

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

TESIS DE MAESTRÍA

**Disminución de Incertidumbres en Sistemas de
Medición Tridimensional Bajo el Principio de
Triangulación Dinámica**

Autor:

Gabriel Trujillo Hernández

Director de Tesis:

Dr. Julio César Rodríguez
Quiñonez

Co-Director de Tesis:

Dra. Vira Tyrsa

*Una tesis presentada en cumplimiento de los requisitos
para el grado de Maestría en Ciencias.*

en la

Facultad de Ingeniería
Campus Mexicali

9 de junio de 2020



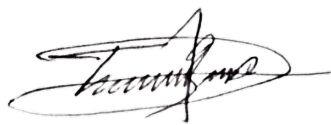
**UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA**
FACULTAD DE INGENIERÍA
MEXICALI

Declaración de Autoría

Yo, Gabriel Trujillo Hernández, declaro que esta tesis titulada, «Disminución de Incertidumbres en Sistemas de Medición Tridimensional Bajo el Principio de Triangulación Dinámica » y el trabajo presentado en esta es mío. Y confirmo que:

- Este trabajo fue realizado por completo o principalmente para obtener un grado de maestría en esta universidad.
- Donde cualquier parte de esta tesis fue revisada por mi comité.
- Donde consulté el trabajo publicado por otros, siempre es atribuido claramente.
- Donde he citado el trabajo de otros, la fuente siempre es proporcionada. Con la excepción de estas citas, esta tesis es por completo mi trabajo.
- Reconocí todas las fuentes principales de ayuda.
- Donde la tesis es basada en trabajo hecho por mí en conjunto con otros, he aclarado exactamente que fue realizado por otros y cuál fue mi contribución.

Firma:



Fecha:

25 de Junio del 2020

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber".

Albert Einstein

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Resumen

Facultad de Ingeniería

Campus Mexicali

Maestría en Ciencias.

Disminución de Incertidumbres en Sistemas de Medición Tridimensional Bajo el Principio de Triangulación Dinámica

por Gabriel Trujillo Hernández

Esta tesis tiene como propósito la incorporación de un eje de rotación β a un sistema bidimensional y la disminución de las incertidumbres que estos presentan en mediciones tridimensionales bajo el principio de triangulación dinámica. Por lo tanto, se definirá e implementará un algoritmo que permita atenuar estas variables para garantizar la factibilidad del equipo en aplicaciones tales como navegación autónoma, monitoreo de salud estructural, biometría, manufactura, etc. En la presente propuesta se describen las modificaciones y adaptaciones en el diseño mecánico, hardware y software que se van a considerar para realizar la integración del eje β . Así como, la determinación de los límites de operación del prototipo utilizando esta experimentación. Por otro lado, también se comparan algoritmos de inteligencia artificial, afín de emplear un método que presente un alto rendimiento en la disminución de las incertidumbres.

Reconocimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por brindar el apoyo tanto económico como material para financiar el desarrollo de la investigación durante estos 2 años.

A la Universidad Autónoma de Baja California (UABC), por brindar la oportunidad de estudiar un posgrado y obtener el grado de Maestro en Ciencias.

A la Facultad de Ingeniería de la UABC Campus Mexicali, por proveer el equipo necesario para el desarrollo de la investigación.

A mi director de tesis, Dr. Julio C. Rodríguez Quiñonez por brindar la oportunidad de participar en este proyecto de investigación, por su tiempo y el apoyo al compartir sus conocimientos del tema.

A mi Co-director de tesis, Dra. Vira Tyrsa por trabajar en conjunto con mi comité de tesis para brindar sugerencias de mejora para el desarrollo de mi investigación.

A mi comité de tesis, conformado por el Dr. Oleg Sergiyenko, Dra. Wendy Flores Fuentes y el Dr. Daniel Hernández Balbuena, por su apoyo y retroalimentación de cada semestre en las presentaciones de avance de tesis.

Índice General

Declaración de Autoría	I
Resumen	III
Reconocimientos	IV
Índice General	V
Índice de Figuras	VIII
Índice de Tablas	XI
Abreviaturas	XII
Símbolos	XIII
1. Introducción	1
1.1. Antecedentes	1
1.2. Principio de Triangulación	3
1.3. Planteamiento del Problema	5
1.4. Justificación y Uso de los Resultados	7
1.5. Objetivos de la Investigación	9
1.5.1. Objetivo General	9
1.5.2. Objetivos Específicos	9
1.6. Metodología	10
2. Marco Teórico	12
2.1. Triangulación Dinámica	12
2.1.1. Posicionador Láser	12
2.1.2. Apertura de Escaneo	13
2.1.3. Detección del Ángulo B_{ij}	14
2.2. Inteligencia Artificial	15
2.2.1. Definición	15

2.2.2.	Algoritmos de Máquinas de Aprendizaje	16
2.2.3.	Aplicaciones	16
2.2.4.	Redes Neuronales Artificiales (ANN)	17
2.2.4.1.	Neurona Biológica	17
2.2.4.2.	Neurona Artificial	18
2.2.4.3.	Estructura y Entrenamiento de una ANN	19
2.3.	Teoría del Control Clásico	21
2.3.1.	Controlador Proporcional	22
2.3.2.	Controlador Derivativo	23
2.3.3.	Controlador Integral	24
2.3.4.	Controlador Proporcional Derivativo	25
2.4.	Tipos de Motores Reductores de Velocidad	26
2.4.1.	Reductores de Velocidad Sin Fin-Corona	27
2.4.2.	Reductores de Velocidad Planetario	27
2.4.3.	Reductores con Grupo Cónico-Espiral o de Velocidad de Engranaje	28
2.4.4.	Reductores de Engranajes Cilíndrico-Helicoidales	29
2.5.	Principio de Codificador	29
2.5.1.	Funcionamiento del Codificador de Cuadratura	30
2.6.	FPGA	32
2.6.1.	Estructura General de las FPGAs	32
2.6.2.	Aplicaciones de las FPGA	33
2.7.	Conclusiones	34
3.	Procedimiento de Investigación	35
3.1.	Ensamble Completo del Eje β	35
3.1.1.	Fabricación del Circuito Impreso	36
3.1.2.	Conteo de Pulsos	38
3.1.3.	Controlador	40
3.1.4.	Modulación de Ancho de Pulso (PWM)	41
3.1.5.	Etapas de Potencia	43
3.2.	Propuesta de Algoritmo	44
3.2.1.	Adquisición de Base de Datos	45
3.2.2.	Entrenamiento de la Red Neuronal Artificial	46
3.2.3.	Implementación de la Red Neuronal Artificial	49
3.3.	Implementación de Ecuación Z	51
3.4.	Implementación de la Interfaz Tridimensional	52
3.5.	Conclusiones	54
4.	Análisis de Resultados	55
4.1.	Análisis de Controlador PD	55
4.2.	Análisis del Entrenamiento de Redes Neuronales	57
4.3.	Análisis de Redes Neuronales	59
4.3.1.	Análisis y Comparación de Nube de Puntos Bidimensionales	60

4.4. Análisis de Nube de Puntos Tridimensionales	62
5. Conclusión	65
Bibliografía	68

Índice de Figuras

1.1. a) Principio de Triangulación b) Ángulo β	4
1.2. Las Piezas Verdes Muestran el Eje de Rotación β	6
1.3. Aplicación del TVS en Robots Autónomos para el Reconocimiento de Superficies Tridimensionales	8
1.4. Eje Auxiliar Utilizado para Medir Superficies Tridimensionales	10
2.1. a) Posicionador Láser b) Apertura de Escaneo	13
2.2. Señales Utilizadas para Calcular B_{ij}	14
2.3. Tipos de Máquinas de Aprendizaje	17
2.4. Neurona Biológica	18
2.5. Neurona Artificial	19
2.6. Red Neuronal	19
2.7. Control Lazo Abierto	21
2.8. Control Lazo Cerrado	21
2.9. Banda Proporcional	23
2.10. Control Derivativo	24
2.11. Control Integral	25
2.12. Reductores de Velocidad Sin Fin-Corona	27
2.13. Reductores de Velocidad Planetario	28
2.14. Reductores con Grupo Cónico-Espiral o de Velocidad de Engranaje	29
2.15. Reductores de Engranajes Cilíndrico-Helicoidales	29
2.16. Disco Perforado	30
2.17. Desfase de 90° Entre el Canal A y Canal B	30
2.18. Giro Hacia las Manecillas del Reloj.	31
2.19. Giro Encontra de las Manecillas del Reloj.	31
2.20. Estructura General de un FPGA.	32
3.1. Propuesta para El Eje β	35
3.2. Propuesta para El Eje β , Donde I) Eje Inicial II) Eje Imaginario III) Eje Final IV) Espejos 45° V) Motor VI) Base Giratoria VII) Codificador	36
3.3. Circuito Impreso para el TVS Bidimensional.	37
3.4. Circuito Impreso para el Eje de Rotacion β	38
3.5. Pulsos en Sentido Horario.	39
3.6. Pulsos en Sentido Antihorario.	39

3.7. a) Modelo de un controlador PD b) La función PID permite modelar diferentes tipos de controladores tales como: PD, PI, P, D y I	40
3.8. Referencia = pulsos equivalentes a la posición deseada, variable a procesar = pulsos del codificador, salida PD= $K_p \cdot e(t) + K_d \cdot \frac{d}{dt} \cdot e(t)$	41
3.9. El motor comienza a girar en $P_c \geq 26.7$, por lo tanto para P_c de 0% y 25% se marcan con una cruz roja debido que no existe giro, sin embargo para 65% y 100% se produce movimiento por tal motivo se marca con una palomita verde.	42
3.10. Metodología de las señales PWM, donde max. resolución = 65535, $0 \leq \text{salida PD} \leq 65535$	43
3.11. Entradas y salidas del puente H L298	44
3.12. Los puntos azules indican la intersección entre las líneas paralelas que representan las distancias de 15 cm a 35 cm y las líneas angulares.	45
3.13. La ANN desarrollada es configurada por tres entradas, siete neuronas, tres capas y funciones de activación $Tanh()$ y $Pureline()$	46
3.14. Interfaz para la configuración de la ANN	47
3.15. ANN configurada utilizando en la paquetería de Matlab <i>nntool</i>	48
3.16. Configuración de parámetros de entrenamiento	48
3.17. n representa el número de interacciones, C_n es el número de carpeta, E son los datos de entrenamiento y P los datos de prueba.	49
3.18. El recuadro verde indica la sección de normalización, mientras que el recuadro rojo indica la sección de desnormalización	50
3.19. Las líneas verdes indican las conexiones de la ecuación Z	51
3.20. Sección programada para compartir información del Compact RIO a nivel computadora.	52
3.21. Programa a nivel computadora para graficar las coordenadas X , Y y Z	53
3.22. Interfaz donde se muestran las nubes de puntos bidimensionales y tridimensionales	53
4.1. a) El sistema responde lento pero preciso b) El sistema responde rápido pero oscilaciones en periodos largos c) El sistema responde rápido pero presenta oscilaciones d) El sistema es rápido y no se presentan oscilaciones	55
4.2. La línea azul representa la linealidad de los valores reales, mientras que los puntos rojos indican el valor controlado.	56
4.3. El valor obtenido de $R = 0.932687$ nos indica que el valor predicho por la ANN se adapta adecuadamente al valor real	57
4.4. Comparación entre el valor real y el valor predicho usando el algoritmo KNN	59
4.5. Comparación entre el valor real y el valor predicho usando el algoritmo ANN.	59
4.6. Nubes de puntos: a) nubes de puntos aplicando el método anterior b) nubes de puntos utilizando el método desarrollado.	60

4.7. Objeto tridimensional conocido como trapezoide.	62
4.8. Nube de puntos obtenida al escanear la pieza trapezoide aplicando el método anterior.	62
4.9. Nube de puntos obtenida al escanear la pieza trapezoide aplicando el método desarrollado.	63
4.10. Nube de puntos del prototipo de una mano humana.	63

Índice de Tablas

4.1. Comparación del ángulo β entre el valor real y valor controlado. . .	56
4.2. RMSE calculado utilizando el algoritmo KNN.	58
4.3. RMSE calculado utilizando el algoritmo ANN.	58
4.4. Mediciones de barrido láser usando el algoritmo KNN.	61
4.5. Mediciones de barrido láser usando el algoritmo ANN.	61
4.6. Porcentaje de mejora en la desviación estándar del algoritmo ANN con respecto al KNN.	61

Abreviaturas

TVS	T echnical V ision S ystem
CCD	C harge C ouple D evice
ANN	A rtificial N eural N etworks
KNN	K - N earest N eighbors
PL	P ositioning L aser
SA	S canning A perture
PD	P roportional- D erivative
PWM	P ulse- W idth M odulation
DC	D irect C urrent
FPGA	F ield- P rogrammable G ate A rray
CCW	C ounter- C lock- W ise
CW	C lock- W ise
LUT	L ook- U p T able
PID	P roportional I ntegral D erivative
RMSE	R oot M ean S quare E rror

Símbolos

Símbolo	Nombre
C_{ij}	Ángulo del Posicionador Láser
B_{ij}	Ángulo de la Apertura de Escaneo
a	Distancia Fija
β	Ángulo del Eje de Rotación β
N_A	No. de Pulsos del Canal A
N_S	No. de Pulsos Contados del Tren de Pulsos de N_A Hasta el Pulso del PR Sensor
N_T	No. de Pulsos Contados del Tren de Pulsos de N_A Hasta el Pulso del Canal A
P_A	Grados por Pulsos del Canal A
W_n	Pesos
X_n	Entradas
F_n	Función de Activación
Y_p	Salida
H	Neurona
e	Error
η	Número de Tasa de Aprendizaje
α	Variable Controlada
κ	Entrada de Referencia
K_p	Ganancia Proporcional
K_i	Ganancia Integral
K_d	Ganancia Derivativa
u_t	Salida Controlada
R_c	Resolución
R	Parámetro de Correlación

P_c	Porcentaje de Ciclo de Trabajo
C_n	Número de Carpeta
E	Datos de Entrenamiento
P	Datos de Prueba
X_t	Datos Normalizados
M_i	Valor Máximo de Entrada
m_i	Valor Mínimo de Entrada
a_e	Escala Mínima
b_e	Escala Máxima

Esta tesis la dedico principalmente a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mi amada esposa Ana Duran, por su amor y paciencia durante este tiempo, gracias por creer en mi e impulsarme a lograr mis metas y sueños.

A mis padres, por su amor, trabajo y sacrificio en estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy.

A mi hermano por estar siempre presente y por el apoyo moral, que me brindó a lo largo de esta etapa de mi vida.

Capítulo 1

Introducción

1.1. Antecedentes

Hoy en día, los sistemas de visión han permitido realizar inspección metrológica sin contacto, lo cual ha revolucionado la manera en la que se capturan las nubes de puntos de superficies. El aumento de la velocidad de escaneo, precisión, resolución y adquisición de datos en tiempo real ha permitido aumentar el rendimiento de aplicaciones tales como: salud estructural, escaneo de prótesis, robots autónomos, procesos de producción y detección de malformaciones físicas en el cuerpo humano [Real-Moreno et al. \[2017\]](#), [Park et al. \[2015\]](#), [Herrera Gil et al. \[2019\]](#), [Kim and Sentis \[2017\]](#). Dentro del sistema de visión se encuentra el Sistema de Visión Técnica (conocido por sus siglas en inglés TVS), el TVS es una tecnología que tiene como función escanear y determinar las coordenadas tridimensionales (X, Y, Z) de objetos dentro del campo de visión. En recientes publicaciones se han desarrollado técnicas para mejorar el rendimiento del sistema tales como: tiempo de vuelo, mediciones de fase y triangulación dinámica.

El método de tiempo de vuelo es una técnica utilizada para estimar las distancia de objetos calculando el tiempo transcurrido entre la emisión y la recepción del haz de láser. En aplicaciones de escaneo el haz de láser es proyectado sobre una superficie, el tiempo desde que se emite el láser hasta que es detectado por el receptor es utilizado para calcular las coordenadas de la superficie del objeto. [Chua et al. \[2016\]](#).

En la publicación [Maddern et al. \[2015\]](#), un dispositivo Lidar basado en el principio de tiempo de vuelo es montado sobre un vehículo para escanear la ciudad durante una trayectoria específica, con el objetivo de mostrar los cambios presentes en distintos periodos de tiempo, los resultados demuestran que el método provee una nube de puntos 3D (tridimensionales) que permite conocer de una manera digital los cambios de las construcciones arquitectónicas y trayectorias con respecto al tiempo.

El método de desplazamiento de fase emite un haz de láser a una frecuencia específica y el reflejo de longitud de onda se desplaza por su impacto en una superficie. El desplazamiento de la onda reflejada permite calcular con precisión la distancia entre los puntos de impacto y el escáner [Danev et al. \[2017\]](#). A diferencia del método de vuelo, el método de desplazamiento de fase puede capturar una mayor cantidad de puntos en un menor tiempo, por lo tanto tiene una mejor resolución y velocidad de escaneo, pero cuenta con un menor alcance de operación. En la publicación [Feng et al. \[2018\]](#), se propone un método de compensación de movimiento basado en profilometría por desplazamiento de fase, el cual es implementado para medir objetos rígidos, debido a los errores de fases inducidos por el movimiento, los cuales causan incertidumbres en las mediciones.

El método de triangulación consiste en el análisis de un triángulo que se presenta en un corto lapso de tiempo y se forma por líneas rectas que unen 3 puntos: emisor, receptor y el punto de reflexión. En un escaner 3D el láser es proyectado sobre una superficie, este a su vez es reflejado hacia una apertura de escaneo [Fig. 1.1](#). Los parámetros conocidos del sistema es el ángulo del posicionador láser (C_{ij}), ángulo de apertura (B_{ij}) y la distancia fija (a) entre posicionador y apertura. Mediante funciones trigonométricas es posible calcular las coordenadas del punto del haz de láser sobre la superficie de objetos. La ventaja de este método a diferencia de los mencionados anteriormente es la precisión y resolución, sin embargo la principal limitación de esta técnica es la deformación del centro energético del haz de láser [Blais et al. \[2004\]](#), [Flores-Fuentes et al. \[2014\]](#). En la publicación [Auerswald et al. \[2019\]](#), el método de triangulación es aplicado para escanear los perfiles de engranes cilíndricos afín de incrementar el rendimiento de los procesos de inspección. Los resultados demuestran una mejora en la precisión y resolución debido a que es posible generar una nube de puntos tridimensional con 1280 puntos y una precisión de $\pm 10\mu m$.

Los sistemas de triangulación dinámica tal como el TVS fueron desarrollados como una opción que permite aumentar la precisión de los sistemas clásicos de triangulación, esto por medio de referencias de conteo de pulsos por rotación, en lugar de utilizar CDD (conocido por sus siglas en inglés como Charge Couple Device) discretos. Sin embargo, el TVS aun presenta incertidumbres, por lo tanto es necesario determinar los factores que pueden afectar la exactitud como son distancias de medición, divergencia del haz de láser y luz ambiental. Para ello, existen distintos estudios que han intentado atenuar las incertidumbres con algoritmos basados en inteligencia artificial logrando resultados satisfactorios en las mediciones. Por ejemplo, en la publicación [Trujillo-Hernández et al. \[2019\]](#), para mejorar el rendimiento del TVS se implementa un algoritmo de Redes Neuronales Artificiales (conocido por sus siglas en inglés como ANN), el cual es comparado con un algoritmo de vecinos cercanos (conocido por sus siglas en inglés KNN), sin embargo la aplicación de ANN demuestra mejores resultados en el escaneo de objetos de superficies, además de una mejora significativa en la velocidad de escaneo del sistema debido que a diferencia del KNN, el cual depende de una base datos para su ejecución, ANN se basa en una función previamente entrenada. Por otro lado en la publicación [Rodríguez-Quinonez et al. \[2013\]](#), el método de triangulación es aplicado en un escáner láser para medir superficies del cuerpo humano y áreas anatómicas, sin embargo el escáner presenta una variación no lineal, por lo tanto un algoritmo ANN basado en el entrenamiento *Levenberg–Marquardt* es implementado para ajustar la variación e incrementar la precisión del sistema. Los resultados presentados demuestran que el método basado en ANN es más eficiente. Por último, en la publicación [Li et al. \[2012\]](#) se presenta un escáner 3D, el cual presenta un error de medición, por lo tanto una ANN es aplicada para predecir el error y disminuir la dispersión de la nube de puntos, los resultados demuestran que la aplicación de la ANN presenta una disminución de dispersión y ruido en la nube de puntos tridimensional generada.

1.2. Principio de Triangulación

El principio de triangulación dinámica consiste en el análisis de un triángulo que existe en un corto tiempo, y se forma por líneas rectas que unen 3 puntos: emisor, receptor, y el punto de reflexión del haz de láser. En el Sistema de Visión Técnica

(conocido por sus siglas en inglés TVS) basado en triangulación dinámica, el posicionador láser (conocido por sus siglas en inglés PL) proyecta un haz de láser sobre la superficie de un objeto, posteriormente es reflejado hacia la apertura de escaneo (conocido por sus siglas en inglés SA), como resultado se forma un triángulo tal como se muestra en la Fig. 1.1 a). Por medio de esta información y con la aplicación de funciones trigonométricas es posible calcular las coordenadas, donde el haz de láser hace contacto con la superficie del objeto [Real-Moreno et al. \[2018\]](#). En la triangulación dinámica el ángulo C_{ij} se forma entre el PL y la distancia a y el ángulo B_{ij} se forma entre SA y la distancia a .

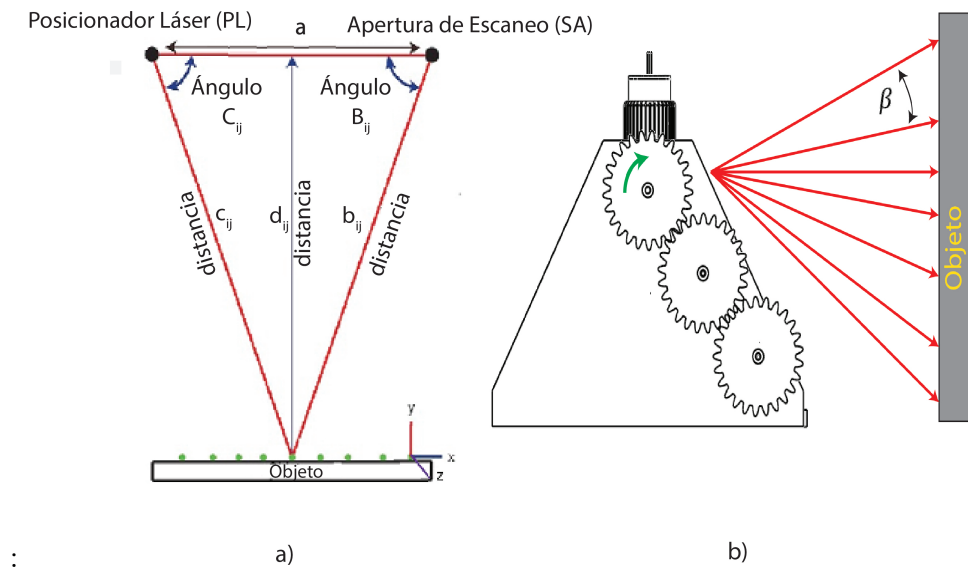


FIGURA 1.1: a) Principio de Triangulación b) Ángulo β

Para calcular las coordenadas (X, Y, Z) de cada punto sobre el objeto a escanear es necesario aplicar las ecuaciones (1.1), (1.2) y (1.3) implementadas en trabajos anteriores [Lindner et al. \[2016\]](#), donde el ángulo β mostrado en la Fig. 1.1 b) es utilizado para calcular las coordenadas de elevación sobre el eje Z .

$$X = (a) \cdot \frac{\sin(B_{ij}) \cdot \sin(C_{ij})}{\sin(B_{ij} + C_{ij})} \quad (1.1)$$

$$Y = (a) \cdot \left(\frac{1}{2} - \frac{\cos(B_{ij}) \cdot \sin(C_{ij})}{\sin(B_{ij} + C_{ij})} \right) \quad (1.2)$$

$$Z = (a) \cdot \left(\frac{\sin(B_{ij}) \cdot \sin(C_{ij}) \cdot \tan(\beta)}{\sin(B_{ij} + C_{ij})} \right) \quad (1.3)$$

La SA es el elemento que tiene la función de recepción en el sistema de triangulación dinámica. Cuando el PL proyecta un haz de láser sobre el objeto a escanear, este se refleja hacia SA, donde un espejo a 45° cambia la dirección del ángulo del haz de láser a 90° . Posteriormente, el haz de láser pasa por un par de lentes convergentes que tienen la función de concentrar las líneas paralelas en un solo punto en dirección hacia el fotoreceptor.

