

**DESARROLLO DE UN SISTEMA DISTRIBUIDO EN SOC-FPGA PARA EL
CONTROL DE UN ACTUADOR ROBÓTICO CON INTERFAZ IIC EN
CONJUNTO CON LA UNIVERSIDAD DE BRASÍLIA - BRASIL.**

Autor

JOSE ALFREDO MENDOZA PEÑALOZA

Director

DIEGO JOSE BARRERA OLIVEIROS

M.Sc.

Codirector

SERGIO ANDRES PERTUZ MENDEZ

M.Sc.

UNIVERSIDAD DE PAMPLONA

FACULTAD DE INGENIERÍAS Y ARQUITECTURAS

**DEPARTAMENTO DE INGENIERÍAS MECÁNICA. MECATRÓNICA E
INDUSTRIAL**

PROGRAMA INGENIERÍA MECATRÓNICA

PAMPLONA, NORTE DE SANTANDER – COLOMBIA

2019

DEDICATORIA

Esta tesis va dedicada a mi familia que me dieron su total apoyo total, confianza y fuerzas para continuar con el trabajo a pesar de las dificultades que pude pasar durante el tiempo de desarrollo de la tesis y también en el transcurso total de mis estudios universitarios, para mi padre Dimas Alberto Mendoza Sierra, mi madre Luz Darys Peñaloza Ibarra, y mi hermano Alberto Jose Mendoza Peñaloza.

También dedico a Dios todo poderoso por darme la guiarne y darme el conocimiento y el entendimiento necesario para la realización de este trabajo.

AGRADECIMIENTOS

Agradezco principalmente a la universidad de Pamplona ya que ellos me brindaron el apoyo económico y jurídico al tener relaciones con universidades en el exterior como es el caso de la universidad de Brasilia también dando agradecimientos a esta última que me brindó apoyo tanto de su personal técnico, así como también prestó sus equipos para la realización del trabajo.

Finalmente agradecer a mi tutor de tesis Diego Barrera Oliveros que me ayudó también en el transcurso de mi formación como ingeniero, y a mi codirector Sergio Pertuz Méndez que me ayudó enormemente para el desarrollo del trabajo de grado y con los problemas que se me fueron presentando a lo largo del transcurso de la pasantía de investigación en la universidad de Brasilia, otras menciones de profesores que me ayudaron durante mi formación son el profesor Cesar Augusto Peña, el profesor Aldo Pardo García, el profesor Diego Mejía Bugallo y el profesor Daniel M. Muñoz Arboleda que me ayudó guiando mi trabajo también en la universidad de Brasilia. Agradezco algunos amigos los cuales nos apoyamos durante el tiempo de formación en la carrera ingeniería en mecatrónica como Anderson Sánchez Rojas, Iván Fernando Vargas y Jader Sánchez Rojas.

Y nuevamente agradeciendo a mi familia por el enorme apoyo durante todo mi proceso de formación y mis estudios en ingeniería en mecatrónica.

RESUMEN

En este trabajo de tesis se expone el trabajo realizado en la pasantía de investigación en la universidad de Brasilia – Brasil, así como también el cómo también del convenio realizado entre la universidad de Pamplona y la universidad de Brasilia. También se expondrán trabajos adicionales realizados. En el desarrollo del fundamento principal de la tesis comenzamos con la construcción de un nuevo actuador robótico compuesto por driver de motor, sensores de posición, sensores de corriente y un módulo de comunicación IIC que serán controlador mediante un sistema de control basado en módulos que trabajan en paralelo también llamado sistema en chip (SoC) implementado una tarjeta FPGA, el actuador robótico esta direccionado para formar parte de un nuevo diseño de mano robótica con mejor funcionalidad y más grados de libertad. También se desarrollarán actividades secundarias como la asistencia a clases de sistemas Bioinspirados que van enfocados las actividades como la participación a congresos y la publicación de artículos científicos en revistas indexadas.

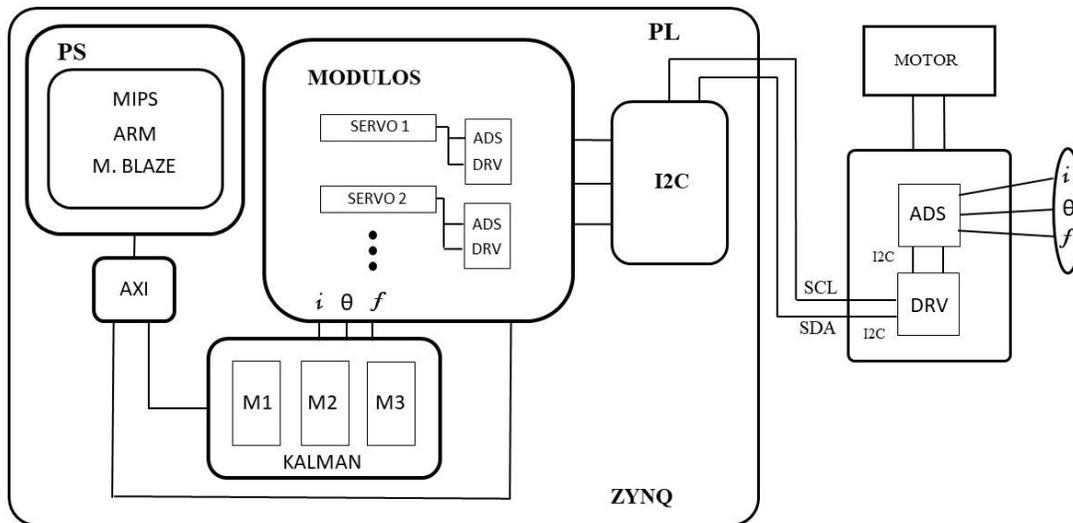


Figura. 1. Representación gráfica del desarrollo del proyecto.

El actuador robótico se desarrollará a partir de diferentes técnicas de manufactura, de CAD/CAM de diseño electrónico y mecánico utilizando equipos de prototipado rápido de placas para componentes electrónicos SMD de LPKF protomat e Impresoras 3D disponibles en los laboratorios del campus universitario de la UNB e IFB en Brasilia, El controlador de este sistema será desarrollado en un SoC-FPGA de Xilinx en la placa de desarrollo Zybo con chipset Zynq-7010 que está compuesta de un procesador ARM Dual-Core Cortex A9 que permite realizar programar procesos en paralelo en su entorno de programación VHDL o Verilog. La ventaja de trabajar en este tipo de ambientes es que se puede controlar el actuador robótico en forma paralela. Esto traduce la capacidad de controlar varios actuadores al mismo tiempo con un solo chip, permitiendo tiempos de procesamiento se produzcan más gasto computacional, sin embargo este problema también se puede abordar y analizar para desarrollar el proceso con el menor gasto computacional posible y que posee una velocidad alta en la ejecución del proceso.

ABSTRACT

In this thesis work, the work carried out in the research internship at the University of Brasilia - Brazil is exposed, as well as the how also of the agreement made between the University of Pamplona and the University of Brasilia. Additional enhanced works will also be exhibited. In the development of the main basis of the thesis we begin with the construction of a new robotic actuator composed of motor driver, position sensors, current sensors and an IIC communication module that will be controller through a control system based on modules that work In parallel also called chip system (SoC) implemented an FPGA card, the robotic actuator is addressed to be part of a new robotic hand design with better functionality and more degrees of freedom. Secondary activities will also be developed, such as attendance at classes of Bio-inspired systems that focus on activities such as participation in conferences and the publication of scientific articles in indexed journals.

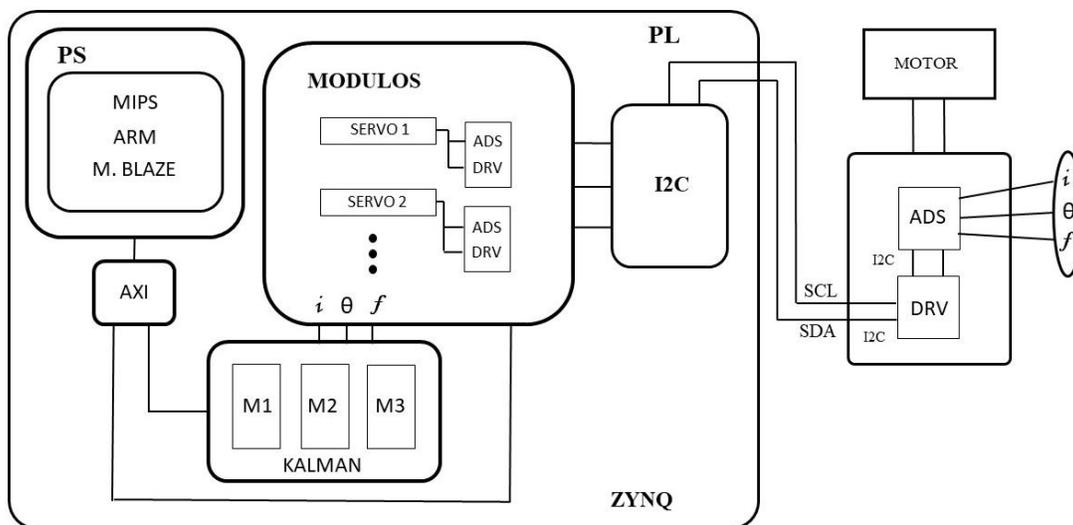


Figure. 1. Graphic representation of project development.

The robotic actuator will be developed from different manufacturing techniques, from CAD/CAM of electronic and mechanical design using rapid plate prototyping equipment for SMK electronic components of LPKF protomat and 3D printers available in the laboratories of the university campus of UNB e IFB in Brasilia, The controller of this system will be developed in an Xilinx SoC-FPGA on the Zybo development board with Zynq-7010 chipset that is composed of an ARM Dual-Core Cortex A9 processor that allows you to program processes in parallel in your VHDL or Verilog programming environment. The advantage of working in these types of environments is that the robotic actuator can be controlled in parallel. This translates the ability to control several actuators at the same time with a single chip, allowing processing times to produce more computational expense, however this problem can also be addressed and analyzed to develop the process with the lowest possible computational expense and that has a high speed in the execution of the process.

TABLA DE CONTENIDO

DEDICATORIA	2
AGRADECIMIENTOS	3
RESUMEN	4
1 INTRODUCCIÓN	10
1.1 Planteamiento del Problema	10
1.2 Justificación	10
1.3 Delimitaciones	10
1.4 Metodología	11
2 INFORME PASANTIA DE INVESTIGACION	14
2.1 Laboratorio y equipos	15
2.2 Estudio de sistemas Bioinspirados para la ingeniería	19
2.3 Diseño mano robótica	21
2.4 Diseño de actuador robótico	21
2.5 Proyecto Bioinspirados	22
2.6 Artículo científico	23
2.7 Convenio entre universidades	23
3 MARCO TEÓRICO	23
3.1 Sistemas en Chip (SoC)	23
3.2 FPGA (Field Programmable Gate Array)	24
3.3 Lenguaje VHDL, VHSIC y HDL	24
3.4 Procesadores	25
3.5 Comunicación AXI	25
3.6 Comunicación IIC o I2C (Inter Integrated Circuit)	25
3.7 Diseño CAD (Computer-Aided Design)	26
3.8 Robótica	26
3.9 Robótica antropomórfica	26
3.10 Robótica móvil	27
3.11 Sistemas de control y Control PID	27
3.12 Sistemas Bioinspirados e inteligencia colectiva	28

3.12.1	Algorithm PSO (Particle Swarm Optimization)	28
3.12.2	Algorithm ABC (Artificial Bee Colony Optimization)	29
3.12.3	Algoritmo DE (Diferential Evolution)	29
3.12.4	Algoritmo SCA (Sine Cosine Algorithm)	30
3.13	Test de hipótesis	30
3.14	Impresión 3D	31
4	DESARROLLO DE ACTUADOR ROBÓTICO	31
4.1	Vivado Xilinx	31
4.2	Mapa de bloques y creación IP	31
4.3	Librería AXI IIC	33
4.4	Librería ADS1115	34
4.5	Librería DRV3330	35
4.5	Diseño CAD y Fabricación	36
4.6	Resultados	38
5	DISEÑO DE MANO ROBÓTICA	43
5.1	Resumen y Componentes	43
5.2	CAD	44
5.3	Fabricación	45
5.4	Resultados	45
6	Proyecto final clase Bioinspirados	46
6.1	Introducción Bioinspirados	46
6.2	Proyecto final Bioinspirados	47
6.3	Preparación de algoritmos para la optimización del sistema de control	48
6.4	Resultados	49
6.5	Test de hipótesis	52
7	CONCLUSIONES	53
8	BIBLIOGRAFÍA	54

ÍNDICE DE FIGURAS

Figura. 1. Representación gráfica del desarrollo del proyecto.	4
Figura. 2. Laboratorio Graco de la universidad de Brasilia.	15
Figura. 3. Laboratorio LEIA de la universidad de Brasilia.	16
Figura. 4. Laboratorio de herramientas de mecatrónica de la UnB.	16
Figura. 5. Maquina CNC para mecanizado.	17
Figura. 6. Planta de calderas doble efecto automatizada.	17
Figura. 7. Brazo robótico de la empresa ABB en la UnB.	18
Figura. 8. Impresora 3D del laboratorio de mecatrónica en la UnB.	18
Figura. 9. Brazo robótico para trabajos de fresado de la empresa ASEA.	19
Figura. 10. Aula de clases de la materia sistemas Bioinspirados para la ingeniería.	20
Figura. 11. Arquitectura de Sistema de control PID.	27
Figura. 12. Modulo IP realizado para la comunicación AXI IIC en VHDL.	32
Figura. 13. Diagrama de bloques para la comunicación del procesador con el hardware de la FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board (zynq).	33
Figura. 14. Descripción de la comunicación IIC del dispositivo IIC ADS1115 de la empresa Texas Instruments.	35
Figura. 15. Descripción de la comunicación IIC del driver DRV3330 de la empresa Texas Instruments.	36
Figura. 16. Vista de explosión del diseño del actuador robótico.	37
Figura. 17. Vista isométrica del diseño del actuador robótico realizado en SolidWorks. ...	37
Figura. 18. Actuador robótico desarmado para la visualización de sus componentes.	38
Figura. 19. Actuador robótico ensamblado para la realización de pruebas.	39
Figura. 20. Montaje del actuador robótico con FPGA Zybo, fuente de poder y osciloscopio para las pruebas de funcionamiento.	39
Fig. 21. Movimiento en un sentido de giro del actuador robótico usando la FPGA Zybo. ...	40
Fig. 22. Señal de stop del emitida por la FPGA Zybo al actuador robótico.	41
Fig. 23. Señal del osciloscopio de SDA y SCL dadas por la FPGA Zybo y por el módulo ADS1115.	42

Fig. 24. Tiempo entre transacciones es de 30.50 micro segundos (uS).	43
Fig. 25. Rediseño del CAD de modelo de mano robótica realizado en SolidWorks.	44
Fig. 26. Modelo de la mano robótica ensamblado para realizar pruebas por estudiantes de maestría y doctorado de la UnB.	45
Fig. 27. Mano robótica funcional para el estudio y análisis utilizando FPGA por parte de los estudiantes de maestría y doctorado de la UnB.	46
Fig. 28. Mano Diagrama de control PID para la simulación del comportamiento del robot móvil en Matlab-Simulink.	47
Fig. 29. Función de transferencia que simula el comportamiento del robot móvil.	48
Fig. 30. Mejor respuesta del controlador PID usando algoritmo ABC.	49
Fig. 31. Mejor respuesta del controlador PID usando algoritmo SCA.	49
Fig. 32. Mejor respuesta del controlador PID usando algoritmo DE.	50
Fig. 33. Datos obtenidos de las cinco rodadas del algoritmo SCA.	50
Fig. 34. Datos obtenidos de las cinco rodadas del algoritmo ABC.	51
Fig. 35. Datos obtenidos de las cinco rodadas del algoritmo DE.	51
Fig. 36. Funcionamiento de robot móvil con el resultado de las contantes de PID del algoritmo DE.	52

1 INTRODUCCIÓN

1.1 Planteamiento del Problema

En la actualidad las nuevas tecnologías de control están en búsqueda de disminuir el tiempo de respuesta de los sistemas y consumo de recursos computacionales, una de las alternativas claras es la implementación de las FPGA (Field Programmable Gate Array) las cuales tienen la capacidad de realizar un procesamiento en paralelo de diferentes procesos y de esta forma entregar un resultado óptimo más rápido. Teniendo en cuenta lo anterior este trabajo se enfocará en la implementación de un actuador robótico utilizando un sistema en chip (SoC) utilizando el protocolo de comunicación IIC, un sensor de corriente y un driver de motor, donde el control se realizará a partir de módulos desarrollados en la FPGA en busca de optimizar recursos y aumentar la velocidad de respuesta del sistema.

¿Se puede aumentar la velocidad de procesamiento de los sistemas utilizando arquitecturas como las que ofrecen las FPGA's?

1.2 Justificación

Dada las necesidades de aumentar la velocidad de los sistemas de control ya que existen procesos que requieren alta capacidad de respuesta se puede notar que las tecnologías que ofrecen las SoC-FPGA's son una buena alternativa ya que al permitir realizar diversos procesos en paralelo los tiempos de proceso disminuyen significativamente utilizando un solo dispositivo.

