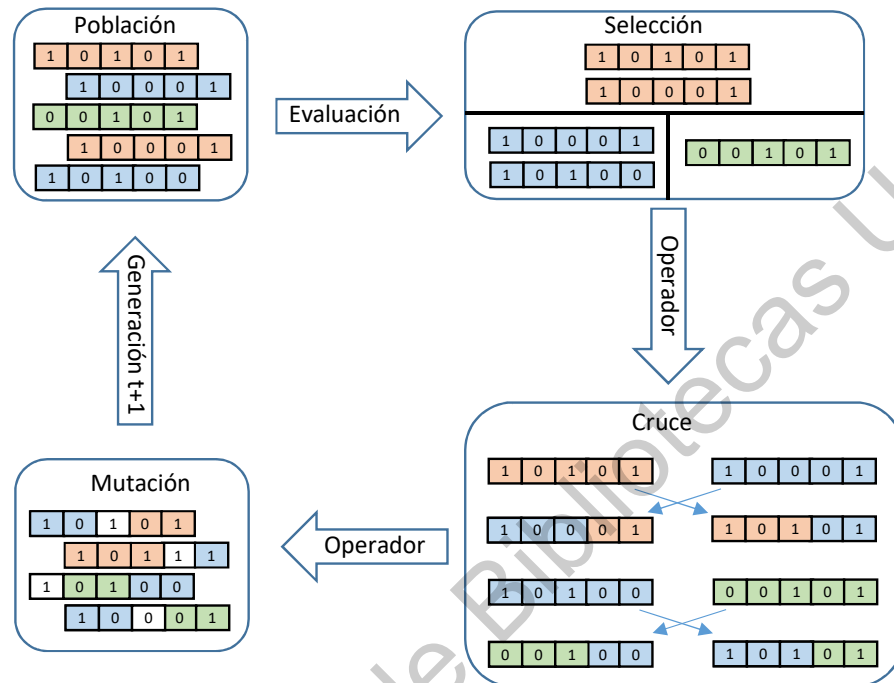


Figura 2.3: Diagrama de flujo del AG.

# HIPÓTESIS

A través de un sistema HIL con base en el modelo dinámico de un robot de arquitectura abierta, diseño de un controlador y un algoritmo generador de perfiles de velocidad se optimizará el seguimiento de trayectorias que será medido en términos del consumo energético.

Dirección General de Bibliotecas UAQ

# OBJETIVOS

## 4.1. Objetivo General.

Diseñar e implementar un sistema de simulación HIL de un robot arquitectura abierta para mejorar el desempeño energético del mismo a través de un algoritmo generador de perfil de velocidad y un controlador robusto con una precisión de  $0.01^\circ$ .

## 4.2. Objetivos Específicos.

- Identificar los parámetros mecánicos del robot, así como de los sensores que involucra el mismo por medio de los manuales de usuario para la construcción del modelo.
- Desarrollar el modelo matemático de la cinemática y dinámica del robot con base a los datos adquiridos para la simulación HIL en tiempo real.
- Diseñar el algoritmo de control para el seguimiento de trayectorias del robot manipulador con el modelo dinámico.
- Comparar el modelo matemático de la simulación HIL con el sistema real para determinar la bondad de ajuste.



## METODOLOGÍA

### 5.1. Modelo del robot con actuadores

El modelo completo del robot manipulador de dos grados de libertad, se obtiene agregando la dinámica de los actuadores. El modelo comprende la dinámica mecánica dada por la ecuación (2.19) obtenida mediante el modelado por Euler-Lagrange Y la dinámica eléctrica para un motor PMSM dada por la ecuación (2.1) y (2.2), obtenidas a partir de la transformación de Park.

De esta manera, si se aplica un voltaje de entrada  $v_d$  y  $v_q$  al motor, se genera el par de entrada que produce el movimiento de cada eslabon del robot.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = [K_{T1}I_D + K_{T2}]I_q \quad (5.1)$$

$$V_q = L_Q\dot{I}_q + R_QI_q + N_pL_DI_D\dot{q} + K_{T2}\dot{q} \quad (5.2)$$

$$V_d = L_D\dot{I}_d + R_DI_d - N_pL_QI_Q\dot{q} \quad (5.3)$$

Donde  $q, \dot{q}, \ddot{q} \in \mathbb{R}^2$  son los vectores de posición, velocidad y aceleración

angular de las uniones. En la ecuación (2.19) se tiene que  $M(q)$  es una matriz simétrica de  $2 \times 2$  definida positiva,  $C(q, \dot{q})$  es la matriz de Coriolis de  $2 \times 2$ ,  $g(q)$  es el vector de  $2 \times 1$  de pares gravitacionales.

En la ecuación (2.1) y (2.2), se tiene que  $V_q$  y  $V_d$  son vectores de  $2 \times 1$  y representa los voltajes de las fases DQ.  $I_q$  e  $I_d$  son vectores de  $2 \times 1$  y representan las corrientes eléctricas asociadas a las fases DQ.  $I_Q$  e  $I_D$  son matrices de  $2 \times 2$  diagonales con elementos  $I_q$  e  $I_d$  respectivamente.  $L_Q$  y  $L_D$ , son matrices constantes diagonales definidas positivas de  $2 \times 2$  con elementos  $L_q$  y  $L_d$  respectivamente, que representan las inductancias de las fases DQ.  $R_Q$  y  $R_D$ , son matrices constantes diagonales definidas positivas de  $2 \times 2$  con elementos  $R_q$  y  $R_d$  respectivamente, que representan las resistencias de las fases DQ.  $N_p$  es una matriz constante diagonal definida positiva de elementos  $n_p$ , que contiene el número de pares de polos del rotor magnético.  $K_B$  es una matriz constante diagonal definida positiva que contiene los coeficientes de la fuerza electromotriz.  $K_{T1}$  y  $K_{T2}$ , son matrices constantes diagonales de par.  $\tau = [K_{T1}I_D + K_{T2}]I_q$  es un vector ( $n \times 1$ ) de pares aplicado a las uniones del robot con  $K_{T1} = N_p(L_D - L_Q)$  y  $K_{T2} = \sqrt{\frac{3}{2}}N_pK_B$ .

## 5.2. Descripción Hardware

En la Tabla 5.1 se muestran los parámetros mecánicos del robot y eléctricos de PMSM para la implementación del sistema HIL.

Tabla 5.1: Parámetros mecánicos del robot y eléctricos de los actuadores.

Parámetros mecánicos del robot				Parámetros eléctricos del PMSM			
Descripción	Símbolo	Valor	Unidad	Descripción	Símbolo	Valor	Unidad
Masa eslabón 1	$m_1$	6.5225	kg	Número de pares de polos	$N_p$	120	
Masa eslabón 2	$m_2$	2.0458	kg	Inductancia de fase $d$	$L_D$	0.00636	H
Longitud eslabón 1	$l_1$	0.26	m	Inductancia de fase $q$	$L_Q$	0.00672	H
Longitud eslabón 2	$l_2$	0.26	m	Resistencia de fase $d$	$R_D$	1.9	$\Omega$
Distancia de centro de masa eslabón 1	$l_{c1}$	0.0983	m	Resistencia de fase $q$	$R_S$	1.9	$\Omega$
Distancia de centro de masa eslabón 2	$l_{c2}$	0.0229	m	Constante de fuerza electromotriz	$K_B$	0.0106	Wb
Inercia eslabón 1	$I_1$	0.1213	kg m <sup>2</sup>	Constante de par	$K_T2$	1.5579	N
Inercia eslabón 2	$I_2$	0.0116	kg m <sup>2</sup>				
Aceleración de la gravedad	$g$	9.81	$\frac{m}{s^2}$				
Par máximo eslabón 1	$\tau_{M1}$	15	Nm				
Par máximo eslabón 2	$\tau_{M2}$	4	Nm				

Los FPGA fueron desarrollados por la empresa Xilinx en 1984, con el propósito de crear chips que actuaran como una cinta en blanco, permitiendo a los usuarios programar la tecnología ellos mismos. El concepto tuvo éxito y otras empresas desarrollaron FPGA debido a su diseño reconfigurable y procesamiento paralelo. Sin embargo, como es una nueva tecnología, cada fabricante mantiene en secreto el rendimiento interno, por lo que el usuario debe adquirir software del fabricante para realizar la síntesis y programarlo (Romanov et al., 2017). Un inconveniente adicional se produce cuando estas empresas desarrollan un nuevo producto, por lo que el software utilizado es obsoleto, sin mencionar que necesitan un procesador específico para ejecutar este software. Hasta hace unos años, se introdujo el proyecto IceStorm, en el que se realizaba ingeniería inversa para obtener el flujo de bits de una familia específica de FPGA (Lattice iCE40). Esto permitió obtener tanto la operación como la distribución interna *cite wolfproject*, completando así el flujo de desarrollo desde el lenguaje descriptivo Verilog hasta la creación del bitstream (Monmasson and Cirstea, 2007; Romanov et al., 2017) basado en herramientas gratuitas multiplataforma.

La estructura del proyecto IceStorm se muestra con un diagrama de flujo en la figura 5.1. En (Wolf and Lasser, 2015), se observa que IceStorm es un proyecto de código abierto que está integrado por diferentes herramientas: (1) Yosys

es una herramienta de síntesis similar a la que usa el fabricante que convierte el código Verilog a un formato diferente, como como FLIG, EDIF, BTOR, SMT-LIB y RTL con soporte Verilog 2005. Además, (2) Arachne-PNR es una herramienta que implementa el lugar y la ruta del proceso de construcción de hardware utilizando archivos BLIF y genera el archivo IceStorm TXT. Actualmente, Arachne-PNR no es compatible y la herramienta NextPNR se convirtió en un reemplazo funcional completo con mejoras significativas. Finalmente, (3) IcePack convierte el archivo TXT en bitstream (Yu et al., 2018; Wolf and Lasser, 2015).

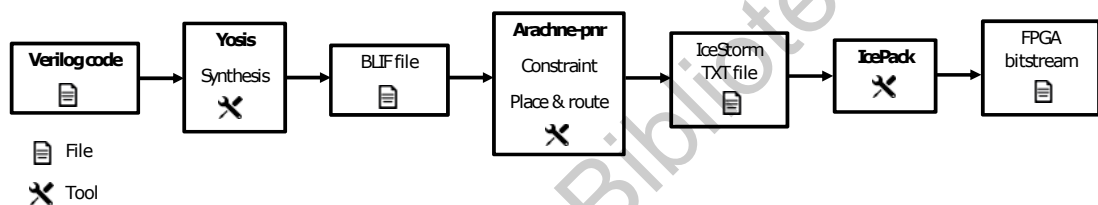


Figura 5.1: Flujo de diseño en IceStorm.

La implementación de los cores se desarrolló en lenguaje Verilog a través de la herramienta IceStorm. Se utilizó un FPGA ICE40UP5K de la familia Lattice con una frecuencia de reloj de 24 MHz. La figura 5.2 muestra los módulos implementados en FPGA para control y monitoreo en tiempo real. El módulo UART es responsable de la comunicación entre la FPGA y la GUI de Python. La comunicación se inicializa a través de la GUI en un formato con la siguiente estructura de 6 bytes para las ganancias  $a_0$ ,  $a_1$  y  $a_2$  del controlador PID discreto, respectivamente, seguido de un byte de configuración del actuador (Digital Analog Convertidor o módulo PWM), utilizando 4 bytes para la trayectoria deseado de cada eslabón y finalmente los puntos del camino a ejecutar. El FPGA ejecuta el monitoreo con un tiempo de muestreo de 0.1 ms y envía una trama a la GUI cada 5 ms, que contiene 8 bytes para monitorear cada corriente de fase  $DQ$  del motor, 4 bytes

para la posición actual de cada eslabón y 4 byte de la señal de control PID de cada eslabón. Toda la información recibida por la GUI (posición actual, corriente consumida) se muestra mediante gráficos. Los primeros 15 bytes, que corresponden a la configuración de ganancias, la salida del controlador PID y el número de puntos en la ruta, se almacenan en un banco de registros. Los siguientes bytes de ruta deseados se almacenan en la RAM. El controlador PID comienza a ejecutar la acción de control cuando se reciben todos los puntos de la ruta. La comunicación se transmite a una velocidad de 115200 bits por segundo y 8 bits de datos sin paridad. Se desarrolló un módulo para leer el codificador cuyo muestreo es de 41,6 ns y se almacena cada 0.1 ms en un registro, la resolución del codificador es de 16 bits.