El PL es el elemento que tiene la función de proyectar un haz de láser hacia el objeto a escanear. El PL apunta en dirección hacia un espejo de 45° que cambia la dirección del ángulo del haz de láser a 90° . Una vez efectuado el cambio de dirección el espejo de 45° gira de 30° a 150° sobre el eje X para barrer la superficie del objeto [Trujillo-Hernández et al. \[2019\]](#).

1.3. Planteamiento del Problema

Actualmente los sistemas de visión han permitido incrementar la eficiencia de diferentes aplicaciones debido a que proveen la ubicación y el modelo digital de las superficies de los objetos. Los sistemas de visión se basan en diferentes métodos, cada uno presenta ventajas y desventajas, sin embargo la triangulación ha tenido buenos resultados en precisión y resolución. Los sistemas de triangulación pueden ser clasificados en sistemas de triangulación estática y sistemas de triangulación dinámica. La triangulación estática se basa en receptores (CCD o arreglos de sensores fijos) para realizar el principio de triangulación y detectar las variaciones de profundidad, cuando los ángulos del emisor y receptor son modificados se conoce como triangulación dinámica. En ambos casos pueden utilizarse para medir en tiempo real y crear una nube de puntos de objetos, sin embargo la triangulación dinámica permite escanear un campo más amplio. En previas investigaciones [Real-Moreno et al. \[2017\]](#), [Trujillo-Hernández et al. \[2019\]](#) el TVS bajo el principio de triangulación dinámica ha demostrado resultados favorables en superficies bidimensionales, sin embargo aún presenta limitaciones tales como: incertidumbres, baja velocidad de escaneo, incapacidad de escanear en lugares de alta intensidad de luz, y no se ha implementado adecuadamente un eje para la obtención del ángulo β , por lo tanto no es posible medir superficies tridimensionales.

a) Las incertidumbres en el sistema de triangulación dinámica se presentan por los siguientes factores: falsos contactos entre los componentes electrónicos y la deformación del haz de láser, por lo tanto es necesario crear un circuito impreso para sustituir la placa de prototipos. Además de establecer un método para encontrar el centro energético del haz de láser [Flores-Fuentes et al. \[2014\]](#).

b) La baja velocidad de escaneo en el TVS se debe a la lenta velocidad de desaturación del fotoreceptor, lo cual limita que el motor de la apertura de escaneo trabaje a su máxima capacidad. Otra causa de la baja velocidad de escaneo se debe a que se utiliza un algoritmo KNN para incrementar la precisión del ángulo B_{ij} , sin embargo la aplicación de KNN utiliza una base de datos formada por atributos de entrada C_{ij} , B_{ij} y $ticks$ (reloj interno del CompactRIO a 40 Mhz) y una salida conocida como *compensación*. La desventaja de este algoritmo es que el cálculo de distancias utilizado para comparar los atributos de entrada es lento [Cambroner and Moreno \[2006\]](#).

c) La alta intensidad de luz en el TVS causa una saturación del fotoreceptor debido a que es sensible a la luz ambiental, por lo tanto es necesario implementar un filtro que permita el paso de la luz con una longitud de onda de 650 nm (luz roja) suprimiendo o atenuando la luz restante.

d) El eje de rotación β como el que se muestra en la Fig. 1.2 no ha sido implementado debido a que se requiere un diseño robusto, el cual permita montar el TVS bidimensional, por lo tanto se requiere seleccionar un motor de corriente directa con caja reductora que provea el torque suficiente para hacer girar el eje, una vez seleccionado el motor es necesario desarrollar un controlador de posición que garantice una precisión de $\pm 0.075^\circ$.

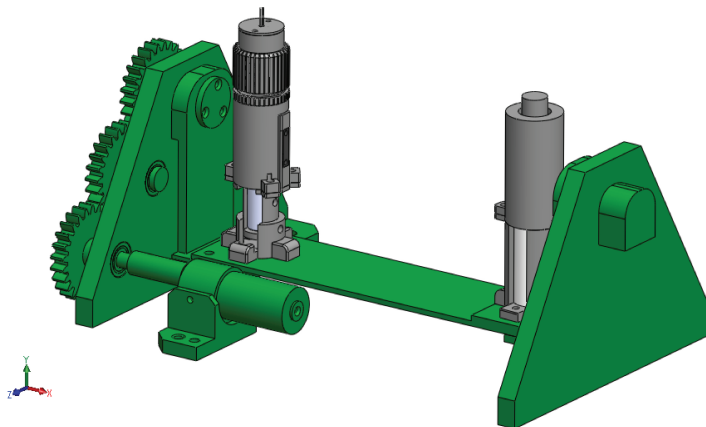


FIGURA 1.2: Las Piezas Verdes Muestran el Eje de Rotación β

Una vez implementado el eje de rotación β y analizando las oportunidades del principio de triangulación dinámica sobre el TVS podemos plantear la siguiente pregunta: ¿Cómo se pueden atenuar las incertidumbres que presenta el TVS, cuando se escanean superficies tridimensionales bajo el principio de triangulación dinámica?

1.4. Justificación y Uso de los Resultados

El desarrollo de los sistemas de visión han permitido el surgimiento de nuevas tecnologías para incrementar el rendimiento de sistemas actuales, existen diversos sistemas de visión, los cuales pueden ser divididos en diferentes categorías tales como: sensores de visión, cámaras inteligentes y visión artificial. Dentro de los sensores de visión se ubica el TVS bajo el principio de triangulación dinámica, el cual ha demostrado resultados satisfactorios en la medición de superficies bidimensionales.

En previas publicaciones [Lindner et al. \[2015\]](#), se han desarrollado nuevos métodos para mejorar la precisión, resolución y velocidad de escaneo del sistema. Sin embargo, aún no ha sido posible analizar incertidumbres en tres dimensiones, debido a que no se cuenta con un tercer eje dependiente, lo cual impide su aplicación en sistemas tales como: robots autónomos, salud estructural o biometría. Por lo tanto, este trabajo de investigación tiene como propósito el diseño y la incorporación del eje β en el TVS para realizar mediciones tridimensionales bajo el principio de triangulación dinámica. Una vez incorporado el eje de rotación β será posible analizar las incertidumbres que se presentan en superficies de tres dimensiones, además podrá ser funcional en un margen más amplio de aplicaciones, donde se requiere conocer el entorno tridimensional de rutas, lugares u objetos. Por ejemplo, en aplicaciones de robots autónomos para localizar superficies del entorno como se muestra en la Fig. 1.3, con el objetivo de encontrar trayectorias seguras que permitan a los robots evitar colisiones con objetos. Otra aplicación importante, donde puede ser utilizado el TVS, es el sector médico para el escaneo de prótesis precisas permitiendo aumentar la precisión y rendimiento de los métodos actuales [Rosicky et al. \[2016\]](#).

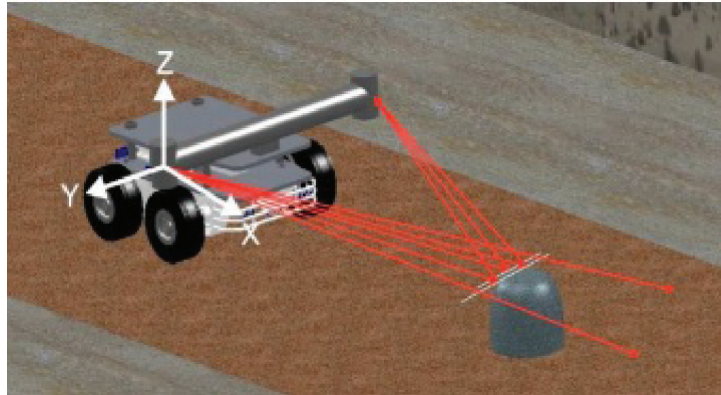


FIGURA 1.3: Aplicación del TVS en Robots Autónomos para el Reconocimiento de Superficies Tridimensionales

Como se mencionó anteriormente, el TVS bajo el principio de triangulación dinámica presenta incertidumbres en las mediciones, por lo tanto es necesario la aplicación de un algoritmo basado en ANN para atenuar los errores de medición. En publicaciones anteriores [Trujillo-Hernández et al. \[2019\]](#), una ANN ha sido aplicada para mejorar la precisión y velocidad de procesamiento de información en tiempo real, debido a que es una función previamente entrenada y a diferencia de otros algoritmos inteligentes no depende de una base de datos para comparar y realizar predicciones. En publicaciones anteriores [Real-Moreno et al. \[2017\]](#), se desarrolló un nuevo método para mejorar la precisión del ángulo B_{ij} utilizando un codificador en SA, los resultados mejoraron la precisión, sin embargo aún presenta incertidumbres. El objetivo de ANN es lograr en el entrenamiento un coeficiente de correlación (R) dentro de 0.90 a 0.94. Una vez obtenido el parámetro R se almacenarán los pesos y bias de las neuronas para implementar un programa en tiempo real que permita corregir el ángulo B_{ij} .

El interés de esta investigación es la aplicación de algoritmos inteligentes que permitan hacer más eficientes los sistemas de visión tridimensionales que ya existen, además de posicionar al TVS dentro de la categoría de sistemas de Visión Artificial.

1.5. Objetivos de la Investigación

1.5.1. Objetivo General

Realizar la medición tridimensional de superficies mediante la incorporación e instrumentación electrónica de un tercer eje (β) al sistema de triangulación dinámica y las adecuaciones de software al sistema existente, así como el análisis e implementación de un algoritmo basado en ANN que permita atenuar las incertidumbres presentadas en el sistema, para la digitalización de superficies tridimensionales.

1.5.2. Objetivos Específicos

- Analizar, diseñar e implementar el mecanismo para la incorporación del eje β en el sistema de barrido láser.
- Implementar un controlador proporcional derivativo y un controlador de modulación por ancho de pulsos para el posicionamiento del motor DC que girará el eje de rotación β .
- Desarrollar un programa en el software LabVIEW que calcule las coordenadas en el eje z, aplicando la ecuación 1.3.
- Implementar una interfaz virtual que permita mostrar nubes de puntos tridimensionales al finalizar el proceso de escaneo.
- Entrenar un algoritmo basado en ANN con el objetivo de alcanzar un R entre 0.90 a 0.94 para evitar el sobre entrenamiento de la red.
- Programar el algoritmo de ANN en el ambiente de programación LabVIEW.
- Realizar pruebas de escaneo de puntos con el algoritmo de ANN y comparar los resultados obtenidos con el algoritmo KNN.
- Eliminar los falsos contactos mediante la implementación de un circuito impreso que sustituya la placa de prototipos existente.

1.6. Metodología

Anteriormente, el cuarto prototipo del TVS requería de un eje auxiliar Z como el que se muestra en la Fig. 1.4, el cual era utilizado para medir superficies tridimensionales, el movimiento era generado por un motor a pasos unido a un tornillo sin fin que permitía desplazar el objeto a lo largo del eje Z. Este funcionamiento no se considera práctico ni eficiente debido a que no se puede emplear en aplicaciones tales como: robots autónomos, salud estructural o biometría, por lo tanto se excluirá el eje lineal y se incorporará un eje de rotación β en el TVS, lo cual permitirá medir superficies tridimensionales sin necesidad de un eje auxiliar. El giro del eje de rotación β será transmitido por medio de un motor DC y un juego de engranes que girarán la flecha. La flecha será unida a un codificador de cuadratura que proveerá pulsos (a través el canal A y B) equivalentes a la posición del motor con una precisión de $\pm 0.075^\circ$. Para el conteo de pulsos se implementará un programa de precisión cuádruple a nivel FPGA en cRIO-9063. Este dispositivo de adquisición de datos garantiza un eficiente conteo de pulsos debido a que opera a un ciclo de reloj de 40 MHz [Culler and Farnsworth \[2019\]](#). Posteriormente, se programará un controlador proporcional derivativo (conocido por sus siglas en inglés PD) en el ambiente de programación LabVIEW. Por último, se implementará un programa de modulación de ancho de pulsos donde el ciclo de trabajo dependerá de la salida del controlador PD.

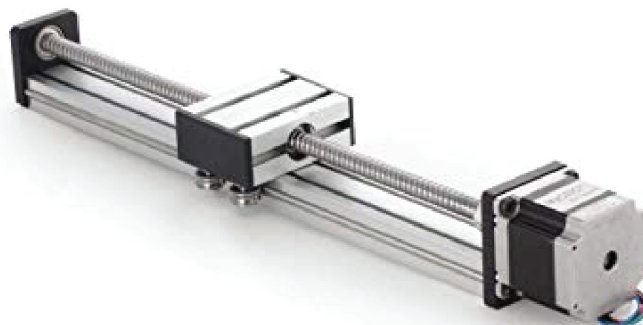


FIGURA 1.4: Eje Auxiliar Utilizado para Medir Superficies Tridimensionales

Después de integrar el eje de rotación β en hardware y software, se realizarán las primeras pruebas para analizar el funcionamiento del sistema y las incertidumbres que se presentan en las mediciones sobre superficies tridimensionales. Posteriormente se construirá una base de datos de los atributos principales para entrenar una ANN con el objetivo de alcanzar un $0.90 \leq R \leq 0.94$. Una vez diseñado el algoritmo ANN, se programará en el ambiente de programación LabVIEW y se realizará una primera prueba, la cual consiste en tomar mediciones en dos dimensiones para conocer la atenuación de las incertidumbres en el sistema mediante el cálculo del error cuadrático medio (conocido por sus siglas en inglés como RMSE). Por último, se realizará una segunda prueba, donde se compararán las incertidumbres en tres dimensiones del TVS utilizando el algoritmo basado en KNN (implementado en el tercer prototipo) y la ANN propuesta.

Capítulo 2

Marco Teórico

2.1. Triangulación Dinámica

El principio de triangulación ha sido desarrollado con el objetivo de presentar una solución al cálculo de coordenadas, a través de la división y medición de un área por triángulos, que forman una malla donde los ángulos y distancias se calculan usando funciones trigonométricas. Para implementar el método de triangulación se requiere de tres variables principales: el ángulo de apertura de escaneo, el ángulo del posicionador láser y la distancia fija entre ellos conocida como a . Una variante de los sistemas de triangulación estática son los sistemas de triangulación dinámica. La triangulación estática se presenta cuando el ángulo de apertura de escaneo y del posicionador de láser no varían, caso contrario se denomina triangulación dinámica [Lindner et al. \[2015\]](#).

2.1.1. Posicionador Láser

Como se muestra, en la Fig. 2.1 a), el posicionador láser está compuesto por un motor DC, un codificador, un espejo a 45° , un láser y una estructura basada en ácido poliláctico (conocido por sus siglas en inglés como PLA). El láser se encuentra ubicado en la parte superior del posicionador láser apuntando hacia el espejo de 45° para proyectarlo sobre las superficies a escanear.

2.1.2. Apertura de Escaneo

Como se muestra Fig. 2.1 b), la apertura de escaneo está compuesta por un motor DC, un codificador, un espejo a 45°, un sensor cero, un par de lentes biconvexas y un fotoreceptor. El funcionamiento ocurre cuando el posicionador láser proyecta el haz de láser sobre la superficie, entonces este es reflejado hacia la apertura de escaneo, cuando el haz de láser se alinea al espejo de 45° lo direcciona ortogonalmente hacia un par de lentes biconvexas para enfocararlo hacia el fotoreceptor. Cabe destacar que la señal del sensor cero permite la detección del haz de láser por el receptor, cuando el espejo de 45° da una revolución completa de 360° (el sistema envía una señal de referencia para iniciar el calculo del ángulo B_{ij}) o si el láser es detectado.

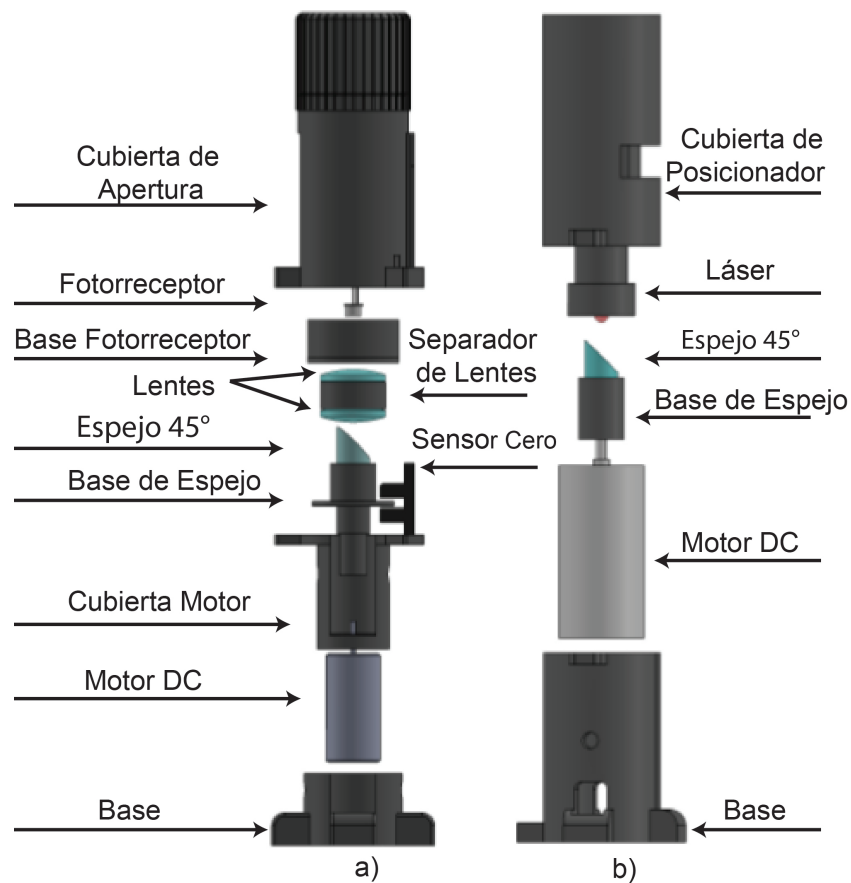


FIGURA 2.1: a) Posicionador Láser b) Apertura de Escaneo

2.1.3. Detección del Ángulo B_{ij}

Para calcular el ángulo de apertura sobre la superficie a escanear, es necesario contar los pulsos del codificador (*canalA*), el cual provee una cantidad de pulsos en cada giro del motor. También se requiere generar un tren de pulsos y obtener la señal que se produce cuando el haz de láser incide sobre el fotoreceptor (PR). Por último, es necesaria la señal de referencia producida por el sensor cero para iniciar un nuevo conteo de pulsos. Las señales mencionadas anteriormente se muestran en la Fig. 2.2.

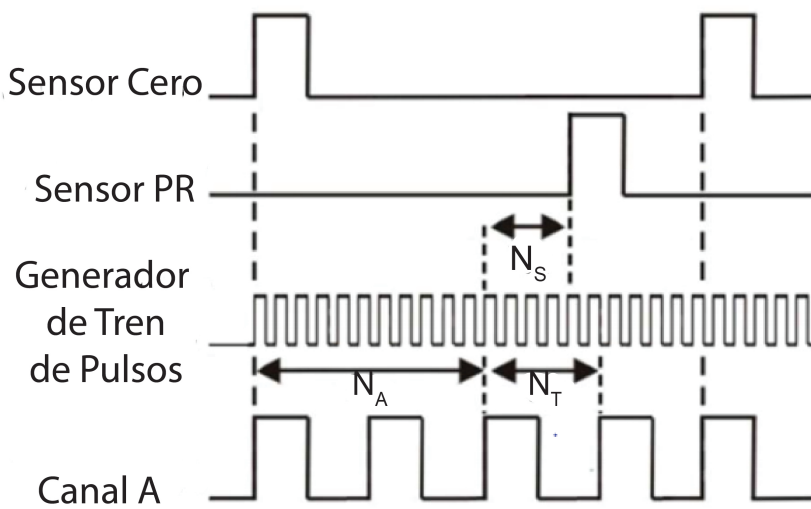


FIGURA 2.2: Señales Utilizadas para Calcular B_{ij}

El método consiste en contar los pulsos del *canalA* y el del generador de tren de pulsos después de recibir un pulso del sensor cero, si después de otro pulso del *canalA*, el haz de láser no ha sido detectado el tren de pulsos se establece en cero, en caso contrario, si el láser es detectado, el número de pulsos contados del generador del tren de pulsos al momento de la detección y el número de pulsos hasta el siguiente pulso del *canalA* son utilizados para calcular el ángulo B_{ij} [Real-Moreno et al. \[2017\]](#). El ángulo B_{ij} puede ser calculado aplicando las ecuaciones 2.1 y 2.2.

$$B_{ij} = P_A \left(N_A + \frac{N_s}{N_t} \right) \quad (2.1)$$

$$P_A = \frac{360^\circ}{P_{360}} \quad (2.2)$$

Donde N_A representa el número de pulsos del canal A antes del pulso del sensor PR, N_s es el número de pulsos contados del generador tren de pulsos de N_A hasta el pulso del PR sensor, N_T es el número de pulsos de un periodo de los pulsos del canal A. P_A son los grados por pulso del canal A y P_{360} son los pulsos del codificados cada revolución.

2.2. Inteligencia Artificial

En los últimos años la inteligencia Artificial (IA) ha despertado un interés debido a que hace posible el aprendizaje de las máquinas en base a la experiencia, lo cual permite realizar tareas como los humanos. Dicho de otra manera intenta simular el funcionamiento del cerebro al tomar decisiones [Boden \[2017\]](#). Una rama de la IA es el aprendizaje de máquinas, el cual es aplicado para procesar datos más rápido o poder realizar modelos estadísticos y proyecciones futuras de datos [Ponce et al. \[2014\]](#).

2.2.1. Definición

A lo largo de la historia no se ha podido emplear una definición precisa de IA, sin embargo científicos han definido el término en base a su criterio. A continuación se presentan algunas de las definiciones:

- a) Ciencia de la obtención de máquinas que logren hacer cosas que requerirían inteligencia si las hiciese el humano [Minsky \[1968\]](#).
- b) Nuevo esfuerzo excitante que logre que la computadora piense como máquinas con mentes, en el sentido completo y literal [Haugeland \[1985\]](#).
- c) Estudio de la computación que observa que una máquina sea capaz de percibir, razonar y actuar [Hirschhorn et al. \[1992\]](#).
- d) Una máquina inteligente es la que realiza el proceso de analizar, organizar, y convertir los datos en conocimiento, donde el conocimiento del sistema es información estructurada adquirida y aplicada para reducir la ignorancia o incertidumbre sobre una tarea específica a realizar por esta [Pajares and Santos \[2006\]](#).
- e) Ciencia que busca la comprensión profunda de la inteligencia [Smarandache and Leyva-Vázquez \[2018\]](#).