De esta forma la propuesta de realizar un actuador robótico que prevé implementar en un nuevo modelo de mano robótica, pueda ofrecer resultados óptimos en poco tiempo es una gran ventaja la cual poseerá alrededor de 12 actuadores de los mencionados que funcionaran en paralelo a una frecuencia alta de aproximadamente 1 KHz, se valora también la capacidad de la comunicación IIC que nos permite conectar una amplia variedad de dispositivos o sensores solo utilizando dos líneas de comunicación (SDA, SCL).

1.3 Delimitaciones

Objetivo General:

Desarrollo de arquitectura en hardware reconfigurable SoC-FPGA de un sistema distribuido para el control de un actuador robótico con interfaz IIC en conjunto con la universidad de Brasilia - Brasil.

Objetivos Específicos:

- Diseño y construcción de actuador robótico.
- Pruebas y caracterización de actuador robótico.
- Desarrollo de arquitectura distribuida en hardware reconfigurable (SoC-FPGA) para el control del actuador robótico.
- Pruebas del actuador robótico integrado con la arquitectura distribuida.
- Integración y realización de manuales del actuador robótico y su librería de control en VHDL y software.

1.4 Metodología

En primer lugar se realiza un estudio sobre el lenguaje de programación VHDL (Very High Speed Integrated Circuit) que es aquel que permite controlar la parte del hardware de los dispositivos programables, la ventaja de realizar este tipo de programación es que es mucho más rápido en realizar operaciones la mayoría de los software convencionales que usan compiladores para interpretar el código escrito en bits para ser enviados al procesador, a diferencia del VHDL es que permite enviar directamente el programa a el procesador destinado para estos procesos, a los dispositivos que se les puede aplicar son principalmente las FPGA (Field Programmable Gate Array), que consisten en sistemas embarcados los cuales poseen un procesador que se encarga de realizar los diferentes procesos programados en este y también posee una parte de implementación en hardware que nos permite realizar programas con la capacidad de funcionar en paralelo diferentes proceso.

Para nuestro caso de estudio de este trabajo se utilizará una FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board, que se usara para la creación de los programas en hardware implementado para la creación de un módulo AXI IIC que mediante la creación de protocolos y registros de comunicación nos permite conectar los programas realizados en el procesador y a la vez que nos permite comunicarnos con dispositivos IIC que para nuestro trabajo serán un driver de motor DRV3330, y un módulo de comunicación IIC que nos permite realizar lectura de sensores ADS1115.

Seguidamente se realiza el diseño de una nueva propuesta de actuador robótico para implementar en una aplicación en un nuevo modelo de mano robótica para diferentes aplicaciones utilizando FPGA, este actuador robótico estará compuesto de un sensor de posición que funciona mediante un eje el cual envía un valor de voltaje de pendiendo de su posición Rotary Position Sensor SV01 fabricado por la empresa Murata, un módulo de comunicación IIC que nos permite leer cuatro canales analógicos y comparador de canales

ADS1115 de Texas Instruments, motor DC para el movimiento del actuador y el eje, un driver para el control del motor enviando un valor de velocidad y los cuatros estados del motor stop, run en dos sentidos de giro y el break, adicionalmente este módulo nos permite leer los fallos de lectura y comunicación IIC presentes durante el uso de este DRV3330 de la Texas Instruments y finalmente un sensor y amplificador de corriente que se encuentra entre el driver DRV3330 y el módulo de comunicación IIC ADS1115 el cual nos permite leer los valores de corrientes generados por el accionamiento del motor DC picos de arranque y demás perturbaciones que afecten al motor INA240 de Texas Instruments.

El diseño del actuador se realizare en el software CAD SolidWorks teniendo como base de medidas el tamaño del sensor SV01, las dimensiones de un motor DC y la placa PCB (Printed Circuit Board) que contiene el dispositivo de comunicación IIC ADS1115, el sensor y amplificador de corriente INA2540, el driver para el control del motor DRV3330.

Seguidamente se realizan las librerías correspondientes para la comunicación con los dispositivos IIC en el procesador de la FPGA el cual permite programarse mediante el lenguaje en código C, en primer lugar se debe crear una librería para la lectura y escritura utilizando el módulo AXI IIC implementado en hardware el donde creamos cuatro funciones, la primera es la función de escritura en donde debemos escribir un registro con la dirección del módulo AXI dada en el software vivado que nos indica que parte de hardware es la que se está utilizando, después esta función nos permite escribir un buffer de datos de hasta 32 bits donde podemos escribir tanto las direcciones de los dispositivos IIC como también los valores de configuración necesarios en los cuales permiten el funcionamiento de los modulos para esto utilizamos la librería de comunicación con módulos AXI de Xilinx “Xil IO.h” con la función “Xil_out32” .

Luego tenemos la función de lectura que consiste primeramente en escribir la dirección del módulo AXI donde se realiza la implementación del hardware junto con un registro, luego se realiza una nueva escritura con la dirección del dispositivo IIC y un vector de configuración que es el que indica que registro del dispositivo que vamos a leer, seguidamente se llama la función “Xil_In32” que permite leer un buffer de datos de hasta 32 bits y seguidamente se almacena en un buffer para los posteriores procesos que se deben realizar.

La siguiente función cumple el papel de un espera o retardo ya que el módulo AXI implementado en hardware al realizar una acción de lectura o escritura este envía una señal de confirmación que indica que se a terminado el proceso y se puede comenzar otro nuevo de esta forma nos aseguramos de que no hayan errores en la comunicación y que no se crucen procesos de escritura y lectura diferentes esto se hace simplemente un ciclo “While” en el cual se realiza un lectura con la función “Xil_In32” que hasta no recibir la señal deseada no salga del ciclo.

Finalmente, la última función de esta primera librería permite limpiar el buffer de datos ya utilizados o residuales para que no afecten el proceso de comunicación con el módulo AXI que se realiza solamente almacenando el valor de cero en dicho buffer.

Adicionalmente realizamos las librerías de comunicación con los dos módulos de comunicación IIC que el ADS1115 que posee un programa “Header” o “Cabecera” donde se definen los vectores de comunicación y diferentes direcciones que puede poseer este módulo dependiendo de su conexión, seguido de eso se crea una clase u objeto en la que definimos algunas constantes y también se crean las funciones necesarias para el funcionamiento del módulo donde tenemos la primera función “ADS1115” la cual permite escribir la dirección del módulo ADS1115 y almacenarla en un buffer como variable global, después tenemos la función de inicio de comunicación, las funciones de leer los canales de forma independiente, después la función de leer los canales en comparación de dos en dos y finalmente las funciones de leer y escribir las cuales tienen la función de llamar las funciones creadas de la librería “Axi IIC.h,” las cuales permiten enviar las señales IIC por medio del módulo de comunicación IIC implementado en VHDL.

Después tenemos la creación de la librería para el control del Driver de motor DRV3330 en las cuales de igual forma que con el ADS1115, se definen las constantes globales con las que trabajaremos con esta librería donde se especifican los vectores de configuración del módulo y de igual forma las direcciones que puede tener el dispositivo dependiendo de su conexión, enseguida creamos una clase en la cual están definidas las funciones con las cuales se debe trabajar el módulo y se crea un nuevo programa en C donde se realizan las funciones en las cuales destacamos las funciones de “Stop” que sirve para detener el movimiento del motor, la función “Break” que sirve para poner el motor en estado de break o dos señales de alto, la función “Drive” que es la que nos permite mover el motor de acuerdo a un valor de velocidad y también definir el sentido de giro del motor, la función “GetFault” que nos permite leer los errores ocurridos durante la comunicación con el dispositivo como errores de dirección o de configuración o cambios en la corriente del dispositivo, y las funciones de escribir y leer registro que solamente llaman las funciones de la librería “Axi IIC.h” creada anteriormente.

Posteriormente se realiza todo el montaje para el proceso de funcionamiento del nuevo modelo de actuador robótico en conjunto con la FPGA que se comunican mediante el protocolo IIC, para realizar las pruebas de funcionamiento para poder verificar que el proceso está funcionando correctamente se usa un osciloscopio donde podremos observar que se emiten y que se reciben tanto de los dispositivos IIC como de la FPGA.

Adicionalmente se realizan algunos trabajos extras, en primer lugar, se trabaja en el rediseño de un prototipo de mano robótica que se encuentra en los laboratorios de universidad de Brasilia “UnB”. Para lo cual se hace uso del software CAD SolidWorks y se hace uso de los materiales del laboratorio de mecatrónica disponibles como las impresoras 3D y de instrumentos el laboratorio de mecánica como lo son el torno para la fabricación de los materiales necesarios para la fabricación de las modificaciones y finalmente pruebas del funcionamiento de la mano robótica.

También como trabajo adicional tenemos la asistencia a una clase de maestría de la universidad de Brasilia, sistemas Bioinspirados para la ingeniería, en la cual se tratan temas como optimización matemática de sistemas o problemas de programación lineal que se pueden resolver mediante soluciones por graficación y ecuaciones o mediante métodos de

tabulación como lo puede ser el método de optimización simplex, luego se aborda el tema de los sistemas Bioinspirados donde podemos estudiar diversos comportamientos de organismo que se mueven como un enjambre como por ejemplo las abejas, hormigas, etcétera. Pasamos después a tipos de algoritmos basados en estos sistemas donde tenemos algoritmos como el PSO “Particle Swarm Optimization”, ABC “optimización por colonia de abejas”, DE “Evolution Differential”, entre otros. Luego seguimos modificando estos algoritmos al agregarles diversidad en la cual las partículas pueden tener algunas características adicionales además de las del comportamiento común de su algoritmo, luego seguimos con un apartado de análisis estadístico los cuales por medio de diferentes test o pruebas podemos determinar que algoritmo fue capaz de optimizar mejor un problema, luego utilizamos los diferentes algoritmos para realizar sintonización de controladores PID y el cálculo de pesos en las redes neuronales artificiales, y finalmente estudiamos algoritmos Bioinspirados multi objetivos de los cuales podemos destacar el algoritmo genético multiobjetivo “NGSA”.

Como parte final de la materia nos pedía realizar un proyecto final que consistiera en un aplicación donde utilizáramos los algoritmos Bioinspirados estudiados durante la clase, que para este caso fue la optimización de un controlador PID para un robot móvil el cual tiene la capacidad de mantenerse a una distancia determinada de una superficie utilizando un sensor de distancia “Sharp”, para lo cual utilizamos los algoritmos PSO, ABC, DE y SCA para la solución de este problema de control y finalmente realizar una comparación para determinar cuál de estos algoritmos funcionaba mejor para la solución de este problema, donde teníamos en cuenta ciertos parámetros como numero de iteraciones máximas, las limites en los cuales deberían estar las constantes del controlador, en el cual se realizaba el siguiente procedimiento en el cual consiste en realizar varios experimentos de cada algoritmo los cuales se ejecutan dependiendo del número de interacciones dados y cada valor de las constantes K_p , K_i y K_d es evaluado en el controlador implementado en el Simulink de Matlab donde se encuentra la función de transferencia haya del sistema mediante la función “Ident” de Matlab luego calculábamos el valor de error cuadrático medio, límite de sobrepico permitido y error de estado estacionario los cuales se realizaba una suma ponderada con ellos y eran de vueltos al algoritmo como respuesta del sistema que se quiere optimizar y de esta forma continua el proceso interactivo hallando nuevos valores de contantes de control PID para la optimización del problema hasta la finalización del número de interacciones del algoritmo.

2 INFORME PASANTIA DE INVESTIGACION

Se desarrollo una pasantía de investigación en la universidad nacional de Brasilia (UnB) desde el 14 de agosto hasta 14 de diciembre del 2019, con la finalidad de realizar el trabajo de grado del estudiante y además fortalecer la relación con dicha universidad para posibles pasantías en el futuro. Durante la pasantía se trabajó en conjunto con los profesores Daniel Muños Arbeloa y el estudiante de doctorado Sergio Pertuz Méndez los cuales ayudaron

enormemente con el trabajo desarrollado estableciendo limites, asesorando y respondiendo dudas del estudiante.

También es importante resaltar la participación como estudiante invitado en la clase de maestría, sistemas Bioinspirados para la ingeniería donde se aprenden nuevos conceptos de control y técnicas para el desarrollo de proyectos de control de sistemas.

Adicionalmente durante esta pasantía se firmó formalmente el convenio entre la universidad de Pamplona y la universidad de Brasilia lo cual permitirá realizar aún más trabajos con en conjunto entre universidades y realizar más intercambio de personal científico como alumnos y profesores investigadores.

2.1 Laboratorio y equipos

Entre los diferentes laboratorios que existen el complejo de la facultad tecnológica de la universidad de Brasilia durante esta pasantía se hizo uso de algunos de los laboratorios y equipos que la universidad posee, entre los laboratorios destacamos el LEIA (Laboratory of Embedded Systems and Integrated Circuits Applications) que es donde se desarrolló la mayoría del trabajo, el laboratorio de herramientas de maestría y doctorado donde se tomaron materiales prestados y se hizo uso de las impresoras 3D, tenemos también el laboratorio de mecánica (Laboratorio do Graco) donde se hizo uso tanto del material disponible como de las diferentes equipos que están disponibles, también se recibió ayuda del personal de laboratoristas y profesores durante el desarrollo del trabajo.



Figura. 2. Laboratorio Graco de la universidad de Brasilia.



Figura. 3. Laboratorio LEIA de la universidad de Brasilia.



Figura. 4. Laboratorio de herramientas de mecatrónica de la UnB.



Figura. 5. Maquina CNC para mecanizado.



Figura. 6. Planta de calderas doble efecto automatizada.



Figura. 7. Brazo robótico de la empresa ABB en la UnB.

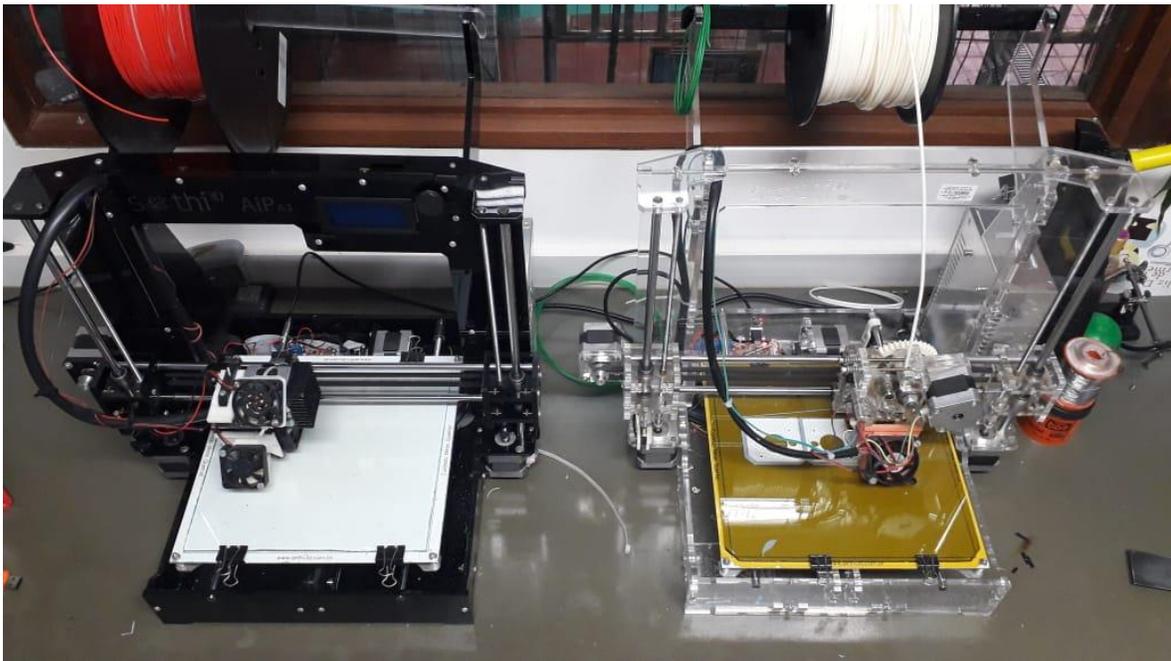


Figura. 8. Impresora 3D del laboratorio de mecatrónica en la UnB.



Figura. 9. Brazo robótico para trabajos de fresado de la empresa ASEA.

2.2 Estudio de sistemas Bioinspirados para la ingeniería

Durante el transcurso de la pasantía el estudiante asistió a la asignatura de maestría de sistemas Bioinspirados para la ingeniería, prestando por el profesor Daniel Muñoz Arboleda.

En esta clase los primeros temas tratados son acerca de optimización de sistemas y problemas de programación lineal lo cuales se pueden resolver de una forma analítica utilizando gráficas y álgebra para despejar ecuaciones o de forma más robusta con métodos de tabulación en los cuales se organiza el problema al sacar las ecuaciones del problema y esquematizarlas en una tabla que mediante un proceso iterativo podemos obtener los resultados deseados ya sea para realizar una maximización o minimización de objetivo, adicionalmente se trabajó con la Toolbox de programas como Matlab y Excel que permiten desarrollar estos problemas de forma más sencilla solo dando las ecuación con su herramienta solver.



Figura. 10. Aula de clases de la materia sistemas Bioinspirados para la ingeniería.