La característica principal del diseño es la modularidad, es decir, si se necesitan agregar más grados de libertad (GDL), es posible clonar el primer GDL hasta que se obtengan los controladores necesarios, al igual que los otros módulos. Internamente, se desarrolló un módulo de pila de Primera entrada, primera salida (FIFO) para controlar el flujo de datos de cada uno de los periféricos, donde cada paquete de datos se compone de un byte de control y dirección del periférico, 2 bytes de direccionamiento y 2 bytes de información, para agregar módulos en el futuro de una manera más rápida y genérica. Los controladores MCP3208 ADC y MCP4921 DAC utilizan un protocolo de interfaz periférica en serie (SPI) independiente para utilizar el procesamiento paralelo.

Para este proyecto, se propone utilizar los recursos DSP de la FPGA ICE40UP5K. La implementación del módulo PID en FPGA se obtiene de acuerdo con la Sección 2.5, ver ecuación (2.27). Por tanto, se utilizan 2 registros a la entrada del error para realizar los retardos de esta señal. Una máquina de estado controla y sincroniza los módulos descritos y un contador es responsable

de realizar los cambios apropiados en la entrada del multiplicador. Para utilizar la señal de control de alta resolución, se agrega un acumulador (registro con sumador) a la salida del bloque DSP. De esta forma la salida se obtiene con mayor sensibilidad y no es necesario cortar la longitud de la señal de control. Por lo tanto, el número y las características de los multiplicadores integrados o segmentos DSP que posee el dispositivo es un aspecto importante a considerar. La mayoría de los FPGA que están disponibles en el mercado tienen multiplicadores con un ancho de bus fijo, por ejemplo, 18 18 bits de longitud. Si se usa un ancho de bus personalizado, entonces la herramienta de síntesis usará recursos lógicos para construir un multiplicador personalizado en lugar de usar los módulos disponibles en hardware. Por lo tanto, la señal de error y las ganancias del controlador deben tener un ancho de bus que coincida con los multiplicadores incorporados y disponibles. En este trabajo, se usaron multiplicadores de 16 16 bits. El error de señal tiene un formato de punto fijo de 16.0 y las ganancias tienen un formato de punto fijo de 8.8. La señal  $e$  representa la señal de error, que es la diferencia entre el punto de referencia y los datos de retroalimentación. De manera similar, las señales  $a_0$ ,  $a_1$  y  $a_2$  son las ganancias del controlador PID, y su valor depende de  $k_p$ ,  $k_i$ ,  $k_d$  y  $T_s$ .

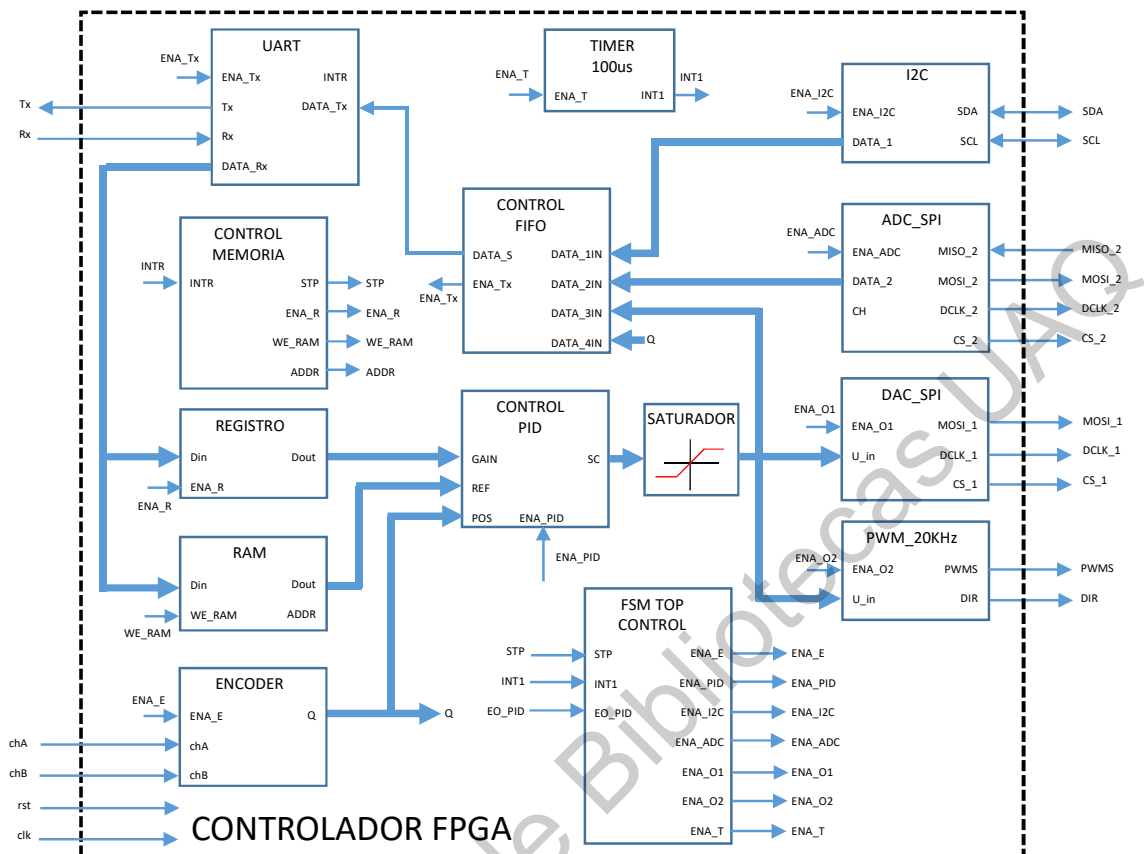


Figura 5.2: Módulos implementados en FPGA.

Por otro lado, la señal  $I_k$  es la salida del filtro, que se envía al actuador. Además,  $CLK$ ,  $RST$ ,  $ST\_PID$  y  $EO\_PID$  denotan las señales de reloj y el reinicio maestro, el inicio del proceso y el fin del proceso, respectivamente. Cabe señalar que las herramientas gratuitas no realizan una inferencia óptima de los bloques DSP contenidos en la FPGA ICE40UP5K. Por esta razón, se utiliza un módulo DSP interno para hacer la estructura PID como se muestra en la figura 5.3 y está configurado para realizar multiplicaciones con signo y acumular en la salida de los multiplicadores; estas sumas no están firmadas. La figura 5.3 presenta los parámetros C1 y C2 que se utilizan para pasar el error del controlador y las ganancias a través de un registro paralelo, y C7 permite el uso de un registro de 32 bits para la salida del multiplicador, llevándolos a un nivel alto. Los

bits C11, C10 y C18, C17 permiten que la parte alta (16 bits más significativos) y la parte baja (16 bits menos significativos) sean la entrada del sumador interno y se configuran tomando C11 y C18 alto y C10 y C17 bajo. Los bits C12 y C19 se mantienen bajos para configurar el bloque DSP como acumulador. Los bits C9, C8 y C16, C15 proporcionan la salida como una salida de registro que tiene la función de acumulador. Estos se configuran tomando los bits C9, C16 bajos y los bits C8 y C15 altos. Los bits C24 y C23 se elevan alto para configurar un multiplicador firmado. Además de la configuración del bloque DSP, la máquina de estado es una parte fundamental para el correcto funcionamiento del controlador PID. Una de las ventajas de las herramientas gratuitas es la posibilidad de configurar de maneras que las herramientas del fabricante no permiten.

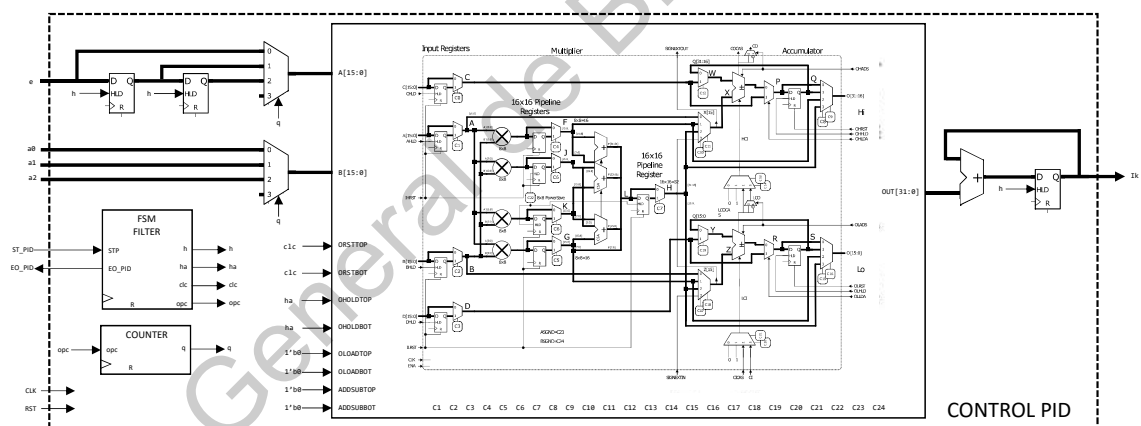


Figura 5.3: Controlador PID implementado en FPGA con base en módulos DSP de ICE40

La GUI desarrollada en Python ejecuta los cálculos para el perfil de velocidad trapezoidal. Esta interfaz es responsable de modificar las ganancias del controlador PID discreto y enviar estos valores  $q_d$ ,  $a_0$ ,  $a_1$  y  $a_2$  a través del puerto serie a la FPGA. El AG también se ejecuta en la GUI. Además, la GUI es responsable de recibir la información de la FPGA, la posición actual del motor,



el error y la señal de control. Estas variables se grafican para observar el comportamiento y el rendimiento tanto del controlador como del robot. Los datos se pueden almacenar por fecha en cada inicio del sistema. La figura 5.4 muestra el esquema de autoajuste y control propuesto.

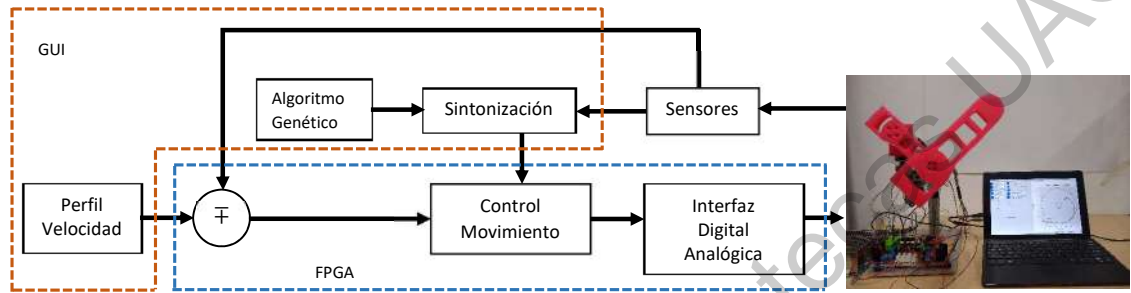


Figura 5.4: Metodología para control y sintonía del robot.

## RESULTADOS

A continuación se muestra el pseudocódigo del AG utilizado para sintonizar el controlador de posición del robot.

```

Configuración de parámetros iniciales;
Definir el tamaño de la población  $P = 10$ ;
Acotar los intervalos de búsqueda  $0 \leq k_p \leq 100$  ;  $0 \leq k_i \leq 200$  ;
 $0 \leq k_d \leq 10$ ;
Creación de la primera generación :  $G[k_p, k_i, k_d]$ ;
Codificar cada elemento de la población creada en un formato 8.8;
while  $J < 20$  do
    Seleccionar P parejas de padres de forma aleatoria ;
    Realice el operador de cruce con los pares generados para obtener
    una población  $G_h$  ;
    Genere una probabilidad de mutación aleatoria  $pm$  ;
    Realice el operador de mutación con la población  $G_h$ ;
    if  $pm < 0.12$  then
        Realizar el operador de mutación con la población  $G_h$ ;
    end
    for para cada elemento  $G$  y  $G_h$  do
        Envíe por USB a FPGA el triple de ganancias  $G[i]$  ;
        Reciba por USB desde la posición FPGA  $\theta$ , error  $e$ , señal de
        control  $i_c$  y acelerómetro ;
        Realice los cálculos de  $IAE$ ,  $ISE$  y  $TV$  ;
        Evalúe las restricciones y la función objetivo  $J$  ;
    end
    Aplicar el operador de selección por elitismo ;
    Genere la nueva población ;
end

```

**Algoritmo 1:** Pseudocódigo del AG.

Los siguientes resultados muestran la respuesta del sistema HIL que se

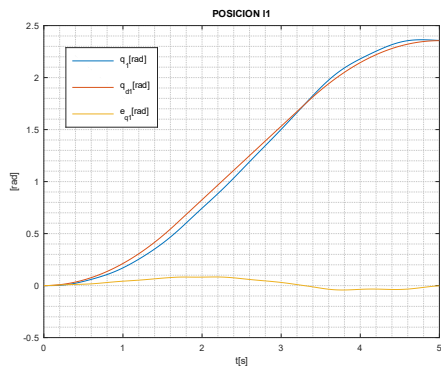
implemento para la simulación del robot manipulador de 2 gdl con PMSM, como actuadores de accionamiento directo.