2.2.2. Algoritmos de Máquinas de Aprendizaje

El aprendizaje de máquinas es una rama de las ciencias de la computación y una aplicación de la IA que permite a los ordenadores adquirir técnicas a través del análisis de datos para conseguir nuevos conocimientos. Los principales grupos de algoritmos de máquinas de aprendizaje son:

a) Aprendizaje por Refuerzos: la función de los algoritmos consiste en optimizar el resultado de un problema por medio de la iteración constante de prueba y error, este se encuentra compuesto por un agente y entorno. El agente tiene el objetivo de entrenarse en el entorno hasta alcanzar un rendimiento definido. Por otro lado, el entorno responde a la acción del agente por medio de acierto o error entre mayor sea la certeza incrementará el desempeño del algoritmo. Los algoritmos más utilizados en aprendizaje por refuerzos son: Q aprendizaje, criterio de optimalidad, fuerza bruta, método de Montecarlo y el método de diferencias temporales.

b) Aprendizaje Supervisado: los algoritmos trabajan con datos etiquetados, los cuales intentan encontrar una función, que dada las entradas estime una salida adecuada, los algoritmos se entrenan con una base de datos históricos, esta es la manera como aprenden para asignar etiquetas adecuadas a un nuevo valor. Los algoritmos pueden ser utilizados para clasificación y regresión, la clasificación intenta categorizar en grupos en base a los datos de entrada mientras que la regresión predice un valor numérico. Los algoritmos más utilizados en aprendizaje supervisado son: árboles de decisión, clasificación de Naive Bayes, regresión por mínimos cuadrados, vecinos cercanos, máquinas de soporte vectorial y redes neuronales.

c) Aprendizaje no Supervisado: no es necesario disponer de datos etiquetados para el entrenamiento, ya que no se busca la reproducción de un resultado conocido, si no el reconocimiento de nuevos patrones o resultados. Los algoritmos más utilizados en aprendizaje no supervisado son: algoritmo de clustering, k-medias, análisis de componentes principales y análisis de componentes independientes.

2.2.3. Aplicaciones

Los algoritmos de máquinas de aprendizaje son utilizados en diversas aplicaciones tales como: reconocimiento facial, reconocimiento de voz, clasificación, regresión, segmentación de audiencia, juegos, autos autónomos, salud, economía y motores

de recomendación. En la Fig. 2.3 se muestran ejemplos de aplicaciones para cada uno de ellos.

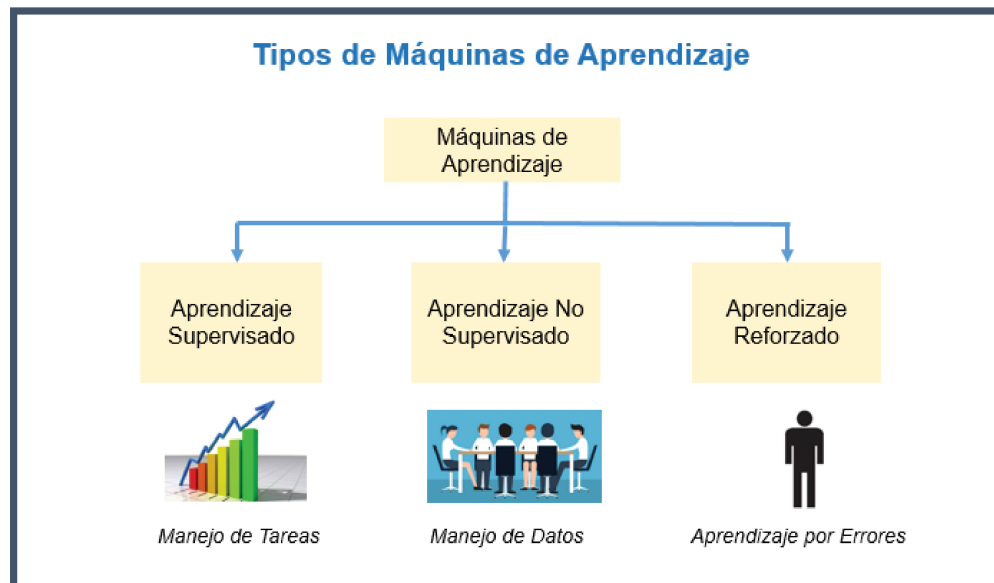


FIGURA 2.3: Tipos de Máquinas de Aprendizaje

2.2.4. Redes Neuronales Artificiales (ANN)

Dentro de las máquinas de aprendizaje supervisados y no supervisados se encuentra las ANNs, su funcionamiento se basa en neuronas artificiales (similares a las neuronas biológicas), las cuales pueden emular la manera en la que el cerebro procesa la información. Dentro de las principales ventajas de las redes neuronales artificiales con respecto a otros algoritmos inteligentes, que destaca la velocidad de procesamiento, debido a que es una función previamente entrenada con una base de datos de nuestro interés.

2.2.4.1. Neurona Biológica

Una neurona biológica es una célula especializada que procesa información a través de estímulos basados en impulsos eléctricos. La Fig. 2.4 muestra las partes que componen una neurona biológica, las cuales se definen como:

- a) Dendritas: Fibras que reciben los impulsos de otras neuronas y las envían al núcleo.
- b) Axones: Fibras que transmiten el potencial de acción (excitación o inhibición) a través de una o más sinapsis.
- c) Sinapsis: Mecanismo entre dos o más neuronas.
- d) Núcleo: Es el componente integrador que genera la respuesta.

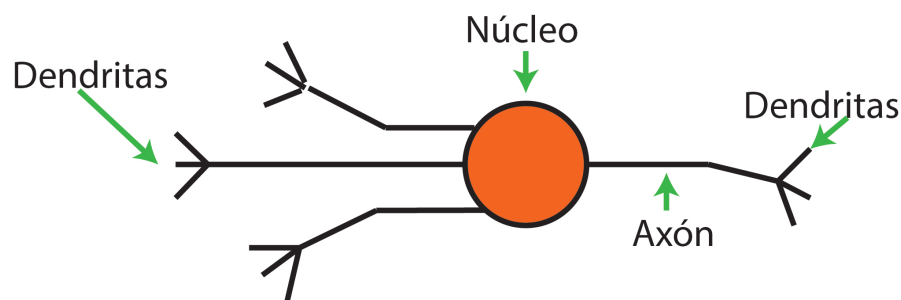


FIGURA 2.4: Neurona Biológica

2.2.4.2. Neurona Artificial

Una neurona artificial es una unidad de procesamiento similar a la neurona biológica que recibe diversas entradas ya sean externas o de otra neurona, cuando se combinan producen una salida. La Fig. 2.5 muestra las partes que componen una neurona artificial y se definen como:

- a) Pesos (W_n): se van modificando de cierta manera para adaptar el error de la salida de tal forma que minimice el error con respecto al valor real, este principio es el que usa la neurona artificial para aprender.
- b) Entradas (X_n): pueden ser clasificadas en dos grupos, binarias o continuas. Las neuronas binarias, sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos (0,1) o (-1,1). Por su parte, las neuronas continuas admiten valores dentro de un determinado rango, que en general suele definirse como (-1, 1).
- c) Función de activación (F_n): calcula la salida de la neurona a partir de los valores de X y en algunos casos de la salida anterior de la neurona artificial.
- d) Salida (Y_p): proporciona el valor de y_p de la neurona, en base a la F_n de la neurona.

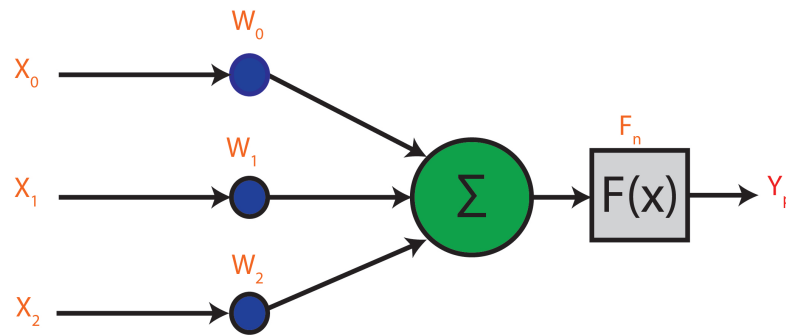


FIGURA 2.5: Neurona Artificial

2.2.4.3. Estructura y Entrenamiento de una ANN

Una ANN es un conjunto de neuronas artificiales conectadas entre sí, las cuales se encuentran en diferentes niveles (capas) como se muestra en la Fig. 2.6. Las capas se pueden dividir en tres tipos:

- a) Capa de entrada: en esta capa se recibe la información de las fuentes externas de la red.
- b) Capa oculta: son internas de la red y no tienen contacto directo con el exterior. Pueden estar interconectadas de distintas maneras, lo que determina junto con su número las distintas tipologías de redes.
- c) Capa de salida: transfieren la información de la red hacia el exterior

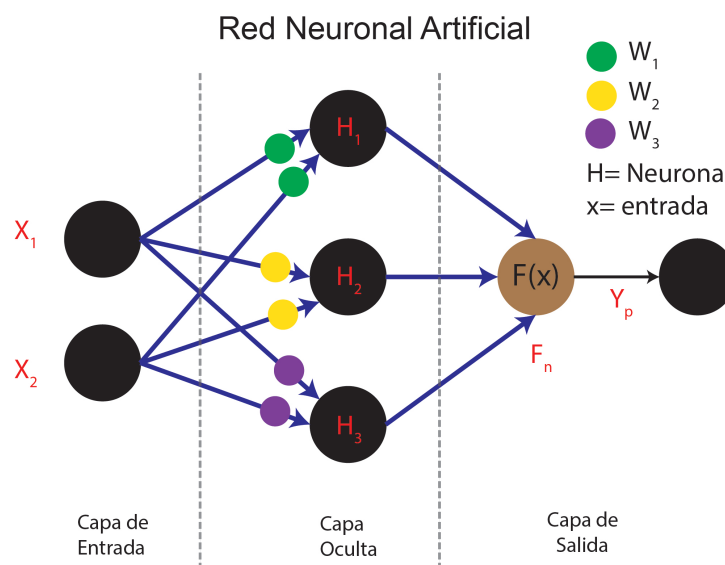


FIGURA 2.6: Red Neuronal

El aprendizaje de una ANN se logra con el algoritmo de propagación hacia adelante y la propagación hacia atrás. La propagación hacia adelante consiste en obtener el valor predicho para compararlo con el valor real de esta manera se calcula el error. Los pesos (W_1, W_2 y W_3) de la primera interacción son inicializados aleatoriamente, posteriormente los datos de la capa de entrada son multiplicados por los W para formar la capa oculta. Este cálculo se realiza aplicando las ecuaciones 2.3, 2.4 y 2.5

$$H_1 = (X_1.W_1) + (X_2.W_1) \quad (2.3)$$

$$H_2 = (X_1.W_2) + (X_2.W_2) \quad (2.4)$$

$$H_3 = (X_1.W_3) + (X_2.W_3) \quad (2.5)$$

La salida de la capa oculta pasa a través de una función de activación F_n para formar Y_p , como se muestra en la ecuación 2.6.

$$Y_p = F_n(H_1, H_2, H_3) \quad (2.6)$$

Por otro lado, la propagación hacia atrás consiste en minimizar el error encontrando su derivada con respecto a cada W y luego restarlo del valor de W . Este cálculo se realiza aplicando las ecuaciones 2.7, 2.8 y 2.9.

$$W_1 = W_1 - \frac{\eta \cdot \partial \cdot e}{\partial \cdot W_1} \quad (2.7)$$

$$W_2 = W_2 - \frac{\eta \cdot \partial \cdot e}{\partial \cdot W_2} \quad (2.8)$$

$$W_3 = W_3 - \frac{\eta \cdot \partial \cdot e}{\partial \cdot W_3} \quad (2.9)$$

Donde η representa la tasa de aprendizaje.

2.3. Teoría del Control Clásico

Los sistemas de control han revolucionado la manera en la que operan una amplia variedad de aplicaciones en sectores industriales tales como: petroleras, gaseras, químicas, farmacéuticas y la red de energía entre otras [Pérez et al. \[2014\]](#). El objetivo de un sistema de control es conseguir, mediante la manipulación de variables de control, un dominio sobre la variable de salida, de modo que alcance valores predeterminados, dicho de otra manera, los sistemas de control regulan el comportamiento de una planta con el fin de controlar su funcionamiento. Los sistemas de control se pueden clasificar en lazo abierto y lazo cerrado.

En los sistemas de control de lazo abierto no existe una retroalimentación de la señal de salida, es decir, la salida no tiene efecto sobre la acción de control, debido a que estos sistemas no utilizan la señal de salida para compararla con la entrada, por lo tanto es necesario asignar una tarea fija a cada combinación de entradas. En estos sistemas de control la precisión depende de la calibración del controlador [Bolton and Ramírez \[2001\]](#). En la Fig. 2.7 se muestra el proceso de un sistema de control de lazo abierto.



FIGURA 2.7: Control Lazo Abierto

En los sistemas de control de lazo cerrado a diferencia de los de lazo abierto, si existe una retroalimentación (positiva y negativa), es decir la salida presenta un efecto sobre la acción de control, por lo tanto se utiliza como una de las entradas del sistema. La retroalimentación se da por medio de sensores, debido que proveen información del comportamiento del sistema (señal controlada) para compararla con la señal de referencia con el fin de determinar el error y proveerlo al controlador. En la Fig. 2.8 se muestra el proceso de un sistema de control de lazo cerrado.

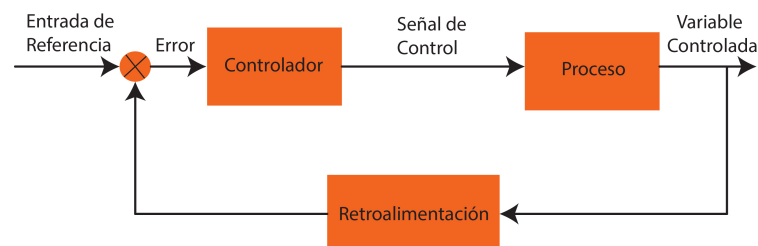


FIGURA 2.8: Control Lazo Cerrado

Dentro de las definiciones básicas de control tenemos: [Ogata \[2003\]](#):

- a) Entrada de referencia: Es el valor esperado a obtener en la señal de salida.
- b) Variable controlada: Es la cantidad o condición que se mide y controla, por lo común la variable controlada es la salida del sistema.
- c) Error (e): Es la diferencia entre la entrada de referencia y variable controlada.
- d) Señal de control: Es la señal que produce el controlador para modificar la variable controlada de tal forma que se disminuya el error.
- e) Proceso: habitualmente también es conocido como planta, y es elemento físico que se pretende controlar (actuador).
- f) Retroalimentación: Se refiere a una operación que en presencia de perturbaciones, tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia y lo continúa haciendo en base a esa referencia.

2.3.1. Controlador Proporcional

La acción de control proporcional intenta minimizar el error del sistema a través de un amplificador de ganancia ajustable. La variable controlada ($y(t)$) se resta de la entrada de referencia ($r(t)$) para calcular el error del sistema como se muestra en la ecuación 2.10. Si en el sistema se genera un error ($e(t)$) grande de igual manera se producirá una señal de control de gran magnitud con el fin de minimizar el error [Sarasola \[2007\]](#).

$$e(t) = y(t) - r(t) \quad (2.10)$$

La salida del controlador $u(t)$ consiste en la multiplicación del e por la ganancia proporcional (K_p) como se muestra en la ecuación 2.11.

$$u(t) = K_p \cdot e(t) \quad (2.11)$$

La ganancia K_p es una constante ajustable utilizada para aumentar la velocidad de respuesta del sistemas y disminuir el error, sin embargo es necesario ajustar la ganancia K_p para eliminar la inestabilidad del sistema. El inverso de la ganancia K_p es un parámetro conocido como banda proporcional (B_p), la cual es la región donde se tiene control sobre el proceso [Solé \[2009\]](#). La banda proporcional consiste en la relación del 100 entre K_p como se muestra en la ecuación 2.12.

$$u(t) = \frac{100}{K_p} \quad (2.12)$$

En la Fig. 2.9 se puede observar que a medida que aumenta el error del sistema, la salida $u(t)$ incrementará linealmente, además podemos concluir que al aumentar la ganancia K_p la banda proporcional disminuye por lo que aumenta rápidamente la respuesta del sistema.

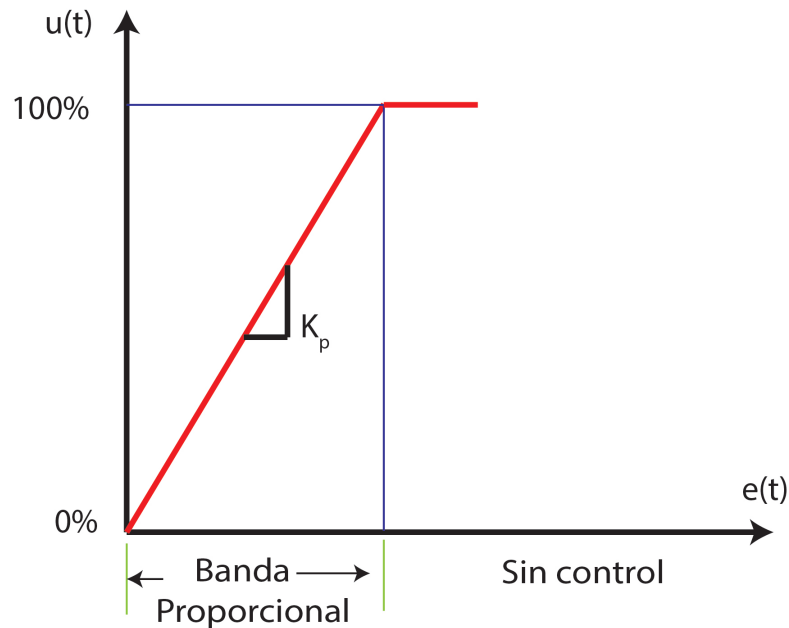


FIGURA 2.9: Banda Proporcional

2.3.2. Controlador Derivativo

La acción derivativa consiste en calcular la tangente en el tiempo actual para anticipar el comportamiento de la señal del error de nuestro proceso en el futuro. La anticipación del e es una aproximación basada en la serie de Taylor, como se muestra en la ecuación 2.13.

$$e(t + t_d) = e(t) + t_d \cdot \frac{de(t)}{dt} + \frac{t_d^2}{2!} \cdot \frac{d^2 \cdot e(t)}{dt^2} + \dots \quad (2.13)$$

Como solamente se requiere realizar una aproximación por la tangente se descartan todos los términos de orden superior, por lo tanto la serie de Taylor se modifica como en la ecuación 2.14.

$$e(t_0 + t_d) = e(t_0) + t_d \cdot \frac{de(t_0)}{dt} \quad (2.14)$$

En la Fig. 2.10 se puede observar el comportamiento del e desde t_0 hasta t_d , si se determina la tangente del punto P_1 con la serie de Taylor es posible estimar el e del Punto P_2 .

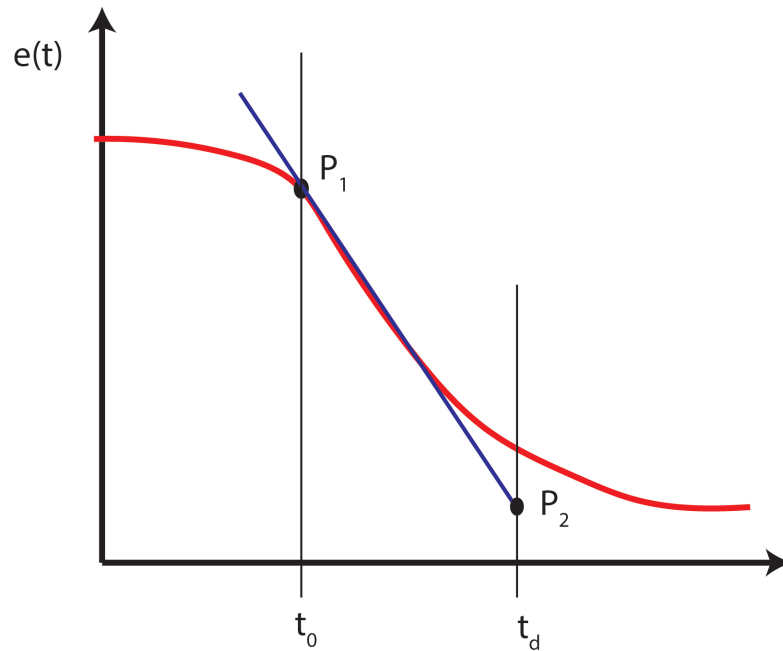


FIGURA 2.10: Control Derivativo

La salida $u(t)$ de un controlador derivativo consiste en el producto de la ganancia derivativa K_d y de la derivada del error con respecto al tiempo tal como se muestra en la ecuación 2.15.

$$u(t) = K_d \cdot \frac{de(t)}{dt} \quad (2.15)$$

2.3.3. Controlador Integral

La acción integral permite encontrar el área bajo la curva (por medio del proceso de integración) conforme el tiempo va transcurriendo. En la Fig. 2.11 el área bajo la curva representa el e que se acumula desde t_0 hasta cada instante de tiempo (t_1, t_2, t_3). Cada vez que se integra el e se aproxima a cero causando que la salida del controlador sea estable $u(t) = C$. Sin embargo, este proceso puede ser lento,

por lo tanto se utiliza una ganancia K_i para acelerar o atenuar la respuesta del sistema.

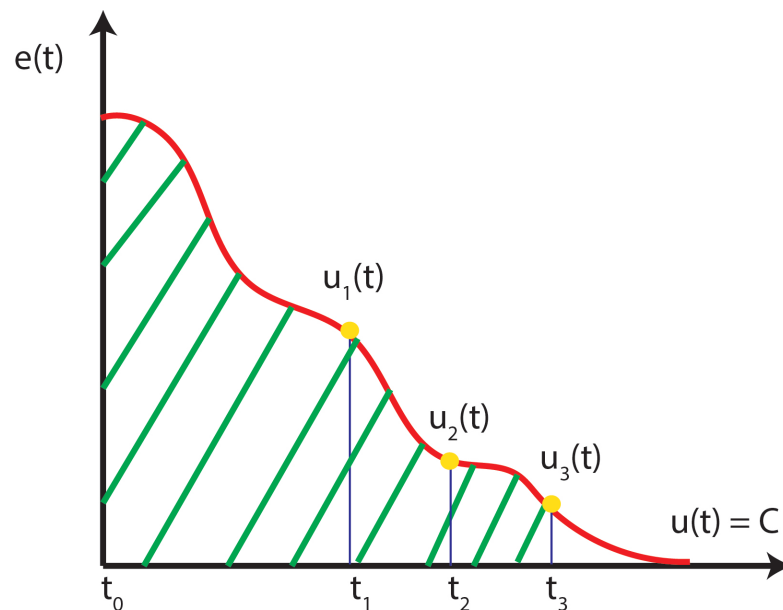


FIGURA 2.11: Control Integral

La acción integral consiste en el producto de la integral del e en un tiempo cero hasta cualquier instante de tiempo multiplicado por K_i como se muestra en la ecuación 2.16

$$u(t) = K_i \cdot \int_0^{t_i} e(t) \cdot dt \tag{2.16}$$

En conclusión este tipo de acción integra el error con respecto al tiempo, es decir va acumulando el valor del error pasado. Cada vez que se integra el error se va aproximando más a la referencia. Sin embargo, este proceso puede ser lento, por lo tanto se utiliza una ganancia K_i para aumentar o disminuir la respuesta.