Seguidamente pasamos a una introducción de cómo es el comportamiento de algunos sistemas encontrados en la naturaleza que a pesar de parecer un comportamiento sencillo estos pueden realmente resolver problemas realmente complejos, como en el caso del algoritmo PSO el cual es un algoritmo que permite que los individuos de una población se pueden desplazar libremente por el espacio teniendo en cuenta que estas partículas poseen un cualidad individual y una social los cuales van a afectar como se desplazan en el entorno, también tenemos el algoritmo ABC que está basado en el comportamiento de las abejas en donde tenemos las abejas que buscan un nuevo objetivo y las abejas seguidoras las cuales siguen a las antes mencionadas dependiendo de los resultados obtenidos por ellas más un cálculo de probabilidad que les permite elegir cual abeja es la mejor seguir en determinados casos, otro algoritmo también importante es el DE o Diferential Evolution que tiene como medida clasificar por rangos las partículas que se encuentran en disposición luego de esto se realiza una mezcla o emparejamiento entre las partículas que dan como resultados nuevas partículas, finalmente se realiza un proceso de segregación en el cual se escogen las más aptas o las que poseen un mejor resultado.

Seguidamente en la materia empezamos a desarrollar métodos de diversidad artificial con la finalidad de que las partículas de cualquiera de los algoritmos mencionados puedan desplazarse de distintas formas al agregar diferentes características como el algoritmo por oposición que causa que las partículas que se encuentren a cierta distancia se dispersen como si se repelieran, también el método de atractivo repulsivo que causa que las partículas se acerquen y se repelan dependiendo de ciertas tasas de probabilidad y también de los resultados por las partículas.

Seguimos después con los test de hipótesis que son aquellos que nos permiten realizar un análisis estadístico de los resultados obtenidos de cada uno de los algoritmos con la finalidad de realizar una comparación en la cual podamos decir de forma estadística que algoritmo posee mejores cualidades que el otro y cuales ofrecen mejores resultados.

Seguimos después con los temas de aplicación de los algoritmos Bioinspirados en los sistemas y procesos de control en primera medida tenemos la explicación de cómo sintonizar un controlador PID a partir de algoritmos y sistemas Bioinspirados estudiados donde podemos obtener resultados en su mayoría muy buenos, también los métodos de como sintonizar los pesos de una red neuronal mediante los algoritmos estudiados donde encontramos buenos resultados obtenidos en los trabajos realizados durante las clases.

Y finalmente tenemos los algoritmos Bioinspirados multi objetivos los cuales se hacen cargo de problemas de optimización en las que debemos encontrar múltiples variables como respuesta del algoritmo estos tienen como característica ser procesos medianamente complejos y que requieren un alto costo computacional, no obstante, se obtiene resultados bastante satisfactorios, en caso de los algoritmos multi objetivo podemos mencionar MODE que el algoritmo multiobjetivo de la evolución diferencial, NSGA el algoritmo genético multiobjetivo.

2.3 Diseño mano robótica

Se realizó como parte de las actividades del trabajo el rediseño de piezas de una mano robótica de los laboratorios de mecatrónica de la universidad de Brasilia, los cambios consistían en reducir el tamaño del diseño actual, así como también adicionarle un nuevo sensor de posición SV01 de Texas Instruments el cual cumple la función de enviar una señal de voltaje dependiendo de la rotación de un eje.

2.4 Diseño de actuador robótico

Como foco principal del trabajo de grado se propone realizar un nuevo tipo de actuador robótico para ser incluido en un nuevo modelo de mano robótica que se desarrollara en la universidad de Brasilia.

El actuador robótico estará compuesto por una PCB (Printed Circuit Board) que contendrá un módulo de comunicación IIC, un driver de motor y un sensor de corriente. Además, llevará un sensor de posición y un motor DC. Este actuador será controlado a partir de un FPGA en la cual realizaremos los programas necesarios para su funcionamiento, en primera medida para hacer el proceso mucho más rápido y aprovechar las características de la FPGA se hace uso de los trabajos posteriores realizados por el equipo de investigación de la UnB y tomamos un programa de comunicación IIC implementado en el hardware de la FPGA realizado en

VHDL, no obstante, solamente así no nos servirá para la aplicación deseada por lo cual se toma el programa y se crea una nueva IP en la herramienta manager de VIVADO donde realizamos la adaptación del código para que se puede utilizar y comunicar con el procesador que posee la FPGA y así tendríamos una interfaz AXI IIC para nuestro trabajo, seguidamente configuramos los componentes que vamos a utilizar en el procesador incluyendo la “Zynq” de la FPGA y la interfaz AXI IIC que anteriormente creamos en la “IP integrator” que es donde configuramos los recursos que vamos a utilizar de la FPGA, luego de generar la síntesis del en el hardware podemos declarar los pines de comunicación IIC SCL y SDA.

Luego de generar el “bitstream” que es el complemento que nos permite programar y depurar los programas en la FPGA, podemos exportarlo y pasar a la herramienta SDK (Software Development Kit) en el cual podemos programar el procesador de la FPGA, entonces podemos comenzar a crear las librerías necesarias para el funcionamiento del de la comunicación IIC y el control de los dispositivos IIC que queremos controlar, en primer lugar creamos la librerías para llamar el módulo de comunicación AXI IIC que creamos anteriormente, para este creamos las funciones de lectura, escritura, espera y limpieza. Luego continuamos con las librerías para el control del dispositivo IIC de lectura, para el cual creamos una librería “Header” en la cual definimos todas sus constantes, parámetros, direcciones y configuraciones necesarias para el funcionamiento del módulo, seguimos luego con la creación de un programa en C donde tendremos las funciones de lectura, escritura, lectura de canal, lectura de comparación e canales. Luego tenemos la creación de la librería para el driver de motor que semejante a lo realizado con el dispositivo IIC a este también se le debe crear un programa “Header” con los mismos argumentos mencionados antes y se crea un programa en C con las funciones de escribir, leer, determinar velocidad, paro de movimiento, función break y una función de lectura para leer el estado del driver donde podemos encontrar varios datos como error de comunicación u otros problema que se presentaron durante su funcionamiento.

Finalmente, luego de lo anterior se realiza el diseño en el software CAD SolidWorks del actuador robótico con los tamaños y componentes mencionados anteriormente.

2.5 Proyecto Bioinspirados

Como parte de los trabajos de la materia del área de sistemas Bioinspirados para la ingeniería, se debe realizar un proyecto final de la materia, para este caso fue el control PID de un robot móvil optimizado mediante algoritmos Bioinspirados. Para esto se toma ayuda de un robot móvil disponible en los laboratorios de mecatrónica de la UnB, para tomar los datos necesarios para la generación del modelo matemático de la planta en forma de función de transferencia conseguida mediante Matlab al incluir los datos obtenidos entran en la función “indet” que nos retorna la función, ya teniendo esto podemos proceder a realizar el modelo de control clásico PID, donde dependiendo de los valores K_p , K_i y K_d , podemos hacer que la función de transferencia se comporte de forma que deseemos al establecer el punto de equilibrio o “setpoint” del sistema, para optimizar los parámetros del control PID usaremos

los algoritmos Bioinspirados para minimizar el error ponderado que está compuesto por error cuadrático medio, límite de sobrepico y el error en estado estacionario, a partir de estos parámetros se genera el error que queremos disminuir utilizando los algoritmos ABC, DE y SCA, con la finalidad de obtener los valores óptimos de controlador con cada uno para después realizar una comparación donde se determina cual brinda una mejor solución y es superior a los otros.

2.6 Artículo científico

Como parte de las actividades y condiciones de esta pasantía de investigación se realizó un artículo científico acerca del proyecto final que se realizó en la materia sistemas Bioinspirados para la ingeniería sobre el control de un robot móvil controlado por PID sintonizado con diferentes métodos de optimización Bioinspirados, esto con el objetivo de realizar una publicación en una revista científica como parte de los resultados de esta pasantía investigativa.

2.7 Convenio entre universidades

Se formalizo el papeleo legal para la realización del convenio entre las universidad de pamplona y la universidad de Brasilia, de esta forma permitiendo así realizar más intercambios de personal para la investigación científica ya sea para el caso de estudiantes o profesores, los cuales tendrán muchos beneficios a la hora de realizar trabajos en la universidad de Brasilia como hospedaje gratuito al igual que alimentación brindados por instituciones pertenecientes a la universidad de Brasilia, residencial colina UnB y Restaurante Universitario (RU) respectivamente. También se brindará apoyo técnico para el desarrollo de los proyectos de investigación como la maquinaria necesaria para la realización del trabajo, y ayuda del personal técnico o profesorado para cualquier trabajo que se desee realizar. De esta forma poder establecer una buena relación entre ambas universidades y continuar con el currículo académico e investigativo de las dos universidades.

3 MARCO TEÓRICO

3.1 Sistemas en Chip (SoC)

Son aquellos sistemas ya sean de comunicación (Wi-Fi, Bluetooth), de adquisición de datos, componentes de audio o video, que se encuentran embarcados en un solo chip compuestos por transistores del rango de tamaño de micros o manos metros. Estos generalmente realizan

trabajos de señales ya sean analógicas o digitales, poseen también varios tipos de memorias como tipo RAM o flash e inclusive un procesador, generalmente los SoC son utilizados como componentes para crear sistemas más complejos. [4] [5]

3.2 FPGA (Field Programmable Gate Array)

Las FPGA son dispositivos que están compuestos por “SoC” y poseen un procesador, pero adicionalmente siendo de los pocos dispositivos tienen un amplio espacio de hardware reconfigurable que los cuales pueden ser modificados para realizar diferentes procesos dependiendo de las necesidades, esto se realiza mediante la programación “VHDL” teniendo en cuenta que estos poseen como ventaja de poder realizar procesos muy complejos trabajando con números binarios y hacer esto en tiempos extremadamente pequeños del orden de los milisegundos pero con el costo de ser algo complejo su lenguaje de programación ya que se programa directamente el hardware, estos tiempos mencionados se logran gracias a lo mencionado anteriormente pero adicionalmente a que estos procesos complejos se pueden ejecutar en paralelo esto quiere decir que todos se realizan al mismo tiempo de forma que varias partes del hardware funcionan a la par para aumentar la velocidad de los procesos, estos programas realizados en hardware reciben el nombre de programas en hardware implementado, sin embargo estas al tener un procesador que puede compilar código en “C” podemos realizar programación en este y realizar comunicación entre los trabajos realizados en el hardware con los hechos en el procesador mediante software intermediarios que permiten configurar tanto sus dos formas de programación como sus entradas y salidas, demás componentes que conforman la FPGA y sus procesos internos. [6] [7]

3.3 Lenguaje VHDL, VHSIC y HDL

Es un lenguaje utilizado para la descripción del comportamiento de circuitos digitales y su automatización definido por la IEEE (Institute of Electrical and Electronics Engineers), las siglas VHDL son una reducción y conjugación de las siglas VHSIC (Very High Speed Integrated Circuit) y HDL (Hardware Description Language) los cuales describen cual es la función que desempeñan cada uno, gracias los dispositivos que permiten la descripción del hardware como las FPGA se pueden realizar aplicación y programas que se ejecuten extremadamente rápido, ya que gracias a las características y posibilidades que permiten los circuitos integrados podemos realizar procesos en paralelo para el funcionamiento de los procesos o de una tarea, la descripción consiste en darle tareas y definir variables de los conjuntos o partes de hardware o circuitos integrados para realizar un proceso, este es generalmente utilizado en PLD (Programmable Logic Device - Dispositivo Lógico Programable), FPGA (Field Programmable Gate Array), ASIC y similares. [1] [2] [3]

3.4 Procesadores

Un procesador es el centro de los sistemas de computación que manda las indicaciones de cómo debe funcionar cada uno de los componentes que los conforman, este está compuesto por un microprocesador que a la vez está compuesto por un chip compuesto por millones de transistores de tamaños entre micros y nano metros, entonces tenemos que entre mayor cantidad de transistores contenga el procesador más grande será su capacidad de realizar tareas en un tiempo más corto es decir aumentar su potencia, el microprocesador es el encargado de la parte matemática de los procesador y este último da las órdenes a los demás componentes. [8]

3.5 Comunicación AXI

La comunicación IP AXI permite conectar uno o más dispositivos maestros mapeados a una memoria AXI o a uno o más dispositivos esclavos mapeados en la memoria en los modelos realizados en FPGA, esta comunicación tiene las funciones de escribir la dirección del dispositivo conectado al canal, la función de escribir datos y enviarlos por el canal, la función de leer la dirección del dispositivo conectado al canal y finalmente leer los datos del dispositivo en el canal. Un módulo AXI conecta al maestro y al esclavo para poder comunicarlos y realizar los procesos correspondientes, de forma que si queremos controlar más dispositivos esclavos se puede realizar con la ayuda de varios módulos de comunicación AXI. [9]

3.6 Comunicación IIC o I2C (Inter Integrated Circuit)

Es un protocolo de comunicación serial creado en 1982 por Philips Semiconductors, que permite enviar una trama de datos mediante conexiones físicas entre dos dispositivos que generan y reciben señales digitales, la comunicación se realiza mediante dos cables de comunicación que son SDA que es el responsable de enviar los datos como direcciones o buffers teniendo ciertas consideraciones, y el SCL que hace la función de reloj para la transmisión de los datos, se debe tener en cuenta que para poder transmitir los datos del SDA se debe partir con una señal en alto en el SCL para que se haga efectiva la comunicación, cada vez que se realice una transacción exitosa el esclavo enviara al maestro un ACK (Acknowledge) como señal de que fue exitosa la comunicación, cabe resaltar que es esta comunicación tiene capacidad de transmisión máxima por paquetes de un byte (8 bits). [10] [11]

El proceso de funcionamiento de la comunicación IIC consiste en preparar una señal de inicio o START que se da cuando el SCL pasa a estado bajo y el SDA a un alto, después de esta señal se debe enviar la dirección del dispositivo esclavo (7 bits) y un bit de R/W de escritura

o lectura que nos indicara que proceso vamos a realizar en el caso de que sea escritura debemos enviar un vector de configuración indicando que registro vamos a modificar, y seguidamente podemos escribir los datos que se desean enviar y además podemos enviar una segunda tanda de datos luego de los enviados previamente, y se finaliza con una señal de STOP que es lo opuesto a la señal de START, todo este proceso es realizado por el dispositivo maestro y el esclavo solo envía una señal ACK cada vez que se cumple una transmisión de datos. Si el trabajo que se desea realizar es de lectura se realiza el mismo proceso hasta el punto donde al escribir la dirección del dispositivo luego ponemos el ultimo bit indicando que haremos una lectura en R/W, luego después el maestro debe enviar un vector de 8 bits indicando que el registro que vamos a leer y posteriormente se realiza el trabajo de lecturas en secciones de 8 bits, para finalizar la lectura se realiza la señal de STOP para finalizar el proceso. [12] [13]

3.7 Diseño CAD (Computer-Aided Design)

El diseño asistido por computadora es consiste en el uso de programas que permiten hacer diseños de componentes de un sistema mediante dibujos y planos, también se le conoce dibujo CAD ya que se basa (en la mayoría de ocasiones) en el dibujo de figuras geométricas en dos dimensiones con la finalidad de realizar piezas más complejas mediante el uso de diferentes herramientas que brindan este tipos de softwares, también permiten generar planos de piezas, realizar simulación de movimiento, análisis de fuerza o térmicos, entre otras aplicaciones. [14] [15]

3.8 Robótica

La robótica se puede definir que es la ciencia que estudia control y realiza un análisis de matemático de los robots, la robótica permite describir el movimiento de los robots mediante el estudio de su comportamiento cinemático y dinámico, adicionalmente estudia la forma en la cual deben ser controlados mediante su cinemática o dinámica para muchos casos de robots esto es lo más idóneo, sin embargo las nuevas técnicas de control inteligente como lógica difusa o redes neuronales promueven el control de robots sin la necesidad de un análisis matemático profundo, el área de estudio de la robótica es muy amplia tanto en la cantidad de aplicaciones como medicina, búsqueda y rescate, entretenimiento, educación, entre otras, así como también es un su cantidad de tipos y clases, robots móviles, autómatas, robots antropomórficos, drones, Bioinspirados, entre otros. [16]

3.9 Robótica antropomórfica

Los robots antropomórficos son aquellos robots Bioinspirados que tienen semejanza con los sistemas y funcionamiento de los cuerpos de los seres humanos, tales como brazos, piernas,

torso, etcétera, estos generalmente tienen aplicaciones médicas tipo prótesis para personas que presentan pérdida de alguna extremidad o también utilizados como robot de precisión para realizar cirugías médicas, también en el campo de la industria son muy utilizados para aplicaciones tipo ensamble, manipulación de objetos pesados o radioactivos entre otras. [17]

3.10 Robótica móvil

Es un área de estudio de la robótica basada en aquellos dispositivos robóticos que se pueden desplazar en el espacio, estos involucran los robots que se pueden mover por el suelo, el agua, el aire e inclusive en el espacio. Estos suelen ser usados en muchas aplicaciones tales como exploración, búsqueda y rescate, entretenimiento, investigación espacial, entre otras. [18] [19]

3.11 Sistemas de control y Control PID

Se define un sistema de control como un sistema que al recibir un impulso o entrada estos pueden generar una señal de salida, generalmente están compuestos de varios componentes que se pueden modelar mediante una función de transferencia que al recibir una señal de entrada esta genera una salida como respuesta a este estímulo. [20] [21]

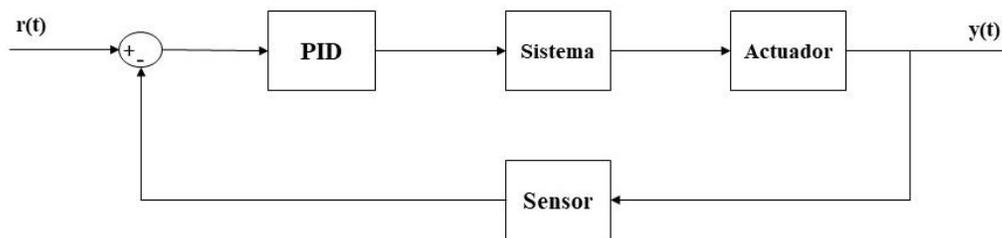


Figura. 11. Arquitectura de Sistema de control PID.

