



# **UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA INSTITUTO DE INGENIERÍA**

**MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA**



**“Optimización de paso de barrido tridimensional a través de Algoritmo Genético”**

**TESIS PARA OBTENER EL GRADO DE  
MAESTRÍA EN CIENCIAS**

**PRESENTA  
MARCO ANTONIO TORREBLANCA SÁNCHEZ**

**DIRECTOR  
DR. OLEG SERGIYENKO**

**CO-DIRECTOR  
DR. ANDREI TCHERNYKH**

**Mexicali, Baja California a 1 de Junio del 2016**

## DEDICATORIAS Y AGRADECIMIENTOS

Dedico esta tesis a dos personas que en particular amo demasiado. La primera persona es mi Madre, persona que a pesar de las adversidades de la vida, siempre me ha sacado adelante con sus esfuerzos, cariño, amor y perseverancia. Me ha enseñado siempre a valorar lo que uno tiene, seguir adelante siempre con objetivos en la vida y terminar siempre lo que uno empieza. La segunda persona es para mí difunto Padrastro, el cual lo considero como un Padre desde mi niñez, siempre estuvo pendiente de mi Educación y crecimiento. Con todo cariño y amor dedico esta tesis a ellos dos.

Agradezco a Isabel, persona a la cual admiro, quiero y estimo demasiado. No tenga forma de agradecerle por su incondicional amor, cariño, comprensión, ternura y apoyo que me ha dado en estos casi dos años de conocerla.

Agradezco al Dr. Oleg por su comprensión y paciencia sobre mis problemas personales que se presentaron durante estos años de Maestría. También por su gran tutoría para concluir esta gran etapa de mi vida.

Agradezco al Dr Andrei y los Doctores que me apoyaron y me dieron clases el cual considero importante para mi formación académica.

**RESUMEN de Tesis de Marco Antonio Torreblanca Sánchez, presentada como requisito parcial para la obtención del grado de MAESTRÍA EN CIENCIAS, EN EL ÁREA DE ELECTRÓNICA, Mexicali Baja California, México a 1 de Junio del 2016.**

## **OPTIMIZACIÓN DE PASO DE BARRIDO TRIDIMENSIONAL A TRAVÉS DE ALGORITMO GENÉTICO.**

**Resumen aprobado por:**



---

**Dr. Oleg Sergiyenko**

La Tesis de Maestría en Ciencias consiste en la optimización de paso de barrido tridimensional en sistema láser escáner del Prototipo TVS a través de Algoritmo Genético. El Capítulo 1 describe el objetivo general, objetivos específicos, descripción breve de las características del Prototipo TVS y el alcance que desea tener en áreas de Navegación Autónoma.

EL Capítulo 2 describe las diferentes tecnologías y metodologías usadas en el desarrollo del Prototipo TVS, la cuales son el funcionamiento del láser escáner, los diferentes tipo de láseres, la metodología de triangulación dinámica, el software de LabVIEW, descripción del Diseño del Prototipo TVS y la herramienta de búsqueda de Algoritmo Genético.

El Capítulo 3 describe la postulación de mejoras que aplican en el Diseño Electrónico, Diseño Mecánico y Diseño de Software del Prototipo TVS. Las propuestas de cada diseño se postulan, principalmente se trabaja en la realización de las mejoras del Diseño de Software. El Capítulo 4 presenta el desarrollo de las propuestas a través de tres herramientas computacionales con el fin de mejorar el Prototipo y la construcción del Algoritmo Genético.

Capítulo 5 presenta los resultados de la evaluación y análisis de las tres herramientas computacionales desarrolladas en la plataforma de LabVIEW. Por último, el Capítulo 6 describe las conclusiones y las recomendaciones para mejorar las condiciones del Prototipo TVS.

## ÍNDICE

### CONTENIDO

<b>CAPÍTULO 1 – INTRODUCCIÓN.....</b>	<b>5</b>
1.1 <i>ESTADO DEL ARTE.....</i>	5
1.2 <i>OBJETIVO GENERAL.....</i>	7
1.3 <i>OBJETIVOS ESPECÍFICOS.....</i>	7
1.4 <i>PLANTEAMIENTO DEL PROBLEMA.....</i>	8
1.5 <i>HIPÓTESIS.....</i>	8
1.6 <i>CONTEXTO DE LA INVESTIGACIÓN.....</i>	8
1.7 <i>LIMITACIONES DE LA INVESTIGACIÓN.....</i>	9
<b>CAPÍTULO 2 – MARCO TEÓRICO.....</b>	<b>10</b>
2.1 <i>ESCÁNER TRIDIMENSIONAL.....</i>	10
2.2 <i>LÁSERES.....</i>	11
2.3 <i>TRIANGULACIÓN DINÁMICA – PRINCIPIOS MATEMÁTICOS.....</i>	12
2.4 <i>SOFTWARE LABVIEW COMO INTERFAZ USUARIO-MÁQUINA.....</i>	15
2.5 <i>TECHNICAL VISION SYSTEM (TVS).....</i>	19
2.5.1 <i>DESCRIPCIÓN DEL DISEÑO ELECTRÓNICO DEL TVS.....</i>	19
2.5.2 <i>DESCRIPCIÓN DEL DISEÑO DE SOFTWARE DEL TVS.....</i>	21
2.5.3 <i>DESCRIPCIÓN ELECTRÓNICA DEL PASO DE BARRIDO DEL TVS.....</i>	26
2.5.4 <i>DESCRIPCIÓN FÍSICA DEL PASO DE BARRIDO.....</i>	30
2.6 <i>ALGORITMO GENÉTICO.....</i>	31
2.6.1 <i>ESTRUCTURA DEL ALGORITMO GENÉTICO.....</i>	32
2.6.2 <i>TÉCNICAS DE SELECCIÓN DEL ALGORITMO GENÉTICO.....</i>	34
2.6.3 <i>TÉCNICAS DE CRUZAMIENTO DEL ALGORITMO GENÉTICO.....</i>	35
2.6.4 <i>TÉCNICAS DE MUTACIÓN DEL ALGORITMO GENÉTICO.....</i>	38
2.6.5 <i>TÉCNICAS DE REEMPLAZO DEL ALGORITMO GENÉTICO.....</i>	39
<b>CAPÍTULO 3 – POSTULACIÓN DE PROBLEMAS DEL PROTOTIPO TVS.....</b>	<b>41</b>
3.1 <i>¿CÓMO MEJORAR EL PASO DE BARRIDO DEL LÁSER ESCÁNER DEL PROTOTIPO TVS?.....</i>	41