2.3.4. Controlador Proporcional Derivativo

A la combinación de las acciones de control proporcional (P) y la derivativa (D) se le conoce como acción de control proporcional derivativa (PD). La expresión

matemática que define a este controlador en el dominio del tiempo viene dada por la ecuación 2.17.

$$u(t) = K_p \cdot e(t) + K_d \cdot \frac{de(t)}{dt} = K_p [e(t) + \frac{K_d}{K_p} \cdot \frac{de(t)}{dt}] = K_p [e(t) + T_d \cdot \frac{de(t)}{dt}] \quad (2.17)$$

La función de transferencia del controlador queda expresada por la siguiente ecuación 2.18

$$\frac{U(s)}{E(s)} = K_p(1 + s \cdot T_d) \quad (2.18)$$

Donde T_d es el tiempo derivativo, habitualmente en minutos.

Al analizar la función de transferencia del controlador PD se puede observar que se introduce un cero en $s = \frac{-1}{K_d}$. El cero del control PD por lo general se diseña para ubicarlo sobre un polo indeseado de lazo abierto y su efecto tiende a modificar considerablemente el comportamiento del sistema hablando en términos de la estabilidad (reduce el sobreimpulso), velocidad (la aumenta) y precisión (la aumenta). Sin embargo, cuando la influencia del controlador es muy grande tiende a ofrecer una respuesta excesivamente lenta.

El procedimiento utilizado para sintonizar las ganancias K_p y K_d es el de prueba y error [Garay \[2009\]](#). Los pasos a seguir son los siguientes:

- a) Eliminar la acción derivativa ($T_d = 0$) y ajustar K_p hasta conseguir la forma de onda deseada.
- b) Aumentar la K_p conseguida e intentar restaurar la respuesta ajustando T_d . Repetir hasta conseguir un valor de K_p tan grande como sea posible.

2.4. Tipos de Motores Reductores de Velocidad

Uno de los problemas en los prototipo 1, 2 y 3 del sistema de visión técnica (conocido por sus siglas en inglés como TVS) es la selección de un motor que disminuya la velocidad del motor DC y aumente el torque para vencer el momento que genera la masa del eje de rotación β . Además, de mantener fijo el ángulo β

sin que existan variaciones al desenergizar el motor. En el prototipo 4 se analiza el funcionamiento, las ventajas y desventajas de las diferentes transmisiones que reducen la velocidad y aumentan el torque.

2.4.1. Reductores de Velocidad Sin Fin-Corona

En este tipo de reductor un motor eléctrico aplica potencia de rotación a un husillo de acero (en forma de sin fin) que al girar hace contacto con una corona dentada (normalmente de bronce). Por cada vuelta del tornillo sin fin, se produce el avance de un diente de la corona y en consecuencia la reducción de la velocidad. La velocidad de una corona sin fin se calcula a través del producto de números de dientes de la corona y el número de entradas del tornillo sin fin. La principal ventaja es que tiene auto freno, sin embargo entre sus desventajas es que se desperdicia potencia porque están sometidos a una alta fricción por su deslizamiento entre la corona y el sin fin. Por lo que, su uso es limitado en muchas aplicaciones. En la Fig. 2.12 se muestra el mecanismo interno de un reductor de velocidad sin fin-corona Acosta and OVIEDO [2016].

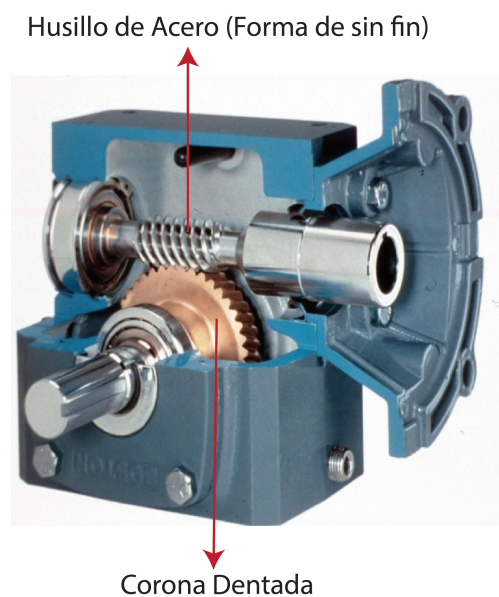


FIGURA 2.12: Reductores de Velocidad Sin Fin-Corona

2.4.2. Reductores de Velocidad Planetario

Este tipo de reductor utiliza un engrane central (conocido como sol), el cual tiene un tamaño superior y gira sobre el eje central, alrededor del engrane central giran

múltiples engranes (conocidos como planetas). A su vez existe un anillo exterior (con dientes en su cara interna) que engrana con los satélites y encierra todo el tren planetario. Una de las principales características de este tipo de reductor es que el eje central puede convertirse en centro de giro para el anillo externo, por lo que permite cambiar la dirección con facilidad. Entre las ventajas podemos encontrar mayor precisión, predictibilidad, durabilidad y mejores niveles de eficiencia. En la Fig. 2.13 se muestra el mecanismo interno de un reductor de velocidad planetario Concli and Gorla [2012].

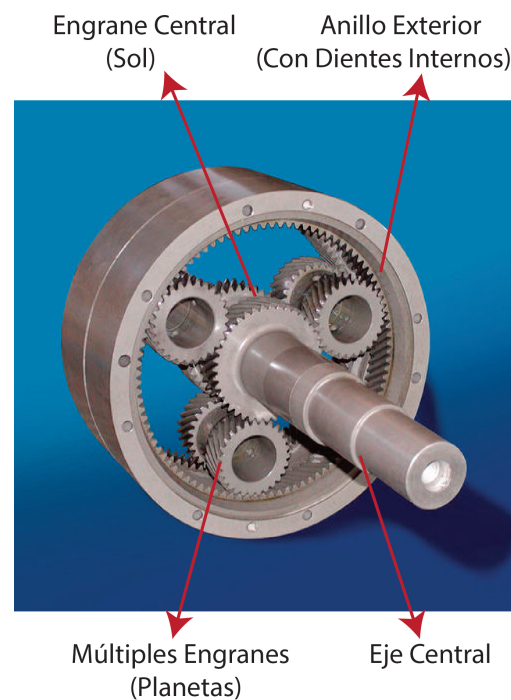


FIGURA 2.13: Reductores de Velocidad Planetario

2.4.3. Reductores con Grupo Cónico-Espiral o de Velocidad de Engranaje

Este tipo de reductor está compuesto por un par o más de engranes, los cuales se colocan en posiciones, donde cada engrane se relacione con el engrane anterior, a través de esta relación se forma una transmisión de movimientos que permite la disminución de la velocidad que genera el motor. La principal ventaja que presentan los reductores de grupo cónico-espiral es que son muy eficientes y requieren poco mantenimiento. En la Fig. 2.14 se muestra el mecanismo interno de un reductor con grupo cónico-espiral o de velocidad de engranaje Neyra Dueñas [2018].

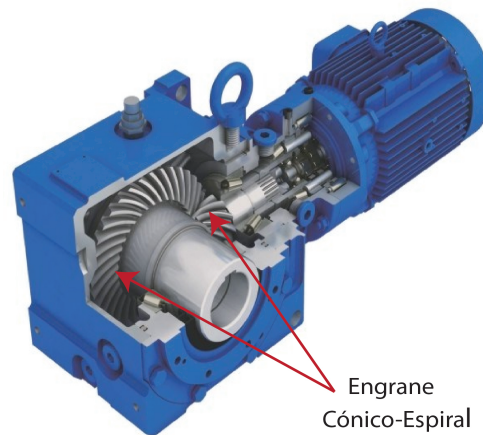


FIGURA 2.14: Reductores con Grupo Cónico-Espiral o de Velocidad de Engranaje

2.4.4. Reductores de Engranajes Cilíndrico-Helicoidales

Este tipo de reductor se basa en engranajes helicoidales, se caracterizan por el dentado oblicuo en relación al eje de rotación, pudiendo ser paralelos o cruzados, por lo general 90° . A diferencia de los reductores con engranajes rectos, los reductores con engranajes helicoidales tienen la capacidad de transmitir más potencia, velocidad y son silenciosos. En la Fig. 2.15 se muestra el mecanismo interno de un reductor de engranajes cilíndrico-helicoidales [Álvarez and Giraldo \[2006\]](#).



FIGURA 2.15: Reductores de Engranajes Cilíndrico-Helicoidales

2.5. Principio de Codificador

Los codificadores para motores eléctricos tienen la función de convertir el giro mecánico en pulsos digitales, los cuales pueden ser utilizados para conocer la velocidad o posicionamiento del motor. Un codificador se compone de un disco conectado

a un eje giratorio. El disco está hecho de vidrio o plástico y sobre su contorno tiene pequeñas dentaduras o perforaciones transparentes u opacas, las cuales permiten el paso de la luz infrarroja generada por un emisor hacia un receptor como se muestra en la Fig. 2.16. Cada vez que la luz se emite al receptor se genera un pulso, es decir cada vez que el disco gira se produce un tren de pulsos, el cual permite calcular la velocidad y posición del actuador. Los codificadores se emplean en una amplia variedad de aplicaciones tales como: robótica, minería, electrodomésticos, transporte y maquinaria entre otras [Mattson \[2009\]](#). Los codificadores se dividen según su diseño básico y funcionalidad en codificadores incrementales, absolutos, ópticos, cuadratura y lineal [Sung and Raudkivi \[1980\]](#).

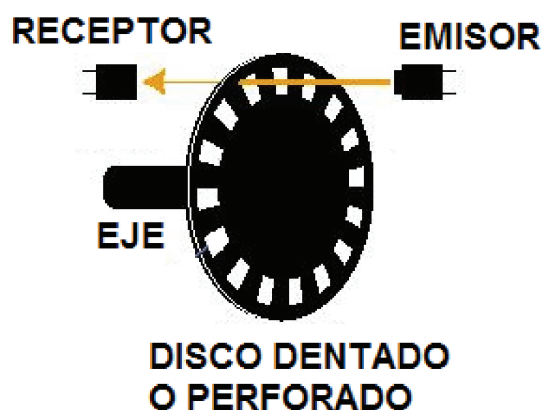
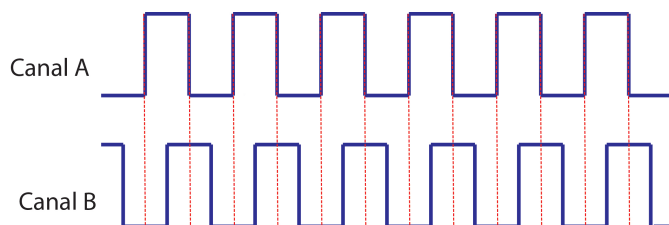


FIGURA 2.16: Disco Perforado

2.5.1. Funcionamiento del Codificador de Cuadratura

Los codificadores de cuadratura normalmente tienen dos canales denominados *A* y *B*, los cuales se encuentran desplazados a 90° uno con respecto al otro [Reyes \[2011\]](#), el desfase de las dos señales se muestran en la Fig. 2.17.

FIGURA 2.17: Desfase de 90° Entre el Canal A y Canal B

Además de conocer la posición y velocidad del actuador, el desfase entre ambos canales permite conocer el sentido de giro. Para ejemplificar lo anteriormente mencionado, tomemos de referencia los eventos alto y bajo del *canal A* como se muestra en la Fig. 2.18, podemos observar que al girar el actuador en sentido de las manecillas del reloj (conocido por sus siglas en Inglés como *CW*) se producirán los eventos (t_1, t_2, t_3, t_4) . Si se compara con el canal B los eventos son inversos.

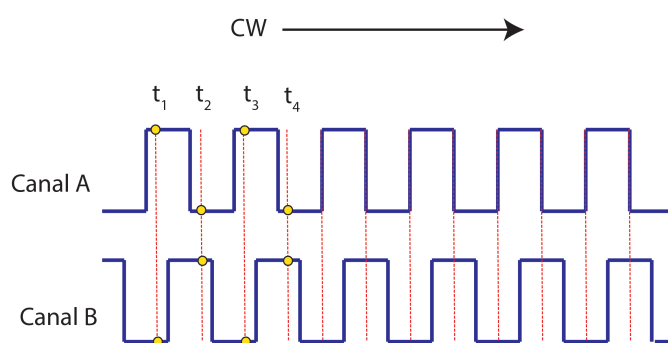


FIGURA 2.18: Giro Hacia las Manecillas del Reloj.

Si se invierte el sentido de giro en contra de las manecillas del reloj (conocido por sus siglas en Inglés como *CCW*) y se toman de referencia los eventos alto y bajo del canal A, podemos observar que ambas señales son idénticas, como se muestra en la Fig. 2.19.

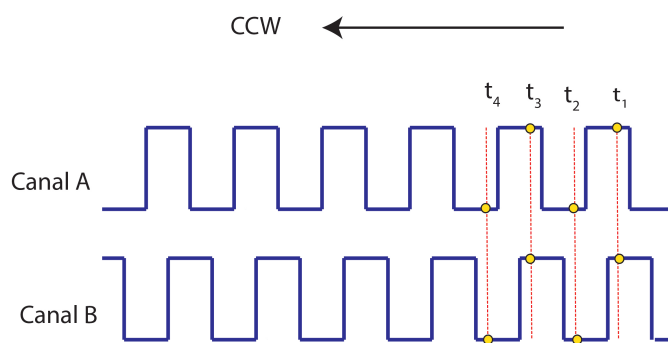


FIGURA 2.19: Giro Encontra de las Manecillas del Reloj.

Por otro lado, para el conteo de pulsos existen diferentes métodos que permiten ajustar la precisión del codificador de cuadratura, de acuerdo a la aplicación que se esté trabajando, entre ellos se encuentran:

- Precisión Simple: se utiliza un solo canal y solamente se registra un único estado (alto o bajo).
- Precisión Doble: se utilizan ambos estados (alto y bajo) en un solo canal.
- Precisión Cuádruple: se utilizan ambos estados (alto y bajo) en ambos canales.

2.6. FPGA

Una matriz de compuertas lógicas programables en campo (conocido por sus siglas en inglés como FPGA) consiste en una matriz bidimensional de bloques configurables que se pueden conectar mediante pistas de diferentes longitudes y conmutadores programables para enlazar bloques a pista o pistas entre sí. La programación de las FPGAs se realiza por medio de los conmutadores que sirven para crear las conexiones entre los distintos bloques [Wolf \[2004\]](#).

2.6.1. Estructura General de las FPGAs

La arquitectura interna de las FPGAs depende del fabricante, a continuación se presentará un esquema genérico como el que se muestra en la [Fig. 2.20](#).

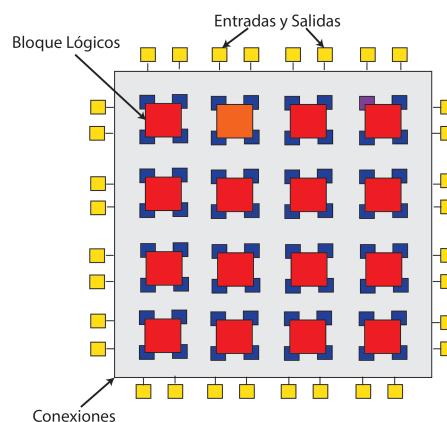


FIGURA 2.20: Estructura General de un FPGA.

Este esquema permite entender el funcionamiento de las conexiones internas y cómo interactúan entre sí. Las partes que componen la arquitectura interna de un FPGA son:

a) Bloques Lógicos Configurables: el bloque lógico consta de una sección combinatorial que permite ejecutar funciones lógicas booleanas tales como: AND, OR y NOT. Además tiene otra sección secuencial que permite sincronizar la salida con una señal de reloj externa e implementar registros. La parte combinatorial puede variar según el fabricante en: bloque basado en LUT (look-up table) o bloque

lógico basado en multiplexores. El bloque basado en LUT almacena una tabla de verdad que se modifica y se guarda en cada celda de las combinaciones de las entradas. Por otro lado, el bloque lógico basado en multiplexores requiere menos lógica que el bloque basado el LUT, debido a que se puede implementar mayor número de bloques lógicos en el mismo espacio o por el mismo número de bloques disponer de más espacio para incrementar las conexiones.

b) Interconexión Entre Bloques Programables: la tecnología utilizada para crear las conexiones entre los canales son antifusible, SRAM y Flash. Una FPGA que se basa en una tecnología de autofusible, solamente se puede programar una sola vez, la desventaja es que no se pueden reutilizar, sin embargo disminuye el tamaño y costo de los dispositivos. Por otro lado, para guardar la configuración de los circuitos se utilizan las SRAM, la cual funciona como un generador de funciones, controlador de mutiplexores y para la interconexión entre bloques. Por último, el tamaño de las celdas flash es más reducido que el de la celda SRAM, son programables pero su velocidad de programación es lenta y no son volátiles, por lo que no necesitan un dispositivo auxiliar para guardar la configuración interna [Mano \[2003\]](#).

c) Bloques de Entradas/Salidas: su principal función es permitir el paso de una señal, dentro o al exterior del dispositivo FPGA. Además, de tener recursos como: salidas configurables, entradas pull-up o pull-down programables, registros de salidas y registros de entradas.

2.6.2. Aplicaciones de las FPGA

Las FPGAs son ampliamente utilizadas en aplicaciones, donde se requiera adquirir señales a alta frecuencia de trabajo. A diferencia de los dispositivos que se programan para realizar instrucciones secuenciales, las FPGAs se pueden considerar que operan en paralelo. Las áreas donde se utilizan más son: sistemas de visión artificial, sistemas de imágenes médicas, radio definida por software, codificación, encriptación, radioastronomía, reconocimiento de voz y aeronáutica [Osio \[2019\]](#), [Bustos Gómez et al. \[2016\]](#).

2.7. Conclusiones

En este capítulo se presentaron las bases teóricas necesarias para desarrollar este trabajo de investigación, se puede concluir que se comprendió a detalle el principio de triangulación dinámica en mediciones tridimensionales, y como predecir las incertidumbres que presenta el método, empleando el algoritmo de *Redes Neuronales Artificiales*. Además, el análisis de los diferentes controladores de posición permitió definir el controlador *PD*, debido a que atenúa el sobreimpulso y mejora la precisión en el ángulo β . Con respecto a la técnica de conteo de pulsos se eligió *conteo cuádruple* debido a que incrementa la resolución cuatro veces (3600 pulsos a 14400 pulsos) mejorando la precisión a 0.025° por pulso. Por último, al analizar el funcionamiento, ventajas y desventajas de las diferentes transmisiones, se optó por utilizar el reductor de velocidad planetario, debido a que presenta una mejor precisión y permite fijar el motor en el ángulo β deseado.

Capítulo 3

Procedimiento de Investigación

3.1. Ensamble Completo del Eje β

En la Fig. 3.1 se muestra el ensamble completo del eje β donde se puede observar la apertura de escaneo y el posicionador láser montados sobre la base giratoria. También se puede ver un motor con reductor de velocidad planetario que se encuentra conectado a una transmisión de 3 engranes que permiten el movimiento del eje β .

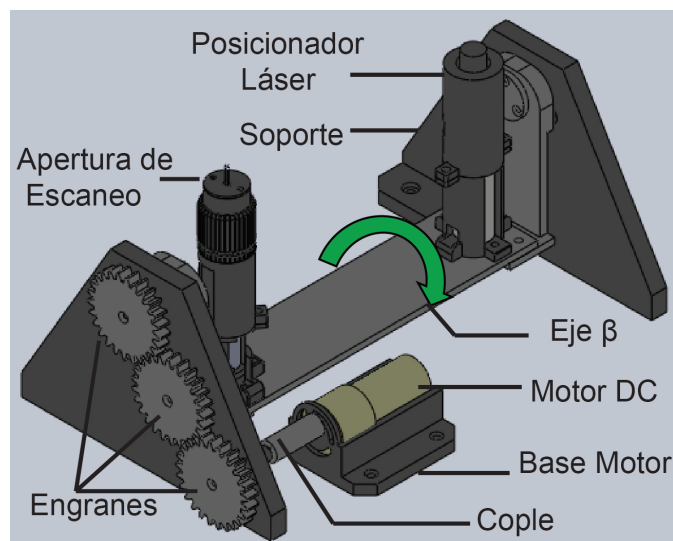


FIGURA 3.1: Propuesta para El Eje β .

La implementación del eje β se realizó utilizando los siguientes materiales:

- 1 Motor PITTMAN modelo *GM8712J015*.

- b) 1 Codificador de cuadratura *HDER – 55l2 – BY09*.
- c) 3 Engranajes rectos de 50 dientes.
- d) 1 Rollo de PLA color negro.
- e) 4 Baleros con un anillo interior de 3/8 in y un anillo exterior de 7/8 in.
- f) 1 Impresora Robo 3D.

Los diseños de las piezas se crearon en el ambiente gráfico *SOLIDWORKS*, después cada una fue convertida en formato *STL* para ser fabricadas en la impresora 3D con material PLA. Una vez realizada cada pieza se ensambló el prototipo del eje β y se montaron la apertura de escaneo y el posicionador láser sobre la base giratoria. Posteriormente, el motor se conecta a la transmisión para aumentar el torque y permitir el giro del eje de rotación β , lo cual genera el movimiento de la base giratoria como se muestra en la Fig. 3.2. Por último, se ensambla el codificador colinealmente a los espejos de 45° para determinar la posición angular del eje β .

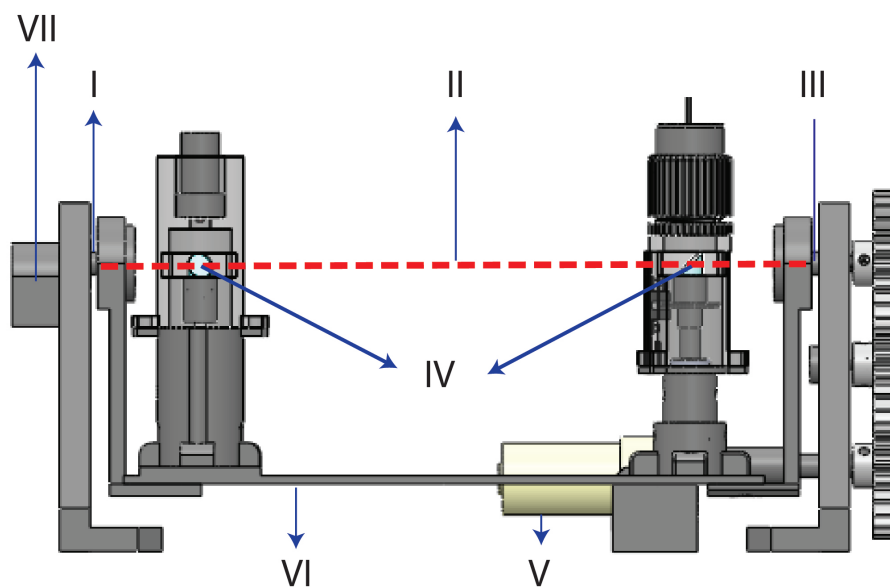


FIGURA 3.2: Propuesta para El Eje β , Donde I) Eje Inicial II) Eje Imaginario III) Eje Final IV) Espejos 45° V) Motor VI) Base Giratoria VII) Codificador

3.1.1. Fabricación del Circuito Impreso

Se diseñó un circuito impreso de doble cara en el software "Proteus" donde los cables fueron sustituidos por pistas de cobre, las cuales mejoran la conducción entre

las conexiones electrónicas de los componentes en el TVS bidimensional y evita falsos contactos, además fue diseñado el segundo circuito impreso que controla el ángulo β .