El control PID es una arquitectura de sistema de control que permite controlar la acción de un sistema es decir controlar como este se comporta a los estímulos o a las perturbaciones, esta arquitectura posee varios modelos en los cuales varía como se acomodan sus componentes y esto depende de la aplicación elegir cual puede ser más útil, pero generalmente se sigue una estructura estándar que se describe mediante, una entrada o “Setpoint” conectada con el bloque de control PID la cual va conectada al sistema que se desea controlar que se representa mediante una función de transferencia en la mayoría de los casos y seguidamente de la salida y una malla cerrada de retroalimentación cerrada que vuelve al PID con el fin de comparar la salida del sistema con el estímulo de entrada del “Setpoint”, esta técnica procura disminuir el error en estado estacionario el cual es la diferencia entre el Setpoint y la respuesta del sistema luego de un tiempo determinado,

mejorar el tiempo de levantamiento, y disminuir el sobre pico de la respuesta del sistema. [22]

$$c(t) = Kp \cdot e(t) + Ki \cdot \int_0^t e(t) \cdot dt + Kd \cdot \frac{de}{dt} \quad (1)$$

La ecuación de controlador PID está conformada por las constantes Kp, Ki y Kd, las cuales van multiplicadas por el error de la diferencia del Setpoint con la respuesta del sistema, la integral del error que comúnmente se llama error acumulado y la derivada del error mencionado respectivamente. [23] [24]

3.12 Sistemas Bioinspirados e inteligencia colectiva

Son aquellos sistemas que pretenden a semejar su comportamiento a los sistemas biológicos y a los sistemas que podemos encontrar en la naturaleza, ya sea en su parte interna o estructural o solamente a su apariencia física, generalmente vienen acompañados de ecuaciones matemáticas que describen su comportamiento, como ejemplo tenemos las redes neuronales. [25]

Como un subconcepto tenemos la inteligencia colectiva que hace referencia a que el comportamiento simple de organismos pequeños que hacen parte de una colonia más grande pueda resolver tareas más complejas a pesar de la simpleza de los individuos que la conforman, como ejemplo tenemos las hormigas, las abejas, el ADN, entre otras. [26] [27]

3.12.1 Algorithm PSO (Particle Swarm Optimization)

Es una técnica de optimización de sistemas basada en el movimiento de partículas en un espacio de búsqueda donde se pretende encontrar el mínimo global de toda la función en la que se encuentra, esta funciona mediante la generación de cierta cantidad de partículas que pueden estar dispersas en varias dimensiones con el objetivo de optimizar más de un parámetro, estas están basadas en una ecuación donde se realiza la posición en la cual se encuentra actualmente la partícula multiplicado por un factor de inercia que permite el desplazamiento llama condición individual, más una condición social la cual tiene la partícula con la mejor posición que optimiza el sistema. [28]

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 \cdot U_{1j}[0,1] \cdot (y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2 \cdot U_{2j}[0,1] \cdot (y_{sj}^{(t)} - x_{sj}^{(t)}) \quad (2)$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (3)$$

Tenemos la ecuación 2 de velocidad la cual toma el parámetro de velocidad actual y lo suma por la multiplicación de una constante “c1” la cual es un valor variable que se define al principio del programa, un valor aleatorio entre cero y uno “U”, los valores “y” y “x” representan la partícula y la dimensión en la que se encuentra respectivamente, más la suma de “c2” que ocurre el mismo caso que con “c1”, nuevamente el vector aleatorio y los valores “ys” y “xs” son la posición de la partícula y su dimensión de la que posee el valor que optimiza la función actualmente. Finalmente, la ecuación 3 indica como se cambia la posición de la partícula actual que es su posición actual más la velocidad antes calculada. [29]

3.12.2 Algorithm ABC (Artificial Bee Colony Optimization)

Es un algoritmo basado en el comportamiento de las colonias de abejas para la búsqueda de polen, las cuales consisten en dos tipos de abejas las exploradoras y las seguidoras, las abejas exploradoras se encargan de buscar nuevas fuentes de polen para la colonia, las cuales al volver a la enjambre indican a las demás abejas donde se encuentra la nueva fuente, las abejas seguidoras son las que se encargan de seguir a las abejas exploradoras siempre que hayan encontrado un buena fuente de alimento esto generalmente se representa con un factor de probabilidad para que la seguidora elija a cual exploradora seguir o también de la intensidad de la exploradora para indicar la posición de la fuente. [30]

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + \phi_{ij} \left(x_{ij}^{(t)} - x_{kj}^{(t)} \right) \quad (4)$$

Tenemos en la ecuación 4 conformada por “x” que indica la variable de la abeja analizada para su cambio de posición tomamos la posición actual e la misma más un numero aleatorio con distribución uniforme “φ” y donde “k” y “j” son aleatoriamente escogidos para determinar a cuál abeja seguir donde “k” es diferente de “i”. [31]

3.12.3 Algoritmo DE (Differential Evolution)

Este algoritmo fue desarrollado a partir del algoritmo genético, tomando sus características más importantes y simplificando partes de sus procesos, este algoritmo consiste en la organización de la partículas a partir de un ranquin que posteriormente será modificado,

luego de ranquear las partículas pasamos a una etapa de mutación donde se cambian los valores de algunas partículas mediante un numero aleatorio, luego se empieza una etapa de crossover en la cual se cruzan las partículas para la creación de otras nuevas, y finalmente una etapa de selección donde se escogen una parte de las partículas de cada ranquin y también las que optimizan el sistema de una mejor manera. [32]

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} + x_{r3,G}) \quad (5)$$

En la ecuación 5 encontramos la ecuación que dictamina la mutación de cada partícula donde tenemos el valor “v” que se puede describir como la velocidad, “x” que corresponde a la posición de la partícula actual, donde r1, r2, r3 son indicadores distintos, y “F” es un valor de peso diferencial constante que va en el rango de cero a dos [0,2]. [33]

3.12.4 Algoritmo SCA (Sine Cosine Algorithm)

El algoritmo SCA está basado en el comportamiento del seno y del coseno dependiendo de un parámetro aleatorio las partículas se mueven en uno de estos dos comportamientos, este tiene la característica de que permite que las partículas se desplacen de forma más suave y no tan errática como otros algoritmos permitiendo así desempeñar otro tipo de aplicaciones. [34]

$$x_i^{t+1} = \begin{cases} x_i^t + r1 \cdot \sin(r2) \cdot |r3P_i^t - x_i^t|, & r4 < 0.5 \\ x_i^t + r1 \cdot \cos(r2) \cdot |r3P_i^t - x_i^t|, & r4 \geq 0.5 \end{cases} \quad (6)$$

Tenemos la ecuación que representa el funcionamiento del algoritmo SCA, donde vemos que la nueva posición de la partícula depende de “r4” el cual es un valor aleatorio, de esta manera se sabe cuál de los dos comportamientos va adoptar la partícula ya sea seno y coseno, los valores de “r2” y “r3” son valores que se cogidos por un valor aleatorio multiplicados por dos pi, “r1” se puede tratar como el radio del círculo que permite el movimiento este depende de un valor constante y por el número de iteraciones del proceso y donde “P” es el vector de valores óptimos para realizar la diferencia con la posición actual de la partícula. [35]

3.13 Test de hipótesis

El test de hipótesis permite realizar un análisis estadístico de los datos de varias poblaciones, que nos permite comparar y saber diferentes características de las poblaciones analizadas, algunos de estos test pueden ser, el test de “Kolmogorov-Smirnov” que nos permite plantear

dos hipótesis H_0 : x es una distribución normal, H_1 : x no es una distribución normal, este test es realizado a poblaciones independientes de forma individual. También tenemos el test “Wilconxon” que nos permite tener dos hipótesis H_0 : Dos muestras independientes, en los vectores X e Y , provienen de distribuciones con medianas iguales, H_1 : X e Y provienen de distribuciones con medianas diferentes, este test se realiza a para realizar la comparación de muestras independientes y podemos observar cual tiene un mejor comportamiento. Finalmente tenemos el test de “Kruskal Wallis” que tienen dos hipótesis H_0 : las muestras independientes de dos o más grupos provienen de distribuciones con medianas iguales H_1 : al menos un grupo proviene de distribuciones con medianas diferentes O el resultado será $p \sim 0$ o $h = 1$ rechazar H_0 con un significado del 5% (nivel de confianza de 95%), este test se realiza a todas las muestras de la población. [36] [37]

3.14 Impresión 3D

Es un tipo de manufactura de prototipado rápido también conocida como fabricación aditiva para la fabricación de piezas de modelado 3D (Tecnología de fabricación por capas), que consiste en que mediante un extrusor caliente permite la salida de un material en forma de hilo salga de forma muy suave y se quede pegado a la superficie de la maquina y mediante su movimiento se puedan realizar formas y figuras de todos tipo de tamaños teniendo en cuenta los límites de la máquina. [38]

4 DESARROLLO DE ACTUADOR ROBÓTICO

4.1 Vivado Xilinx

En primer lugar, para poder realizar el proyecto debemos tener el programa Vivado de la empresa Xilinx que es la que nos permite controlar tanto el procesador de la FPGA así como también su parte de hardware para implementación en este posee librerías de muchas FPGA's lo cual permite realizar muchas aplicaciones mediante estos dispositivos utilizando este programa, también posee un buen soporte y ejemplos de funcionamiento de sus diferentes librerías para la comunicación con sus herramientas mapa de bloques, para nuestra aplicación usaremos la FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board.

4.2 Mapa de bloques y creación IP

El mapa de bloques se debe crear ya sea para poder utilizar el procesador de la FPGA o para crear la conexión entre el procesador y los programas creados en el hardware de

implantación, con este se pueden hacer numerosas aplicaciones utilizando los diferentes componentes que posean las FPGA's, como tipos de comunicación (IIC, Bluetooth, Wi-Fi) o usos de sus propias características como pulsadores, switches o leds.

Para la creación del bloque IP que posee los códigos en VHDL desarrollados por los estudiantes de maestría de la UnB nos dirigimos al manager IP de software VIVADO en el cual nos podemos crear librerías basados en este tipo de código realizado en VHDL, luego se le asigna un nombre a la nueva IP en este caso, "AXI_IIC3_0" ya que es la tercera versión realizada de la misma, luego al crear el paquete IP seleccionamos crear una nueva AXI4 IP la cual genera ciertas configuraciones en el código de muestra que nos permite incluir más fácilmente los programas realizados anteriormente, se nos muestra una vista predeterminada de la IP y seguimos con editar IP, en este punto debemos adicionar a la carpeta los programas desarrollados en VHDL para la comunicación IIC de esta forma editamos el código muestra de AXI4 IP para comunicar ambos programas, finalmente le damos los últimos parámetros de configuración como lo son seleccionar el dispositivo de compatibilidad para este caso la FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board (zynq), la frecuencia mínima y máxima del input Clk (25MHz y 150MHz) y la frecuencia del Bus Clk (1KHz y 1MHz) y los puertos e interfaces (SCL y SDA).

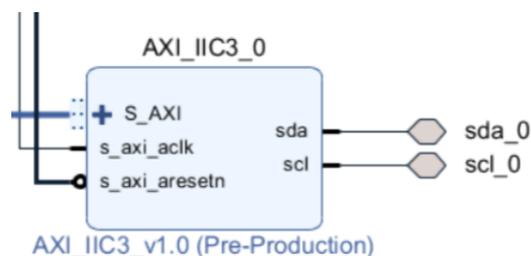


Figura. 12. Modulo IP realizado para la comunicación AXI IIC en VHDL.

Para nuestro caso de aplicación realizaremos un mapa o diagrama de bloques compuesto por el procesador, una interfaz AXI de comunicación IIC y los demás bloque pertinentes para el funcionamiento de los componentes propuestos, debemos tener en cuenta que la interfaz AXI IIC mencionada es una programa implementando en el hardware de la FPGA por razones de optimización de recursos, teniendo esto en cuenta es necesario agregar el programa al diagrama de bloque con el proceso de creación de librerías IP de Vivado, para realizar este proceso solo debemos tener el programa en VHDL crear una nueva IP y añadir los programas VHDL con ciertas modificaciones y de esta forma poder agregarlo al mapa de bloques para la comunicación con el procesador "Zynq".

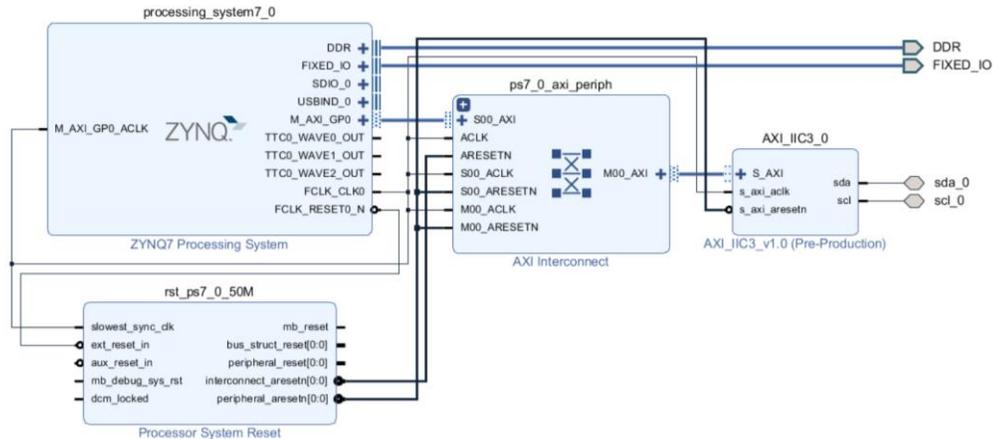


Figura. 13. Diagrama de bloques para la comunicación del procesador con el hardware de la FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board (zynq).

Para enviar las señales SCL y SDA a través del hardware de la FPGA necesitamos declarar los pines de comunicación esto se realiza al completar los procesos de síntesis e implementación en el software VIVADO que permiten realizar un archivo de restricción donde podemos declarar los pines de la FPGA siendo de SCL el pin V12 y SDA el pin V13 de la FPGA Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board.

4.3 Librería AXI IIC

Luego de haber realizado el diagrama de bloques procedemos a realizar una síntesis en la FPGA y un “bitstream” que nos permite añadir las direcciones de las de los dispositivos del diagrama de bloques. Entonces procedemos a ejecutar el SDK que una herramienta de Vivado que nos permite programar el procesador.

Entonces una vez en el SDK podemos realizar el programa de la librería AXI IIC que permite comunicarnos con el procesador y de la misma forma con el bloque de hardware implementado AXI IIC, en este se crearan dos programas, un “header” y otro programa en C donde estará el programa de las funciones, el primero solamente definirá algunas constantes necesarias y definirá las funciones que vamos a utilizar que son cuatro, write (escribir), read (leer), wait (esperar) y clear (limpiar), en el segundo programa realizaremos las funciones comenzando por la de escribir “xil_iic_write” donde utilizaremos la función “Xil_out32” de la librería “Xil io. h” la cual es una librería que nos ofrece la herramienta SDK para enviar conjuntos de datos de forma fácil al procesador, en este caso esta función permite enviar hasta un tamaño de 32 bits, de esta forma la función de escribir recibirá los parámetros de la dirección del módulo AXI IIC, la dirección del dispositivo de comunicación IIC, un buffer de datos para enviar, y un tamaño de buffer, en primer momento se llama la función “Xil_out32” la cual toma como parámetros la dirección del módulo AXI IIC más un

registrador “r0_offset” y como segundo valor la dirección del dispositivo IIC, posteriormente se envía la dirección del módulo AXI IIC más un registrador “r1_offset”, más el valor del buffer que se desea enviar.

Luego tenemos la función para leer “xil_iic_read” la cual funciona mediante la escritura de con la función “Xil_out32” para escribir la dirección del módulo AXI IIC más el registrador “r0_offset” junto con la dirección del dispositivo IIC, seguidamente utilizamos la función “Xil_In32” la cual permite adquirir los datos mediante un nuevo buffer, esta función solo necesita como parámetro la dirección del módulo AXI IIC más un registrador “r2_offset” para leer los datos enviados desde el dispositivo esclavo IIC.

4.4 Librería ADS1115

Entre los dispositivos que utilizaremos como componentes para el actuador robótico es el ADS1115 el cual es un dispositivo que permite leer cuatro canales analógicos, con tamaño SMD y además poder leer la diferencia entre cada uno de ellos, este dispositivo se comunica a partir del protocolo de comunicación IIC y es fabricado por la empresa Texas Instruments.