La figura 6.1a a muestra la posición  $q_1$  y trayectoria deseada del eslabón 1 de HIL, la figura 6.1a a muestra la posición  $q_2$  y trayectoria deseada del eslabón 2 de HIL, donde la trayectoria que sigue el sistema se denota con  $q_{d1}$  y  $q_{d2}$  respectivamente, aplicado al GA para el autoajuste del control de posición.

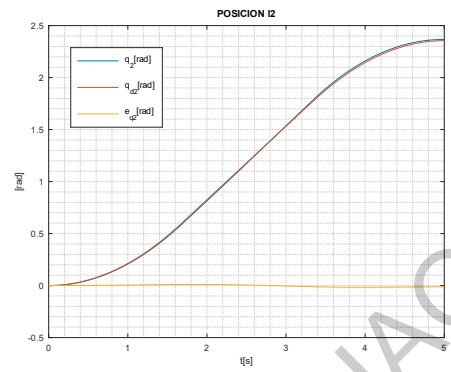
La figura 6.1c y 6.1d muestran las corrientes de la fase q de cada eslabón, respectivamente. Se observa que el controlador no satura esta señal de control en cada caso. La figura 6.1e y 6.1f muestran las corrientes de la fase d de cada eslabón, respectivamente. Se puede apreciar que las variaciones son muy pequeñas y cumple con la condición del diseño de control ( $i_d = 0$ ).

La figura 6.2a muestra los pares aplicados a cada eslabón y se aprecia que de acuerdo a las características de los actuadores, no sobre-pasa los pares máximos y mínimos,  $\pm 15 Nm$  y  $\pm 4 Nm$  de cada eslabón, respectivamente. y la figura 6.2b muestran las velocidades angulares de cada eslabón de HIL, respectivamente.

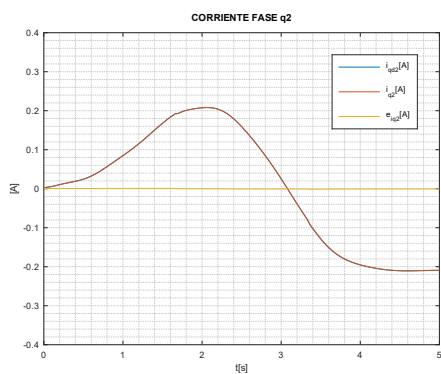
La figura 6.3a muestra la respuesta de la función IAE para cada eslabón de HIL, a través de las diferentes generaciones. La figura 6.3b muestra la respuesta de la función ISE para cada eslabón, a través de las diferentes generaciones. La figura 6.3c muestra la respuesta de la función TV para cada eslabón, a través de las diferentes generaciones. De acuerdo a la configuración propuesta, el número de generaciones es 40; sin embargo, se observa que a partir de la generación 20 se alcanza un valor mínimo constante de la función objetivo y la respuesta del robot es la deseada según la figura 6.3d. Esto se ve reflejado, en la respuesta de cada eslabón, ya que el error del seguimiento de la trayectoria es menos al 3% y la respuesta de la señal de control es suave y no saturada.



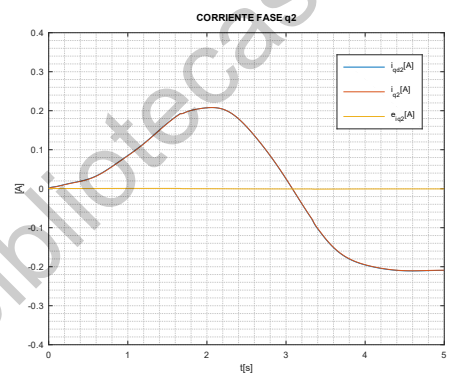
(a)



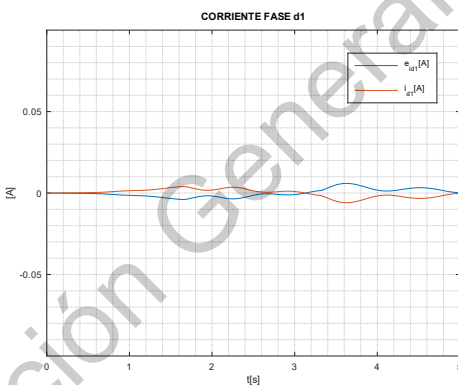
(b)



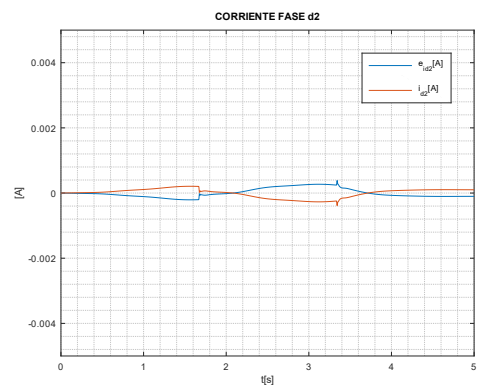
(c)



(d)

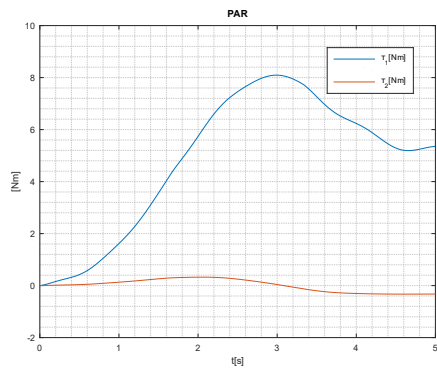


(e)

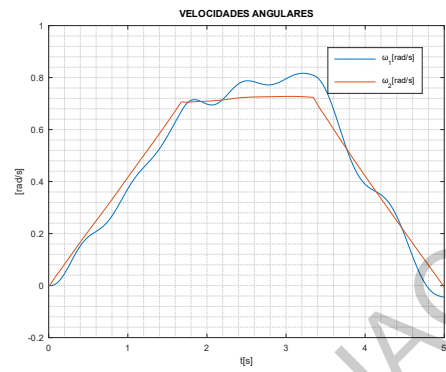


(f)

Figura 6.1: Respuesta de cada eslabón de HIL aplicando el AG para la sintonización y perfil de velocidad (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente de la fase q eslabón 1, (d) corriente de la fase q eslabón 2, (e) corriente de la fase d eslabón 1 (f) corriente de la fase d eslabón 2.

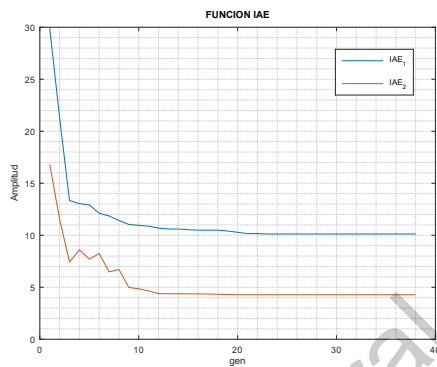


(a)

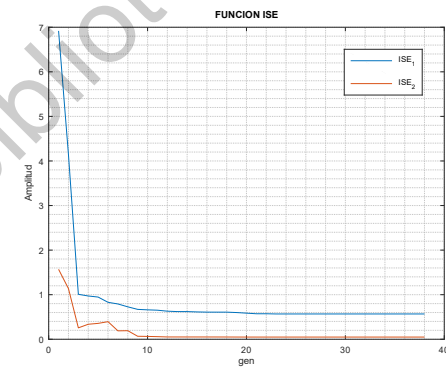


(b)

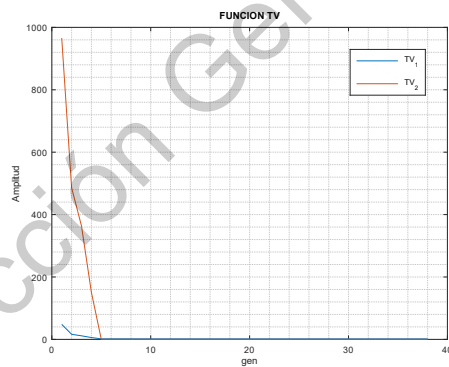
Figura 6.2: Respuesta del HIL aplicando el AG para la sintonización y perfil de velocidad (a) par aplicado a cada eslabón, (b) velocidad medida en cada eslabón.



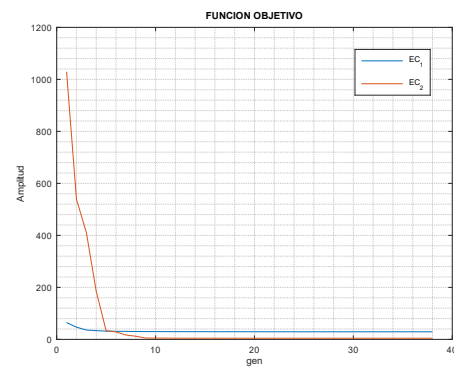
(a)



(b)



(c)

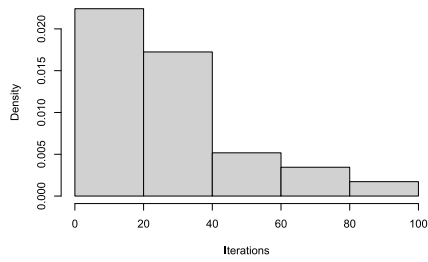


(d)

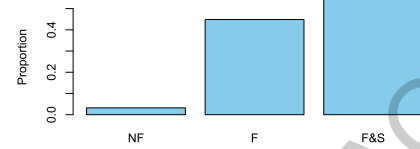
Figura 6.3: Respuesta de función objetivo para cada eslabón de HIL, (a) IAE, (b) ISE, (c) TV, (d) Función objetivo.

Durante el experimento, se realizaron 15 réplicas, que consisten en la estimación de los parámetros  $a_0$ ,  $a_1$  y  $a_2$  del controlador PID discreto utilizando el GA propuesto. Las réplicas registran un cierto número de iteraciones antes de que el algoritmo se detenga debido a la condición de parada establecida. Los valores alcanzaron la función objetivo y las restricciones planteadas así como un indicador de si se encontró una solución deseable, factible o no. Se desea lograr un tiempo de fraguado inferior a 200 ms y con un rebasamiento inferior al 5%. Primero, se realizó un análisis estadístico de la finalización del algoritmo. La figura 6.4 a representa la distribución del número de iteraciones que se requieren antes de que el algoritmo se detenga por la magnitud de la función objetivo menor que 70, según la literatura. Debido a que hay un sesgo hacia la derecha, la mediana es una herramienta estadística útil para describir la centralidad de las observaciones, que se puede interpretar como el número de iteraciones necesarias para la convergencia del algoritmo.