Para la implementación del circuito impreso que recibe los pulsos del codificador (de la apertura de escaneo), sensor cero y el cual tiene la función de amplificar la señal del fotoreceptor del TVS bidimensional, se realizó el diseño de las conexiones. Una vez fabricado se colocaron los componentes en su posición y fueron soldados como se muestra en la Fig. 3.3. La lista de materiales (conocido por sus siglas en inglés como BOM) requeridos para este procedimiento fueron los siguientes:

- a) 1 Amplificador operacional TL084
- b) 1 Compuerta inversora 74LS14
- c) 2 Resistencias de 220Ω
- d) 2 Resistencias de $4.7 k\Omega$
- e) 1 Resistencia de $1 k\Omega$
- f) 2 Porta integrados de 14 pines
- g) 6 Conectores terminal macho para entrada de terminal
- h) 6 Conectores terminal hembra para terminal
- i) 3 Conector pin hembra de 7, 6 y 4 entradas
- j) Estaño y caudín

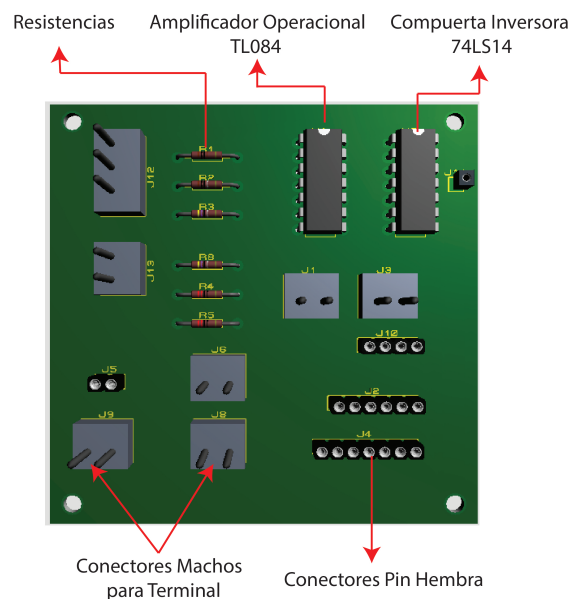


FIGURA 3.3: Circuito Impreso para el TVS Bidimensional.

Para la implementación del circuito impreso que alimenta con 5 V el codificador (del eje β) y lo protege contra cortos circuitos se diseñó un circuito impreso de una cara. El procedimiento para la fabricación del circuito impreso inicio en cortar la placa fenólica a la medida de 5 cm x 7 cm, posteriormente se configuró la cortadora láser *ProtoMatS63* y se cargó el diseño en formato Gerber. También se colocaron las brocas requeridas para hacer los barrenos de las terminales de los componentes. Después se agregó el recubrimiento para proteger al circuito impreso del entorno por ejemplo humedad, vibraciones o contaminantes químicos. Por último, como se muestra en la Fig. 3.4 se colocaron y soldaron los siguientes componentes:

- a) 2 Capacitores de 4.7 pF
- b) 1 Regulador de voltaje a 5 V LM340T5
- c) 2 Fusibles de 0.25 Amperes
- d) 2 Porta fusibles
- e) 3 Conectores de bloques (2 de tres entradas y 1 de 3 entradas)
- f) 12 Pines

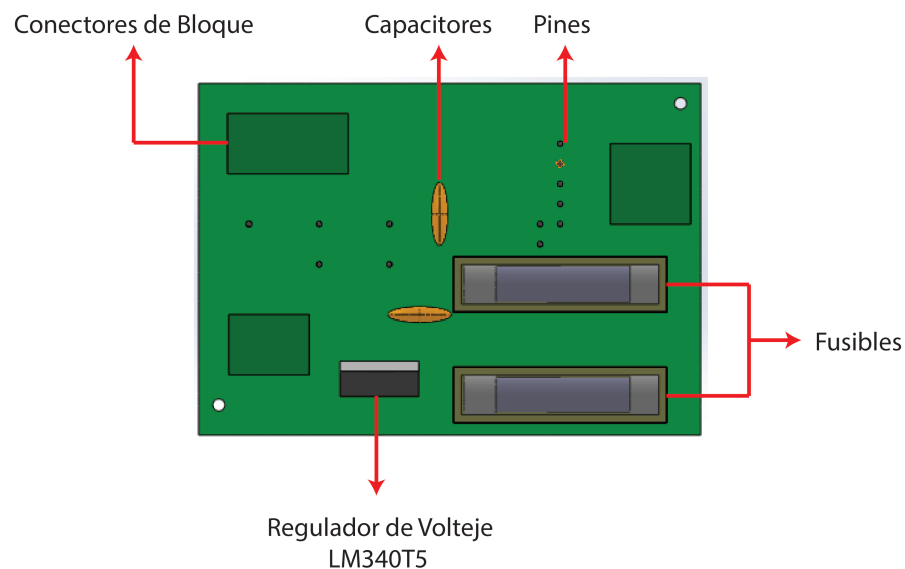


FIGURA 3.4: Circuito Impreso para el Eje de Rotacion β .

3.1.2. Conteo de Pulsos

El codificador de cuadratura *HEDR - 55L2 - BY09* tiene dos canales desfasados 90° que se conectan a dos entradas digitales del *NI9403*, cada canal provee

3600 pulsos por rotación en precisión simple, sin embargo en este trabajo de investigación se diseñó un programa a nivel FPGA para utilizar precisión cuádruple, es decir medir los estados 0 y 1 en ambos canales. Por ejemplo en la Fig. 3.5, si tomamos de referencia el canal A podemos observar que existen cuatro estados desde t_1 hasta t_4 (en un ciclo cuadrado) con respecto al canal B. Por lo tanto, podemos observar que al comparar ambos canales se presentan cuatros estados por ciclo, lo cual cuadruplica los pulsos del codificador a 14400 pulsos por revolución.

Por otro lado, como se mencionó en el marco teórico el desfase de 90° entre ambos canales permiten conocer la dirección de giro del motor. En la Fig 3.5, si tomamos como referencia el canal A podemos apreciar que cuando existe un cambio de estado de 0 a 1, el canal B tendrá un valor de 0, y cuando se presente un estado de 1 a 0, el canal B tendrá un valor de 1, bajo estas condiciones el motor girará en sentido horario.

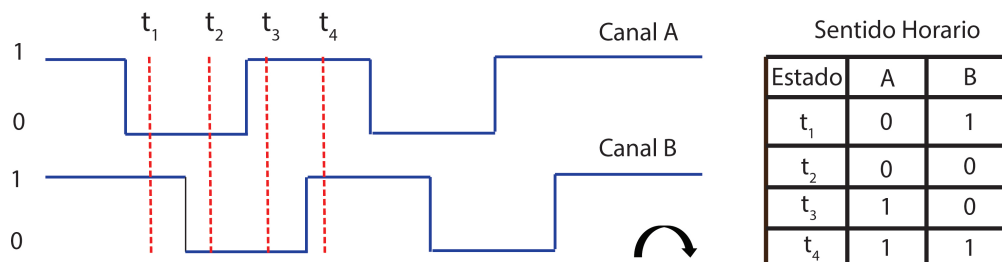


FIGURA 3.5: Pulsos en Sentido Horario.

En la Fig. 3.6, si nuevamente se toma de referencia el canal A podemos apreciar que cuando se presenta un cambio de estado de 0 a 1, el canal B tendrá un valor de 1, y cuando se presente un estado de 1 a 0, el canal B tendrá un valor de 0, bajo estas condiciones el motor girará en sentido antihorario.

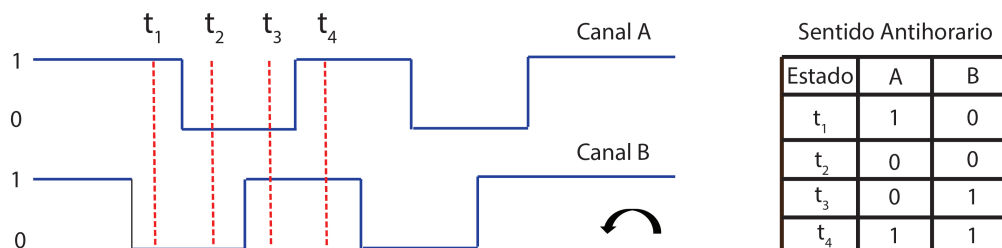


FIGURA 3.6: Pulsos en Sentido Antihorario.

Como parte del desarrollo se creó a nivel FPGA el programa conteo de pulsos para leer las señales cuadradas (canal A y canal B) del codificador y conocer la dirección de giro del motor. Cuando el motor gira en sentido horario se incrementa

el conteo de pulsos, los cuales disminuyen si el motor gira en sentido antihorario. Al contar 14400 pulsos (es decir una revolución completa) el contador se reinicia a 0°. Para calcular los grados (G) por pulsos se divide 360° sobre la cantidad de pulsos totales en una revolución como se mencionó en la ecuación 2.2. Por lo tanto, para los 14400 pulsos que provee nuestro codificador se obtiene $P_A = 0.025^\circ$, este valor se toma como una constante que multiplica a los pulsos contados por el codificador (P_c) como se muestra en la ecuación 3.1. Por otro lado, cabe mencionar que se desarrolló un programa a nivel tiempo real para realizar las conversiones de pulsos a grados, debido a que el FPGA se encuentra dedicado a la adquisición y transmisión de señales.

$$G = (P_c)(P_A) \tag{3.1}$$

3.1.3. Controlador

Para el desarrollo del programa que permite un control optimo en la posición angular β se utilizó la paquetería de control que proporciona el software LabVIEW, los parámetros de entrada y salida de esta función son los mismos que en un control clásico PID como se muestra en la Fig. 3.7 a) y Fig. 3.7 b). Para este trabajo de investigación la referencia se obtiene por el número de pulsos requeridos para llegar a la posición angular deseada, mientras que la variable a procesar es la retroalimentación en pulsos que provee el codificador de cuadratura. La diferencia que existe entre la referencia y la variable a procesar da como resultado el error (e) del sistema como se muestra en la ecuación 2.10.

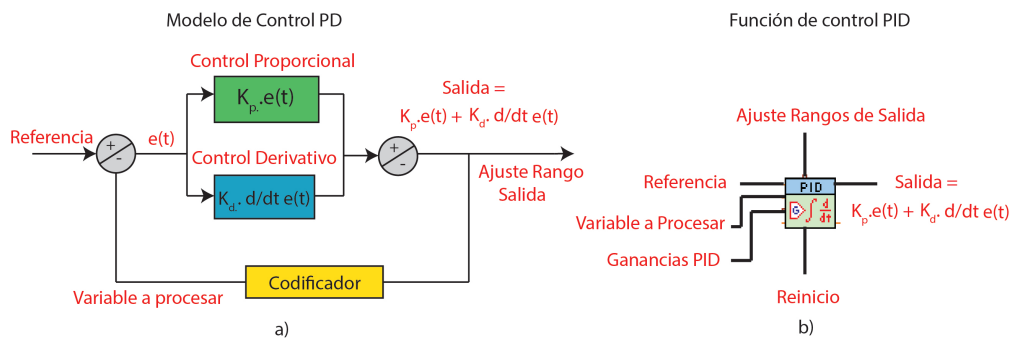


FIGURA 3.7: a) Modelo de un controlador PD b) La función PID permite modelar diferentes tipos de controladores tales como: PD, PI, P, D y I

La función PID calcula e y lo multiplica por la ganancia K_p para obtener el control proporcional, a su vez se calcula la $\frac{d}{dt} \cdot e(t)$ y se multiplica por la ganancia K_d para obtener el control derivativo, mientras que la ganancia K_i se deshabilita colocando un cero debido a que en el controlador PD no requiere la acción integral. La suma o resta de las salidas de ambos controladores generan la salida del controlador PD, sin embargo si las ganancias no son ajustadas adecuadamente se generan sobreimpulsos y bajo amortiguamiento en el sistema. Por lo tanto, para encontrar las ganancias que permitan alcanzar una precisión de $\pm 0.075^\circ$ se realizaron pruebas variando las ganancias del controlador, donde empíricamente se determinó que con $K_p = 10$, $K_i = 0$ y $K_d = 1$ se obtiene la precisión requerida. Por otro lado, los rangos de la salida generada por el controlador PD se calibraron desde la mínima y máxima potencia del motor (desde que comienza a girar hasta la máxima velocidad) para que la señal proveída por el controlador PD garantice el movimiento del motor en su mínima potencia y se evite la saturación, los rangos establecidos en este trabajo es de un rango mínimo de 17500 y un rango máximo de 65535, los cuales se eligen dependiendo de la resolución del ancho de pulso (conocido por sus siglas en inglés como PWM) que serán explicados en la siguiente subsección. Para el motor PITTMAN modelo *GM8712J015* (voltaje nominal = 38.2 v, relación de transmisión = 96:1) con una alimentación de voltaje de 20 v la función PID fue configurada como se muestra en la Fig. 3.8.

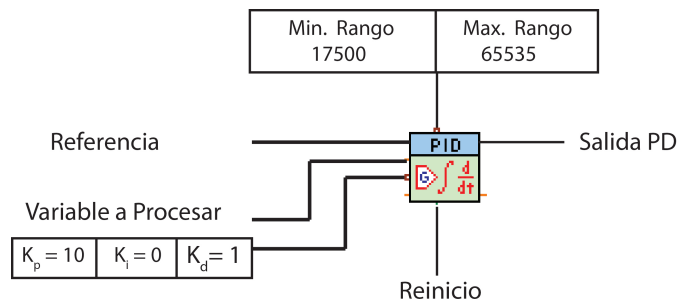


FIGURA 3.8: Referencia = pulsos equivalentes a la posición deseada, variable a procesar = pulsos del codificador, salida PD = $K_p \cdot e(t) + K_d \cdot \frac{d}{dt} \cdot e(t)$

3.1.4. Modulación de Ancho de Pulso (PWM)

Para controlar la aceleración y desaceleración se decidió utilizar ajustes entre tiempo de *encendido/apagado* de las bobinas del motor DC, en función del controlador PD. Por lo tanto se programaron dos PWMs (de 16 bits) a nivel FPGA utilizando dos salidas digitales del módulo *NI 9403*, la máxima resolución de los

PWMs es de $2^{16} = 65535$, por tal razón la salida máxima del controlador PD fue establecida a 65535 y el valor mínimo se estableció a 17500 debido que a valores menores el motor no es capaz de vencer la fricción y el momento que genera el peso del sistema. Si convertimos la mínima y máxima resolución (R_c) de los PWMs a ciclos de trabajo (τ) expresado en porcentaje (P_c), al aplicar la ecuación 3.2 podemos afirmar que nuestro motor comenzará a girar cuando el $26.7\% \leq \tau \leq 100\%$ tal como se muestra en la Fig. 3.9.

$$P_c = \frac{(R_c)(100)}{65535} \quad (3.2)$$

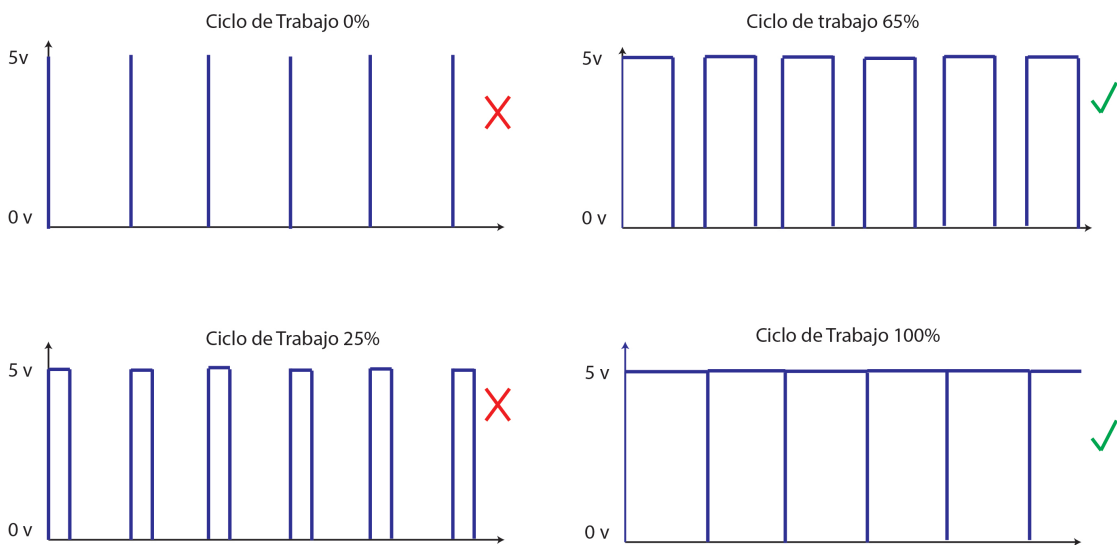


FIGURA 3.9: El motor comienza a girar en $P_c \geq 26.7$, por lo tanto para P_c de 0% y 25% se marcan con una cruz roja debido que no existe giro, sin embargo para 65% y 100% se produce movimiento por tal motivo se marca con una palomita verde.

Como se comentó al inicio de esta subsección se generaron dos salidas del PWM, una es para controlar el giro del motor en *CCW* y la segunda para controlar el giro del motor en *CW*. El control de las salidas de los PWMs se basa en las siguientes tres condiciones:

- Si el motor se encuentra dentro de una histéresis de $-0.075 \leq e \leq 0.075$ se detendrá, es decir el PWM se va a 0 en sus dos canales.
- Si $e \leq -0.075$ el PWM 1 se enciende y hace girar el motor en dirección de *CW*.
- Si el $e \geq 0.075$ el PWM 2 se enciende y hace girar el motor en dirección de *CCW*.

El P_e de ambos canales depende de la magnitud del e , por ejemplo si queremos posicionar nuestro motor a 30° en dirección de CW , el PWM 1 se encenderá para intentar posicionarlo. Supongamos que el motor se pasó a 35° , generando un $e = 5^\circ$. Entonces el PWM 2 se encenderá para girar el motor en dirección de CCW , el motor se detendrá hasta que $-0.075^\circ \leq e \leq 0.075^\circ$. Por último, el tiempo alto de los PWMs se define como la salida del controlador PD, mientras que el tiempo bajo como la diferencia de la salida del controlador PD menos la resolución de 16 bits (65535). En la Fig. 3.10, se muestra el funcionamiento de las señales PWMs sobre el sistema de control.

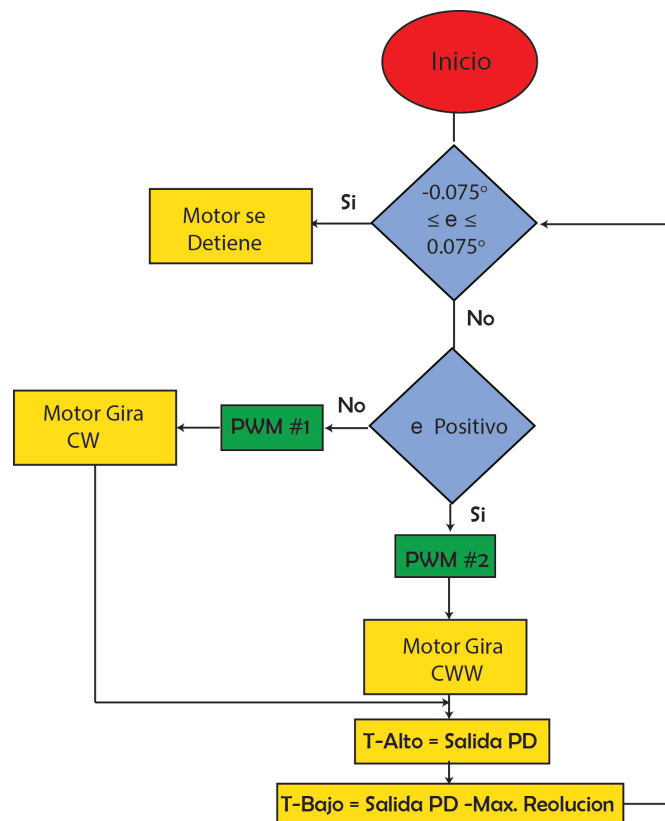


FIGURA 3.10: Metodología de las señales PWM, donde max. resolución = 65535, $0 \leq \text{salida PD} \leq 65535$.

3.1.5. Etapa de Potencia

La potencia producida por el modulo *NI 9403* genera salidas PWMs con amplitud de 5 v, con este voltaje es imposible hacer girar el motor *GM8712J015*, el cual opera de 7 v a 38.2 v. Por lo tanto se utilizó un puente H modelo L298 para incrementar el voltaje a 20 v y cambiar la dirección de giro del motor. El PWM

1 Y PWM 2 generados por el NI 9403 se conectan en las entradas ($IN1$ y $IN2$) del L298 respectivamente. El módulo tiene una entrada para la alimentación de los componentes donde se conecta la fuente fija de 5 v, también tiene la entrada de potencia donde se conecta la fuente variable de 20 v, posteriormente las tierras de las fuente de 5 v y 20 v se juntan para formar la tierra común. Por último, se conectan las salidas ($OUT1$ y $OUT2$) al motor. En la Fig. 3.11 se muestra los pines de alimentación, entradas y salidas del L298.

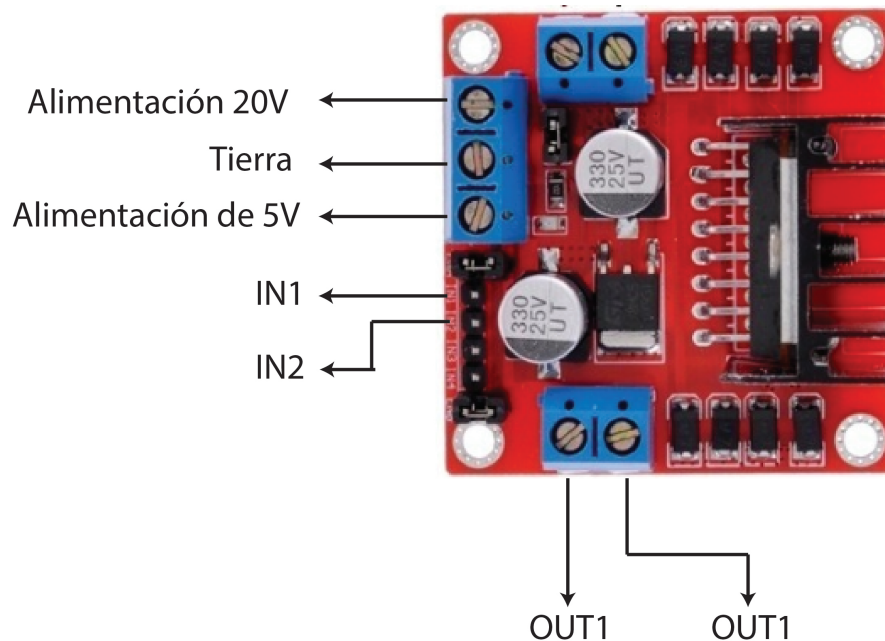


FIGURA 3.11: Entradas y salidas del puente H L298

3.2. Propuesta de Algoritmo

En trabajos previos [Real-Moreno et al. \[2018\]](#) se realizaron pruebas en dos dimensiones con el TVS aplicando el algoritmo KNN para mejorar la precisión del sistema, sin embargo aun se presentan incertidumbres, por lo que en esta investigación se propone la aplicación de un nuevo algoritmo basado en ANN que tiene el objetivo de predecir el error $e(t)$ introducido al TVS para compensar la incertidumbre. Para la implementación de la ANN fue necesario crear una base de datos de los atributos principales del sistema tales como: el ángulo de apertura, ángulo del posicionador láser y los Ticks (pulso del reloj del CompactRIO a 40 MHz). Posteriormente, la ANN fue entrenada con el software Matlab hasta alcanzar un

coeficiente de correlación (R). Posteriormente, se tomaron las matrices de pesos para programar la ANN en el software LabVIEW.