3.2 PROPUESTAS GENERALES PARA LA SOLUCIÓN DE PROBLEMAS DEL TVS.....	42
3.3 PROBLEMAS ACTUALES DEL DISEÑO DE SOFTWARE DEL TVS .....	43
3.4 PROPUESTAS PARA EL DISEÑO DE SOFTWARE DEL TVS.....	45
<b>CAPÍTULO 4 – DESARROLLO DE PROPUESTAS.....</b>	<b>46</b>
4.1 ALGORITMO GENÉTICO ROBUSTO.....	46
4.1.1 DESCRIPCIÓN DEL PANEL FRONTAL DEL ALGORITMO GENÉTICO ROBUSTO .....	48
4.1.2 PSEUDOCÓDIGO DEL ALGORITMO GENÉTICO ROBUSTO .....	50
4.2 MEJORA DE LA INTERFAZ GRAFICA DE USUARIO DEL TVS .....	57
4.2.1 DESCRIPCIÓN DE LA INTERFAZ GRAFICA DEL TVS VERSIÓN 2.0 .....	58
4.2.2 CÓDIGO DE LA INTERFAZ GRAFICA DEL TVS VERSIÓN 2.0 .....	60
4.3 SIMULADOR DE ESCENARIOS VIRTUALES .....	66
4.3.1 DESCRIPCIÓN DE SIMULACIÓN DEL POSICIONAMIENTO LÁSER .....	67
4.3.2 DESCRIPCIÓN DE SIMULACIÓN DE ESCENARIOS .....	70
4.3.3 DESCRIPCIÓN DE SIMULACIÓN DE MAPEO DE FIGURAS .....	73
4.3.4 RESTRICCIONES DEL SIMULADOR DE ESCENARIOS VIRTUALES .....	74
4.3.5 DESCRIPCIÓN DEL PANEL FRONTAL DEL SIMULADOR DE ESCENARIOS VIRTUALES..	76
4.3.6 DESCRIPCIÓN DEL CÓDIGO DEL SIMULADOR DE ESCENARIOS VIRTUALES .....	79
<b>CAPÍTULO 5 – RESULTADOS DE LOS PROGRAMAS PARA MEJORAR EL PASO DE BARRIDO.....</b>	<b>84</b>
5.1 PRUEBAS Y RESULTADOS DE ALGORITMO GENÉTICO ROBUSTO .....	85
5.2 PRUEBAS Y RESULTADOS DE ALGORITMO GENÉTICO ROBUSTO .....	92
5.3 PRUEBAS Y RESULTADOS DE ALGORITMO GENÉTICO ROBUSTO .....	93
<b>CAPÍTULO 6 – CONCLUSIONES.....</b>	<b>98</b>
<b>REFERENCIAS .....</b>	<b>101</b>
<b>ANEXOS.....</b>	<b>105</b>

## CAPÍTULO 1 – INTRODUCCIÓN

### 1.1 ESTADO DEL ARTE

En la actualidad existe una gran variedad de mecanismos electromecánicos físicos llamados robots, desarrollados para efectuar tareas que el ser humano no puede hacer por su nivel de complejidad o simplemente no estamos fisiológicamente diseñados para ello. Por ejemplo, realizar mediciones o adquisiciones de lecturas métricas a escalas pequeñas (mili, micro, nano, etc.) o desempeñar labores en un ambiente peligroso puede resultar difícil o hasta imposible. Las principales razones para el avance tecnológico, es compensar las limitaciones mencionadas del ser humano y facilitar el trabajo de tareas difíciles con sistemas robustos e inteligentes, donde en las últimas décadas se ha incrementado la tendencia en desarrollar sistemas capaces de realizar movimientos autónomos.