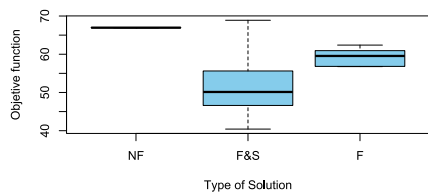
En este sentido, se encontró que la mediana de las iteraciones necesarias fue de 26 iteraciones y, usando Bootstrap, el intervalo de confianza asintótico normal se estimó en el 95% de este estadístico; resultó que el número promedio de iteraciones requeridas para la convergencia fue de entre 19 y 33 iteraciones (el intervalo exacto fue de 19,29 a 32,71). Otra observación interesante es que, incluso si el algoritmo se detiene debido a una condición de parada establecida, esto no garantiza en ningún momento que se encontrará un resultado factible o deseable. Además, se encontró que el algoritmo no encontró una solución viable 3.22% de las veces, 44.84% se encontró una solución factible pero no deseada, y 54.84% de las veces, se encontró una solución posible y deseada. alcanzado (figura 6.4 b). De acuerdo con las réplicas, los intervalos de confianza de 95% se calcularon para el evento donde se encontró una solución al menos factible y



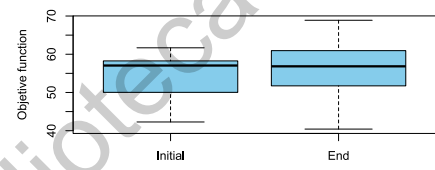
(a)



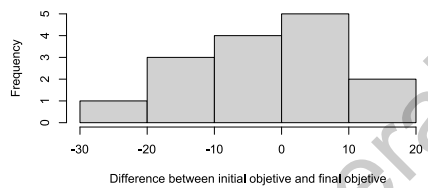
(b)



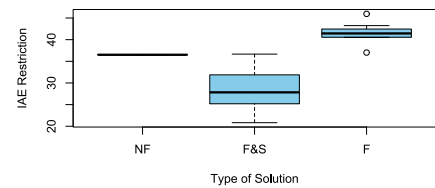
(c)



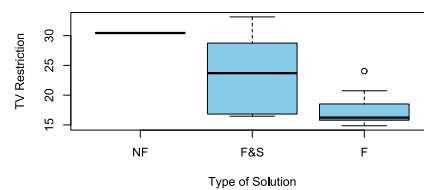
(d)



(e)



(f)



(g)

Figura 6.4: Análisis estadístico del AG propuesta la distribución (a) distribución del número de iteraciones, (b) análisis de solución no factible (NF), factible (F) y solución factible y adecuada (F&S), (c) desviación estándar estimada del objetivo alcanzado, (d) valores obtenidos de la función objetivo, (e) cambio de la función objetivo, (f) cambio de la restricción de Error Absoluto Integral (IAE) y (g) cambio de la restricción de TV.

otra para el evento, donde se encontró una solución alcanzable y deseada. De esta manera, el algoritmo obtiene una confianza del 95 % y la solución deseada entre el 36.30 % y el 72.22 % del tiempo, y al menos una solución factible entre el 81.49 % y el 99.83 % del tiempo.

Respecto a las funciones objetivo, la figura 6.4 c muestra que el conjunto de soluciones factibles y deseables logra objetivos altamente variables, ya que la desviación estándar estimada del objetivo logrado en este tipo de solución es de 9.3468 unidades. Mientras que los otros tipos de soluciones tienen variaciones menores, además, la figura 6.4 c muestra que el cubo de soluciones que limita el conjunto de soluciones factibles y deseables contiene a su manera solo soluciones posibles y soluciones no viables. Además, la figura 6.4 d resalta los valores que alcanzó la función objetivo en la primera generación del algoritmo y la última edad del algoritmo. Se puede observar que las soluciones del algoritmo son más variables al final de su ejecución y que el rango del objetivo inicial está contenido en el objetivo final, lo que puede ser una explicación de la rápida convergencia del algoritmo para este caso particular. En este sentido, la figura 6.4 e muestra el cambio entre el valor de la función objetivo al inicio del algoritmo y al final del mismo, donde se puede mostrar que la mayoría de los cambios se concentran alrededor de 0. Por lo tanto, cuando se aplica una prueba t pareada para contrastar la hipótesis de que estadísticamente, el algoritmo no genera el valor de la función objetivo con el supuesto de que si hay un cambio estadísticamente significativo, se encontró que el cambio medio estimado fue  $-1.8288$  unidades; pero no hay suficiente evidencia estadística para descartar la hipótesis de que estadísticamente el algoritmo no genera el valor de la función objetivo (donde el estadístico t calculado fue  $-0.6733$  con 14 grados de libertad y un valor  $p$  de 0,5117). Sin embargo, esto no se repitió en ambas restricciones.



La figura 6.4 f muestra que los rangos de la restricción IAE alcanzados por cada objetivo son mutuamente excluyentes. Al mismo tiempo, la figura 6.4 g muestra que el comportamiento observado por la restricción TV fue similar al observado en la función objetivo, donde también se observa que el cubo de soluciones que limita el conjunto de factibles y Las soluciones deseables contienen a su manera las únicas soluciones posibles y las no viables. No obstante, como en el caso de la función objetivo, no se encontró que hubiera un cambio estadísticamente significativo en las restricciones antes y después del algoritmo. En el caso de la restricción IAE, se estimó un desplazamiento medio en  $-1.9892$  con un estadístico  $t$  asociado de  $-0.6256$  de 14 grados de libertad con un valor  $p$  de  $0.5417$ ; y, para el caso de la restricción de TV, se estimó un cambio medio de  $0,1603393$  con un estadístico  $t$  asociado de  $0,0618$  de 14 grados de libertad y un valor de  $p$  de  $0,9516$ .

El algoritmo no encontró una solución factible el 3,22% de las veces, como se muestra en el análisis estadístico antes mencionado. En caso de que el algoritmo no converja, el resultado se manifiesta visualmente, ya que tanto el gran consumo de energía como las vibraciones presentes en el sistema. Por lo tanto, es posible implementar una función de parada de emergencia en la GUI por seguridad como en cualquier sistema de control.

En la figura 6.5 se muestra el robot prototipo que se contruyó para la realización de las pruebas y aplicación de técnicas de control.

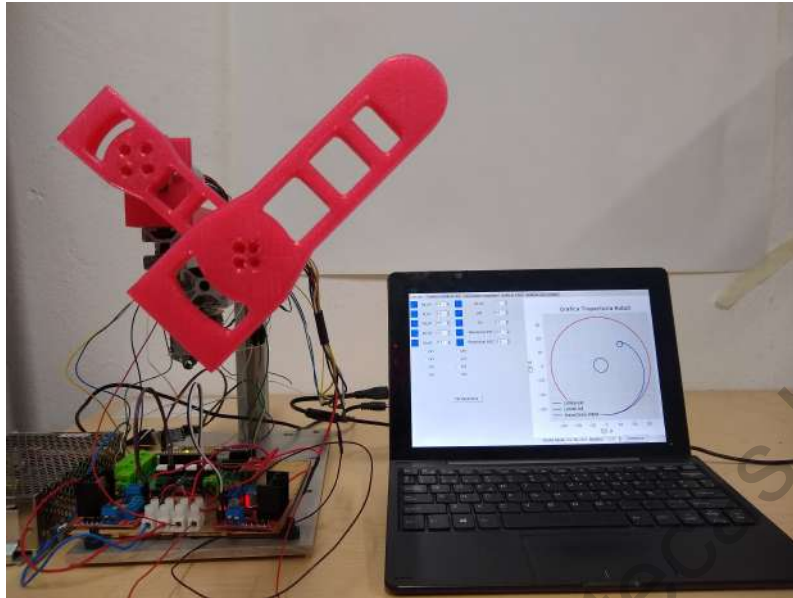
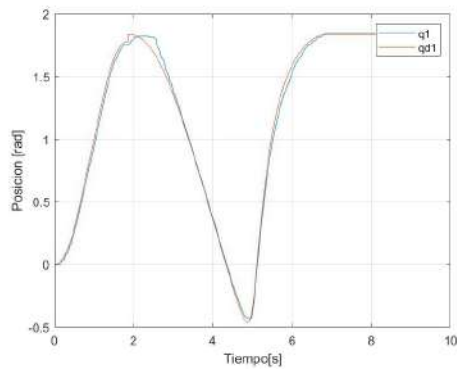


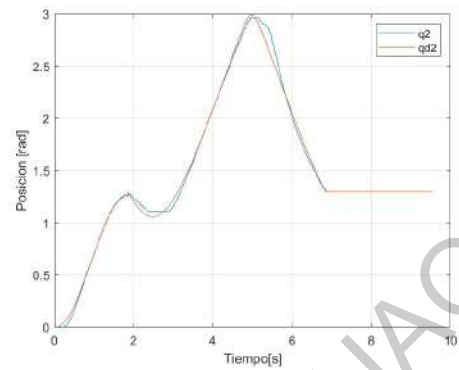
Figura 6.5: Banco de pruebas utilizado para aplicación de técnicas de control.

Los resultados mostrados a continuación, se aplicaron sobre la plataforma construida del robot de 2 gdl.

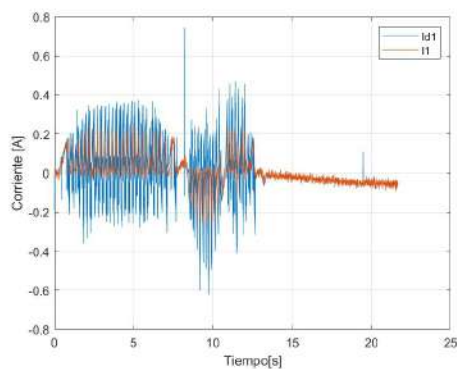
En la figura 6.6 muestra la respuesta del robot sin aplicar el algoritmo genético para la sintonización del controlador sobre la plataforma del robot de 2 gdl construido. Puede observarse que tiene errores para el seguimiento de trayectorias y las no linealidades del sistema como el efecto de la caja de engranes repercuten en el rendimiento de este.



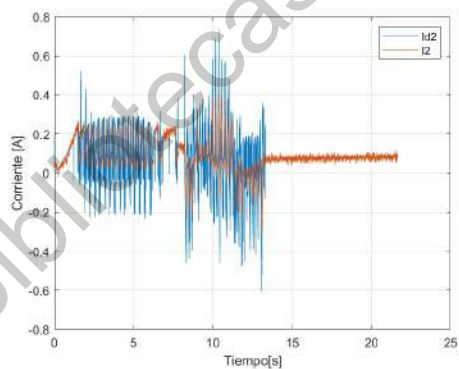
(a)



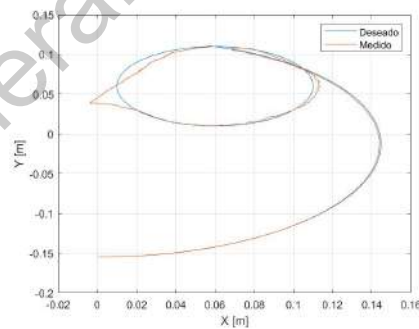
(b)



(c)



(d)



(e)

Figura 6.6: Respuesta del robot sin aplicar el AG para la sintonización (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente del actuador 1 (d) corriente del actuador 2 (e) trayectoria realizada.

En la figura 6.7 se muestra la respuesta del sistema cuando se aplicó el AG para la sintonización del controlador. Se puede observar que se tiene un

mejor seguimiento de la trayectoria y los consumos de corriente de cada eslabón son menores en comparación con la figura 6.6.