3.2.1. Adquisición de Base de Datos

El principal paso para el entrenamiento del algoritmo ANN es la creación de una base de datos que almacene las características principales del sistema de una manera estructurada y organizada. Para este trabajo de investigación se implementó una base de datos utilizando un grid como se muestra en la Fig. 3.12, donde se marcan cinco líneas paralelas separadas 5 cm cada una, estas líneas se encuentran a una distancia de 15 cm hasta 30 cm con respecto a la distancia ortogonal que existe entre la apertura de escaneo y el posicionador láser. Además, el grid cuenta con trazos angulares en el rango de visión de 50° a 130° con respecto a la apertura de escaneo. Las intersecciones que existen entre las líneas que indican los ángulos y las distancias paralelas son los puntos utilizados para entrenar el algoritmo.

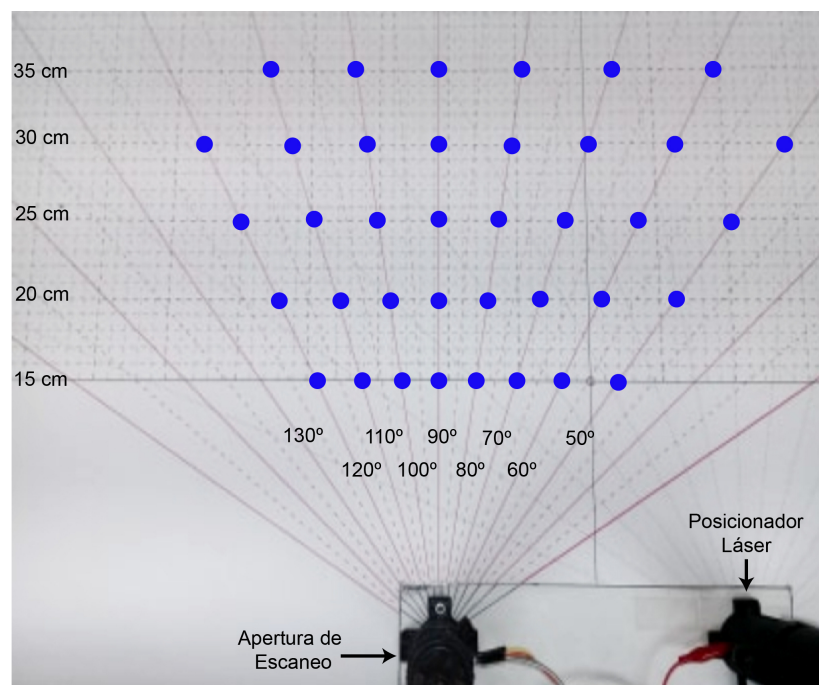


FIGURA 3.12: Los puntos azules indican la intersección entre las líneas paralelas que representan las distancias de 15 cm a 35 cm y las líneas angulares.

Para la creación de la base de datos fue necesario desmontar el TVS bidimensional del eje β y medir las variables: ángulo de apertura de escaneo (B_{ij}), ángulo del posicionador láser (C_{ij}), y el mínimo valor que puede leer nuestro sistema

(*TICKs*). Para obtener las mediciones de las variables se posicionó el haz láser sobre la intersección. Una vez posicionado el láser, se utilizó un programa desarrollado en LabVIEW, donde se miden por 15 segundos las variables B_{ij} , C_{ij} y *TICKs*, al finalizar el tiempo se almacenan en una matriz. Este procedimiento se utiliza para medir todas las intersecciones y se obtiene una matriz de 1000 mediciones de muestras.

3.2.2. Entrenamiento de la Red Neuronal Artificial

Se diseñó una ANN que consiste en las entradas (B_{ij}), (C_{ij}) y (*TICKs*) y una salida llamada compensación (error predicho por la ANN) que se suma o resta al ángulo B_{ij} . Además, la capa oculta está compuesta por siete neuronas, siete bias y una función de activación $Tanh()$ para cada neurona. Por último, la capa de salida está formada por una neurona, una bia y una función de activación $Pureline()$. En la Fig. 3.13 se muestra el diseño de la ANN.

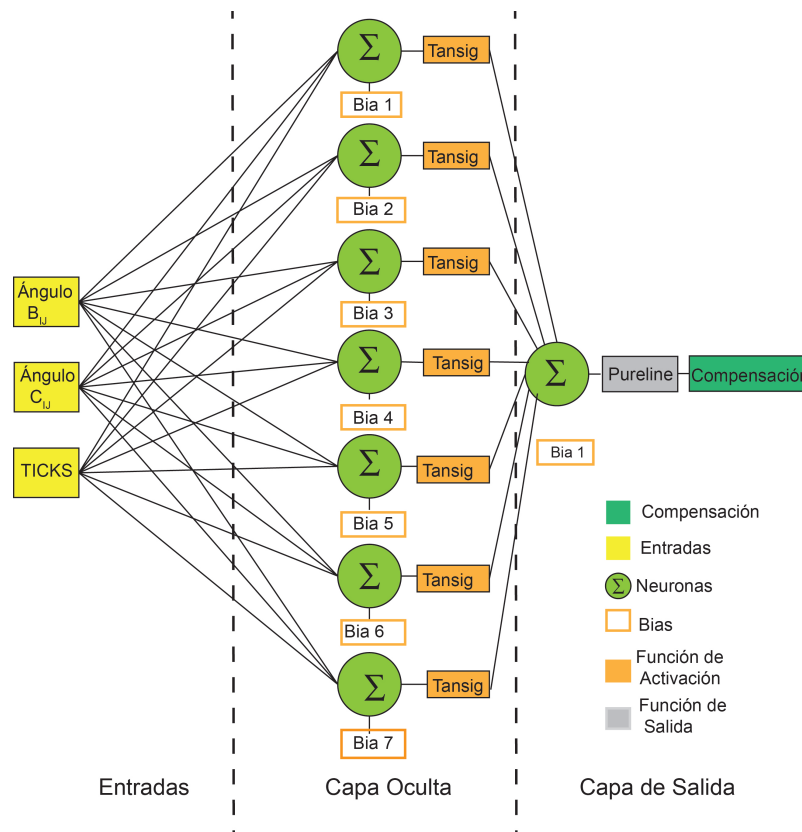


FIGURA 3.13: La ANN desarrollada es configurada por tres entradas, siete neuronas, tres capas y funciones de activación $Tanh()$ y $Pureline()$

Para la configuración y entrenamiento de la ANN se utilizó la paquetería del software Matlab conocida como *nntool*, con la cual se obtienen las matrices de pesos y los vectores bias necesarios para implementar la ANN. El primer paso es importar la matriz de la base de datos al espacio de trabajo de Matlab, posteriormente se selecciona el tipo de red *feedforward backpropagation* y el tipo de entrenamiento basado en el método de *Levenberg–Marquardt*, después se elige el número de capas de la ANN (para este caso 2). En la capa 1 (capa oculta) se asignan 7 neuronas y una función de activación *Tansig* (también conocida como función de activación *TanH*). Por otro lado, para la capa 2 (capa de salida) se asigna una neurona y una función de activación *Pureline*. En la Fig. 3.14 se muestra la interfaz de configuración de la ANN.

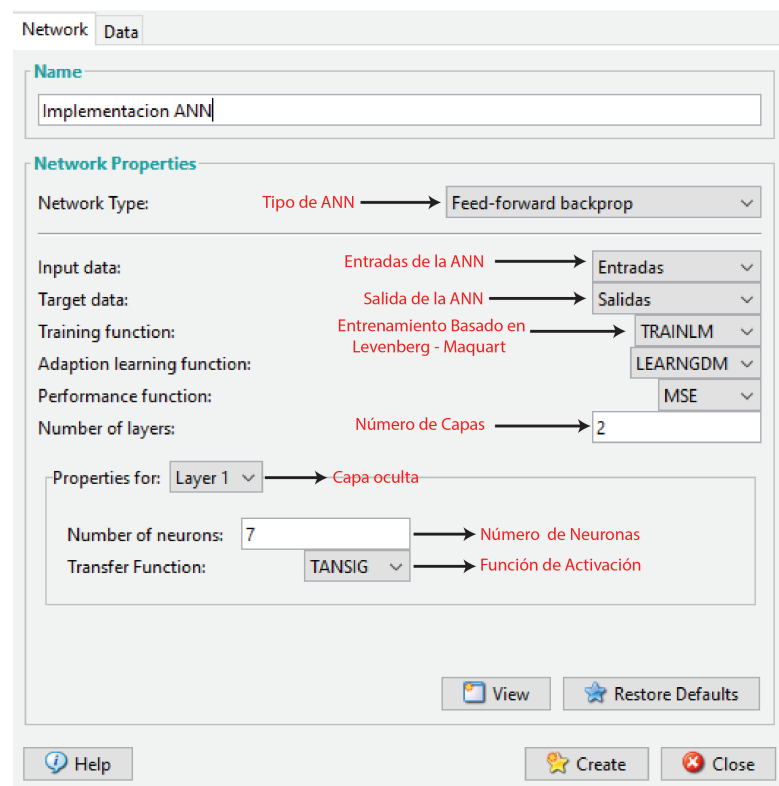


FIGURA 3.14: Interfaz para la configuración de la ANN

Una vez configurada la ANN, se pulsa el botón de crear, y enseguida aparecerá como se muestra en la Fig. 3.15, donde **Input** representa las entradas con las cuales será entrenada nuestra ANN, **Hidden Layer** es la capa oculta que está conformada por los pesos y las bias que se suman para calcular la función de activación *Tansig*. Posteriormente, el resultado pasa a ser el peso de la capa de salida, el cual se suma con la bia para calcular la función de activación *Pureline*

y producir la salida de la ANN que compensará el error en el TVS.

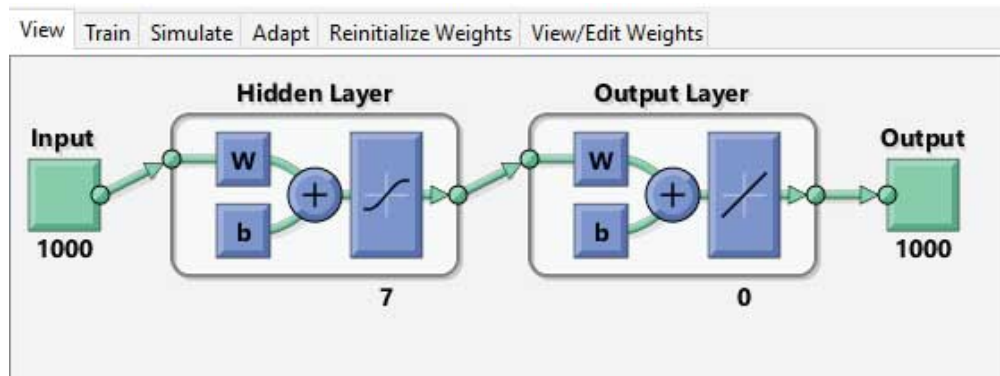


FIGURA 3.15: ANN configurada utilizando en la paquetería de Matlab *nntool*

Para el entrenamiento del algoritmo se configuraron los parámetros como se muestra en la Fig. 3.16, donde *epoch* es el número de épocas (para este caso 1000) en las que se entrenará la ANN siempre que no se haya logrado el *goal* (meta). El parámetro *mu* representa el parámetro de adaptación que es usado en el entrenamiento de *Levenberg – Maquart*. *mu – dec* representa el factor de decremento mientras que *mu – inc* representa el factor de incremento, por último *mu – max* es el valor máximo que puede alcanzar *mu*. Por otro lado, *max – fail* es el número máximo de fallos en el entrenamiento y *min – grad* es el mínimo rendimiento del gradiente.

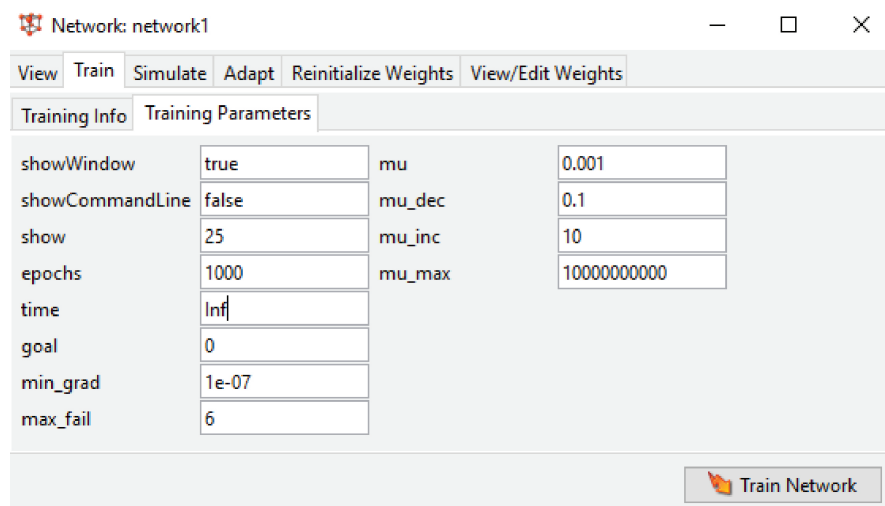


FIGURA 3.16: Configuración de parámetros de entrenamiento

Se entrenó el algoritmo hasta alcanzar un $0.90 \leq R \leq 0.94$ para evitar el sobreentrenamiento es decir, evitar que la ANN aprenda solo en casos particulares con los cuales fue entrenada y no pueda ajustarse a otros casos.

Por otro lado, para evaluar el rendimiento del algoritmo y garantizar que los resultados son independientes de la partición del entrenamiento y prueba se utiliza la función de validación cruzada, la cual permite dividir la base de datos en 10 secciones. En la Fig. 3.17 se muestra las particiones de datos, en donde para cada interacción se asigna una carpeta para la prueba y nueve para el entrenamiento. Una vez divide la base de datos, el algoritmo es entrenado y probado en cada interacción, de esta manera se calcula el error cuadrático medio (conocido por sus siglas en inglés como RMSE). Por último, se promedia el RMSE de cada interacción para obtener el valor validado.

		Kfold									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Interacciones	n=1	P	E	E	E	E	E	E	E	E	E
	n=2	E	P	E	E	E	E	E	E	E	E
	n=3	E	E	P	E	E	E	E	E	E	E
	n=4	E	E	E	P	E	E	E	E	E	E
	n=5	E	E	E	E	P	E	E	E	E	E
	n=6	E	E	E	E	E	P	E	E	E	E
	n=7	E	E	E	E	E	E	P	E	E	E
	n=8	E	E	E	E	E	E	E	P	E	E
	n=9	E	E	E	E	E	E	E	E	P	E
	n=10	E	E	E	E	E	E	E	E	E	P

FIGURA 3.17: n representa el número de interacciones, C_n es el número de carpeta, E son los datos de entrenamiento y P los datos de prueba.

3.2.3. Implementación de la Red Neuronal Artificial

Al finalizar el proceso de entrenamiento y validación de la ANN se toman las matrices de pesos y las bias de cada capa, posteriormente se programa la ANN considerando que el valor de entrada del ángulo B_{ij} oscila de 40° a 120° . Además, el valor de entrada del ángulo C_{ij} varía de 31° a 87° y el valor de la entrada de los $Ticks$ cambia de 7200 a 264479. Por otro lado, la función de activación $TanH$ oscila de (-1,1). Por lo tanto, no se puede calcular la función debido a que los valores de las entradas de la ANN se encuentran fuera de rango de la función $Tanh$, por

lo que es necesario normalizar las entradas a la misma escala que la función de activación *Tanh*. Para normalizar los datos es necesario aplicar la ecuación 3.3.

$$X_t = a_e + \frac{(X_i - m_i)(b_e - a_e)}{M_i - m_i} \quad (3.3)$$

Donde X_t representa los valores normalizados de las entradas de la ANN, X_i son las entradas de la ANN sin normalizar, m_i es el valor mínimo de cada entrada sin normalizar, M_i es el valor máximo de cada entrada sin normalizar. Por último, los valores a_e y b_e son la escala mínima y máxima respectivamente a la que serán normalizadas las entradas.

En la Fig. 3.18 se muestra el programa desarrollado en LabVIEW, donde el vector de entrada es normalizado para ser multiplicado por la matriz de pesos de la capa oculta, posteriormente se suma el vector bias. El resultado obtenido es un vector que pasa por una función de activación *Tanh*, una vez calculada se multiplica por el vector de pesos de la capa de salida y se suma con el vector bias. Por último, es necesario desnormalizar el valor obtenido de la ANN para obtener el error en el rango real. Para la desnormalización se utiliza la ecuación 3.4.

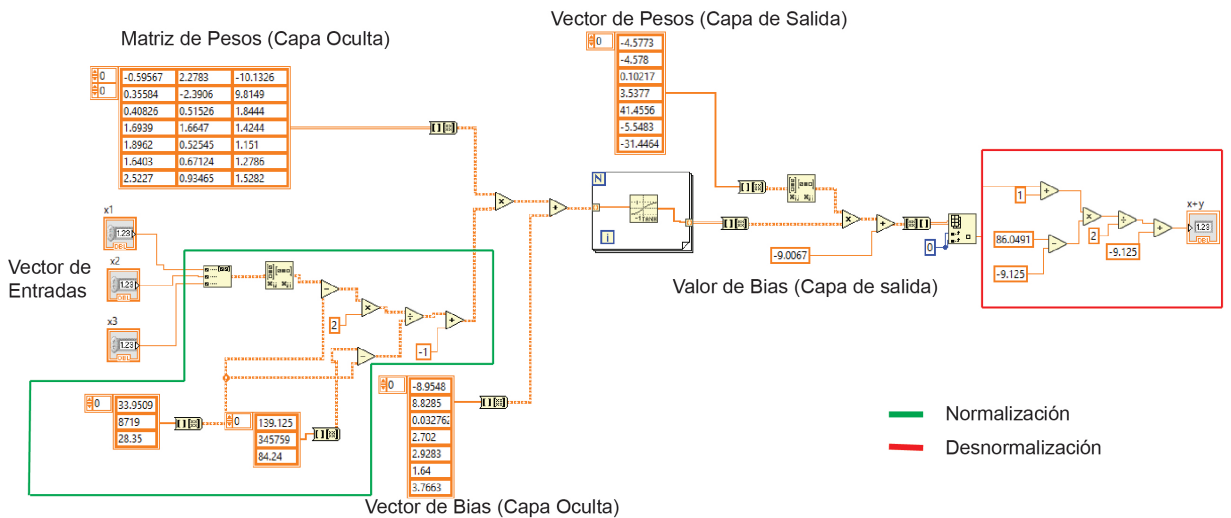


FIGURA 3.18: El recuadro verde indica la sección de normalización, mientras que el recuadro rojo indica la sección de desnormalización

$$X_i = \frac{(X_t - a_e)(M_i - m_i)}{b_e - a_e} + m_i \quad (3.4)$$

3.3. Implementación de Ecuación Z

La implementación del eje de rotación β sobre el Sistema de Visión Técnica (conocido por sus siglas en inglés como TVS) hace posible escanear superficies tridimensionales usando la ecuación 3.5.

$$Z_i = (a) \frac{(\sin C)(\sin B)(\tan \beta)}{\sin(B + C)} \quad (3.5)$$

Donde a representa la distancia fija que existe entre la apertura de escaneo y el posicionador láser, C es el ángulo del posicionador láser, B es el ángulo de la apertura de escaneo y β representa el ángulo del eje de rotación implementado.

La ecuación Z fue programada dentro del SubVI, donde se encuentran programadas las ecuaciones 1.1 y 1.2, las cuales permiten el escaneo de superficies bidimensionales en los ejes X y Y . Para medir superficies tridimensionales fue necesario incluir en el programa la ecuación Z como se muestra en la Fig. 3.19.

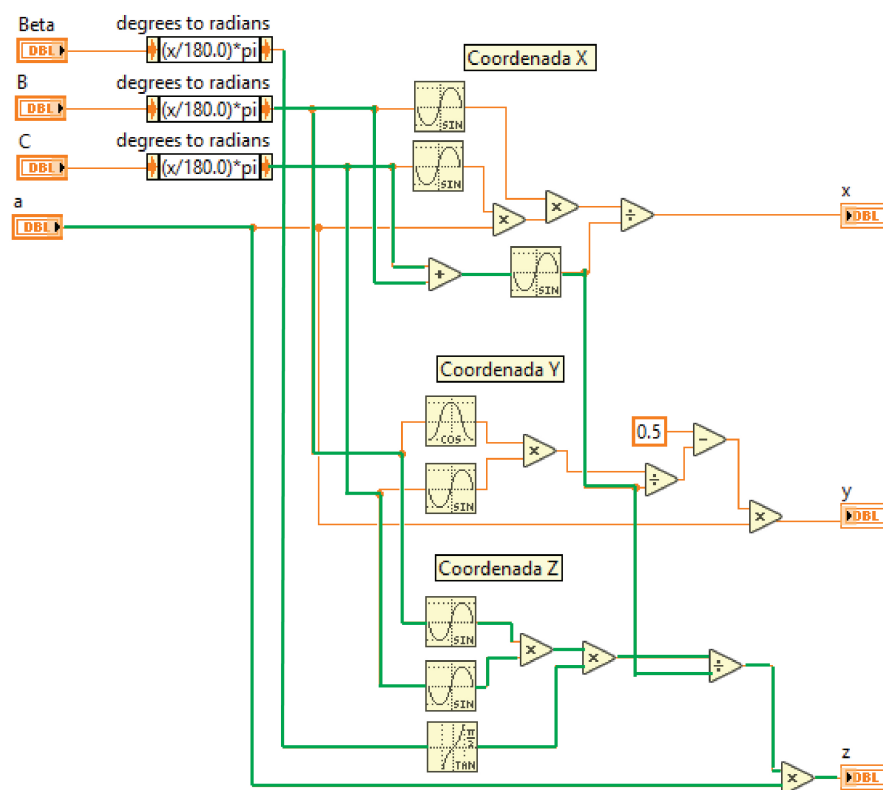


FIGURA 3.19: Las líneas verdes indican las conexiones de la ecuación Z

3.4. Implementación de la Interfaz Tridimensional

Uno de los principales problemas que surge es la implementación de una interfaz gráfica que permita visualizar superficies tridimensionales debido a que el Compact-RIO no tiene acceso a las funciones de entorno gráfico, por lo que se desarrolló un programa a nivel computadora con ayuda de variables compartidas que permiten transmitir información del dispositivo a la computadora, lo cual hace posible graficar las coordenadas X , Y y Z al finalizar el proceso de escaneo de superficies. Para crear el programa fue necesario crear una variable compartida en el programa principal y agregar la sección de código que se muestra en la Fig. 3.20.

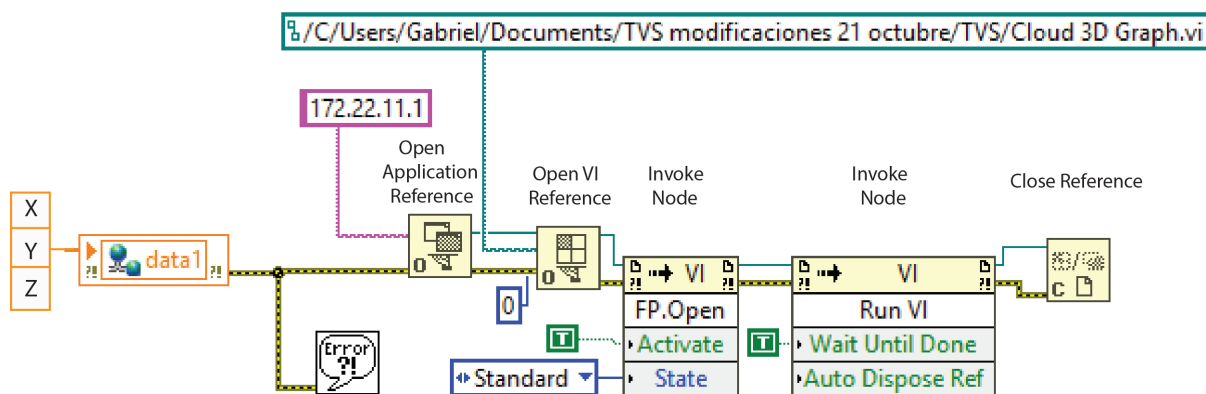


FIGURA 3.20: Sección programada para compartir información del Compact RIO a nivel computadora.