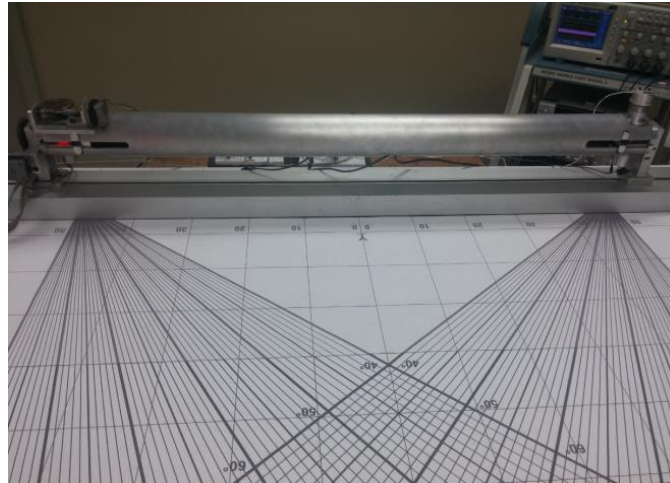
Dichos sistemas autónomos, dependiendo de la aplicación, pueden llegar a integrarse con otros sistemas para lograr un objetivo y manipular su entorno, un ejemplo de ello son los sistemas llamados láser escáner tridimensional [1], utilizados para analizar objetos o escenas, con el fin de crear modelos digitales tridimensionales utilizados en una amplia gama de aplicaciones y áreas de investigación.

Científicos e Ingenieros trabajan día con día para lograr aumentar la precisión, calidad y rendimiento de los sistemas tecnológicos operados en la actualidad, sin embargo haciendo énfasis en su disponibilidad o accesibilidad, el aumento significativo de características o mejora, puede figurar un incremento en su precio, haciendo este tipo de tecnología muy costoso para ser adquiridos por un mercado del sector público, por ende, en la mayoría de los casos solo puede estar disponible para mercados del sector privado e industrial.

La investigación en Universidades e Instituciones Académicas contribuye con el avance tecnológico de sistemas complejos y robustos, facilitando de tal forma el camino para reducir el costo de su adquisición. El desarrollo de nuevas técnicas, métodos para optimizar su funcionamiento, y además el uso de materiales de bajo costo, hace posible el acceso a esta tecnología sea más sencillo.

En el Instituto de Ingeniería de la Universidad Autónoma de Baja California campus Mexicali, existe un prototipo láser escáner (Figura 1.1.0). Fue desarrollado con la finalidad de utilizarse en

áreas de Monitoreo de Estructuras, Medicina y Navegación. En el área de Medicina tiene como aplicación realizar mediciones de parámetros biométricos [2] y en el área de Navegación ser utilizado en robot móviles [3]. Ahora bien, el prototipo tiene oportunidades y mejoras para llegar aplicarse en otras diferentes áreas, tales como Metrología, Ingeniería y Exploración Espacial [4].



**Figura 1.1.0** Prototipo de láser escáner (Vista Frontal).

El equipo de la línea de investigación de Optoelectrónica y Mediciones se encuentra trabajando en desarrollar un robot competente para el área de Navegación Autónoma, el robot realizará su trayectoria de forma autónoma con la ayuda del prototipo láser escáner. Dicho láser escáner tiene la capacidad de inspeccionar el entorno físico con el fin de ser integrado a un robot para trazar una ruta segura durante su trayecto [5], de tal manera, podrá evadir objetos que representen una amenaza de colisión y continuar con su objetivo. Sin embargo, el prototipo todavía tiene una amplia oportunidad de mejora y desarrollo en cuestiones mecánicas, software y electrónicas que ayudarían sin duda alguna al funcionamiento y desenvolvimiento dentro de la aplicación destinada. Por lo tanto, se ha analizado y detectado que mejorando el paso de barrido del prototipo láser escáner, podemos lograr una mejor ejecución en la funcionalidad del prototipo.

Mi problema de tesis tiene la finalidad de mejorar el proceso de paso de barrido del prototipo mencionado, haciendo un enfoque en la parte encargada en la búsqueda y localización de obstáculos en el escenario, mejorando de tal forma la velocidad y sin perder precisión en la calidad de captura, entonces cumpliendo con este objetivo el prototipo se optimiza. Esto es posible a través de una selección analítica de valores óptimos locales en el paso de barrido con la ayuda de una herramienta de software llamada Algoritmos Genéticos.

## **1.2 OBJETIVO GENERAL**

En esta investigación se propone desarrollar un diseño basado en Métodos de Optimización con el objetivo de generar un ángulo de paso de barrido capaz de incrementar la precisión de captura de información de puntos detectados dentro del escenario y aparte aumentar la velocidad del escaneo del Sistema Láser (dicho sistema es conocido como TVS por sus siglas en inglés – Technical Vision System), de tal forma esto se traduce en mejorar el rendimiento de los actuales recursos del software (LabVIEW), recursos electrónicos y movimientos mecánicos que hace posible funcionar al sistema.