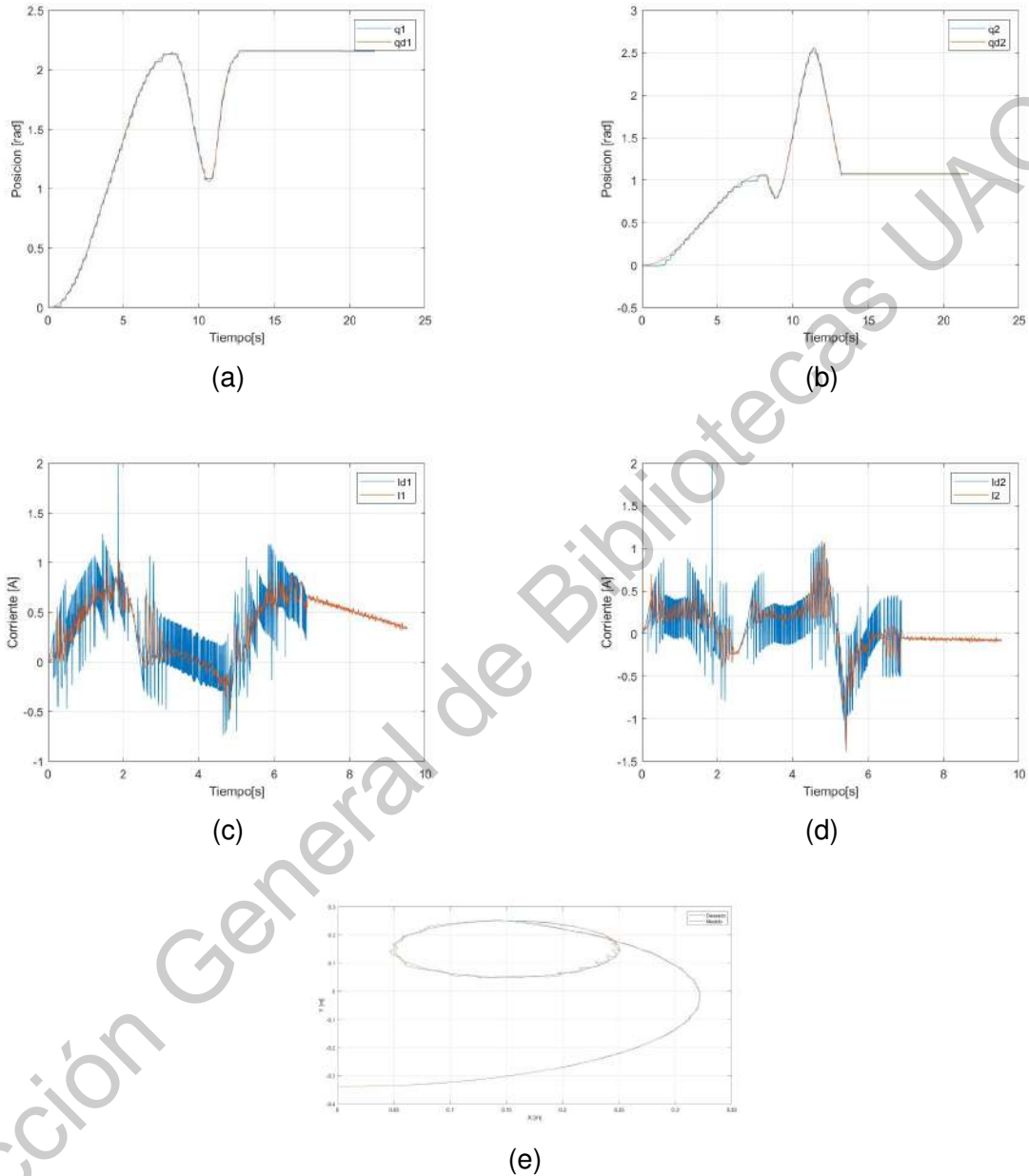


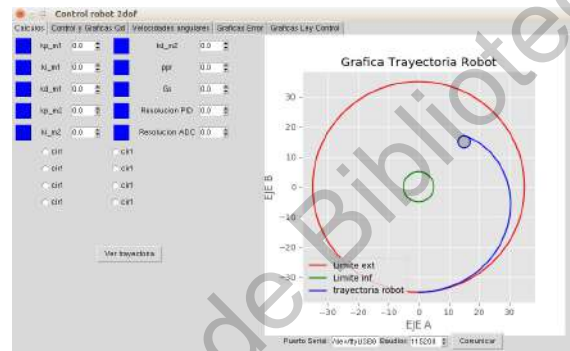
Figura 6.7: Respuesta del robot aplicando el AG para la sintonización (a) posición eslabón 1, (b) posición eslabón 2, (c) corriente del actuador 1 (d) corriente del actuador 2 (e) trayectoria realizada.

En la Tabla 6.1 se muestran los recursos lógico utilizados para la implementación del controlador propuesto (dos lazos de corriente y uno de posición

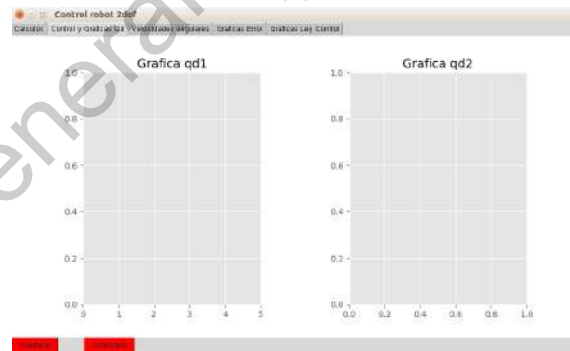
para cada uno de los eslabones) en FPGA. A demás se muestran los recursos utilizados en la literatura en controladores similares.

Tabla 6.1: Revisión literaria de recursos lógicos y dispositivos FPGA empleandos en controladores similares al propuesto.

Trabajo	Fabricante	Dispositivo	Flipflops	LUTs	Multiplicadores/DSP	SPRAM256K	HFOsc	Minimum Period
Propuesto	Lattice	ICE40UP5k	1514/5280	1221/5280	8/8	4/4	1	37.2 ns
Martínez-Prado et al. (2018)	Xilinx	Spartan-3	2998/15,360	3792/15,360	24/24	-	-	12.174 ns
Morales-Velazquez et al. (2010)	Xilinx	Spartan-3	-	-	-	-	-	-
Jokić et al. (2020)	Altera	Cyclone II DE2	1788	998	35/70	-	-	-
Ordóñez Cerezo et al. (2019)	Lattice	ICE40hx4k	-	-	-	-	-	-
García-Martínez et al. (2017)	Xilinx	Zynq XC7Z01	-	-	-	-	-	-
Ponce et al. (2015)	Xilinx	Spartan-3	-	-	-	-	-	-
Ricci and Meacci (2018)	Altera	Max10	3079	5017	6	-	-	-
García-Martínez et al. (2019)	Xilinx	Zynq 7	-	-	-	-	-	-



(a)



(b)

Figura 6.8: Vista principal de la GUI desarrollada en PYTHON.

En la figura 6.8 se muestra la GUI desarrollada con software libre en lenguaje PYTHON, tiene como finalidad modificar las ganancias del controlador, visulaizar información de los sensores de posición y corriente de cada eslabón.

Se desarrolló el sistema eléctrico para acondicionamiento de las señales de los sensores y las señales de la etapa de potencia como se muestra en la figura 6.9.

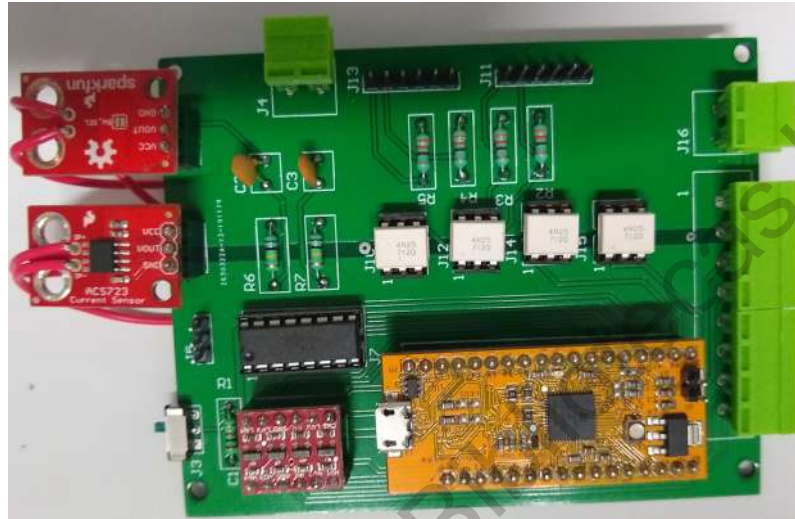


Figura 6.9: Tarjeta electrónica desarrollada para acondicionamiento de señales.

# CONCLUSIONES

En la presente investigación se desarrolló un sistema de control empleado para el seguimiento de trayectorias de un robot manipulador de dos grados de libertad, con base en herramientas de software y hardware de código abierto. El trabajo muestra el desarrollo y análisis completo del modelo del robot. Se presenta el modelo dinámico del motor PMSM como actuador directo de cada eslabon. Se presenta el modleo completo (Robot y actuadores) para la implementación del sistema HIL. Además, se desarrolló una metodología para el ajuste automático del controlador PID empleado para el seguimiento eficiente de la trayectoria. El sistema de control se implementó en un FPGA ICE40UP5k de bajo costo de la familia Lattice. Todos los módulos propuestos son portables para cualquier fabricante de FPGA. Según la Tabla 2.2, se hizo un recuento de las aplicaciones de control donde se realizó una actualización del sistema de control, mostrando que este es uno de los primeros trabajos donde se utilizaron herramientas de código abierto para programar FPGA.

De acuerdo a la Tabla 6.1, también se observa que las herramientas de código abierto tienen un mayor rendimiento, debido a la distribución y gestión de recursos lógicos. Adicionalmente, tiene la ventaja de ser multiplataforma, pudiendo ejecutarse en sistemas operativos Linux, Windows o MAC OS. Se desarrolló una GUI de código libre en Python para monitorear las variables de cada sen-

sor de cada eslabón. Esta GUI también es responsable tanto de configurar las ganancias a través de la comunicación en serie como de calcular los perfiles de velocidad trapezoidal. El algoritmo de cálculo de perfil se puede reemplazar fácilmente para probar diferentes perfiles de velocidad. El costo del tablero de control es de alrededor de \$35 dólares. Las figuras 6.1, 6.2, 6.3 y 6.7 muestran que el controlador y algoritmo de sintonización, es capaz de seguir cualquier trayectoria con un error menor al 3% y una respuesta suave de la señal de control. Con esta evidencia queda aceptada la hipótesis de este trabajo. Como trabajo futuro se pretende agregar acelerómetros a cada eslabón del robot manipulador con la finalidad de ofrecer una alternativa para detección de fallas y monitoreo de vibraciones y ofrecer la función de mantenimiento predictivo.



---

## Referencias

- Alvarez-Gonzalez, F., Griffo, A., Sen, B., and Wang, J. (2017). Real-time hardware-in-the-loop simulation of permanent-magnet synchronous motor drives under stator faults. *IEEE Transactions on Industrial Electronics*, 64(9):6960–6969.
- Bandyopadhyay, R., Chakraborty, U. K., and Patranabis, D. (2001). Autotuning a pid controller: A fuzzy-genetic approach. *Journal of Systems Architecture*, 47(7):663–673.
- Belloni, F., Chiumeo, R., Gandolfi, C., and Villa, A. (2017). Performance test of a pq universal compensator through control hardware in the loop simulation. In *2017 6th International Conference on Clean Electrical Power (ICCEP)*, pages 502–508. IEEE.
- Borangiu, T., Anton, F., and Anton, S. (2010). Open architecture for robot controllers. In *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, pages 181–186. IEEE.
- Brant, A. and Lemieux, G. G. (2012). Zuma: An open fpga overlay architecture. In *2012 IEEE 20th international symposium on field-programmable custom computing machines*, pages 93–96. IEEE.
- Concha Sánchez, A., Figueroa-Rodríguez, J. F., Fuentes-Covarrubias, A. G., Fuentes-Covarrubias, R., and Gadi, S. K. (2020). Recycling and updating an educational robot manipulator with open-hardware-architecture. *Sensors*, 20(6):1694.

- Correa, J. E., Toombs, N., and Ferreira, P. M. (2017). A modular-architecture controller for cnc systems based on open-source electronics. *Journal of Manufacturing Systems*, 44:317–323.
- Craig, J. J. (1977). *John Craig.*, volume 1.
- Emre, S., Haoyong, Y., and Kouhei, O. (2015). A practical tuning method for the robust pid controller with velocity feed-back.
- Fang, Y., Hu, J., Liu, W., Shao, Q., Qi, J., and Peng, Y. (2019). Smooth and time-optimal s-curve trajectory planning for automated robots and machines. *Mechanism and Machine Theory*, 137:127–153.
- Flores-Morán, E., Yáñez-Pazmiño, W., and Barzola-Monteses, J. (2018). Genetic algorithm and fuzzy self-tuning pid for dc motor position controllers. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 162–168. IEEE.
- Fu, K. S., Gonzalez, R. C., and Lee, C. G. (1987). Control, sensing, vision and intelligence. *Robotics*, McGraw, New York.
- Garcia-Valdovinos, L. G., Velarde-Sanchez, A., Sosa-Cruz, R., and Garcia-Zarco, L. A. (2016). Open software/hardware architecture for advanced force/position control of an industrial robot. In *2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6. IEEE.
- García-Martínez, J. R., Reséndiz, J. R., Prado, M. Á. M., and Miguel, E. E. C. (2017). Assessment of jerk performance s-curve and trapezoidal velocity profiles. In *2017 XIII International Engineering Congress (CONIIN)*, pages 1–7. IEEE.