Para la creación de la librería usaremos dos programas un “Header” y un programa en C++, el primero está compuesto por las constantes necesarias para el funcionamiento del módulo IIC entre estas se definen su dirección, los vectores de configuraciones y registros, las direcciones para leer cada canal y sus comparadores entre cada uno y la definición de una clase donde se declaran las funciones que vamos a utilizar en la librería, como son leer un solo canal, leer el diferencial entre dos canales y obtener una lectura del estado del dispositivo. Para el segundo programa nos concentraremos en la lectura de los canales individuales ya que son los que necesitamos para la ejecución del proyecto, de esta forma nos concentraremos en la función de leer un solo canal “readADC_SingleEnded”, la es una función con un solo parámetro de entrada que es el número del canal que deseamos leer, esta consiste en seleccionar la dirección previamente definida en el “Header” dependiendo del número del canal definido como entrada, de esta manera seguimos con la llamada de una función llama definir registro la cual recibe como parámetros la dirección del ADS1115 y un vector de configuración que indica que registro vamos a leer este está definido en el “Header”, la función mencionada lo que hace es organizar los datos en un solo buffer y enviarlos a la por la función “xil_iic_write” descrita anteriormente, luego realizamos una pausa durante ocho micro segundos y llamamos la función leer registro, la cual toma como parámetros la dirección del ADS1115 y un vector de configuración de registro la cual permite leer los datos enviados por el ADS1115.

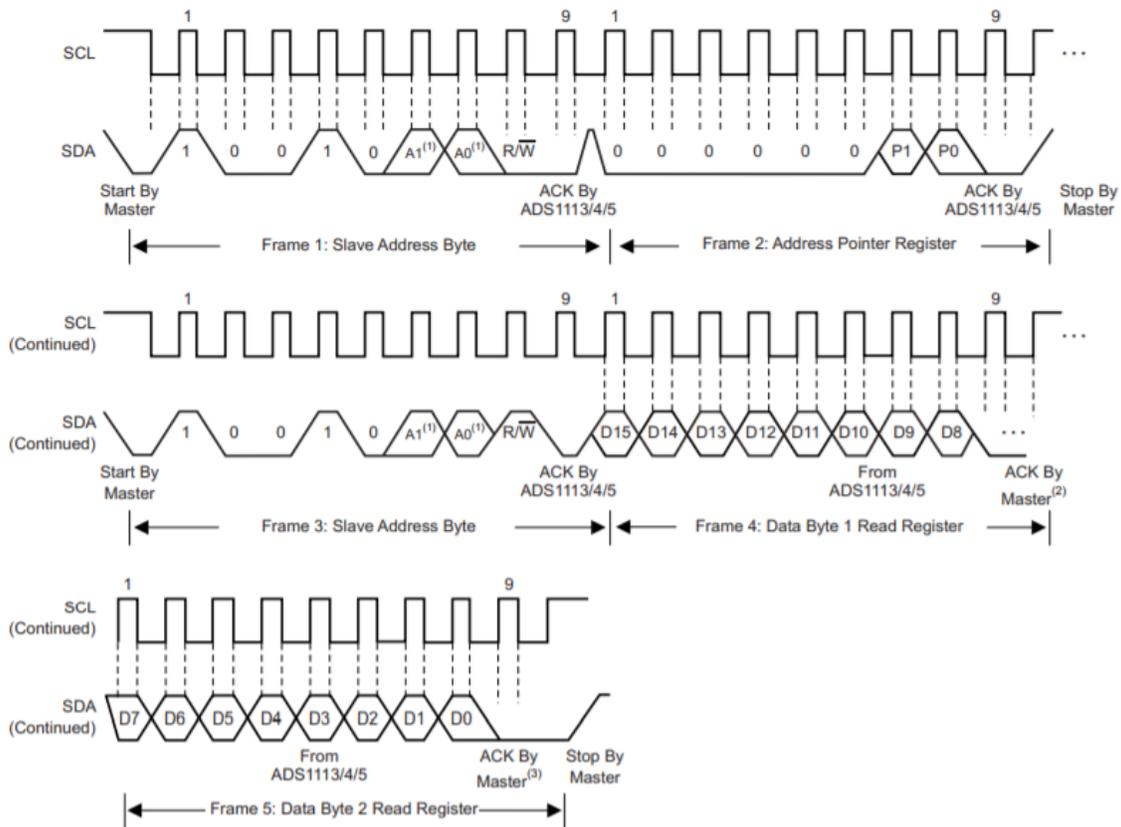


Figura. 14. Descripción de la comunicación IIC del dispositivo IIC ADS1115 de la empresa Texas Instruments.

4.5 Librería DRV3330

El dispositivo DRV3330 es un driver SMD que nos permite controlar motores de corriente directa con un voltaje máximo de siete voltios, este dispositivo posee una comunicación IIC para la transmisión de datos con otros dispositivos.

Para la creación de la librería de este dispositivo crearemos dos programas un “Header” y un programa en C++, el primero contendrá la definición de parámetros para el correcto funcionamiento de la comunicación IIC, como condiciones de inicio y parada de la comunicación, condición de falla y de stop, además se definen las una clase con las funciones que vamos a utilizar para la comunicación como son “drive” para mover el motor en alguna de las dos direcciones de dependiendo de un valor de velocidad, “Stop” para detener el movimiento, “Break” para la condición de alto en los dos canales del motor, y “GetFault” que sirve para leer los erros de comunicación con el dispositivo.

El segundo programa consta de la descripción de las funciones declaradas para la comunicación comenzando el DRV3330 ya que este puede cambiar su dirección de dependiendo del modo de conexión de uno de su pines, después tenemos la función “drive” la

cual solo requiere un valor entero de velocidad, y consiste en tomar el valor de la velocidad en el cual dependiendo de su signo determina el sentido de giro del motor y su velocidad varia en números enteros de cero hasta sesenta y tres luego llamamos a la función escribir la cual recibe el valor de un registro “0x00”, más un buffer en este caso la velocidad y el tamaño del buffer, esta función lo que hace es ordenar los datos dados y enviar los como buffer a la función “xil_iic_write” descrita anteriormente. Luego tenemos la función “GetFault” la cual consiste en enviar un registro “0x01”, un buffer “0x00” más un tamaño de buffer “1” lo cual se envía a la función “xil_iic_read” para poder leer el estado del dispositivo IIC. Luego seguimos con la función “Stop” la cual consiste en enviar un registro “0x00”, un buffer “0x00” y un tamaño de buffer “1” usando la función “xil_iic_write”, finalmente tenemos la función “break” que envía un registro “0x00”, un buffer “0x03” y un tamaño de buffer “1” para poder realizar la comunicación con el dispositivo IIC mediante la interfaz AXI IIC.

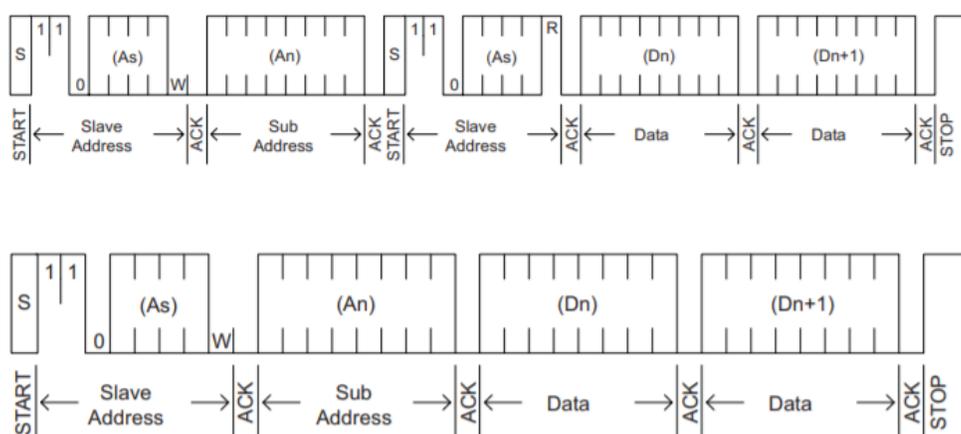


Figura. 15. Descripción de la comunicación IIC del driver DRV3330 de la empresa Texas Instruments.

4.6 Diseño CAD y Fabricación

Para el diseño CAD del actuador robótico fue realizado en el software SolidWorks, donde se hizo un modelo tipo caja por la cual sobre sale un eje con un acople lo cual nos permitirá ensamblar diferentes herramientas o ponerse acomodar en diferentes sistemas, realizamos un espacio para acoplar el motor, y al otro lado acomoda una PCB la cual contiene los dispositivos IIC ADS1115, DRV3330 y un sensor de corriente INA2540 mediante tornillos, más se agregan engranes rectos al motor y al eje permitiendo la movilidad del sistema, en una de las paredes de los eje se adiciona un sensor de posición SV01, y finalmente a la salida del eje por cada extremo de la caja se adicionan rodamientos que facilitan la movilidad del mismo.

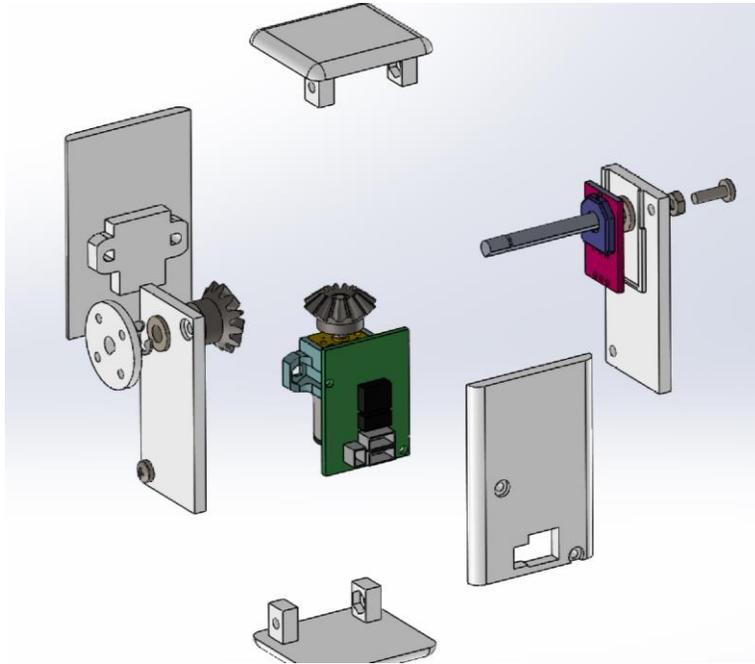


Figura. 16. Vista de explosión del diseño del actuador robótico.

Este diseño se compone por una placa PCB la cual contiene los dispositivos SMD ADS1115 para la lectura de canales analógicos, el driver de motor de corriente directa DRV3330 y un sensor de corriente INA240. Luego tenemos en una pared el sensor de posición SV01, dos engranes de cónicos de doce dientes, un motor de corriente directa, un eje de latón, dos rodamientos y dos soportes para ejes en los extremos, todo acompañado por tornillos y tuercas de tres milímetros.

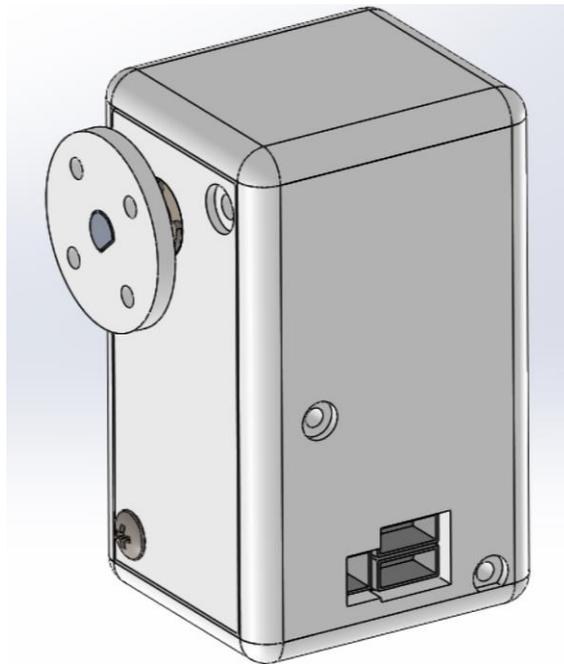


Figura. 17. Vista isométrica del diseño del actuador robótico realizado en SolidWorks.

Para la fabricación del actuador robótico hacemos uso de variedad de herramientas y personal facilitados por la universidad de Brasilia, como material para la fabricación como puede ser latón, aluminio, PLA, ABS y PEGT (estos últimos materiales para impresora 3D), mas también se hizo uso de los recursos económicos de la universidad para la compra de los motores de corriente directa, los diferentes sensores y dispositivos para el proyecto, se hizo uso de herramientas de prototipado rápido como lo fue la impresora 3D y de los elementos de fabricación más clásicos como el torno.

4.7 Resultados

Como resultados obtenemos el modelo de un actuador robótico funcional el cual puede controlar su velocidad y su sentido de giro, también permite leer datos de posición o velocidad del actuador y leer datos de corrientes de consumo causados por el motor de corriente directa, Implementando una FPGA la cual genera un gran beneficio en velocidad de funcionamiento, confiabilidad y mayor resistencia a fallas.

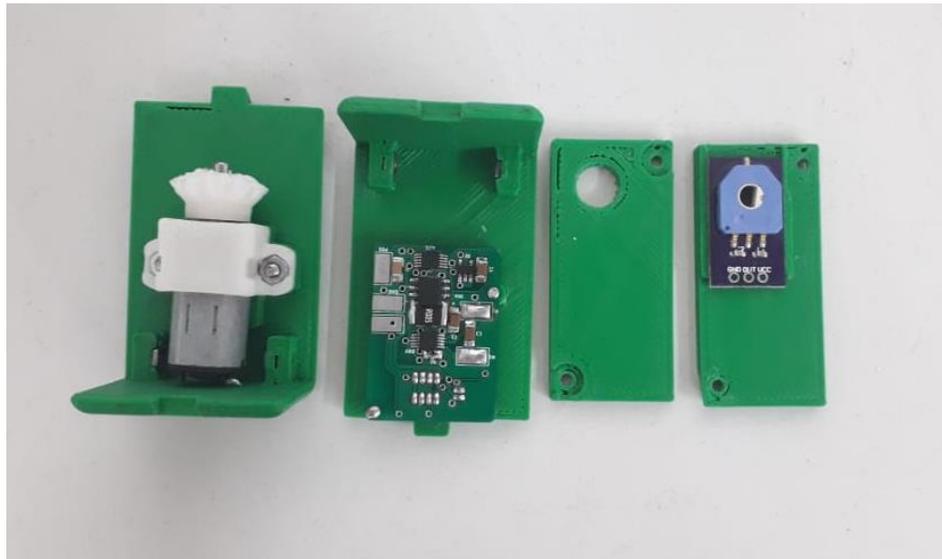


Figura. 18. Actuador robótico desarmado para la visualización de sus componentes.

Este actuador posee unas dimensiones con un ancho 28.6 milímetros, un largo de 32.5 milímetros y un alto de 53.92 milímetros, mas un eje de latón con diámetro de 4 milímetros y un largo de 46 milímetros, todos los tornillos y tuercas usados son de medida M2.5, y también se posee un acople para el eje de con diámetro de 20 milímetros con cuatro orificios para tornillos de 2 milímetros.



Figura. 19. Actuador robótico ensamblado para la realización de pruebas.

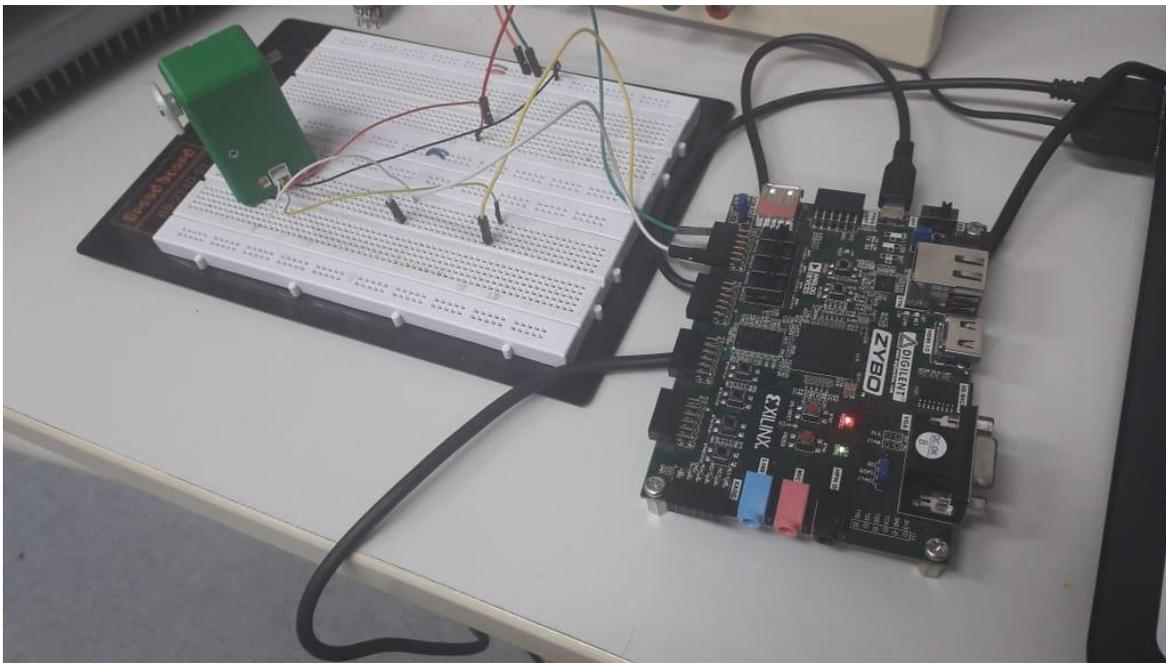


Figura. 20. Montaje del actuador robótico con FPGA Zybo, fuente de poder y osciloscopio para las pruebas de funcionamiento.