## **1.3 OBJETIVOS ESPECÍFICOS**

1. Mejorar el funcionamiento del programa actual que utiliza el Sistema de Barrido Láser (TVS) para reducir/evitar que el programa se congele o deje funcionar cuando este en operación.
2. Agregar funciones nuevas al TVS que mejoren su funcionamiento para interactuar de una forma eficiente en aplicaciones dirigidas a Navegación Autónoma.
3. Desarrollar Simulación robusta del Algoritmo Genético para evaluar su eficiencia.
4. Desarrollar Simulación en 2D que represente cualquier escenario físico, en donde el TVS podría interactuar.
5. Analizar y Evaluar los diferentes tipos de Métodos que constituyen al Algoritmo Genético para seleccionar el paso de barrido óptimo.
6. Hacer pruebas de funcionamiento en la Simulación en 2D.
7. Integrar la Simulación del Algoritmo Genético a la Simulación del escenario en 2D para evaluar la interacción entre ambas simulaciones.
8. Investigar en publicaciones, artículos y revistas de prestigio que discutan sobre temas de interés acerca de Métodos de Optimización que puedan ayudar al desarrollo del modelo matemático.
9. Formalizar y sustentar teóricamente el modelo matemático propuesto.
10. Hacer pruebas y mediciones del simulador en LabVIEW.
11. Crear una metodología para el análisis de los resultados.

## **1.4 PLANTEAMIENTO DEL PROBLEMA**

El prototipo utilizado para la elaboración de esta tesis, es un sistema de barrido por láser conocido como TVS (Technical Vision System, por sus siglas en inglés), dicho sistema realiza un reconocimiento digital de un entorno físico cualquiera, con el fin de ser utilizado para crear un camino seguro y evadir cualquier obstáculo que represente un peligro inminente al momento de trasladarse de manera autónoma de un Destino “A” hacia un destino “B” [6].

El TVS tiene subsistemas mecánicos y electrónicos que hacen posible el funcionamiento de la aplicación para la que fue diseñada [7] [8], actualmente dicho sistema tiene varias limitaciones (hardware y software) que representan oportunidad de mejora. En esta tesis nos enfocaremos en dos problemas.

1. Tiempo de respuesta en cada selección de ángulo de paso de barrido.
2. Adquisición de puntos mapeados dentro de un escenario, dependiendo de la selección de ángulo de paso de barrido.

## **1.5 HIPÓTESIS**

Como se ha mencionado, el sistema de barrido por láser puede llegar a ser implementado en áreas de Navegación Autónoma por sus características. Se puede describir a grandes rasgos como un diseño robusto configurado para adquirir de forma digital información de un mapa interpretadas como coordenadas en 3D de cualquier escenario o entorno que se desee analizar, la información puede ser decodificada e implementada en sistemas móviles que reconozcan caminos seguros y seleccionando el mejor recorrido para transportar individuos o cosas de manera autónoma sin sufran ningún peligro durante el trayecto.

## **1.6 CONTEXTO DE LA INVESTIGACIÓN**

El trabajo desarrollado de esta investigación fue realizado en las instalaciones del Laboratorio de Optoelectrónica y Mediciones del Instituto de Ingeniería de la Universidad Autónoma de Baja California Campus Mexicali, bajo la tutoría y revisión del Dr. Oleg Sergiyenko. También se desarrollaron pruebas de programación en las instalaciones de Centro de Investigación Científica y

de Educación Superior de Ensenada bajo la tutoría del Dr. Andrei Tchernykh con la finalidad de ampliar el conocimiento en Métodos de Optimización Computacionales.

El diseño de los programas, la implementación del Algoritmo en el prototipo de TVS, la evaluación y experimentación de cada uno de los programas, la corrección de errores de programación fueron hechos en las instalaciones del laboratorio mencionado para obtener resultados del compartimento del algoritmo principal. Este algoritmo se mejoró en base a la creación de otros programas con el fin de someterlo a diferentes pruebas que fueron útiles para el desarrollo y crecimiento del algoritmo principal, por lo tanto, se invirtió un tiempo considerable para encontrar una buena adaptación y sincronización entre el prototipo y el algoritmo.

### ***1.7 LIMITACIONES DE LA INVESTIGACIÓN***

Las limitaciones que se presentaron durante el desarrollo de esta investigación fueron con el funcionamiento del Prototipo TVS (Hardware y Software), y en la creación de distintos escenarios para ser usado en pruebas de experimentación. En la parte del Prototipo TVS tuve dificultades iniciales al encender el sistema por cables que se encontraban haciendo falso contacto electrónicamente y componentes electrónicos que no estaban correctamente posicionados en el tablero de pruebas (Protoboard), de tal forma, esta condición inhabilitaba el funcionamiento del sistema. La configuración de la instalación electrónica hace que continuamente el sistema sufra de falsos contactos, haciendo difícil iniciar o continuar con pruebas experimentales para el desarrollo del Algoritmo, es decir, se planeaba iniciar con pruebas pero en ocasiones el sistema no respondía, entonces me tomaba tiempo lograr la reparación y poder continuar con la operación normal del sistema. Su instalación electrónica tiene gran oportunidad de mejora para corregir este problema, en el Capítulo 4 se menciona las recomendaciones.

Otro detalle fue en la elaboración de distintos escenarios, el cual considero una limitante importante debido a que el sistema TVS no tiene las condiciones adecuadas para escanear escenarios de manera rápida, aparte no existe recursos suficientes y disponibilidad para crear un gran número de escenarios que pudieran estar interactuando físicamente con el sistema TVS, por lo tanto, para continuar con la investigación esto se corrigió de manera virtual con la creación de un Simulador de Escenarios 2D.

## CAPÍTULO 2 – MARCO TEÓRICO

### 2.1 ESCÁNER TRIDIMENSIONAL

La naturaleza del prototipo cae en la categoría de escáner tridimensional, por lo tanto, estos son dispositivos electrónicos que analizan el entorno de un objeto o escena espacial para crear una nube de puntos (representación discreta) de una superficie continua. La nube de puntos (matriz) son capturados para realizar cálculos métricos, vectorizar entidades, reconstrucción de modelos tridimensionales, etc. [9].