- García-Martínez, J. R., Rodríguez-Reséndiz, J., and Cruz-Miguel, E. E. (2019). A new seven-segment profile algorithm for an open source architecture in a hybrid electronic platform. *Electronics*, 8(6).
- Ghorbani, M., Pasand, M., Bayati, A. G., and Baheri, N. (2018). Real-time hardware-in-the-loop test for a small upper stage embedded control system. In *2018 Real-Time and Embedded Systems and Technologies (RTEST)*, pages 96–103. IEEE.
- Heo, H.-J., Son, Y., and Kim, J.-M. (2019). A trapezoidal velocity profile generator for position control using a feedback strategy. *Energies*, 12(7):1222.
- Hernández-Guzmán, V. M., Silva-Ortigoza, R., and Carrillo-Serrano, R. V. (2013). *Control Automático: Teoría de diseño, construcción de prototipos, modelado, identificación y pruebas experimentales*. Colección CIDETEC.
- Jimenez-Fernandez, A., Jimenez-Moreno, G., Linares-Barranco, A., Dominguez-Morales, M. J., Paz-Vicente, R., and Civit-Balcells, A. (2012). A neuro-inspired spike-based pid motor controller for multi-motor robots with low cost fpgas. *Sensors*, 12(4):3831–3856.
- Jokić, D., Lubura, S., Rajs, V., Bodić, M., and Šiljak, H. (2020). Two open solutions for industrial robot control: The case of puma 560. *Electronics*, 9(6):972.
- Kain, S., Schiller, F., and Dominka, S. (2011). Methodology for reusing real-time hil simulation models in the commissioning and operation phase of industrial production plants. *Edited by Ganesh R. Naik*, page 143.
- Kelly, R. and Santibáñez, V. (2003). *Control de movimiento de robots manipuladores*. Pearson educación.

- Kumar, A., Mittal, A., Arya, R., Shah, A., Garg, S., and Kumar, R. (2017). Hardware in the loop based simulation of a robotic system with real time control and animation of working model. In *2017 International Conference on Inventive Systems and Control (ICISC)*, pages 1–5. IEEE.
- Martínez-Prado, M.-A., Rodríguez-Reséndiz, J., Gómez-Loenzo, R.-A., Herrera-Ruiz, G., and Franco-Gasca, L.-A. (2018). An fpga-based open architecture industrial robot controller. *IEEE Access*, 6:13407–13417.
- Monmasson, E. and Cirstea, M. N. (2007). Fpga design methodology for industrial control systems—a review. *IEEE transactions on industrial electronics*, 54(4):1824–1842.
- Morales-Velazquez, L., de Jesus Romero-Troncoso, R., Osornio-Rios, R. A., Herrera-Ruiz, G., and Cabal-Yeppez, E. (2010). Open-architecture system based on a reconfigurable hardware–software multi-agent platform for cnc machines. *Journal of Systems Architecture*, 56(9):407–418.
- Mu, S., Goto, S., Shibata, S., and Yamamoto, T. (2019). Intelligent position control for pneumatic servo system based on predictive fuzzy control. *Computers & Electrical Engineering*, 75:112–122.
- Nicola, M. and Nicola, C.-I. (2020). Sensorless fractional order control of pmsm based on synergetic and sliding mode controllers. *Electronics*, 9(9):1494.
- Ordóñez Cerezo, J., Castillo Morales, E., and Cañas Plaza, J. M. (2019). Control system in open-source fpga for a self-balancing robot. *Electronics*, 8(2):198.
- Paiz, C., Pohl, C., Radkowski, R., Hagemeyer, J., Pormann, M., and Rückert, U. (2009). Fpga-in-the-loop-simulations for dynamically reconfigurable applications. In *2009 International Conference on Field-Programmable Technology*, pages 372–375. IEEE.

- Ponce, P., Molina, A., Tello, G., Ibarra, L., MacCleery, B., and Ramirez, M. (2015). Experimental study for fpga pid position controller in cnc micro-machines. *IFAC-PapersOnLine*, 48(3):2203–2207.
- Pritschow, G., Altintas, Y., Jovane, F., Koren, Y., Mitsuishi, M., Takata, S., Van Brussel, H., Weck, M., and Yamazaki, K. (2001). Open controller architecture—past, present and future. *CIRP Annals*, 50(2):463–470.
- Reyes, F. (2011). *Robótica-Control de robots manipuladores*. Alfaomega grupo editor.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., and Martínez, M. (2013). Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo pid: Estado actual y perspectivas. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 10(3):251–268.
- Ricci, S. and Meacci, V. (2018). Simple torque control method for hybrid stepper motors implemented in fpga. *Electronics*, 7(10):242.
- Rodríguez, B., González, F., Naya, M. Á., and Cuadrado, J. (2020). Assessment of methods for the real-time simulation of electronic and thermal circuits. *Energies*, 13(6):1354.
- Romanov, A., Romanov, M., and Kharchenko, A. (2017). Fpga-based control system reconfiguration using open source software. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EICon-Rus)*, pages 976–981. IEEE.
- Sciavicco, L. and Siciliano, B. (2000). *Modelling and control of robot manipulators*. pringer-verlag. New York.

- Shu, T., Gharaaty, S., Xie, W., Joubair, A., and Bonev, I. A. (2018). Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor. *IEEE/ASME Transactions on Mechatronics*, 23(3):1159–1170.
- Wolf, C. and Lasser, M. (2015). Project icestorm. *URI: <http://www.clifford.at/icestorm>*.
- Xin, W., Ran, L., Yanghua, W., Yong, P., and Bin, Q. (2013). Self-tuning pid controller with variable parameters based on particle swarm optimization. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*, pages 1264–1267. IEEE.
- Yang, X., Chen, X., Xia, R., and Qian, Z. (2018). Wireless sensor network congestion control based on standard particle swarm optimization and single neuron pid. *Sensors*, 18(4):1265.
- Yasuda, K. (2017). The future vision of industrial robot. In *2017 29th International Symposium on Power Semiconductor Devices and IC's (ISPSD)*, pages 7–12. IEEE.
- Yu, H., Lee, H., Lee, S., Kim, Y., and Lee, H.-M. (2018). Recent advances in fpga reverse engineering. *Electronics*, 7(10):246.
- Zhang, S., Wang, S., Jing, F., and Tan, M. (2019). A sensorless hand guiding scheme based on model identification and control for industrial robot. *IEEE Transactions on Industrial Informatics*, 15(9):5204–5213.
- Zhao, H. and Mai, S. (2013). Simulation of asic/fpga cochlear implant speech processor with simulink and modelsim cosimulation method. In *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, pages 1–4. IEEE.

**ANEXOS PRODUCTOS DEL  
TRABAJO DE INVESTIGACIÓN**

Dirección General de Bibliotecas UAQ

## **A. Artículo indizado**

Cruz-Miguel, E. E., Rodríguez-Reséndiz, J., García-Martínez, J. R., Camarillo-Gómez, K. A., Pérez-Soto, G. I. (2019). Field-programmable gate array-based laboratory oriented to control theory courses. *Computer Applications in Engineering Education*, 27(5), 1253-1266.

Dirección General de Bibliotecas UAQ





## Field-programmable gate array-based laboratory oriented to control theory courses

Edson E. Cruz-Miguel<sup>1</sup> | Juvenal Rodríguez-Reséndiz<sup>1</sup> | José R. García-Martínez<sup>1</sup> |  
Karla A. Camarillo-Gómez<sup>2</sup> | Gerardo I. Pérez-Soto<sup>1</sup>

<sup>1</sup>Facultad de Ingeniería,  
Universidad Autónoma de Querétaro,  
Querétaro, Mexico

<sup>2</sup>Department of Mechanical Engineering,  
Tecnológico Nacional de México en  
Celaya, Guanajuato, Mexico

### Correspondence

Juvenal Rodríguez-Reséndiz, Facultad de  
Ingeniería, Universidad Autónoma de  
Querétaro, Querétaro, Mexico.  
Email: juvenal@uaq.edu.mx

### Funding information

Consejo Nacional de Ciencia y  
Tecnología; PRODEP

### Abstract

This article presents a methodology to improve the performance of learning of students in the major of Automation Engineering, through the development of a multidisciplinary project that involves the use of contemporary technological tools and knowledge of previous subjects. It is sought to add in their educational training, theory and practice for the student, learning, new knowledge, and skills within their training as an engineer. Despite advances in teaching methods and the inclusion of contemporary tools in the learning process, the success rate is low. A qualitative and quantitative analysis of student learning is made by integrating a multidisciplinary project based on contemporary tools compared with other reviewed works. This laboratory based on a field-programmable gate array helps the student to acquire skills related to the implementation of reconfigurable logic and instrumentation applied to control theory. The development of the final project involves the design of a graphic user interface, hardware description, communication protocols, and implementation of motion control, it is looking for is that the student can relate theory to practice in order to motivate and increase skills and knowledge, as well as solving both didactic and industrial problems. The results obtained show that the students are satisfied with the course and that the experience with the integration of this type of multidisciplinary projects directly impacts the motivation and learning of the student, considerably increased developed based on the specific indicators that the Querétaro State University has accredited by Accreditation Board for Engineering and Technology since 2016.

### KEYWORDS

ABET, assessment student, embedded system, FPGA, motion control

## 1 | INTRODUCTION

Universities must train students to add value to society by training them in scientific and

complementary disciplines—such as commitment to their country, responsibility, and ethics—and promote the development of skills, attitudes, and qualities in them.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. *Computer Applications in Engineering Education* published by Wiley Periodicals, Inc.

*Comput Appl Eng Educ.* 2019;27:1253–1266.

wileyonlinelibrary.com/journal/cae | 1253

## **B. Artículo indizado**

Cruz-Miguel, E. E., García-Martínez, J. R., Rodríguez-Reséndiz, J., Carrillo-Serrano, R. V. (2020). A New Methodology for a Retrofitted Self-tuned Controller with Open-Source FPGA. *Sensors*, 20(21), 6155.

Dirección General de Bibliotecas UAQ

Article

## A New Methodology for a Retrofitted Self-tuned Controller with Open-Source FPGA

Edson E. Cruz-Miguel , José R. García-Martínez , Juvenal Rodríguez-Reséndiz \* ,  
and Roberto V. Carrillo-Serrano 

División de Investigación y Posgrado, Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro 76010, Mexico; ecruz30@alumnos.uaq.mx (E.E.C.-M.); jose.gm@uaq.mx (J.R.G.-M.); roberto.carrillo@uaq.mx (R.V.C.-S.)

\* Correspondence: juvenal@uaq.edu.mx; Tel.: +52-442-192-1200

Received: 27 August 2020; Accepted: 24 October 2020; Published: 29 October 2020



**Abstract:** Servo systems are feedback control systems characterized by position, speed, and/or acceleration outputs. Nowadays, industrial advances make the electronic stages in these systems obsolete compared to the mechanical elements, which generates a recurring problem in technological, commercial and industrial applications. This article presents a methodology for the development of an open-architecture controller that is based on reconfigurable hardware under the open source concept for servo applications. The most outstanding contribution of this paper is the implementation of a Genetic Algorithm for online self tuning with a focus on both high-quality servo control and reduction of vibrations during the positioning of a linear motion system. The proposed techniques have been validated on a real platform and form a novel, effective approach as compared to the conventional tuning methods that employ empirical or analytical solutions and cannot improve their parameter set. The controller was elaborated from the Graphical User Interface to the logical implementation while using free tools. This approach also allows for modification and updates to be made easily, thereby reducing the susceptibility to obsolescence. A comparison of the logical implementation with the manufacturer software was also conducted in order to test the performance of free tools in FPGAs. The Graphical User Interface developed in Python presents features, such as speed profiling, controller auto-tuning, measurement of main parameters, and monitoring of servo system vibrations.

**Keywords:** retrofitted; open-hardware FPGA; vibration analysis; genetic algorithm; instrumentation and sensors; adaptive and predictive control; controller self-tuning

### 1. Introduction

In the last years, the demand for motion controllers has increased in industrial applications, such as robot manipulators, computerized numerical control (CNC) machines and frequency converters [1,2]. In this context, efforts have been made by researchers to improve the performance of these control systems. However, most control systems present limitations to modify or replace control algorithms [3]. These limitations are presented by the control system architecture that can be classified as: (1) closed architecture, where it is not possible to access the control algorithm and communication protocols, (2) hybrid architecture, where the control strategies cannot be modified, but it is possible to add devices to the system, and (3) the open architecture, where both hardware and software and control strategies can be modified [2,4]. Unlike the mechanical system, the electrical and software stages become obsolete in less time, which represents an area of opportunity in servo systems [3]. The Field Programmable Gate Arrays (FPGAs) are devices with a large number of programmable logic elements, such as composite, flip-flops, and wires, and use hardware description language (HDL). Both the




### **C. Artículo indizado, Co-Autor**

García-Martínez, J. R., Rodríguez-Reséndiz, J., Cruz-Miguel, E. E. (2019).  
A New Seven-Segment Profile Algorithm for an Open Source Architecture in a  
Hybrid Electronic Platform. *Electronics*, 8(6), 652.