Se realizan pruebas del actuador robótico para llevar registro y tener constancia del funcionamiento del mismo para comprobar las señales de comunicación IIC emitidas por la FPGA Zybo usaremos un osciloscopio que permite visualizar y decodificar en hexadecimal las señales tipo IIC, de esta manera podemos llevar documentación y hacer medidas de nuestra comunicación.

Seguidamente nos desplazamos a el software de vivado a su herramienta de código en C, el entorno SDK donde se está realizando la comunicación con el módulo AXI IIC creado en VHDL donde se realizó la prueba de las señales emitidas por el SCL y SDA con la programación en la FPGA donde se aprecia que funciona adecuadamente, en la primera imagen se ve como es la señal cuando se mueve el motor en un sentido de giro, y en la segunda podemos ver cómo es la señal cuando se manda el stop al motor.

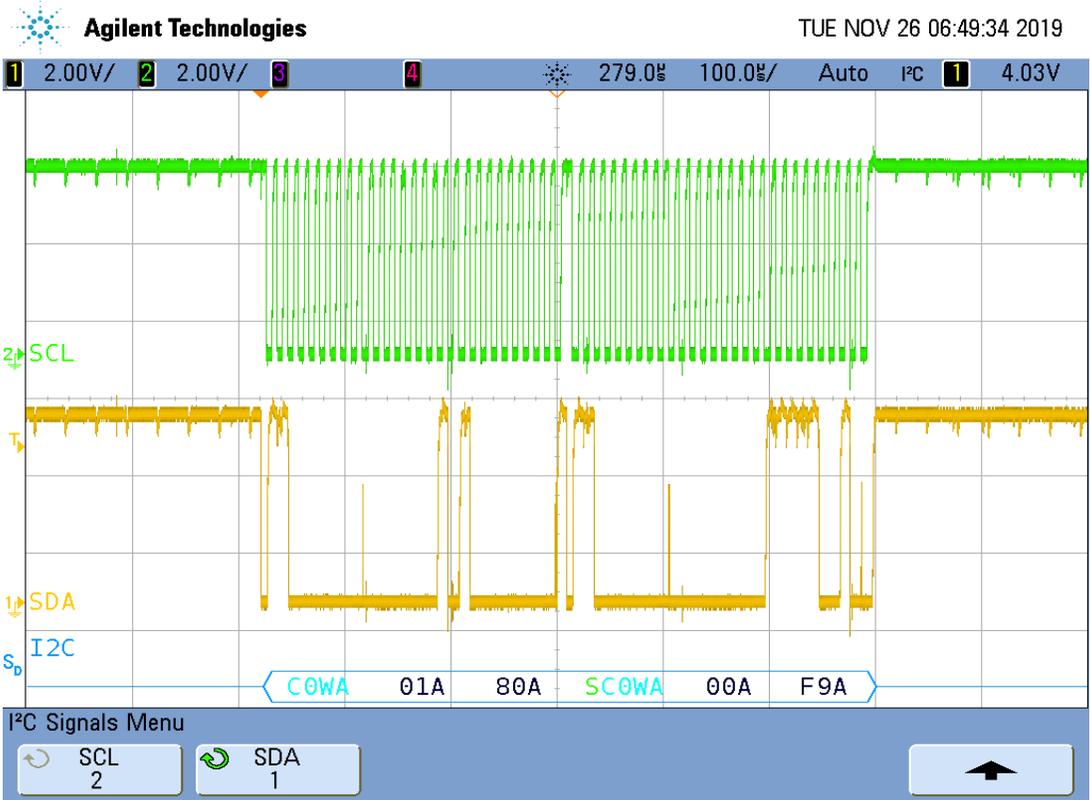


Fig. 21. Movimiento en un sentido de giro del actuador robótico usando la FPGA Zybo.

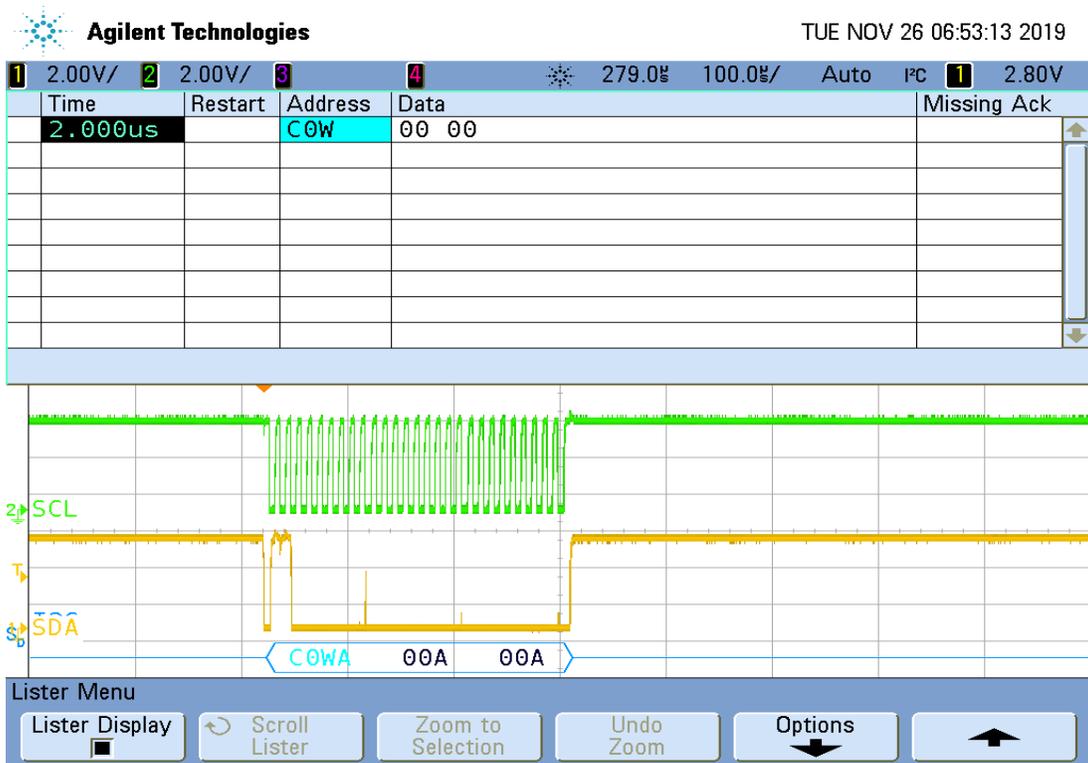


Fig. 22. Señal de stop del emitida por la FPGA Zybo al actuador robótico.

Luego tenemos los resultados obtenidas por el osciloscopio después de modificar el código en VHDL para la creación de un nuevo bloque de comunicación AXI IIC y haciendo un nuevo código en C para la comunicación con el dispositivo ADS1115 donde obtenemos las siguientes imágenes.

Sin embargo, las primeras pruebas realizadas fueron mediante el código en VHDL y una implementación en VIVADO para probar solamente las señales emitidas por la FPGA Zybo en los pines configurados como SCL y SDA en el cual obtuvimos un resultado positivo en el que se enviaban las señales deseadas atreves de los pines.

Después se realizó el nuevo código en C en la herramienta SDK de VIVADO para poder trabajar de forma óptima y probar la comunicación deseada mediante el IIC, a la cual se le realizaron varias modificaciones.

Cabe resaltar que en estos programas envinaban la señal completa independientemente si recibían el acknowledge por parte del dispositivo IIC o no, se plantea para trabajos futuros incluir esta característica de que solo se envíen los mensajes respectivos si el dispositivo IIC envía el acknowledge.

Seguidamente trasladamos el código corregido tanto los documentos creados en VHDL para la creación del módulo de comunicación AXI IIC y el programa desarrollado en C para realizar pruebas con la PCB con los componentes y módulos IIC SMD, de los cuales obtenemos como resultado de que funciona de forma óptima y no presenta ningún inconveniente.

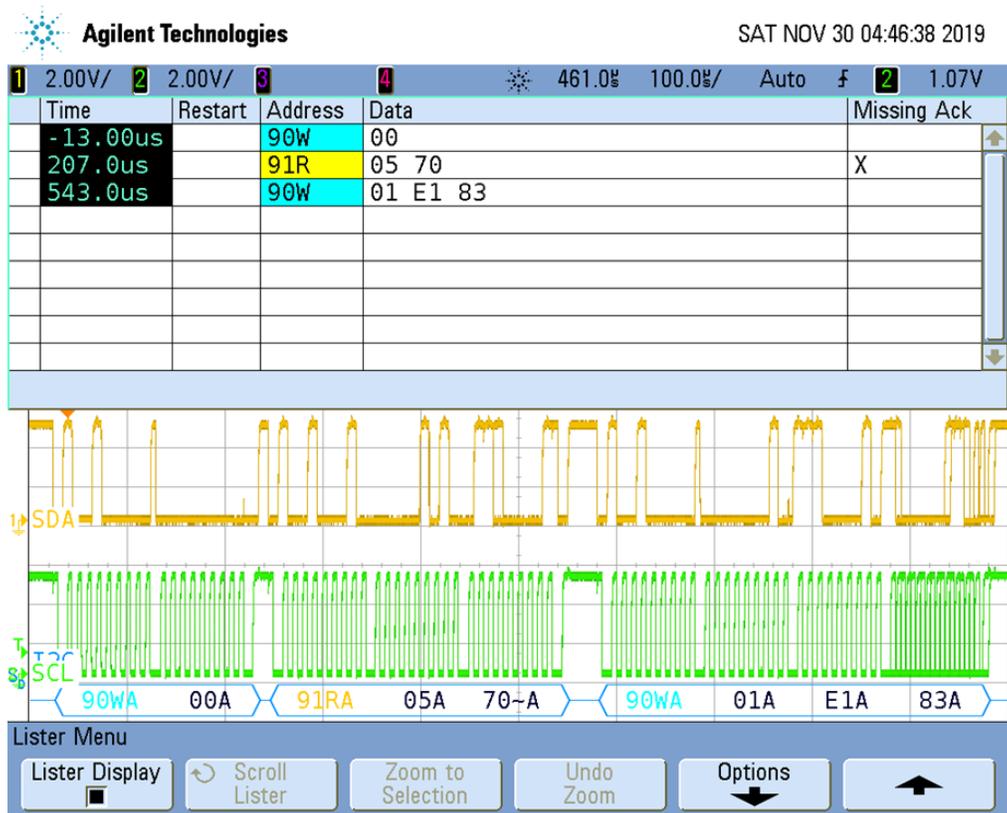


Fig. 23. Señal del osciloscopio de SDA y SCL dadas por la FPGA Zybo y por el módulo ADS1115.

Se muestra que el tiempo entre las transacciones es de 30.5 microsegundos que es un dato importante para posteriores comparaciones con otros programas como por ejemplo con los realizados en VHDL.



Fig. 24. Tiempo entre transacciones es de 30.50 micro segundos (uS).

Finalmente tenemos la propiedad intelectual de la aplicación del módulo AXI IIC realizado por los estudiantes de maestría de la universidad de Brasilia y librerías para el control de dispositivos IIC como ADS1115 y DRV3330, estos programas pueden ser utilizados por otros integrantes del grupo de trabajo de maestría y doctorado de la universidad de Brasilia.

5 DISEÑO DE MANO ROBÓTICA

5.1 Resumen y Componentes

Como parte de los trabajos extras realizados en la universidad de Brasilia, se propone realizar un rediseño de un modelo de mano robótica disponible en los laboratorios de mecatrónica en la universidad de Brasilia, los componentes que se agregaran al nuevo diseño de mano robótica son un sensor de posición SV01 fabricado por la empresa Murata y un eje que atraviese los nudillos de la mano robótica hechos con latón.

5.2 CAD

Para el diseño CAD recibimos unos archivos de la mano robótica anterior los cuales deben ser modificados para el nuevo modelo, tenemos como trabajo reducir el tamaño del diseño ya que se desea reducir el diseño existente, y también se debe acoplar un nuevo sensor de posición ya que el establecido era un tipo de potenciómetro el cual era muy frágil y de gran tamaño que causaba inconvenientes a la hora de ensamblar.

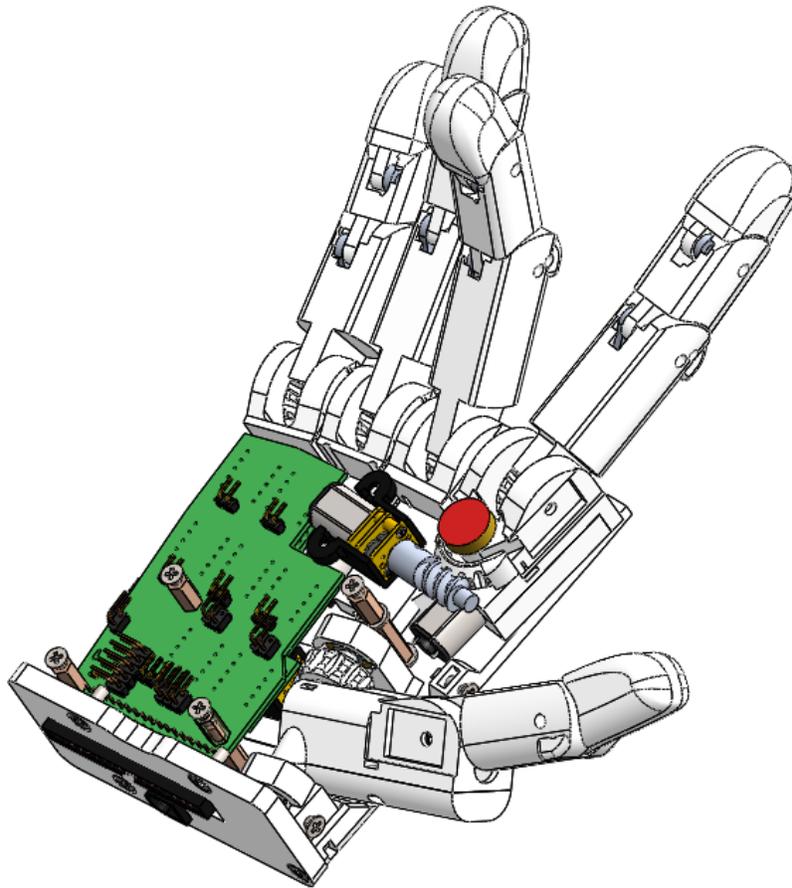


Fig. 25. Rediseño del CAD de modelo de mano robótica realizado en SolidWorks.

Se realiza una disminución significativa de grosor de los dedos de la mano robótica y se acopla un nuevo sensor de posición más robusto como lo es el sensor SV01 de la empresa Murata y resistente a la fricción gracias a la adición de rodamientos en el sistema, todo esto sostenido mediante ejes pequeños que se acoplan al diseño y al sensor para un funcionamiento más robusto.

5.3 Fabricación

la fabricación de los componentes de la mano robótica se lleva a cabo principalmente por impresoras 3D las cuales realizan los cinco dedos de la mano, la palma y el soporte trasero de la mano de acuerdo al diseño CAD realizado, y finalmente con la ayuda de los laboratoristas del laboratorio de mecánica nos ayudan con la fabricación de los ejes de latón que conforman parte de los nudillos y se acoplan con el sensor de posición SV01.

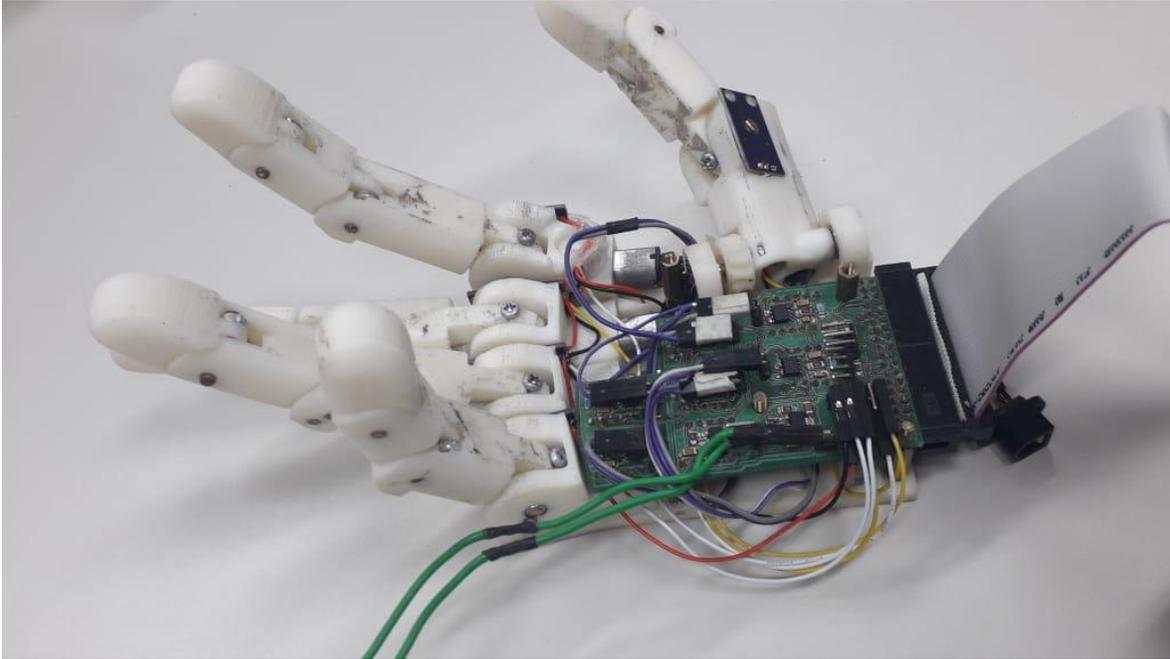


Fig. 26. Modelo de la mano robótica ensamblado para realizar pruebas por estudiantes de maestría y doctorado de la UnB.