El escáner tridimensional comúnmente se cataloga en dos tipos, los de contacto mecánico y los ópticos sin contacto. Los de contacto mecánico son aquellos que examinan el objeto apoyando el elemento de medida (Palpador) sobre la superficie del mismo, típicamente una punta de acero duro o punta de zafiro. Una serie de sensores internos permiten determinar la posición espacial del palpador. Se usan en su mayoría en control dimensional en procesos de fabricación y pueden conseguir precisiones típicas de 0,01mm. La desventaja de este tipo es que requiere el contacto físico con el objeto para ser escaneado y al momento de escanear el objeto puede modificar o dañar la superficie del objeto.

El escáner tridimensional óptico sin contacto es una opción rápida para la medición de un gran número de puntos. Los métodos sin contacto se han preferido para grandes superficies o de forma libre, objetos con muchas características, objetos flexibles, u objetos frágiles, porque las superficies no son tocadas por una sonda. Este tipo de escáner se divide a su vez en dos métodos identificados como pasivos y activos.

**Pasivos:** También conocidos como visión estéreo densa (DSV, por sus siglas en inglés), los escáneres pasivos son aquellos escáneres que pueden capturar fotografías de alta resolución en superficies iluminadas con alto lumen de luz externa o ambiente, en lugar de irradiar luz. Técnicas como DSV utilizan múltiples cámaras y software de procesamiento, y son útiles para la exploración de gama media. Son fáciles de configurar, tiene tiempos de medición rápidos, y algunas versiones comerciales proporcionan correspondencia superficie automatizado. Los inconvenientes incluyen ser menos precisos que las técnicas de corto alcance debido a los efectos de oclusión, y sólo trabajando en piezas con textura que no emiten ninguna clase de radiación por sí mismos, pero en lugar se fía de detectar la radiación reflejada del ambiente [10].

**Activos:** Este método es utilizado comúnmente para la exploración de objetos. Conceptualmente, la energía se proyecta sobre una parte física, que se refleja para ser detectada por un dispositivo electrónico (típicamente una cámara o sensor). El análisis de la posición de la imagen de la energía reflejada determina las coordenadas (X, Y, Z) del objeto.

En la Figura 2.1.0 podemos observar el diagrama definido por diferentes tipos de escáner, el diagrama se ramifica hasta llegar al recuadro de triangulación (encerrado en un rectángulo rojo). El Prototipo desarrollado por los Doctores Julio Rodríguez y Luis Básaca está basado en el método de triangulación. La triangulación se apoya con sistemas ópticos (láser) donde pertenece al tipo método de óptico sin contacto. Para conocer una descripción más detallada de este tipo de tecnología, en la sección de anexo se adjunta la Tabla A.1

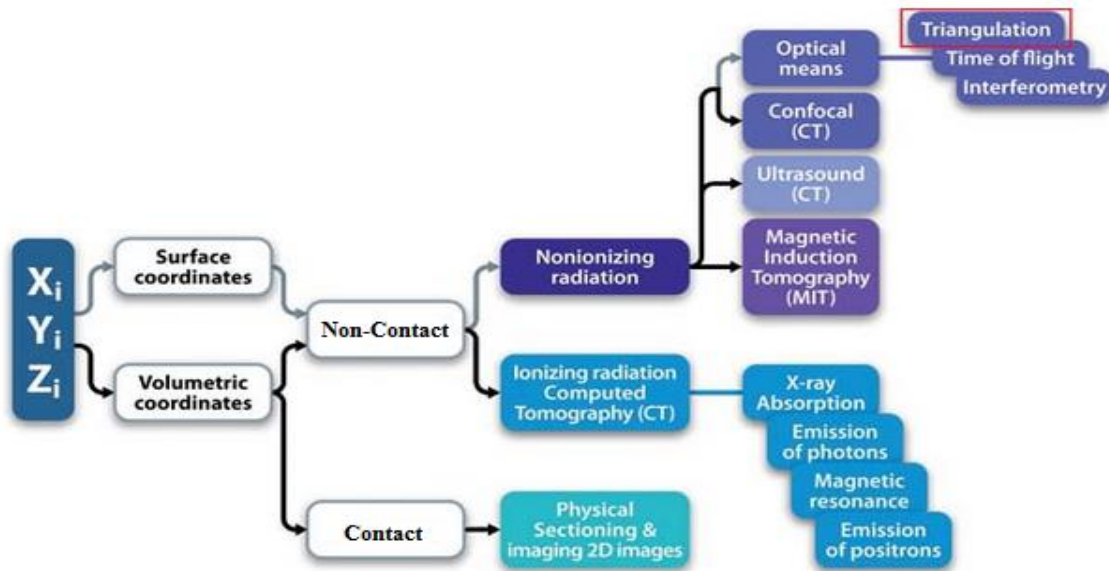


Figura 2.1.0 Diagrama de tipos de tecnología de Escáner Tridimensional [42].