Dirección General de Bibliotecas UAQ

Article

## A New Seven-Segment Profile Algorithm for an Open Source Architecture in a Hybrid Electronic Platform

José R. García-Martínez <sup>†</sup>, Juvenal Rodríguez-Reséndiz <sup>\*†</sup> and Edson E. Cruz-Miguel <sup>†</sup>

División de Investigación y Posgrado, Facultad de Ingeniería, Universidad Autónoma de Querétaro (UAQ), Cerro de las Campanas, S/N, Col. Las Campanas, Querétaro C.P. 76010, Mexico; jose.gm@uaq.mx (J.R.G.-M.); ecruz30@alumnos.uaq.mx (E.E.C.-M.)

\* Correspondence: juvenal@uaq.edu.mx; Tel.: +52-442-192-1200

† These authors contributed equally to this work.

Received: 28 April 2019; Accepted: 6 June 2019; Published: 10 June 2019



**Abstract:** The velocity profiles are used in the design of trajectories in motion control systems. It is necessary to design smoother movements to avoid high stress in the motor. In this paper, the rate of change in acceleration value is used to develop an S-curve velocity profile which presents an acceleration and deceleration stage smoother than the trapezoidal velocity profile reducing the error at the end of the duty-cycle pre-established in one degree of freedom (DoF) application. Furthermore, a new methodology is developed to generate a seven-segment profile that works with negative velocity and displacement constraints applying an open source architecture in a hybrid electronic platform compounded by a system on a chip (SoC) Raspberry Pi 3 and a field programmable gate array (FPGA). The performance of the motion controller is measured through the comparison of the error obtained in real-time application with a trapezoidal velocity profile. As a result, a low-cost platform and an open architecture system are achieved.

**Keywords:** low-cost platform; FPGA; S-curve; motion control; robotics; SoC

### 1. Introduction

The velocity profiles have been studied broadly in recent years to design point-to-point trajectories in robot manipulators, conveyor belts, computer numerical control (CNC) machinery or whatever system with the use of direct current (DC) and alternating current (AC) motors [1,2]. Velocity profiles have an essential role in motion control since it is possible to accomplish a target position reducing the vibrations and the energy consumption, increasing the precision and the durability of the systems [3,4]. Nowadays, a great variety of velocity profiles exist, but their accuracy depends on the velocity's demeanor. Since, if the velocity changes abruptly, the behavior of acceleration could cause discontinuities in the trajectory [5]. The rate of change in acceleration is denominated as jerk [6], whether acceleration changes too fast, the vibrations increase their frequency causing damage to the structure of the mechanical system [7]. Therefore, when something like this happens, it is possible to deduce that the jerk value is too big, which means that the energy consumption is high.

There are different velocity profiles applied to specific processes, the common ones are the triangular velocity profile, parabolic velocity profile and the trapezoidal velocity profile [8]. The triangular velocity profile is a piece-wise defined function given by two linear segments corresponding to the acceleration and deceleration of the actuator, the jerk value is high because of the acceleration changes radically. On the other hand, the parabolic velocity profile presents a smoother velocity curve than the triangular, making the jerk value less than triangular profile but neither of them maintain a velocity constant phase [9,10], it means that they accelerate and decelerate the actuator immediately. The trapezoidal velocity profile consists of three phases: acceleration,







## **D. Artículo indizado, Co-Autor**

García-Martínez, J. R., Cruz-Miguel, E. E., Carrillo-Serrano, R. V., Mendoza-Mondragón, F., Toledano-Ayala, M., Rodríguez-Reséndiz, J. (2020). A PID-Type Fuzzy Logic Controller-Based Approach for Motion Control Applications. *Sensors*, 20(18), 5323.

Dirección General de Bibliotecas UAQ

Article

## A PID-Type Fuzzy Logic Controller-Based Approach for Motion Control Applications

José R. García-Martínez <sup>1,†</sup> , Edson E. Cruz-Miguel <sup>1,†</sup> , Roberto V. Carrillo-Serrano <sup>1,†</sup> ,  
Fortino Mendoza-Mondragón <sup>2,†</sup>  and Manuel Toledano-Ayala <sup>1,†</sup>   
and Juvenal Rodríguez-Reséndiz <sup>1,\*</sup> 

<sup>1</sup> Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro 76010, Mexico;

jose.gm@uaq.mx (J.R.G.-M.); ecruz30@alumnos.uaq.mx (E.E.C.-M.); roberto.carrillo@uaq.mx (R.V.C.-S.); toledano@uaq.mx (M.T.-A.)

<sup>2</sup> Laboratorio de Investigación en Control Reconfigurable, Querétaro 76120, Mexico; f.mendoza@licore.org

\* Correspondence: juvenal@uaq.edu.mx; Tel.: +52-442-192-1200

† These authors contributed equally to this work.

Received: 14 August 2020; Accepted: 14 September 2020; Published: 17 September 2020



**Abstract:** Motion control is widely used in industrial applications since machinery, robots, conveyor bands use smooth movements in order to reach a desired position decreasing the steady error and energy consumption. In this paper, a new Proportional-Integral-Derivative (PID)-type fuzzy logic controller (FLC) tuning strategy that is based on direct fuzzy relations is proposed in order to compute the PID constants. The motion control algorithm is composed by PID-type FLC and S-curve velocity profile, which is developed in C/C++ programming language; therefore, a license is not required to reproduce the code among embedded systems. The self-tuning controller is carried out online, it depends on error and change in error to adapt according to the system variations. The experimental results were obtained in a linear platform integrated by a direct current (DC) motor connected to an encoder to measure the position. The shaft of the motor is connected to an endless screw; a cart is placed on the screw to control its position. The rise time, overshoot, and settling time values measured in the experimentation are 0.124 s, 8.985% and 0.248 s, respectively. These results presented in part 6 demonstrate the performance of the controller, since the rise time and settling time are improved according to the state of the art. Besides, these parameters are compared with different control architectures reported in the literature. This comparison is made after applying a step input signal to the DC motor.

**Keywords:** fuzzy control; robot; PID controller; S-curve motion profile; applied artificial intelligence

### 1. Introduction

Linear motor motion controllers are presented in many industrial applications, including sliding door closers, assembly, conveyor systems, electronic manufacturing, material handling, industrial test, and robotic applications [1]. Motion control is a sub-field of automation that involves controlling mechanical movements of load and it is applied directly to the actuator to manage physical variables, such as torque, acceleration, velocity and position of an axis or axes, depending of the degree of freedom (DoF) of the system [2]. Motion control is applied to avoid the stress that is produced by a fast movement and to reduce the vibrations that are caused by the high rate of change in acceleration; also, trajectories are created to reach a desired position that the actuators must achieve [3,4]. Most commercial motion controllers that are available for industrial processes are based on classic controllers, such as Proportional-Integral-Derivative (PID) controller, and they are of closed architecture [5].

## **E. Artículo revista arbitrada**

Miguel, E. C., Reséndiz, J. R., Gutiérrez, C. A. G. (2018). IMPLEMENTACIÓN Y CO-SIMULACIÓN DE UN CONTROLADOR PID EN TIEMPO REAL CON FPGA DE BAJO COSTO Y CÓDIGO LIBRE (IMPLEMENTATION AND CO-SIMULATION OF PID CONTROLLER RT BASED ON FPGA OF LOW COST AND OPEN SOURCE). Pistas Educativas, 40(130).

Dirección General de Bibliotecas UAQ



## **IMPLEMENTACIÓN Y CO-SIMULACIÓN DE UN CONTROLADOR PID EN TIEMPO REAL CON FPGA DE BAJO COSTO Y CÓDIGO LIBRE**

*IMPLEMENTATION AND CO-SIMULATION OF PID CONTROLLER  
RT BASED ON FPGA OF LOW COST AND OPEN SOURCE*

**Edson Cruz Miguel**

Universidad Autónoma de Querétaro  
*ecruz30@alumnos.uaq.mx*

**Juvenal Rodríguez Reséndiz**

Universidad Autónoma de Querétaro  
*juvenal@uaq.edu.mx*

**Carlos A. González Gutiérrez**

Universidad del Valle de México  
*calberto.gonzalez@uvmnet.edu*

### **Resumen**

Los FPGA's (Field Programmable Gate Array) son dispositivos que tienen un gran número de elementos lógicos programables, por medio de lenguaje descripción de hardware (HDL) es posible desarrollar aplicaciones de nivel industrial. La disponibilidad de éstas en el mercado y la tendencia de herramientas libres han hecho que su uso prolifere en distintas áreas de la industria, aeronáutica, robótica, automatización. Los FPGA a través de HDL se convierten en verdadero hardware y esto se transforma en alta velocidad de procesamiento. La inclusión de dispositivos FPGA a procesos de automatización tiene sus razones en la capacidad del dispositivo en atender de forma paralela distintos procesos. El artículo presenta el diseño, implementación y validación de una plataforma de bajo costo para control de un motor CD (corriente directa) basado en FPGA. La plataforma permite desarrollar de control de velocidad y posición con controladores PID (Proporcional Integral Derivativo) industriales. En este trabajo se abordan la identificación y diseño controlador a través de un código libre

## F. Constancia ponencia



TECNOLÓGICO  
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO  
EN CELAYA

OTORGA EL PRESENTE

RECONOCIMIENTO

A

EDSON CRUZ MIGUEL, JUVENAL RODRÍGUEZ RESÉNDIZ,  
CARLOS A. GONZÁLEZ GUTIÉRREZ

POR HABER IMPARTIDO LA PONENCIA DEL ARTÍCULO  
“IMPLEMENTACIÓN Y CO-SIMULACIÓN DE UN CONTROLADOR  
PID EN TIEMPO REAL CON FPGA DE BAJO COSTO Y CÓDIGO  
LIBRE” DEL 17 AL 19 DE OCTUBRE, EN EL MARCO DEL CONGRESO  
INTERNACIONAL DE SISTEMAS MECATRÓNICOS CISMÉ 2018.

CELAYA, GTO., A 18 DE OCTUBRE DE 2018.

MTRO. IGNACIO LÓPEZ VALDOVINOS  
DIRECTOR



## **G. Artículo revista arbitrada**

Martínez, J. R. G., Reséndiz, J. R., Prado, M. Á. M., Miguel, E. E. C. (2017, May). Assessment of jerk performance s-curve and trapezoidal velocity profiles. In 2017 XIII International Engineering Congress (CONIIN) (pp. 1-7). IEEE.

Dirección General de Bibliotecas UAQ



## Assessment of Jerk Performance S-curve and Trapezoidal Velocity Profiles

José Román García Martínez  
Autonomous University of Queretaro  
UAQ  
Queretaro, Mexico  
jossroman@gmail.com

Miguel Ángel Martínez Prado  
Autonomous University of Queretaro  
UAQ  
Queretaro, Mexico  
Miguel.prado@uaq.mx

Juvenal Rodríguez Reséndiz  
Autonomous University of Queretaro  
UAQ  
Queretaro, Mexico  
juvenal@uaq.edu.mx

Edson Eduardo Cruz Miguel  
Autonomous University of Queretaro  
UAQ  
Queretaro, Mexico  
eecruz@gmail.com

**Abstract**— nowadays, there are several techniques for the acceleration and deceleration of computer numerical control (CNC) machine tools and industrial robots (Robot manipulators) in order to plan smoother trajectories avoiding the jerk and another kind of stress. The aim of this article is to show a comparison between the trapezoidal and s-curve velocity profile used in robotics and CNC machinery. After doing that, the reader will be able to select a velocity profile in order to design their trajectories.

**Keywords**— Trapezoidal, s-curve, polynomial, trajectory, profile.

### I. INTRODUCTION

Today, in many systems of movement, performance requirements include small motion times and small settlement times. Typical examples are pick and place machines, hard disk drives, XY-tables and many robots [1]. The traditional motion laws with piecewise constant acceleration presents discontinuities such that regulators cannot follow, whatever the performances of the actuators. These discontinuities can excite considerably the structure under study in transitory phases, also they are responsible for a large part of the deterioration of the dynamic behavior. From the available parameters in recent robots and modern CNCs, it is known that the maximum change of acceleration value (per axis) can restrict the oscillatory behavior of the load [2]. The rate of change of acceleration is defined as jerk. If the jerk value is reduced, it is fairly possible reducing the vibrations, acting directly on the smoothness degree of motion.

Industrial designers handle frequently reference profiles to depict varying scanning and point-to-point motions. These profiles are commonly designed as piecewise finite order polynomials. Finite order polynomials typically contain

constant motion phases (velocity, acceleration, jerk, a derivative of jerk, etc.) [3]. A special characteristic of these profiles is that they hold mostly low-frequency energy.