5.4 Resultados

Como resultado obtenemos el nuevo modelo de mano robótica el cual acopla un nuevo sensor de posición y un diseño más pequeño y robusto mediante la adición de ejes fabricados de material de latón los cuales son superiores que al diseño que se poseía anteriormente, cumpliendo así con los requerimientos dados por el tutor de trabajo de grado. Este modelo de mano robótica será implementado en nuevas aplicaciones de control utilizando control de impedancia y sistemas Bioinspirados.

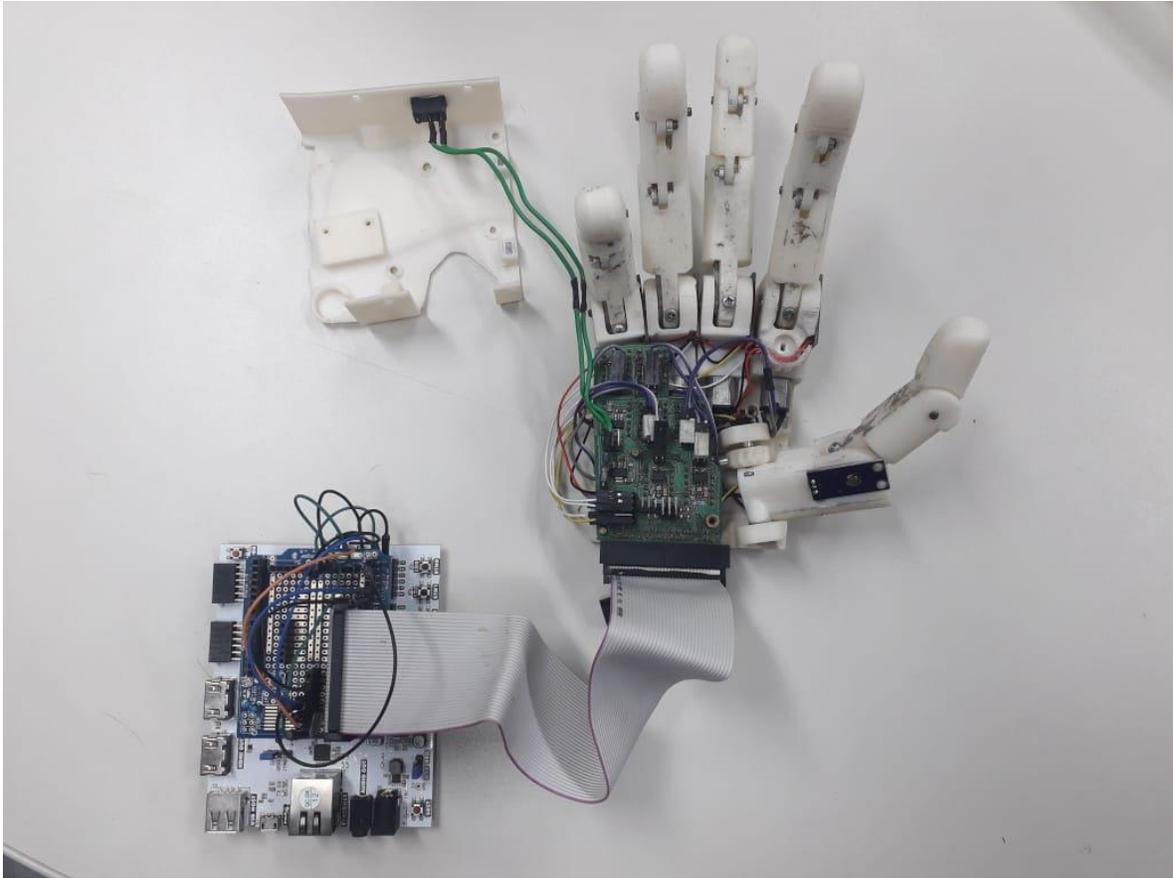


Fig. 27. Mano robótica funcional para el estudio y análisis utilizando FPGA por parte de los estudiantes de maestría y doctorado de la UnB.

6 Proyecto final clase Bioinspirados

6.1 Introducción Bioinspirados

Como parte de los trabajos adicionales se hace parte de la clase de sistemas Bioinspirados para la ingeniería, esta materia se trata de como nuevos algoritmos de optimización que permiten la solución de problemas de optimización, en el comienzo de la materia tratamos los temas de optimización de sistemas o problemas de programación lineal mediante la solución analítica que consiste en realizar graficas sobre la ecuaciones que podemos obtener del problema y encontrar los puntos de cruce mediante solución de ecuaciones y otro método de solución de problemas que consiste en la tabulación de las ecuaciones y solución del problema mediante un proceso iterativo llamado método simplex, después tenemos la introducción y los algoritmo de basados en el comportamiento de sistemas biológicos o de la naturaleza como el algoritmo PSO, ABC, DE, basados en hormigas, ballenas y demás, seguidamente tratamos la temática de diversidad artificial que consiste en ampliar la dinámica de los algoritmos dependiendo de otras formas de desplazamiento como lo pude

ser la atracción y repulsión de partículas, o el movimiento por oposición, seguimos después con el tema de test de hipótesis el cual consiste en un análisis estadístico de los resultados de la población de los algoritmos para poder determinar que algoritmo se comporta mejor que otros, luego estudiamos la aplicación de los algoritmos Bioinspirados como soporte para optimizar sistemas de control como los basado en PID o las propuestas de control inteligente tales como las redes neuronales artificiales, y finalmente analizamos las propuestas de algoritmos multiobjetivo que son capaces de resolver varias problemáticas a la vez como ejemplo tenemos el algoritmo genético NSGA o el Differential Evolution multiobjetivo (DEMO).

6.2 Proyecto final Bioinspirados

El proyecto final para este materia de Bioinspirados el es uso de los algoritmos de optimización estudiados durante las clases de la materia Bioinspirados implementados para la optimización de un controlador PID de un robot móvil que tiene como aplicación estar acierta distancia de un objetivo mediante un sensor Sharp y en control de una de sus ruedas, se probaran los algoritmo ABC que es el algoritmo basado en el comportamiento de las abejas, el algoritmo DE el cual está basado en los algoritmos genéticos teniendo algunas simplificaciones y finalmente una nueva propuesta de algoritmo llamado SCA que está basado en el comportamiento cíclico del seno y el coseno.

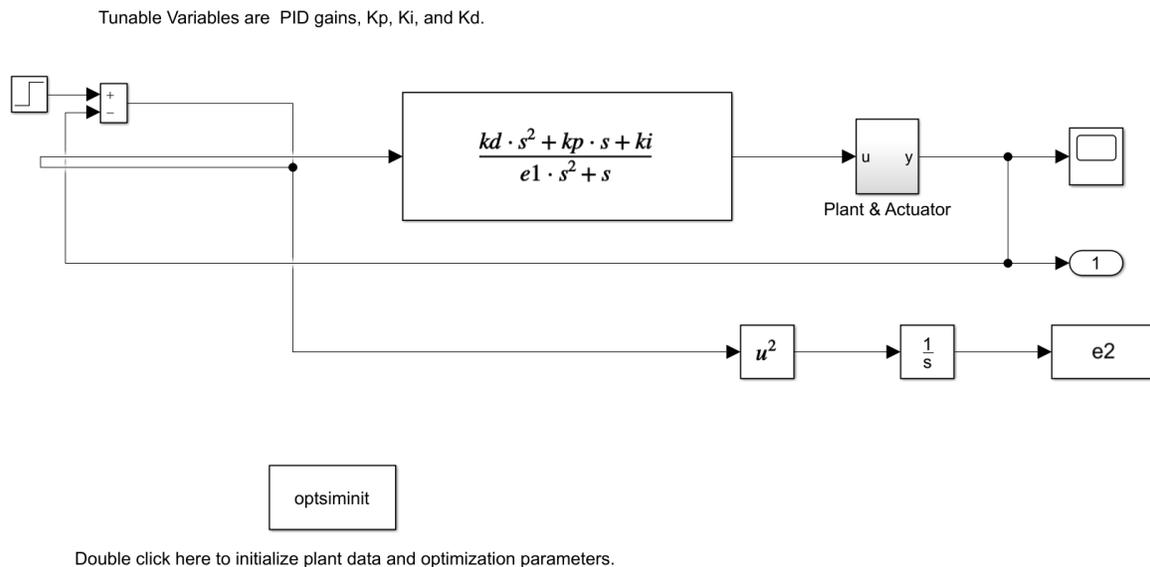


Fig. 28. Mano Diagrama de control PID para la simulación del comportamiento del robot móvil en Matlab-Simulink.

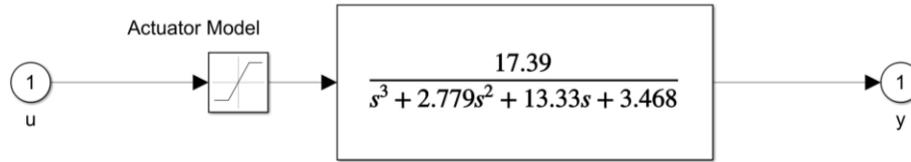


Fig. 29. Función de transferencia que simula el comportamiento del robot móvil.

6.3 Preparación de algoritmos para la optimización del sistema de control

Para la ejecución del algoritmo ponemos las condiciones de que se ejecuten ochenta veces y tengan un número de treinta partículas, con repetición de cinco veces por algoritmo, para permitir que tengan el tiempo suficiente para la optimización del sistema ya que los problemas de control pueden ser más difíciles de optimizar que las funciones de prueba, con un límite inferior de cero punto uno ya que si se deja con valor de cero genera un error y la variable de control proporcional no puede ser cero, y un límite superior de diez ya que un valor superior volvería la función inestable e oscilatoria, y un valor de dimensiones de optimización de tres ya que estamos optimizando los valores de controlador PID, K_p , K_i y K_d , de esta forma llamamos cada uno de los algoritmos planteados, se realiza la ejecución del código para realizar la primera optimización y obtenemos los primeros valores de controlador.

Estos valores obtenidos donde estos se ejecutan y se evalúan en la función “tracklsq” en la cual llamamos un programa en Simulink que simula el comportamiento del robot móvil y genera una respuesta de controlador que al volver a la función permite calcular los errores de estado estacionario, límite de sobrepico, valor de tiempo de levantamiento.

$$F = \left(\sum |20 - y_{out}|^2 \right) \cdot 0.3 \cdot |y_{out} - 20| \cdot 0.7 \quad (7)$$

De esta forma obtenemos el error ponderado (7) el cual enviamos al algoritmo este valor y de esa forma empezamos la comparación con el valor mínimo global el cual se almacena para comenzar de nuevo el proceso interactivo de optimización del sistema PID para el control del robot móvil controlando un solo motor y un sensor Sharp para conocer la distancia con el objetivo.

6.4 Resultados

Como resultados obtenemos que los valores de controlador obtenidos por cada uno de los algoritmos donde podemos apreciar mediante la media y mediana cuál de ellos posee un mejor valor de error ponderado y determinar que el que posee mejor comportamiento es el algoritmo DE Differential Evolution ya que los demás algoritmos (ABC y SCA) ciertamente obtuvieron valores aceptables como se aprecia también en sus graficas significativamente no superan el algoritmo mencionado en un principio.

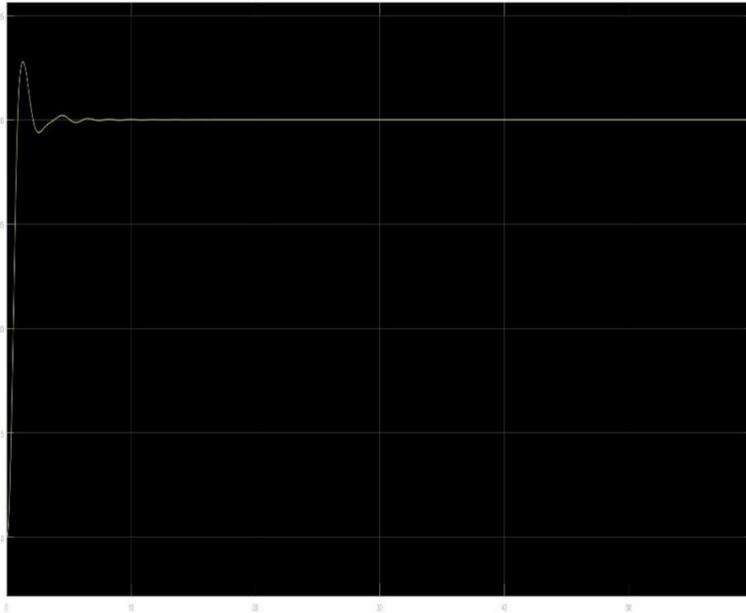


Fig. 30. Mejor respuesta del controlador PID usando algoritmo ABC.

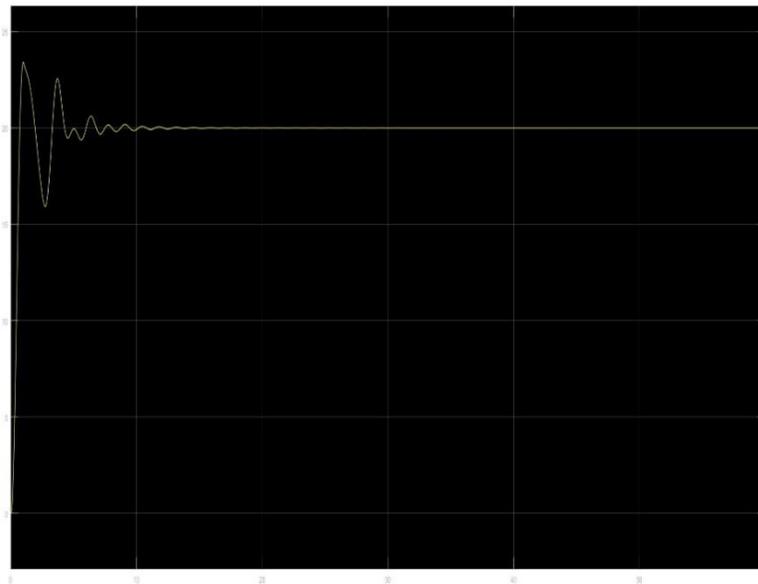


Fig. 31. Mejor respuesta del controlador PID usando algoritmo SCA.

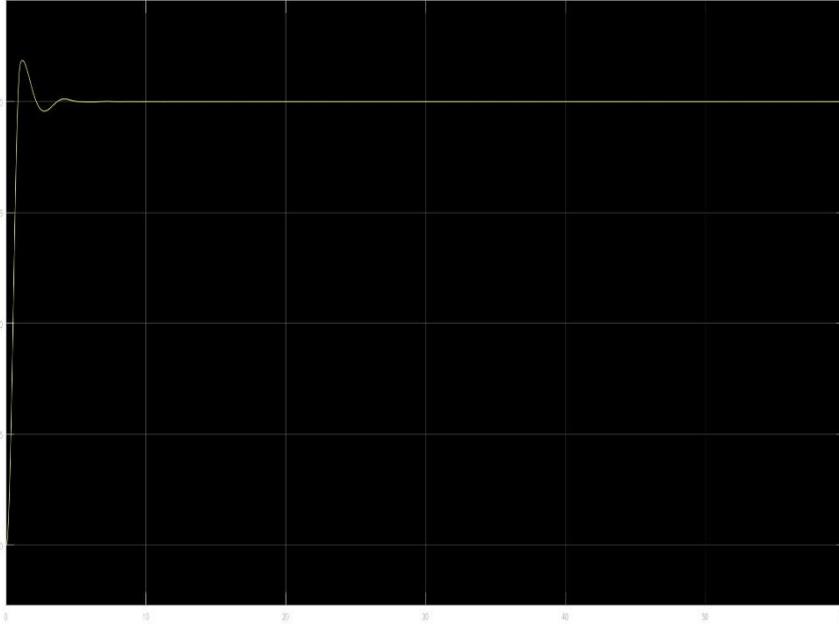


Fig. 32. Mejor respuesta del controlador PID usando algoritmo DE.