## 2.2 LÁSERES

Un instrumento capaz de generar ondas de luz usando una estrecha banda del espectro se le llama láser. Un láser típico emite luz en un estrecho y poco divergente haz de longitud de onda bien definida (correspondiente a un color particular si el láser opera en el espectro visible). Los láseres tienen la propiedad de ser similares a los transistores, generan o amplifican luz al igual que los transistores generan o amplifican señales electrónicas de audio, radio o frecuencia. La palabra láser

es el acrónimo en inglés de las palabras amplificación de la luz por emisión inducida de radiación (Light Amplification by Stimulated Emission of Radiation).

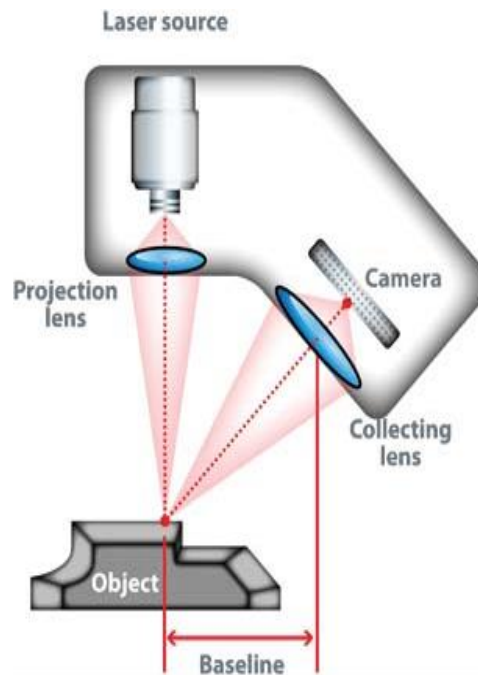
Las aplicaciones comunes del láser son los lectores de los códigos de barras y, obviamente, los punteros láser. En la industria, los láseres se usan para cortar el acero y otros metales y para grabar estampados, como por ejemplo las letras en los teclados de ordenador. También se utilizan en aplicaciones médicas y militares. El funcionamiento típico de un láser consta de cuatro etapas:

- a) **Bombeo:** Se provoca mediante una fuente de radiación como puede ser una lámpara, el paso de una corriente eléctrica o el uso de cualquier otro tipo de fuente energética que provoque una emisión.
- b) **Emisión espontánea de radiación:** Los electrones que vuelven al estado fundamental emiten fotones. Es un proceso aleatorio y la radiación resultante está formada por fotones que se desplazan en distintas direcciones y con fases distintas generándose una radiación monocromática incoherente.
- c) **Emisión estimulada de radiación:** La emisión estimulada, base de la generación de radiación de un láser, se produce cuando un átomo en estado excitado recibe un estímulo externo que lo lleva a emitir fotones y así retornar a un estado menos excitado. El estímulo en cuestión proviene de la llegada de un fotón con energía similar a la diferencia de energía entre los dos estados. Los fotones así emitidos por el átomo estimulado poseen fase, energía y dirección similares a las del fotón externo que les dio origen. La emisión estimulada descrita es la raíz de muchas de las características de la luz láser. No sólo produce luz coherente y monocroma, sino que también "amplifica" la emisión de luz, ya que por cada fotón que incide sobre un átomo excitado se genera otro fotón.
- d) **Absorción:** Proceso mediante el cual se absorbe un fotón. El sistema atómico se excita a un estado de energía más alto, pasando un electrón al estado meta-estable. Este fenómeno compete con el de la emisión estimulada de radiación.

### ***2.3 TRIANGULACIÓN DINÁMICA – PRINCIPIOS MATEMÁTICOS***

La técnica más utilizada en los sistemas con láser escáner es la triangulación. La triangulación láser es una técnica estereoscópica [11], donde la distancia del objeto es calculada por medio de una posición de luz láser y una cámara. Un haz de luz láser se proyecta desde un espejo sobre el objeto a

ser escaneado (Figura 2.3.0). El objeto refleja la luz, la cual es detectada por una cámara localizada a una distancia conocida del emisor de la luz láser. Existen dos tipos de triangulación por láser; triangulación estática y triangulación dinámica. Para realizar el cálculo de alguna coordenada espacial por medio de la triangulación estática existen 3 variables que deben ser conocidas: El ángulo de emisión del láser, el ángulo detectado por el arreglo de foto-detectores (usualmente una cámara) y la distancia entre el emisor de luz láser y la cámara [12]. Una de las ventajas más comunes son que el receptor tiene una limitación en su campo de visión (FOV por sus siglas en inglés), también es estos sistemas realizan el escaneo en un solo plano dimensional.



**Figura 2.3.0** Ejemplo de un Sistema de triangulación estática [43].

Por esa razón, se descartó la triangulación estática en el prototipo, y sus diseñadores decidieron desarrollar su propio método basándose en la triangulación dinámica. Su función se basa en teoremas de senos para realizar los cálculos de las coordenadas del sistema láser. A continuación se presentan ecuaciones que describen el funcionamiento de la triangulación dinámica del láser escáner en sistema del TVS, cuyos elementos son presentados en la Figura 2.3.1.