Donde *data1* es la variable compartida que permite transmitir las coordenadas X , Y y Z a la computadora. El bloque *Open Application Reference* crea una referencia para acceder a la información del Compact-RIO utilizando la dirección del dispositivo para este caso 172.22.11.1, posteriormente el bloque *Open VI Reference* permite abrir la referencia del VI indicando la ubicación de la carpeta, después *Invoke Node* abre y ejecuta el programa que se encuentra a nivel computadora. Por último, el bloque *Close Reference* es utilizada para cerrar la referencia asociadas al VI. Por otro lado, el programa de VI que se ejecuta a nivel computadora y se utiliza para graficar las coordenadas tridimensionales se muestra en la Fig. 3.21.

Donde la variable compartida provee las coordenadas X , Y y Z que se están calculando en el Compact-RIO en un solo vector, por lo tanto se separan con las

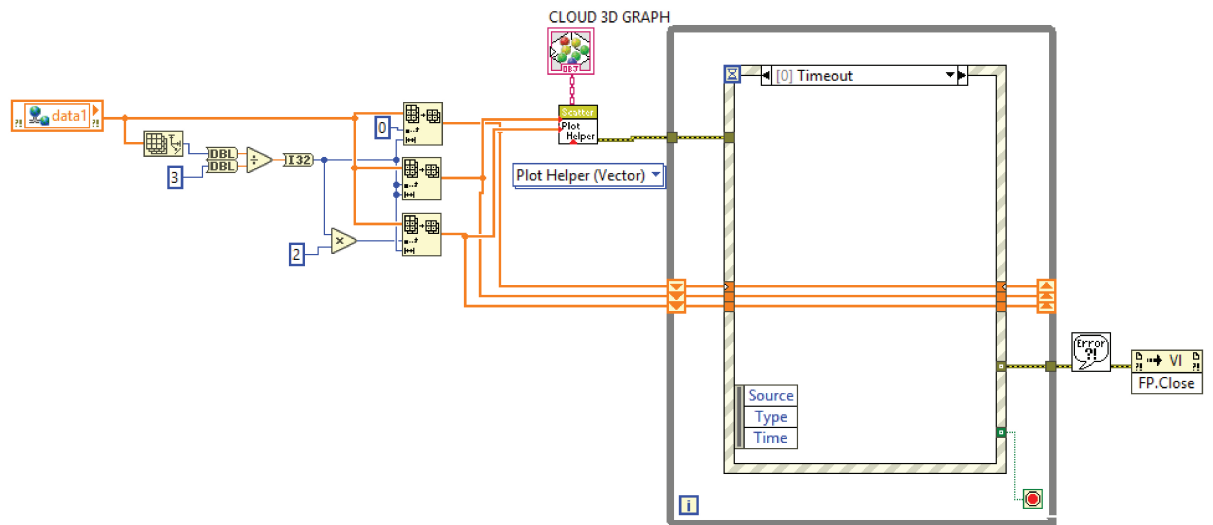


FIGURA 3.21: Programa a nivel computadora para graficar las coordenadas X , Y y Z

funciones *ArraySubset* para enviarse a la función *3D scatter* como tres vectores independientes. Posteriormente se muestran las nubes de puntos bidimensionales o tridimensionales en la interfaz *cloud 3D scatter* que se aprecian en la Fig. 3.22. Por otro lado, dentro del ciclo *while* se encuentra una estructura de caso basada en eventos, el cual tiene la función de guardar la nube de puntos o cerrar sin guardar, una vez finalizado el tiempo de escaneo.

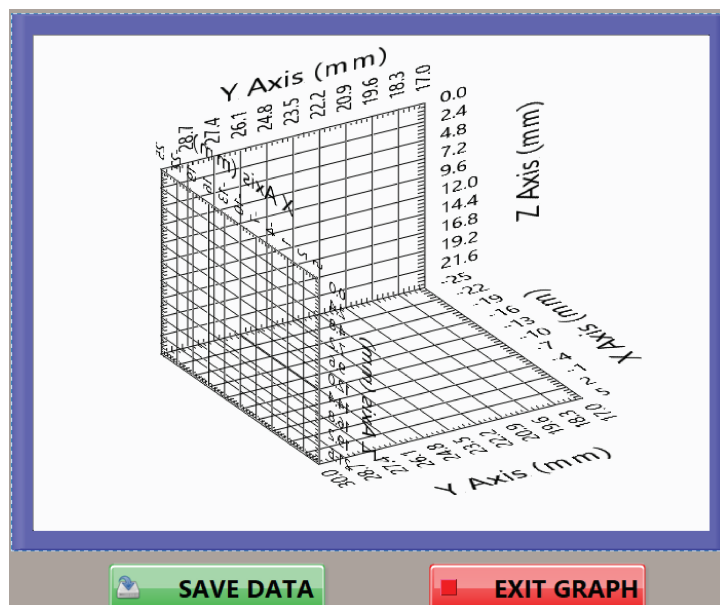


FIGURA 3.22: Interfaz donde se muestran las nubes de puntos bidimensionales y tridimensionales

3.5. Conclusiones

En esta capítulo se describió el procedimiento para realizar este trabajo de investigación, se explica el diseño, fabricación y ensamble del sistema, así como del eje de rotación β . Posteriormente se indicó el procedimiento y materiales necesarios para la fabricación de los circuitos impresos del TVS. Después se describió el método de conteo de pulsos con precisión cuádruple, el cual consiste en leer los flancos altos y bajos del canal A y canal B del codificador de cuadratura. También se habla acerca del diseño y configuración del controlador PD utilizado para posicionar el eje β y como se relaciona con el *PWM* para manipular el ciclo de trabajo de 0% a 100%. A su vez se habla de la etapa de potencia utilizada para aumentar el voltaje del *PWM* a 20 v y permitir el movimiento del motor DC. Por otro lado, se explica el procedimiento detallado para la creación de una base de datos de 1000 muestras utilizada para entrenar una ANN previamente configurada, las matrices de pesos y las bias se toman para desarrollar la ANN en el software LabVIEW. Por último, se desarrolla una interfaz gráfica con el uso de variables compartidas que permite transmitir información del Compact-Rio a la computadora para visualizar nubes de puntos bidimensionales y tridimensionales.

La implementación del eje de rotación β a través del conteo de pulsos generados por el codificador de cuadratura, controlador PD, modulación de ancho de pulso y la etapa de potencia permitió el giro controlado del ángulo β con una precisión de $\pm 0.075^\circ$. Este avance hizo posible el escaneo de superficies tridimensionales. Por otro lado, con la fabricación de los circuitos electrónicos se logró atenuar las incertidumbres ocasionadas por los falsos contactos entre los componentes. Sin embargo, aún se presentaban incertidumbres en el cálculo del ángulo de la apertura de escaneo y la deformación del centro energético del haz láser. Por lo tanto, la aplicación del algoritmo basado en ANN permitió predecir el error y compensarlo. Por último, la implementación de la interfaz gráfica creada en el ambiente de programación LabVIEW permitió visualizar las nubes de puntos tridimensionales generadas en el escaneo de objetos. En el capítulo 4 se explicarán los resultados obtenidos en base al procedimiento realizado en esta sección.

Capítulo 4

Análisis de Resultados

4.1. Análisis de Controlador PD

Para la sintonización del sistema de control PD se realizaron pruebas donde se ajustaron las constantes K_p y K_d hasta atenuar el error angular β dentro de un rango de $-0.075^\circ \leq e \leq 0.075^\circ$. En la Fig. 4.1 se muestra el comportamiento del sistema de control al aplicarse una entrada tipo escalón de 7 unidades (línea roja) utilizando diferentes ganancias.

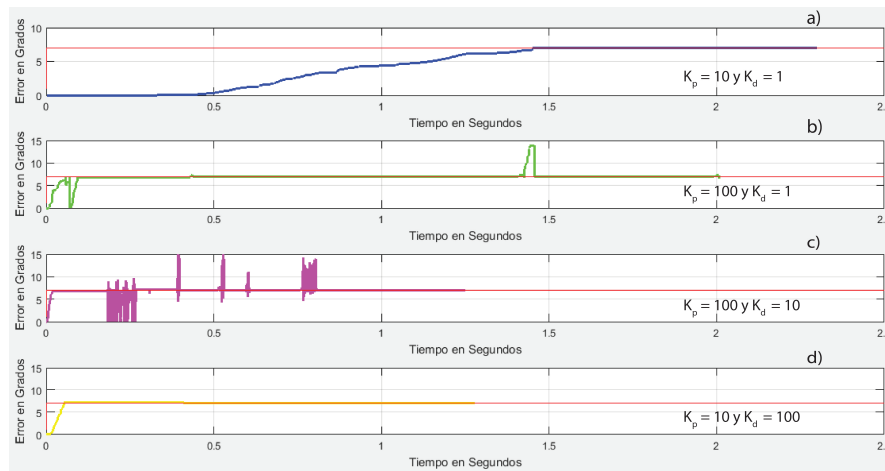


FIGURA 4.1: a) El sistema responde lento pero preciso b) El sistema responde rápido pero oscilaciones en periodos largos c) El sistema responde rápido pero presenta oscilaciones d) El sistema es rápido y no se presentan oscilaciones

Para el Sistema de Visión Técnica (conocido por sus siglas en inglés como TVS) es posible utilizar la opción de la Fig. 4.1 a) y Fig. 4.1 d), debido a que ambos modelos son precisos y aunque el primer modelo es más lento que el segundo, ambos

presentan la velocidad de respuesta necesaria para que el TVS pueda posicionarse en un tiempo aceptable en el ángulo β . Sin embargo, en esta investigación se utilizó el modelo de la Fig. 4.1 a) ya que el modelo de la Fig. 4.1 d) presenta un frenado rápido y por tratarse de un mecanismo fabricado con material PLA se genera mayor desgaste sobre el cople que sujeta el rotor del motor causando oscilaciones en el sistema. Una vez, implementado el controlador PD se posicionó el eje de rotación β cada 3° para analizar la precisión que presenta dentro de un campo de visión (conocido por sus siglas en inglés como FOV) de -15° a 15° . Cabe señalar que el eje de rotación β puede abarcar un rango más amplio de FOV si se sustituye el motor con caja reductora por uno que genere más potencia y se cambia el cople que une la flecha del engrane y el motor ya que son fabricados con material PLA y tiende a barrerse. En la Tabla 4.1 se muestra el valor real y el valor obtenido utilizando el controlador.

Comparación del Ángulo β											
Valores Reales	-15.00°	-12.00°	-9.00°	-6.00°	-3.00°	0.00°	3.00°	6.00°	9.00°	12.00°	15.00°
Valores Controlados	-15.00°	-12.07°	-8.97°	-6.10°	-3.05°	0.05°	3.02°	6.05°	9.10°	12.00°	15.05°

TABLA 4.1: Comparación del ángulo β entre el valor real y valor controlado.

Se realizó una gráfica de correlación utilizando la función `corrcoef()` del software Matlab para analizar la linealidad del valor real con el valor controlado. En la Fig. 4.2 se muestra que los valores controlados se adaptan adecuadamente a los valores reales logrando un coeficiente de correlación (R) de 0.99

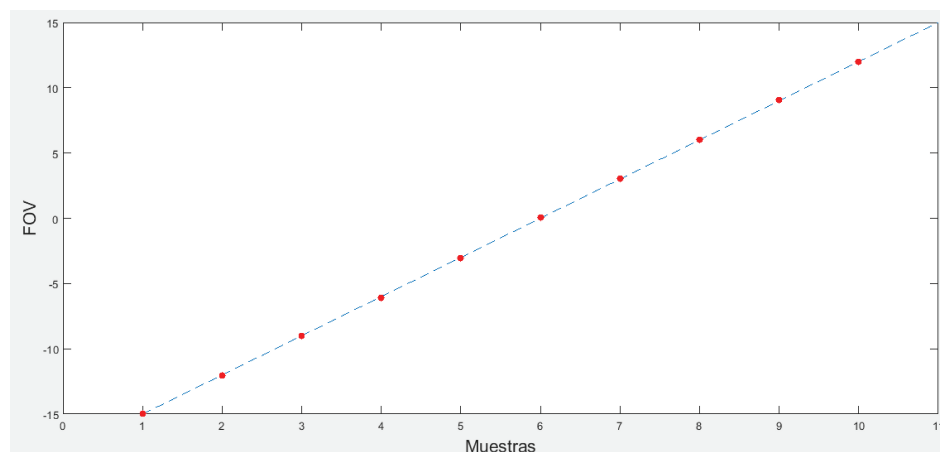


FIGURA 4.2: La línea azul representa la linealidad de los valores reales, mientras que los puntos rojos indican el valor controlado.

Por último, se calculó el error cuadrático medio (conocido por sus siglas en inglés como RMSE), donde el valor obtenido es de 0.0578° . Con este resultado se puede concluir que se presenta un error relativamente bajo entre el valor real y el valor controlado.

4.2. Análisis del Entrenamiento de Redes Neuronales

En previas investigaciones [Real-Moreno et al. \[2018\]](#), el TVS utiliza un algoritmo de vecinos cercanos (conocido por sus siglas en inglés como KNN) que predice el error y lo introduce en el sistema para compensarlo. Sin embargo, el algoritmo presenta un error cuadrático medio (conocido por sus siglas en inglés como RMSE) de 0.5638° , además de ser lento cuando procesa la información debido a que se necesita cargar una base de datos. Por lo que, en esta investigación se intenta encontrar una función rápida y adaptable a los eventos exteriores que mejore el *RMSE*. Por lo tanto, se entrenó un algoritmo basado en ANN hasta alcanzar un coeficiente de correlación (R) entre 0.90 y 0.94. Los resultados del R obtenidos al entrenar el algoritmo en la paquetería del software *Matlab* se muestran en la Fig. 4.3.

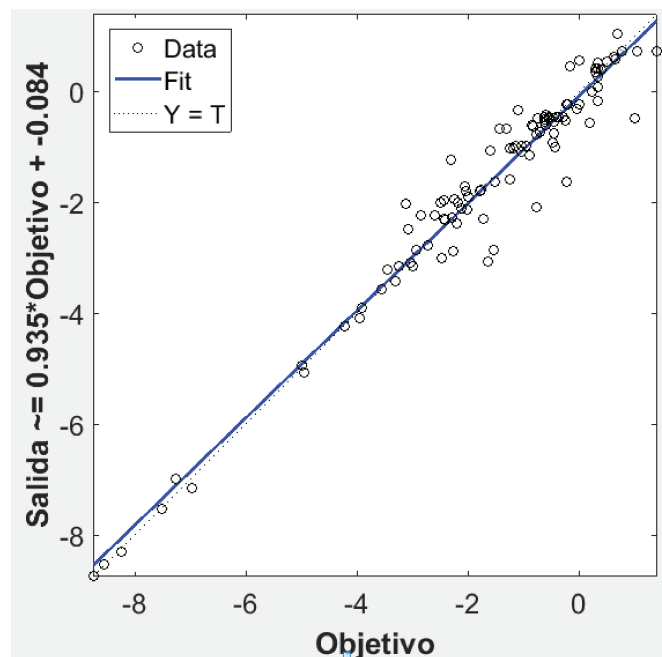


FIGURA 4.3: El valor obtenido de $R = 0.932687$ nos indica que el valor predicho por la ANN se adapta adecuadamente al valor real

Para el entrenamiento y validación del algoritmo se utilizó la función de validación cruzada, la cual permite dividir la base de datos en diez carpetas (las carpetas pueden variar) para entrenar y probar los algoritmos diez veces con diferentes carpetas de datos. Por ejemplo, en la primera interacción se utilizó la primera carpeta de datos para probar el algoritmo y a partir de los datos de la segunda carpeta se entrenó el algoritmo. En la segunda interacción se utilizó los datos de la segunda carpeta para probar el algoritmo y se entrenó con todas las carpetas a excepción de la carpeta dos, y así sucesivamente hasta probar y entrenar los algoritmos con todas las carpetas de la base de datos. Cabe mencionar que para cada interacción del algoritmo KNN y el ANN se calculó el RMSE, los resultados se muestran en la Tabla 4.2 y Tabla 4.3 respectivamente.

KNN ($RMSE = 0.5638^\circ$)

Interacciones	RMSE	Interacciones	RMSE
1	0.6362 ^o	6	0.7489 ^o
2	0.5205 ^o	7	0.4998 ^o
3	0.5313 ^o	8	0.4489 ^o
4	0.5656 ^o	9	0.6074 ^o
5	0.5086 ^o	10	0.5708 ^o

TABLA 4.2: RMSE calculado utilizando el algoritmo KNN.

ANN ($RMSE=0.3686^\circ$)

Interacciones	RMSE	Interacciones	RMSE
1	0.4002 ^o	6	0.2925 ^o
2	0.3293 ^o	7	0.3085 ^o
3	0.2790 ^o	8	0.4028 ^o
4	0.3369 ^o	9	0.3483 ^o
5	0.6614 ^o	10	0.3267 ^o

TABLA 4.3: RMSE calculado utilizando el algoritmo ANN.

Al promediar el valor RMSE de las diez interacciones para el algoritmo KNN se obtuvo un $RMSE = 0.5638^\circ$, mientras que para el algoritmo ANN se logró un $RMSE = 0.3686^\circ$. Por lo tanto, podemos afirmar que la aplicación del algoritmo basado en ANN predice el error con mayor precisión que el algoritmo KNN, además es más rápido porque no depende de una función para la predicción.

4.3. Análisis de Redes Neuronales

Se tomaron 100 mediciones sobre el grid en el plano X y Y utilizando el programa basado en KNN (método anterior) implementado en el TVS. La Fig. 4.4 muestra el comportamiento del valor predicho con respecto al valor real.

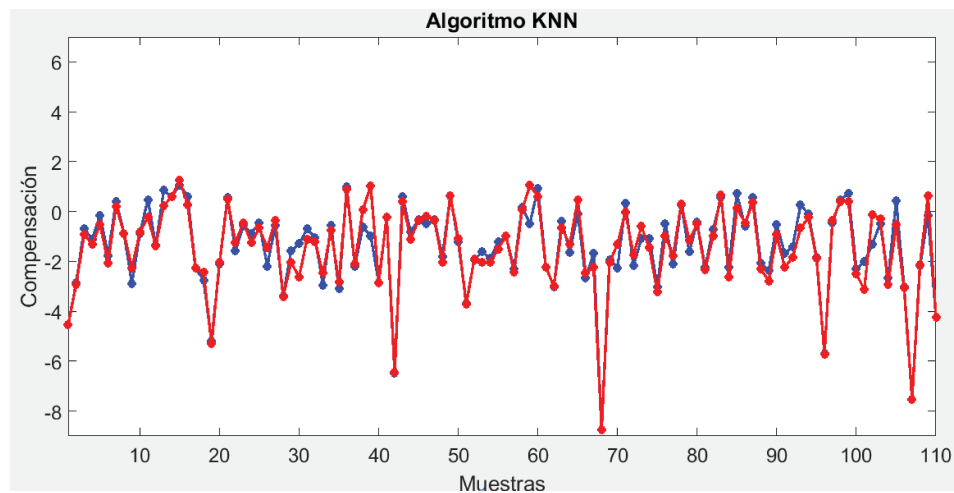


FIGURA 4.4: Comparación entre el valor real y el valor predicho usando el algoritmo KNN

Por otro lado, se tomaron las mismas mediciones con el TVS utilizando un algoritmo ANN (método desarrollado). La Fig. 4.5 muestra el comportamiento del valor predicho y el valor real.

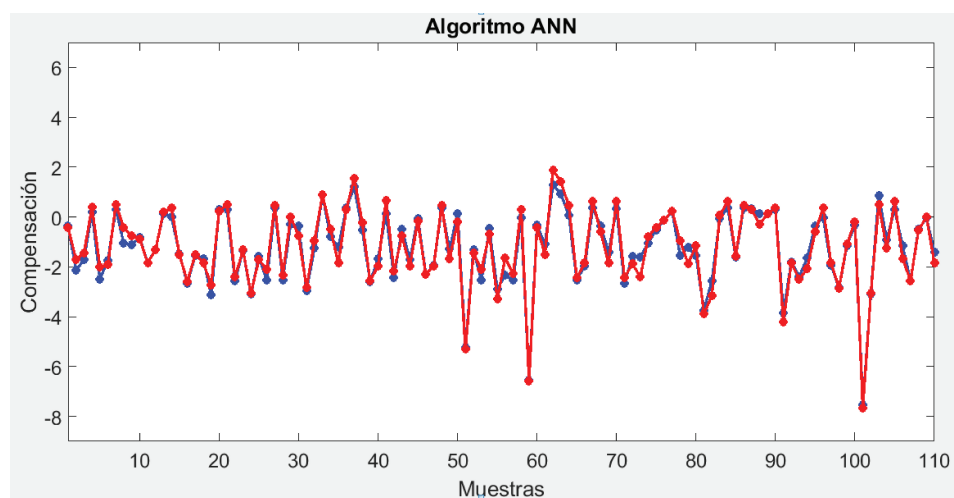


FIGURA 4.5: Comparación entre el valor real y el valor predicho usando el algoritmo ANN.

Al comparar las gráficas 4.4 y 4.5 se muestra que la aplicación del método desarrollado predice mejor el error del sistema, debido a que el valor predicho tiene un mejor compartimento con respecto al valor real.

4.3.1. Análisis y Comparación de Nube de Puntos Bidimensionales

Se tomaron mediciones del barrido láser aplicando el método anterior sobre superficies bidimensionales a distancias de 15 cm, 20 cm, 25 cm y 30 cm en el eje Y, mientras que en el eje X de -5 cm hasta 25 cm. La Fig. 4.6 a) muestra las nubes de puntos obtenidas en cada distancia aplicando el método anterior. Por otro lado, se volvieron a tomar las mismas mediciones pero esta vez utilizando el método desarrollado, los resultados de las nubes de puntos de cada distancia se muestran en la Fig. 4.6 b).