Al realizar un análisis de las gráficas obtenidas por las mejores respuestas de cada algoritmo podemos observar que las constantes del algoritmo ABC son bastantes buenas no tienen una obligación demasiado grande se asienta en poco tiempo y no tiene un sobre pico alto, al analizar la grafica del algoritmo SCA notamos una gran oscilación antes de poder llegar a la estabilidad aunque se estabilice casi al mismo tiempo que la del ABC esta posee también un sobre pico mas grande, y finalmente la grafica obtenida por el algoritmo DE podemos notar que esta posee una pequeña oscilación para luego entrar a su estado estacionario en el setpoint con un sobrepico bastante bueno. Cabe resaltar un punto importante es que todos los algoritmos dieron una respuesta de constantes PID en la que conseguía llegar al Setpoint.

KP	KI	KD	Resposta de erro
0,71815836	0,91786561	0,1	3,12E+05
1,83658634	10	1,27369179	2,20E+05
2,69626838	9,81152206	2,1625916	4,25E+05
1,13905258	1,80325132	0,66849967	2,07E+05
2,2834482	8,47757392	1,01949928	1,88E+05

Fig. 33. Datos obtenidos de las cinco rodadas del algoritmo SCA.

KP	KI	KD	Resposta de erro
1,78825376	4,81406904	0,55710253	1,15E+05
2,48040868	2,83001212	0,70946351	1,17E+05
2,17009728	3,94408502	0,61748338	1,16E+05
2,42570945	2,73969871	0,70533179	1,14E+05
1,53894349	4,45313533	0,44264295	1,12E+05

Fig. 34. Datos obtenidos de las cinco rodadas del algoritmo ABC.

KP	KI	KD	Respostas de erro
1,84415768	4,56365884	0,54785065	1,06E+05
1,82475065	4,61204475	0,5466149	1,05E+05
1,82393513	4,61286419	0,54672228	1,05E+05
1,8248644	4,61374804	0,54676313	1,05E+05
1,82302914	4,61309888	0,54684755	1,05E+05

Fig. 35. Datos obtenidos de las cinco rodadas del algoritmo DE.

Al realizar un análisis de los datos obtenidos podemos observar que todas las respuestas alcanzan un rango de error cercano sin embargo podemos notar que el algoritmo DE conseguía una media y media mejor que los otros algoritmos esto quiere decir que este algoritmo en cada una de sus puestas en acción consigue casi siempre la mejor respuesta o las constantes de PID que mejor optimizaban el sistema. Tenemos también que en la parte de implementación el mejor valor del algoritmo DE reanalizo la tarea correctamente, esto permite concluir que los algoritmos Bioinspirados son capaces de optimizar sistemas de control de complejidad media y se espera probar con otro tipo de problemas de control.



Fig. 36. Funcionamiento de robot móvil con el resultado de las contantes de PID del algoritmo DE.

6.5 Test de hipótesis

Los test de hipótesis ofrecen un análisis estadístico que permite discernir de forma matemática cuál de los algoritmos puede ser considerado mejor que los otros, se realizan tres distintos test de hipótesis con la finalidad de encontrar estadísticamente cual dio mejores resultados. Al realizar el test Kolmogorov-Smirnov tiene la función de decirnos si las poblaciones pertenecen a grupos de distribución normal o no en el caso de los algoritmos DE, ABC y SCA, obtuvieron el resultado de que no poseen una distribución normal, luego pasamos al test KruskalWallis el cual indica que si las poblaciones pasadas por el pertenecen a un grupo con medias iguales o no, de esta forma al ejecutar el test tenemos de que por lo menos una de las poblaciones po se media diferentes que los otros algoritmos, Finalmente pasamos al siguiente test Wilconxon el cual indica en comparación de una población a otra tomando como referencia cual tiene la mejor media obtenemos que el DE contra el ABC indica que el DE ofrece un mejor valor para optimizar el sistema, y la comparación entre el DE y SCA indica de la misma forma que el DE ofrecer una solución óptima superior.

7 CONCLUSIONES

Uno de los principales resultados de la pasantía de la investigación fue la resolución del convenio entre la universidad de Pamplona y la universidad de Brasilia que permitirá el intercambio de estudiantes y profesores para realizar pasantías, cursos o clases entre las universidades sino también compartir el conocimiento, para los próximos pasantes que vayan a la universidad de Brasilia podrán tener beneficios como alimentación y vivienda gratuita además de otros benéficos como facilidad para entrar a realizar una maestría o un doctorado.

Como conclusiones tenemos que los resultados obtenidos del desarrollo del actuador robótico son satisfactorios y permiten continuar con el trabajo e implementarlos en la manos robóticas para realizar tipos de control más complejos y realizar sistemas más robustos, la implementación del módulo AXI IIC para la comunicación de dispositivos IIC es un contribución para el desarrollo de nuevos proyectos que deseen trabajar con este tipo de comunicación ya que facilita este proceso y se ahorra un gran trabajo para el procesador ya que este se encuentra implementado en el hardware de la FPGA, las librerías realizadas para el control de los dispositivos ADS1115 y DRV3330 serán usadas para implementaciones más grandes para el control de varios actuadores robóticos a la vez ya que la futura implementación del actuador es formar parte de una mano robótica. Es un gran beneficio implementar una FPGA en estos tipos de sistemas ya que generalmente se trabajan con muchos datos y conllevan procesos de control bastantes pesados que pueden saturar un microcontrolador, ya que los programas que consume que mas consumen recursos pueden ser realizado en VHDL o en software implementado ya que estos pueden trabajar en paralelo y en muy poco tiempo de ejecución, se prevé como trabajo futuro implementar el actuador robótico en un nuevo modelo de mano robótica como parte de las actividades de estudiantes de maestría y doctorado.

Tenemos también que los resultados de la remodelación de la mano robótica fue buena contribución ya que gracias a las mejoras añadidas se pudo realizar un mejor control de la mano robótica de forma más robusta, para dicha mano se implementó un control de impedancia el cual consiste en regular el voltaje de los motores dependiendo del consumo de corriente, gracias a esto se planea realizar una publicación en revista científica en los primeros meses del año 2020.

Finalmente se destaca que los algoritmos pudieron optimizar de buena forma el problema de control planteado para un robot móvil con PID el cual fueron utilizados por otros alumnos de maestría que compararon sus valores de optimización encontrados que eran muy parecidos y contribuyo así trabajo al realizar estas comparaciones, de esta manera también se plantea la posibilidad de publicación de un artículo en una revista científica en los próximos meses del siguiente año.

8 BIBLIOGRAFÍA

- [1] Sachin Maheshwari, V. A. Bartlett, Izzet Kale, Modelling, simulation and verification of 4-phase adiabatic logic design: A VHDL-Based approach, *Integration*, Volume 67, July 2019, Pages 144-154. <https://doi.org/10.1016/j.vlsi.2019.01.007>
- [2] Du-Juan Dong, Wen Liu, Mei-Juan Cai, Jin-Xing Wang, Xiao-Fan Zhao, Steroid hormone 20-hydroxyecdysone regulation of the very-high-density lipoprotein (VHDL) receptor phosphorylation for VHDL uptake, *Insect Biochemistry and Molecular Biology*, Volume 43, Issue 4, April 2013, Pages 328-335. <https://doi.org/10.1016/j.ibmb.2013.02.001>
- [3] Godofredo R. Garaya, Andrei Tchernykh, Alexander Yu. Drozdov, Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education, *Journal of Computational Science*, Volume 36, September 2019, 100652. <https://doi.org/10.1016/j.jocs.2017.04.004>
- [4] Sanjeeb Mishra, Neeraj Kumar Singh, Vijayakrishnan Rousseau, SoC Design Fundamentals and Evolution, *System on Chip Interfaces for Low Power Design*, 2016, Pages 1-11. <https://doi.org/10.1016/B978-0-12-801630-5.00001-3>
- [5] Xuecheng Du, Chaohui He, Shuhuan Liu, Analysis of sensitive blocks of soft errors in the Xilinx Zynq-7000 System-on-Chip, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 940, 1 October 2019, Pages 125-128, <https://doi.org/10.1016/j.nima.2019.06.015>
- [6] Gina R. Smith, *FPGA Development Phases*, FPGAs 101, 2010, Pages 43-55. <https://doi.org/10.1016/B978-1-85617-706-1.00003-5>
- [7] Zoran Jakšić, Nicola Cadenelli, A highly parameterizable framework for Conditional Restricted Boltzmann Machine based workloads accelerated with FPGAs and OpenCL, *Future Generation Computer Systems*, Volume 104, March 2020, Pages 201-211, <https://doi.org/10.1016/j.future.2019.10.025>
- [8] Wen Bai, Yadi Chen, Di Wua, Efficient temporal core maintenance of massive graphs, *Information Sciences*, Available online 9 November 2019. <https://doi.org/10.1016/j.ins.2019.11.003>
- [9] Oussama Kerdjidj, Abbes Amira, Khalida Ghanem, An FPGA implementation of the matching pursuit algorithm for a compressed sensing enabled e-Health monitoring platform, *Microprocessors and Microsystems*, Volume 67, June 2019, Pages 131-139. <https://doi.org/10.1016/j.micpro.2019.03.007>
- [10] Vinh Nguyena, Andrew Dugenske, An I2C based architecture for monitoring legacy manufacturing equipment, *Manufacturing Letters*, Volume 15, Part B, January 2018, Pages 67-70. <https://doi.org/10.1016/j.mfglet.2017.12.018>

- [11] F. Gomez-Bravo, J. Medina García, Plataforma Experimental para el Estudio de la Vulnerabilidad Hardware en los Robots Móviles: el Bus I2C como Caso de Estudio Experimental Platform for Studying Hardware Vulnerabilities on Mobile Robots: I2C Bus, a Case of Study, *Revista Iberoamericana de Automática e Informática Industrial RIAI*, Volume 14, Issue 2, April–June 2017, Pages 205-216. <https://doi.org/10.1016/j.riai.2017.02.002>
- [12] Jesús Lázaro, Armando Astarloa, I2CSec: A secure serial Chip-to-Chip communication protocol, *Journal of Systems Architecture*, Volume 57, Issue 2, February 2011, Pages 206-213. <https://doi.org/10.1016/j.sysarc.2010.12.001>
- [13] Jim Drew, Chapter 48 - Compact I2C-controllable quad synchronous step-down DC/DC regulator for power-conscious portable processors, *Analog Circuit Design*, Volume 3: Design Note Collection, 2015, Pages 105-106. <https://doi.org/10.1016/B978-0-12-800001-4.00048-X>
- [14] Katrin Hecka, Helena Paterno, Fatigue resistance of ultrathin CAD/CAM ceramic and nanoceramic composite occlusal veneers, *Dental Materials*, Volume 35, Issue 10, October 2019, Pages 1370-1377. <https://doi.org/10.1016/j.dental.2019.07.006>
- [15] Daniel O. Bittner MD, Prognostic Value of Coronary CTA in Stable Chest Pain: CAD-RADS, CAC, and Cardiovascular Events in PROMISE, *JACC: Cardiovascular Imaging*, Available online 13 November 2019. <https://doi.org/10.1016/j.jcmg.2019.09.012>
- [16] Zongze Lia, A preliminary study of a novel robotic system for pedicle screw fixation: A randomised controlled trial, *Journal of Orthopaedic Translation*, Available online 16 September 2019. <https://doi.org/10.1016/j.jot.2019.09.002>
- [17] K. Rahul, Hifjur Raheman, Vikas Paradkar, Design and development of a 5R 2DOF parallel robot arm for handling paper pot seedlings in a vegetable transplanter, *Computers and Electronics in Agriculture*, Volume 166, November 2019, 105014. <https://doi.org/10.1016/j.compag.2019.105014>
- [18] Dechao Chena, Shuai Li, Liefu Liao, A recurrent neural network applied to optimal motion control of mobile robots with physical constraints, *Applied Soft Computing*, Volume 85, December 2019, 105880. <https://doi.org/10.1016/j.asoc.2019.105880>
- [19] Pouya Panahandeh, Khalil Alipour, Bahram Tarvirdizadeh, Alireza Hadi, A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots, *Mechanical Systems and Signal Processing*, Volume 134, 1 December 2019, 106319. <https://doi.org/10.1016/j.ymsp.2019.106319>
- [20] Lucian Ribeiro da Silva, Rodolfo César Costa Flesch, Julio Elias Normey-Rico, Controlling industrial dead-time systems: When to use a PID or an advanced controller, *ISA Transactions*, Available online 12 September 2019. <https://doi.org/10.1016/j.isatra.2019.09.008>

- [21] Meena E. Girgis, Rania A. Fahmy, Ragia I. Badr, Optimal fractional-order PID control for plasma shape, position, and current in Tokamaks, *Fusion Engineering and Design*, Available online 15 October 2019, 111361. <https://doi.org/10.1016/j.fusengdes.2019.111361>
- [22] Aldo Jonathan Muñoz-Vázquez, A fuzzy fractional-order control of robotic manipulators with PID error manifolds, *Applied Soft Computing*, Volume 83, October 2019, 105646. <https://doi.org/10.1016/j.asoc.2019.105646>
- [23] Ignacio Carlucho, Mariano De Paula, Gerardo G. Acosta, Double Q-PID algorithm for mobile robot control, *Expert Systems with Applications*, Volume 137, 15 December 2019, Pages 292-307. <https://doi.org/10.1016/j.eswa.2019.06.066>
- [24] Yashan Xing, Jing Na, Ramon Costa-Castelló, Composite PID Control with Unknown Dynamics Estimator for Rotomagnet Plant, *IFAC-PapersOnLine*, Volume 51, Issue 4, 2018, Pages 817-822. <https://doi.org/10.1016/j.ifacol.2018.06.174>
- [25] Chinnu Sabua, Christine Rejob, Sabna Kotta, Bioinspired and biomimetic systems for advanced drug and gene delivery, *Journal of Controlled Release*, Volume 287, 10 October 2018, Pages 142-155. <https://doi.org/10.1016/j.jconrel.2018.08.033>
- [26] Gianluca Elia, Alessandro Margherita, Digital entrepreneurship ecosystem: How digital technologies and collective intelligence are reshaping the entrepreneurial process, *Technological Forecasting and Social Change*, Volume 150, January 2020, 119791. <https://doi.org/10.1016/j.techfore.2019.119791>
- [27] Yu-Lin Jenga, Yong-Ming Huang, Dynamic learning paths framework based on collective intelligence from learners, *Computers in Human Behavior*, Volume 100, November 2019, Pages 242-251. <https://doi.org/10.1016/j.chb.2018.09.012>
- [28] Li Junxiang, Chen Jianqiao, Solving time-variant reliability-based design optimization by PSO-t-IRS: A methodology incorporating a particle swarm optimization algorithm and an enhanced instantaneous response surface, *Reliability Engineering & System Safety*, Volume 191, November 2019, 106580. <https://doi.org/10.1016/j.ress.2019.106580>
- [29] Jun Tao, Gang Sun, Xinyu Wang, Robust optimization for a wing at drag divergence Mach number based on an improved PSO algorithm, *Aerospace Science and Technology*, Volume 92, September 2019, Pages 653-667. <https://doi.org/10.1016/j.ast.2019.06.041>
- [30] K. Martin Sagayam, D. Jude Hemanth, ABC algorithm based optimization of 1-D hidden Markov model for hand gesture recognition applications, *Computers in Industry*, Volume 99, August 2018, Pages 313-323. <https://doi.org/10.1016/j.compind.2018.03.035>
- [31] Sara Najari, Gyul Gróf, Samrand Saeidi, Modeling and optimization of hydrogenation of CO₂: Estimation of kinetic parameters via Artificial Bee Colony (ABC) and

- Differential Evolution (DE) algorithms, *International Journal of Hydrogen Energy*, Volume 44, Issue 10, 22 February 2019, Pages 4630-4649. <https://doi.org/10.1016/j.ijhydene.2019.01.020>
- [32] Mohamed Abd Elaziz, Lin Li, K. P. N. Jayasena, Multiobjective big data optimization based on a hybrid salp swarm algorithm and differential evolution, *Applied Mathematical Modelling*, Available online 7 November 2019. <https://doi.org/10.1016/j.apm.2019.10.069>
- [33] Pinar Civicioglu, Erkan Besdok, Bernstein-search differential evolution algorithm for numerical function optimization, *Expert Systems with Applications*, Volume 138, 30 December 2019, 112831. <https://doi.org/10.1016/j.eswa.2019.112831>
- [34] Yansong Hao, Liuyang Song, Lingli Cui, A three-dimensional geometric features-based SCA algorithm for compound faults diagnosis, *Measurement*, Volume 134, February 2019, Pages 480-491. <https://doi.org/10.1016/j.measurement.2018.10.098>
- [35] Seyedali Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowledge-Based Systems*, Volume 96, 15 March 2016, Pages 120-133. <https://doi.org/10.1016/j.knosys.2015.12.022>
- [36] Yuanhui Xiao, A fast algorithm for two-dimensional Kolmogorov–Smirnov two sample tests, *Computational Statistics & Data Analysis*, Volume 105, January 2017, Pages 53-58. <https://doi.org/10.1016/j.csda.2016.07.014>
- [37] Samuel N. Cohen, Timo Henckel, Gordon D. Menzies, Switching cost models as hypothesis tests, *Economics Letters*, Volume 175, February 2019, Pages 32-35. <https://doi.org/10.1016/j.econlet.2018.11.014>
- [38] Hideki Kitamori, Iori Sumida, Evaluation of mouthpiece fixation devices for head and neck radiotherapy patients fabricated in PolyJet photopolymer by a 3D printer, *Physica Medica*, Volume 58, February 2019, Pages 90-98.