$$D_{ij} = a \frac{(\sin B_{ij})(\sin C_{ij})}{\sin[180^\circ - (B_{ij} + C_{ij})]} \quad (2.3.1)$$

$$X_{ij} = a \frac{(\sin B_{ij})(\sin C_{ij})(\cos \sum_{j=1}^n B_j)}{\sin[180^\circ - (B_{ij} + C_{ij})]} \quad (2.3.2)$$

$$Y_{ij} = a \left( \frac{1}{2} - \frac{(\sin B_{ij})(\sin C_{ij})}{\sin[180^\circ - (B_{ij} + C_{ij})]} \right) \text{ at } B_{ij} \leq 90^\circ \quad (2.3.3)$$

$$Y_{ij} = a \left( \frac{1}{2} + \frac{(\sin B_{ij})(\sin C_{ij})}{\sin[180^\circ - (B_{ij} + C_{ij})]} \right) \text{ at } B_{ij} \geq 90^\circ \quad (2.3.4)$$

$$Z_{ij} = a \frac{(\sin B_{ij})(\sin C_{ij})(\cos \sum_{j=1}^n B_j)}{\sin[180^\circ - (B_{ij} + C_{ij})]} \quad (2.3.5)$$

Dónde:

$B_{ij}$  = Es el ángulo del Sistema de Apertura.

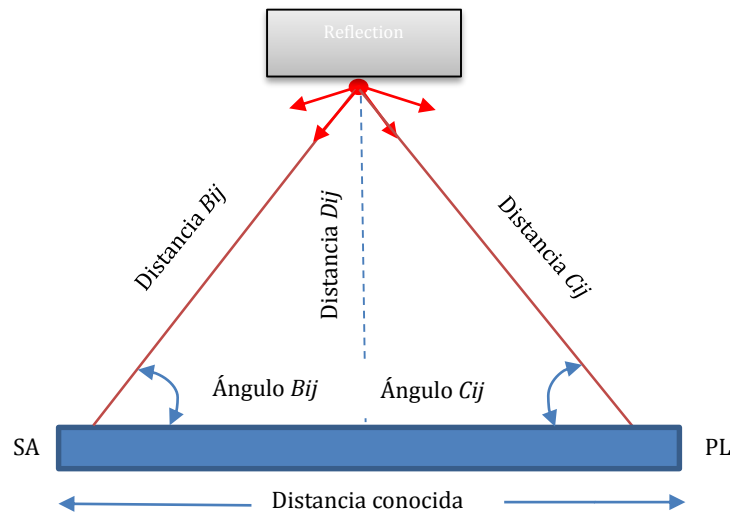
$C_{ij}$  = Es el ángulo del Posicionador Láser.

$X_{ij}$  = La Distancia conocida en el eje "X" del escenario.

$Y_{ix}$  = La Distancia conocida en el eje "Y" del escenario.

$Z_{ij}$  = La Distancia conocida en el eje "Y" del escenario.

$D_{ij}$  = La Distancia conocida entre el objeto detectado y el Prototipo TVS.



**Figura 2.3.1** Principio de Triangulación Dinámica.

Estas ecuaciones presentadas hacen posible el funcionamiento de la Triangulación Dinámica y ha sido comprobado experimentalmente por sus diseñadores en el campo de Medicina [2] [7]. En la Figura 2.3.1 podemos observar de forma gráfica el principio de Triangulación Dinámica usado en el funcionamiento del TVS, donde la distancia más importante por conocer del Sistema es la Distancia  $D_{ij}$ . En el apartado 2.5.3 y 2.5.4 se menciona a fondo acerca de las partes que consta la triangulación dinámica del TVS.

## 2.4 SOFTWARE LABVIEW COMO INTERFAZ USUARIO-MÁQUINA

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) es un software desarrollado desde el año 1986 por la compañía de National Instruments [13] utilizado frecuentemente en el área de la industria e investigación [14]. Este software es una plataforma de programación para crear aplicaciones o programas que sirvan en la adquisición de medición, control, diseño, instrumentación, pruebas y simulación de cualquier sistema, su lenguaje de programación reemplaza las líneas de texto comunes que usan los softwares de programación (tales como ABAP, C++, JAVA, MATLAB, Visual Basic, entre otros) por iconos gráficos, es conocida como Lenguaje G por su entorno gráfico en su área de programación.

Un programa creado en LabVIEW es conocido como Instrumento Virtual (VI - Virtual Instruments), su extensión informática para ser reconocido por el Sistema Operativo es la extensión .VI. Su entorno visual se divide en dos ventanas llamadas Panel Frontal y Diagrama de Bloque, el primero mencionado es la interface humano máquina en la cual se encuentran controles e indicadores que permiten al usuario interactuar con el instrumento virtual. Los controles e indicadores simulan la apariencia de un instrumento real, por lo tanto, se pueden personalizar modificando su posición, tamaño estructura, etc., para lograr coincidir el entorno visual con el sistema que se esté desarrollando.