Generally, a profile is planned such that the resonance dynamics of the system is not excited[1]. The most used profiles in the industry are, the trapezoidal and the s-curve velocity profile. The main reason is because of its smoothness, in the first case the acceleration is constant in two periods of time (from  $t_0$  to  $t_1$  and  $t_2$  to  $T$  in Fig. 1). The second case consist in making the jerk constant for several periods of time, avoiding that the jerk goes to the infinite.

### II. POLYNOMIAL TRAPEZOIDAL MODEL

The trapezoidal velocity profile consists in accelerate constantly a motor with a constant velocity and then decelerated it constantly to zero; the motor can reach fast motions in short time. Nevertheless as shown in Fig. 1, the instant  $t_1$  is achieved when the velocity reaches the high value, the acceleration jumps from its constant value to zero. The leap happens at other time instants,  $t_0$ ,  $t_1$ ,  $t_2$  and  $T$  when the velocity changes its orientation [4].

## H. Constancia poster

2018.pdf



### XIV International Engineering Congress – CONIIN 2018

The Querétaro State University through the Engineering Faculty grant the present acknowledgment to:

**Marco Antonio Gómez Guzmán (Autor)**  
Jonathan Armando Troncoso Ramos (Co-autor)  
Dr. Roberto Valentín Carrillo Serrano (Co-autor)  
Edson Eduardo Cruz Miguel (Co-autor)

FOR THE COLLABORATION:

**POSTER SESSION:** Recent advances  
of four refrigeration cycles

QUERÉTARO, MEX.  
MAY 2018

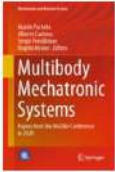
**Dr. Manuel Toledano Ayala**  
PRINCIPAL  
ENGINEERING FACULTY

**Dr. Gonzalo Macías Bobadilla**  
GENERAL COORDINATOR CONIIN  
ENGINEERING FACULTY

[ingenieria.uaq.mx/coniin](http://ingenieria.uaq.mx/coniin)

## I. Artículo revista arbitrada


Padron, B., Cruz-Miguel, E. E., Correa, J. C. J., Carbone, G. A Force Control Architecture for Grasping with a Robotic Finger. In International Symposium on Multibody Systems and Mechatronics (pp. 214-221). Springer, Cham.



[International Symposium on Multibody Systems and Mechatronics](#)  
MuSMe 2021: [Multibody Mechatronic Systems](#) pp 214-221 | [Cite as](#)

### A Force Control Architecture for Grasping with a Robotic Finger

Authors Authors and affiliations

Bart Padron , Edson E. Cruz-Miguel, Juan Carlos Jauregui Correa, Giuseppe Carbone

Conference paper  
First Online: 23 October 2020

12 Downloads

Part of the [Mechanisms and Machine Science](#) book series (Mechan. Machine Science, volume 94)

#### Abstract

This paper addresses the design of control architecture for a robotic finger, achieved by a proper and specific kinematic and dynamic modeling. The proposed model permits force feedback prediction, based on the input motor torque, preventing the usage of force sensors for achieving a proper grasping force control. Hardware and software of the proposed control are developed and implemented in a prototype. The force prediction accuracy is tested experimentally while grasping an object. Preliminary tests show that the robotic finger can accomplish the needed grasping with proper force feedback control.

#### Keywords

Force controller   Robotic finger   Grasping   Experimental testing

## J. Constancia seminario doctoral



UNIVERSIDAD  
AUTÓNOMA  
DE QUERÉTARO



FACULTAD  
DE INGENIERÍA



DIPFI  
POSGRADO  
INGENIERÍA

Se otorga la presente  
**CONSTANCIA** a:

**M.C. EDSON EDUARDO CRUZ MIGUEL**

Por su participación como PONENTE  
en el SEMINARIO DOCTORAL con la conferencia:

**"CONTROLADORES ROBUSTOS  
DE ARQUITECTURA ABIERTA PARA SERVOSISTEMAS  
BAJO EL CONCEPTO OPEN SOURCE"**

*Campus Querétaro*

*8 de mayo de 2020  
Parque Biotecnológico  
Facultad de Ingeniería*

**Dr. Manuel Toledano Ayala**  
DIRECTOR  
FACULTAD DE INGENIERÍA

## **K. Formación de recurso humano**

Registro de protocolo de alumno de Maestría en Ciencias Bernardo Daniel Trujillo Ibarra con el tema: Plataforma de control difuso para generación de trayectorias en un robot poliarticulado.

Dirección General de Bibliotecas UAQ





Universidad Autónoma de Querétaro  
División de Investigación y Posgrado  
Facultad de Ingeniería

C. U., febrero 05, 2020  
Acta 273/2020

**Dra. Ma. Guadalupe Flavia Loarca Piña,**  
**Directora de Investigación y Posgrado UAQ**  
**Presente:**

Con relación a la solicitud del alumno **Bernardo Daniel Trujillo Ibarra**, inscrito en la Maestría en Ciencias (INSTRUMENTACIÓN Y CONTROL AUTOMÁTICO) con número de expediente 281306, para cambio de Director de Tesis, enviado al H. Consejo de Posgrado e Investigación de esta Facultad, le informo el acuerdo que se tomó y quedó asentado en el Acta de sesión ordinaria No. 273 celebrado el día 05 de febrero de 2020.

Se acepta el cambio inicialmente **Dr. Juvenal Rodríguez Reséndiz**, ahora quedará el **M. en C. Edson Eduardo Cruz Miguel**.

Sin otro particular, me es grato enviarle un cordial saludo.

Atentamente

"El Ingenio para Crear. No para Destruir"  
  
**Dr. Juan Carlos Jáuregui Corrao**  
**Jefe de la DIFI.**

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO  
FACULTAD DE INGENIERÍA  
DIVISIÓN DE INVESTIGACIÓN  
Y POSGRADO

c.c.p. Coordinador del Programa  
Alumno  
Archivo.

JC.G.

## **L. Formación de recurso humano**

Titulación de alumno de Maestría en Ciencias Jose Roman García Martínez como sinodal con el tema: Diseño de un controlador difuso y un generador de perfiles *S-curve* para el control de un robot poliarticulado.

Dirección General de Bibliotecas UAQ



No. de Acta:  
2018FIN016687

# UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

## SECRETARÍA ACADÉMICA

### ACTA DE EXAMEN DE GRADO

En la ciudad de Querétaro Qro., siendo las 11:00 horas del día 10 de Diciembre de 2018, en las instalaciones de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro, se reunió el jurado designado por la Secretaría Académica en la autorización número 5024 de fecha 29 de Noviembre de 2018, DR. JUVENAL RODRIGUEZ RESENDIZ, DIRECTOR; DR. EDGAR ALEJANDRO RIVAS ARAIZA, SECRETARIO; DR. ROBERTO VALENTIN CARRILLO SERRANO, VOCAL; MTR. EDSÓN EDUARDO CRUZ MIGUEL, SINODAL; MTR. MIGUEL ANGEL MARTINEZ PRADO, SINODAL, para efectuar el examen de grado al

### C. JOSE ROMAN GARCIA MARTINEZ

Para obtener el Grado de

### MAESTRIA EN CIENCIAS (INSTRUMENTACION Y CONTROL AUTOMATICO)

El acto se realizó de conformidad con el Instructivo de Exámenes Profesionales vigente, expedido por la institución, con base en: La tesis "Diseño de un controlador difuso y un generador de perfiles S-curve para el control de un robot poliarticulado", realizado por el sustentante, y el resultado fue:

*Aprobado por unanimidad de votos*

Acto seguido se tomó protesta y para constancia del examen se levanta la presente acta, que firman de conformidad los participantes:

 DR. JUVENAL RODRIGUEZ RESENDIZ PRESIDENTE	 DR. EDGAR ALEJANDRO RIVAS ARAIZA SECRETARIO
 DR. ROBERTO VALENTIN CARRILLO SERRANO VOCAL	 MTR. EDSÓN EDUARDO CRUZ MIGUEL SUPLENTE
 MTR. MIGUEL ANGEL MARTINEZ PRADO SUPLENTE	 JOSE ROMAN GARCIA MARTINEZ SUSTENTANTE
 DRA. MA. GUADALUPE FLAVIA LOARCA PIÑA Director de Investigación y Posgrado	 DR. AURELIO DOMINGUEZ GONZALEZ Secretario Académico

## **M. Formación de recurso humano**

Titulación de alumno de Maestría en Ciencias Jose Roman García Martínez como sinodal con el tema: Desarrollo de un control de posición para un robot *ten-segrity* clase 2.

Dirección General de Bibliotecas UAQ



# UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

## SECRETARÍA ACADÉMICA



### ACTA DE EXAMEN DE GRADO

En la ciudad de Querétaro Qro., siendo las 14:00 horas del día 14 de febrero de 2020, en las instalaciones de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro, se reunió el jurado designado por la Secretaría Académica en la autorización número 5677 de fecha 11 de diciembre de 2019, DR. GERARDO ISRAEL PEREZ SOTO, PRESIDENTE; DR. JUVENAL RODRIGUEZ RESENDIZ, SECRETARIO; MTRO. CARLOS GUSTAVO MANRIQUEZ PADILLA, VOCAL; DRA. KARLA ANHEL CAMARILLO GOMEZ, SINODAL; MTRO. EDSON EDUARDO CRUZ MIGUEL, SINODAL, para efectuar el examen de grado al

**C. OSCAR ALEJANDRO ZAVALA PEREZ**

Para obtener el grado de

**MAESTRIA EN CIENCIAS (INSTRUMENTACION Y CONTROL AUTOMATICO)**

El acto se realizó de conformidad con el Reglamento de Estudiantes vigente, expedido por la institución, con base en la tesis "Desarrollo de un control de posición para un robot tensegrity clase 2", realizado por el sustentante, y el resultado fue:

No. de acta:  
2020FIN020969

*Aprobado por unanimidad de votos con mención honorífica*

Acto seguido se tomó protesta y para constancia del examen se levanta la presente acta, que firman de conformidad los participantes:

 DR. GERARDO ISRAEL PEREZ SOTO PRESIDENTE	 DR. JUVENAL RODRIGUEZ RESENDIZ SECRETARIO
 MTRO. CARLOS GUSTAVO MANRIQUEZ PADILLA VOCAL	 DRA. KARLA ANHEL CAMARILLO GOMEZ SUPLENTE
 MTRO. EDSON EDUARDO CRUZ MIGUEL SUPLENTE	 OSCAR ALEJANDRO ZAVALA PEREZ SUSTENTANTE
Director de Investigación y Posgrado  DRA. MA. GUADALUPE FLAVIA LOARCA PIÑA	Secretario Académico  DR. AURELIO DOMINGUEZ GONZÁLEZ

Dirección General de Bibliotecas

## N. Formación académica



## Ñ. Formación académica



Se otorga la presente

**CONSTANCIA** a:

**EDSON EDUARDO CRUZ MIGUEL**

Por su participación en:  
El Curso de Actualización Docente  
en la temporada 2019-2 del Seminario Doctoral  
de la Facultad de Ingeniería el semestre  
agosto-diciembre 2019 con una duración de 26 horas.

20 de enero del 2020  
Auditorio Parque Biotecnológico  
Facultad de Ingeniería

**Dr. Irineo Torres Pacheco**  
COORDINADOR DOCTORADO EN INGENIERÍA  
FACULTAD DE INGENIERÍA

**Dr. Arturo González Gutiérrez**  
COORDINADOR DEL SEMINARIO DOCTORAL  
FACULTAD DE INGENIERÍA

## O. Formación académica



UNIVERSIDAD  
AUTÓNOMA  
DE QUERÉTARO



FACULTAD  
DE INGENIERÍA



DIPFI  
POSGRADO  
INGENIERÍA

Se otorga la presente  
**CONSTANCIA** a:

**EDSON EDUARDO CRUZ MIGUEL**

Por su participación en:  
El Curso de Actualización Docente  
en la temporada 2020-1 del Seminario Doctoral  
de la Facultad de Ingeniería durante el semestre  
enero-julio 2020 con una duración de 26 horas.

10 de agosto del 2020  
Auditorio Parque Biotecnológico  
Facultad de Ingeniería

**Dr. Irineo Torres Pacheco**  
COORDINADOR DOCTORADO EN INGENIERÍA  
FACULTAD DE INGENIERÍA

**Dr. Arturo González Gutiérrez**  
COORDINADOR DEL SEMINARIO DOCTORAL  
FACULTAD DE INGENIERÍA