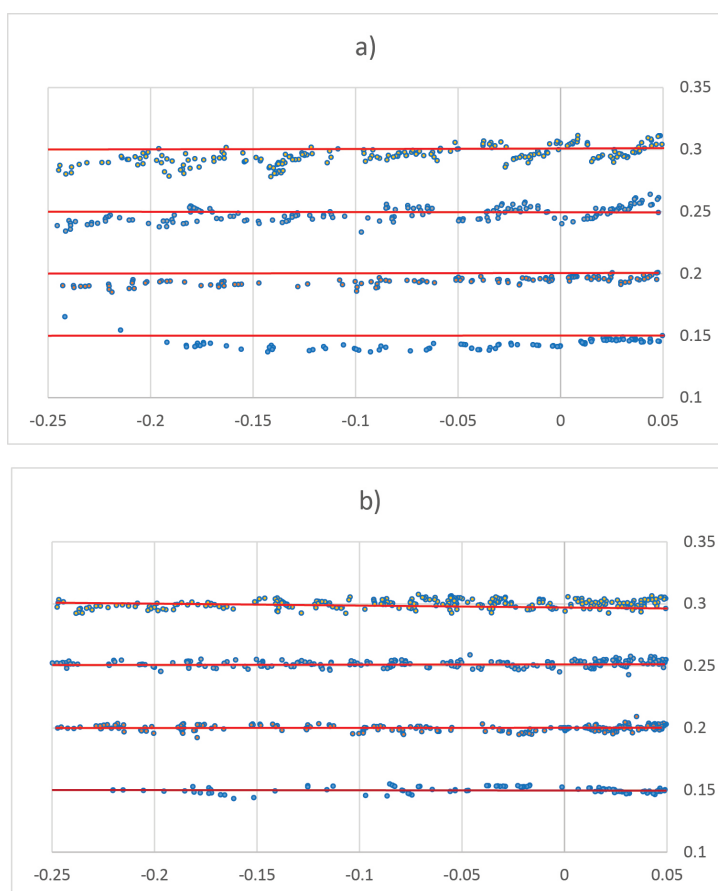


FIGURA 4.6: Nubes de puntos: a) nubes de puntos aplicando el método anterior
b) nubes de puntos utilizando el método desarrollado.

Al comparar las gráficas 4.6 a) y 4.6 b) se observa que al aplicar el método desarrollado se presenta una menor dispersión y desfase de los puntos con respecto a cada distancia. Para comprobar esta afirmación, se calculó la desviación estándar de las nubes de puntos del método anterior y el método desarrollado. La Tabla 4.4 muestra la desviación estándar con respecto a cada distancia y es posible observar

que los resultados obtenidos a 20 cm, 25 cm y 30 cm son similares. Sin embargo, a 15 cm es más alta que las otras distancias escaneadas.

Distancia	Cantidad de Mediciones	Desviación Estandar
15 cm	101	0.0080
20 cm	125	0.0066
25 cm	173	0.0061
30 cm	204	0.0069

TABLA 4.4: Mediciones de barrido láser usando el algoritmo KNN.

Por otro lado, la Tabla 4.5 muestra la desviación estándar de las nubes de puntos utilizando el método desarrollado, donde se puede apreciar en los resultados una mejor aproximación y disminución de dispersión de los puntos con respecto a cada distancia.

Distancia	Cantidad de Mediciones	Desviación Estándar
15 cm	103	0.0034
20 cm	337	0.0055
25 cm	498	0.0056
30 cm	447	0.0053

TABLA 4.5: Mediciones de barrido láser usando el algoritmo ANN.

Por último, en la Tabla 4.6 se pueden observar los porcentajes de mejora de la desviación estándar del método anterior con respecto al método desarrollado sobre las distancias 15 cm, 20 cm, 25 cm y 30 cm.

Distancia	Método Anterior	Método Desarrollado	Porcentaje de Mejora %
15 cm	0.0080	0.0034	57.5 %
20 cm	0.0066	0.0055	16.66 %
25 cm	0.0061	0.0056	8.19 %
30 cm	0.0069	0.0053	23.18 %

TABLA 4.6: Porcentaje de mejora en la desviación estándar del algoritmo ANN con respecto al KNN.

4.4. Análisis de Nube de Puntos Tridimensionales

Para el escaneo de superficies tridimensionales se utilizó un objeto fabricado en la impresora 3D tal como el que se muestra en la Fig. 4.7.

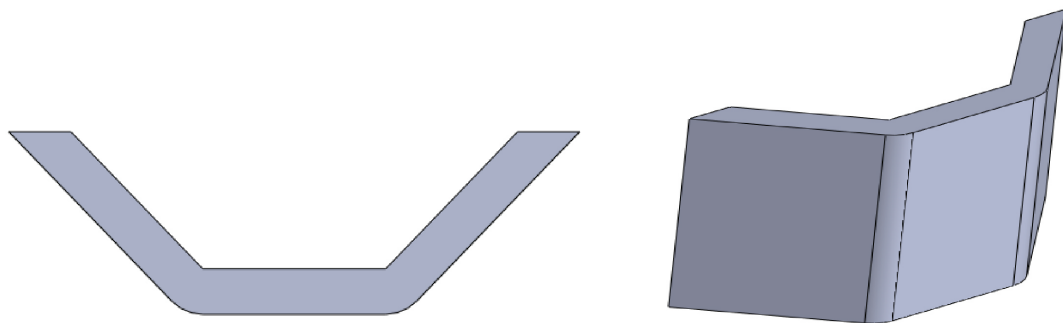


FIGURA 4.7: Objeto tridimensional conocido como trapecoide.

El objeto con superficies tridimensionales fue escaneado aplicando el método anterior utilizando una velocidad en el movimiento del haz de láser de 5 pasos/segundo, una frecuencia en la apertura de escaneo de 20 Hz y un avance de 2 pasos cada barrido en el ángulo β . La nube de puntos tridimensionales obtenida se muestra en la vista frontal e isométrica respectivamente en la Fig. 4.8 a) y Fig. 4.8 b), donde se indica en círculos rojos la deformación ocasionada debido a la baja precisión con la que se predice el ángulo de apertura B_{ij} y la deformación del centro energético del haz de láser que se produce a causa del efecto de triangulación.

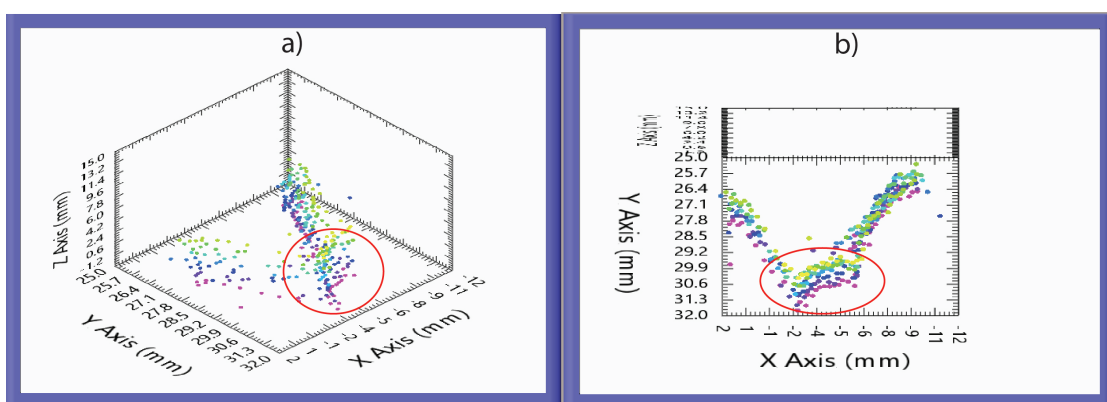


FIGURA 4.8: Nube de puntos obtenida al escanear la pieza trapecoide aplicando el método anterior.

Se realizó una segunda medición utilizando los mismos parámetros de velocidad y avance pero esta vez aplicando el método desarrollado. En la Fig. 4.9 a) y Fig. 4.9 b) se muestra la vista frontal e isométrica respectivamente, donde se puede apreciar en los círculos rojos una mejora en la medición de la pieza debido al incremento en la precisión con la que se predice el error del ángulo B_{ij} , sin embargo el efecto de triangulación aun causa ligeras deformaciones.

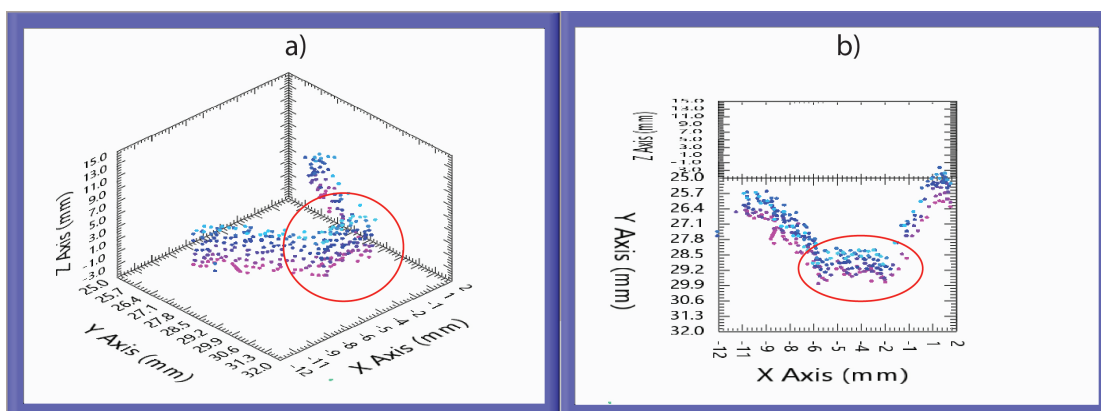


FIGURA 4.9: Nube de puntos obtenida al escanear la pieza trapecoide aplicando el método desarrollado.

Por último, aplicando el método desarrollado se tomaron mediciones tridimensionales de un prototipo de una mano humano que se fabricó en la impresora 3D utilizando una velocidad en el movimiento del haz de láser de 0.78 pasos cada segundo, una frecuencia de escaneo de 20 Hz y un avance en el ángulo β de 0.5° . En la Fig. 4.10 a) y Fig. 4.10 b) se muestra la vista frontal e isométrica respectivamente, de la nube de puntos del prototipo de la mano humana.

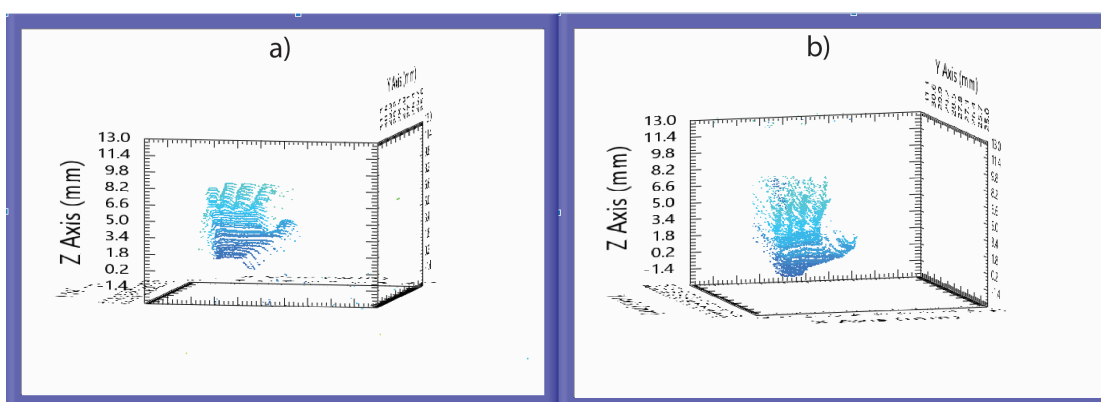


FIGURA 4.10: Nube de puntos del prototipo de una mano humana.

Los resultados de la nube de puntos del prototipo de la mano humana demuestra que la aplicación del método desarrollado permitió caracterizar algunos parámetros biométricos tales como: el palmar, dorsal y los dedos. Sin embargo aún es

necesario encontrar un algoritmo que elimine el ruido que se genera alrededor. Por ejemplo, en el modelo de la mano humano se presentó ruido en la parte de los dedos debido a la deformación del centro energético del haz de láser a causa del efecto de triangulación.

Capítulo 5

Conclusión

En este trabajo se presentó el diseño y el control del eje de rotación β sobre el TVS, esta modificación permitió la medición de superficies tridimensionales utilizando el principio de triangulación dinámica. Además, para mejorar la precisión se analizó e implementó un algoritmo rápido y eficiente basado en redes neuronales artificiales (conocido por sus siglas en inglés como ANN), el cual permite atenuar las incertidumbres en el escaneo de piezas en dos y tres dimensiones. Para lograr lo anteriormente mencionado fue necesario realizar lo siguiente:

a) Se diseñó y fabricó el mecanismo de rotación del eje β utilizando la impresora 3D con material PLA para montar el TVS bidimensional. En la experimentación se observó un funcionamiento adecuado en el mecanismo de giro. Los baleros que se incorporaron en cada barra redujeron la fricción de los ejes, sin embargo en el tren de engranes se sigue presentando desgaste debido a que son fabricados con PLA.

b) Se implementó un programa a nivel FPGA basado en el método de precisión cuádruple para contar los flancos de subida y bajada del canal A y el canal B, lo cual permitió incrementar la resolución a 0.025° cada pulso. Para controlar la posición se desarrolló un programa a nivel tiempo real de un controlador PD, que a su vez controla un PWM para aumentar y disminuir la velocidad de giro del motor. En base a la metodología utilizada para el control del posicionamiento del eje de rotación β se puede concluir que se logró atenuar el error entre -0.075° y 0.075° dentro de un campo de visión (conocido por sus siglas en inglés como FOV) de -15° a 15° . Sin embargo, es posible ampliar el rango del FOV si se sustituye el motor con caja reductora por uno de mayor potencia.

c) El TVS no era capaz de medir la coordenada Z porque no se tenía el valor del ángulo β , con la metodología empleada para obtener el ángulo β y la programación de la ecuación Z fue posible que el nuevo prototipo realizará mediciones tridimensionales.

d) Se entrenó un algoritmo basado en redes neuronales artificiales (ANN) hasta alcanzar un coeficiente de correlación de 0.932687, por medio de la paquetería del software Matlab. Una vez entrenada y obtenido las matrices de pesos de la ANN, se desarrolló un programa en tiempo real en el software LabVIEW. Durante la programación se tuvieron problemas con las variables de entrada (ángulo B_{ij} , ángulo C_{ij} y *TICKS*), porque no se utilizaba un proceso de preprocesamiento (normalización y denormalización) ocasionando que la ANN predijera valores fuera de rango, debido a que la función de activación de la capa oculta *tanh* que opera entre los valores de -1 a 1. Al aplicar el preprocesamiento se predijo el error del sistema dentro de los rangos.

e) Se realizaron 100 mediciones en dos dimensiones dentro del campo de operación del TVS aplicando el algoritmo KNN (método anterior) y el algoritmo ANN (método desarrollado), donde se pudo observar un error cuadrático medio (RMSE) de 0.5638° y 0.3686° respectivamente. La aplicación del método desarrollado permite que el sistema funcione más rápido debido a que se basa en una función previamente entrenada para predecir el error, a diferencia del método anterior que requiere cargar una base de datos para predecir el error más cercano, causando que el sistema sea más lento. Por otro lado, se realizaron barridos láser a 15 cm, 20 cm, 25 cm y 30 cm aplicando el método anterior y método desarrollado. En la Fig. 4.6 a) y Fig. 4.6 b) se muestran las nubes de puntos bidimensionales respectivamente. Además, se puede concluir que la aplicación del método desarrollado disminuyó la dispersión de los puntos con respecto a cada distancia, también se puede notar que conforme comienza a aumentar la distancia de escaneo se presenta una mayor dispersión y menor precisión.

f) Se realizaron mediciones tridimensionales sobre una pieza conocida como trapecoide aplicando el método anterior y método desarrollado como se muestra en la 4.8 y el método 4.9 respectivamente. En base a las nubes de puntos podemos observar que al aplicar el método desarrollado se permitió una mejor caracterización de la superficie. Además, por tratarse de una función previamente entrenada el escaneo fue más rápido, 8 pasos cada segundo a diferencia que el método anterior que se tenía que escanear a una velocidad de 2 pasos cada segundo.

g) Se realizó el escaneo de un prototipo de mano humana, donde se puede apreciar la caracterización de su superficie interna, además es posible notar las superficies de los dedos y espacios entre ellos.

h) Se fabricaron dos circuitos electrónicos (TVS bidimensional y eje de rotación β) para sustituir la placa de prototipos existente, esto permitió la eliminación de falsos contactos en el sistema electrónico y garantiza un escaneo sin paros a causa de las desconexiones.

Bibliografía

- Acosta, D. A. A. and OVIEDO, J. E. S. (2016). *Ingeniería inversa, desarrollo tecnológico y construcción del conjunto corona-eje para un reductor de tornillo sin fin*. PhD thesis, Universidad Tecnológica de Pereira. Facultad de Tecnologías. Tecnología Mecánica.
- Álvarez, C. A. and Giraldo, J. A. (2006). Diseño de una gama de reductores de engranes helicoidales y ejes paralelos. *Scientia et Technica*, 12(32):231–236.
- Auerswald, M. M., von Freyberg, A., and Fischer, A. (2019). Laser line triangulation for fast 3d measurements on large gears. *The International Journal of Advanced Manufacturing Technology*, 100(9-12):2423–2433.
- Blais, F. et al. (2004). Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1):231–243.
- Boden, M. A. (2017). *Inteligencia artificial*. Turner.
- Bolton, W. and Ramírez, F. J. R. (2001). *Ingeniería de control*. Alfaomega Mexico.
- Bustos Gómez, F. A., Pantevis Perilla, H. N., et al. (2016). Implementación de un sistema de reconocimiento de voz en fpga como interfaz hombre-máquina en aplicaciones de robótica.
- Cambroner, C. G. and Moreno, I. G. (2006). Algoritmos de aprendizaje: knn & kmeans. *Inteligencia en Redes de Comunicación, Universidad Carlos III de Madrid*, 23.
- Chua, S., Wang, X., Guo, N., Tan, C., Chai, T., and Seet, G. L. (2016). Improving three-dimensional (3d) range gated reconstruction through time-of-flight (tof) imaging analysis. *Journal of the European Optical Society-Rapid publications*, 11.

- Concli, F. and Gorla, C. (2012). Computational and experimental analysis of the churning power losses in an industrial planetary speed reducer. In *9th International Conference on Advances in Fluid Mechanics-Advances in Fluid Mechanics IX, WIT Transactions on Engineering Sciences*, volume 74, pages 287–298.
- Culler, E. C. and Farnsworth, J. A. (2019). Higher frequencies in stall flutter moment development. *Journal of Fluids and Structures*, 85:181–198.
- Danev, R., Tegunov, D., and Baumeister, W. (2017). Using the volta phase plate with defocus for cryo-em single particle analysis. *Elife*, 6:e23006.
- Feng, S., Zuo, C., Tao, T., Hu, Y., Zhang, M., Chen, Q., and Gu, G. (2018). Robust dynamic 3-d measurements with motion-compensated phase-shifting profilometry. *Optics and Lasers in Engineering*, 103:127–138.
- Flores-Fuentes, W., Rivas-Lopez, M., Sergiyenko, O., Rodríguez-Quñonez, J. C., Hernández-Balbuena, D., and Rivera-Castillo, J. (2014). Energy center detection in light scanning sensors for structural health monitoring accuracy enhancement. *IEEE Sensors Journal*, 14(7):2355–2361.
- Garay, V. G. (2009). Acciones basicas de control.
- Haugeland, J. (1985). Artificial intelligence: the very idea.
- Herrera Gil, L. et al. (2019). Diseño y desarrollo de una férula de miembro superior escaneada e impresa en 3d.
- Hirschhorn, J. N., Brown, S. A., Clark, C. D., and Winston, F. (1992). Evidence that snf2/swi2 and snf5 activate transcription in yeast by altering chromatin structure. *Genes & development*, 6(12a):2288–2298.
- Kim, K. S. and Sentis, L. (2017). Human body part multicontact recognition and detection methodology. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1908–1915. IEEE.
- Li, P., Hu, Y., and Chen, T. (2012). Error compensation for 3d laser scanning system based on neural network. In *2012 24th Chinese Control and Decision Conference (CCDC)*, pages 3875–3878. IEEE.

- Lindner, L., Sergiyenko, O., Rodríguez-Quñonez, J. C., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., Murrieta-Rico, F. N., and Tyr-sa, V. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*.
- Lindner, L., Sergiyenko, O., Rodríguez-Quñonez, J. C., Tyr-sa, V., Mercorelli, P., Fuentes, W. F., Murrieta-Rico, F. N., and Nieto-Hipolito, J. I. (2015). Continuous 3d scanning mode using servomotors instead of stepping motors in dynamic laser triangulation. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pages 944–949. IEEE.
- Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging experience for large-scale lidar localisation in changing cities. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1684–1691. IEEE.
- Mano, M. M. (2003). *Diseño digital*. Pearson Educación.
- Mattson, M. (2009). *CNC programming: principles and applications*. Cengage Learning.
- Minsky, M. L. (1968). Matter, minds, models.
- Neyra Dueñas, E. J. (2018). Análisis comparativo de métodos para el diseño de engranajes cilíndricos de dientes rectos.
- Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación.
- Osio, J. R. (2019). *Procesamiento digital de imágenes médicas sobre plataformas FPGAs*. PhD thesis, Universidad Nacional de La Plata.
- Pajares, G. and Santos, M. (2006). Inteligencia artificial e ingeniería del conocimiento.
- Park, S., Park, H. S., Kim, J., and Adeli, H. (2015). 3d displacement measurement model for health monitoring of structures using a motion capture system. *Measurement*, 59:352–362.
- Pérez, M. A., Baltazar, V. B., Ramírez, J. P., and Valle, F. M. (2014). Diseño de controladores p, pi y pid para el comportamiento dinámico de un servo-sistema hidráulico, basado en desarrollo experimental. *Investigación*, 2(16):21.

- Ponce, J., Torres, A., Aguilera, F., Silva Sprock, A., Flor, E., Casali, A., Scheihing, E., Tupac, Y., Torres, D., Ornelas, F., Hernández, J.-A., D., C., Vakhnia, N., and Pedreño, O. (2014). *Inteligencia Artificial*.
- Real-Moreno, O., Castro-Toscano, M. J., Rodríguez-Ouiñonez, J. C., Hernández-Balbuena, D., Flores-Fuentes, W., and Rivas-Lopez, M. (2018). Implementing k-nearest neighbor algorithm on scanning aperture for accuracy improvement. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3182–3186. IEEE.
- Real-Moreno, O., Rodríguez-Quiñonez, J. C., Sergiyenko, O., Basaca-Preciado, L. C., Hernandez-Balbuena, D., Rivas-Lopez, M., and Flores-Fuentes, W. (2017). Accuracy improvement in 3d laser scanner based on dynamic triangulation for autonomous navigation system. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1602–1608. IEEE.
- Reyes, F. (2011). *Robótica-Control de robots manipuladores*. Alfaomega grupo editor.
- Rodríguez-Quinonez, J. C., Sergiyenko, O., Gonzalez-Navarro, F. F., Basaca-Preciado, L., and Tyrsa, V. (2013). Surface recognition improvement in 3d medical laser scanner using levenberg-marquardt method. *Signal Processing*, 93(2):378–386.
- Rosicky, J., Grygar, A., Chapcak, P., Bouma, T., and Rosicky, J. (2016). Application of 3d scanning in prosthetic & orthotic clinical practice. In *Proceedings of the 7th international conference on 3D body scanning technologies*, pages 88–97.
- Sarasola, A. F. (2007). *Control de los sistemas continuos*. Universidad de Oviedo.
- Smarandache, F. and Leyva-Vázquez, M. (2018). *Fundamentos de la lógica y los conjuntos neutrosóficos y su papel en la inteligencia artificial*. Infinite Study.
- Solé, A. C. (2009). *Instrumentos industriales: su ajuste y calibración*. Marcombo.
- Sung, C. H. and Raudkivi, U. (1980). A new encoding scheme for an absolute digital encoder in units of degrees. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, (1):5–9.
- Trujillo-Hernández, G., Rodríguez-Quiñonez, J. C., Ramírez-Hernández, L. R., Castro-Toscano, M. J., Hernández-Balbuena, D., Flores-Fuentes, W., Sergiyenko, O., Lindner, L., and Mercorelli, P. (2019). Accuracy improvement by artificial

neural networks in technical vision system. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 5572–5577. IEEE.

Wolf, W. (2004). *FPGA-based system design*. Pearson education.