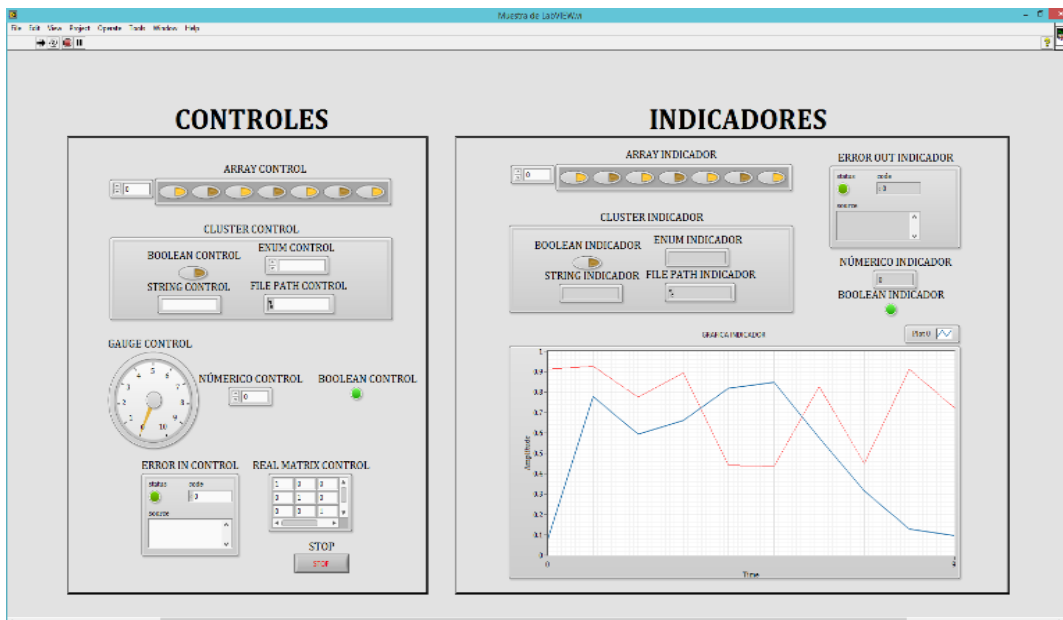
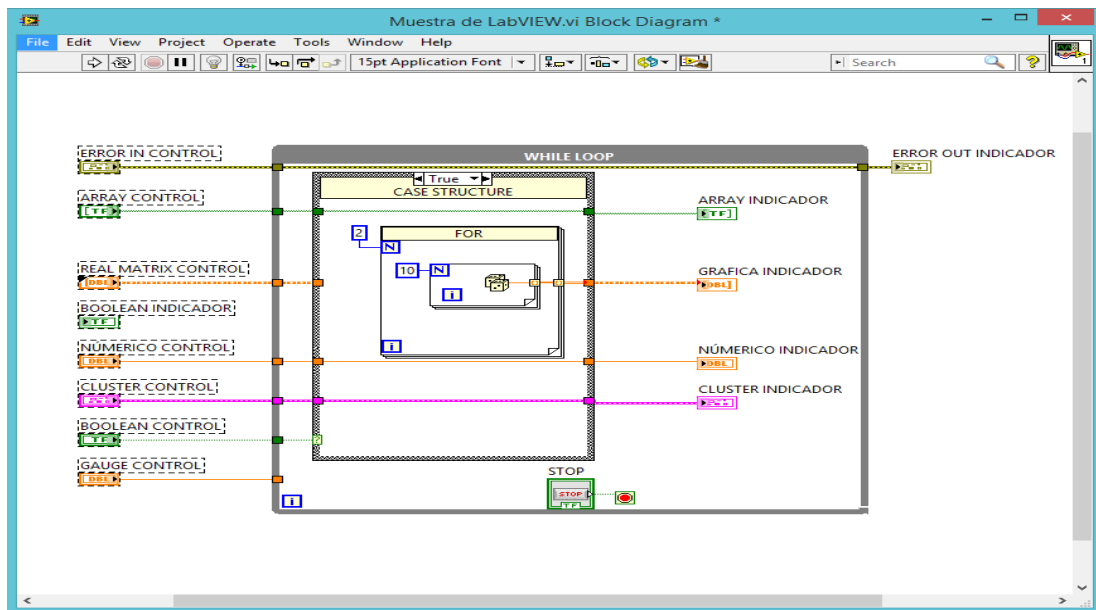


Figura 2.4.0 Ejemplo del Panel Frontal del Software LabVIEW.

En la Figura 2.4.0 se observa un ejemplo de la pantalla Front Panel con sus respectivos controles e indicadores que tiene un VI, este puede constar de diferentes tipos de información, lo más comunes son numéricos, booleanos, cadenas, arreglos, clúster, matrices, etc., se pueden visualizar los tipos de información como control o indicador. También existen controles e indicadores para campos muy específicos que sirven principalmente para el área de Instrumentación Electrónica, por nombrar tres de ellos como simple ejemplo, encontramos que existen controles e indicadores para el área de control, visión y robótica.

En el Diagrama de Bloques se encuentra el código de programación basado en iconos que se conectan para formar un flujo de datos, cada control e indicador del panel frontal tiene un terminal que lo representa en el Diagrama de Bloques, por lo general, todo es conectado al inicio de una terminal para ser vinculado por medio de un alambre y mandar toda la información contenida hacia otra terminal, estos alambres hacen la conexión entre varias terminales para realizar las instrucciones que el programador diseña.

En la Figura 2.4.1 se presenta un ejemplo de la pantalla Diagrama de Bloques, en este ejemplo se muestra el ordenamiento frecuente de los controles e indicadores, los controles se posicionan en la parte izquierda de la pantalla y los indicadores en la parte derecha, además el ejemplo contiene tres funciones básicas en la programación de LabVIEW, estas son “While Loop”, “Case Structure” y Ciclo “For”, dichas funciones son usualmente usadas en otros Lenguajes de Programación.



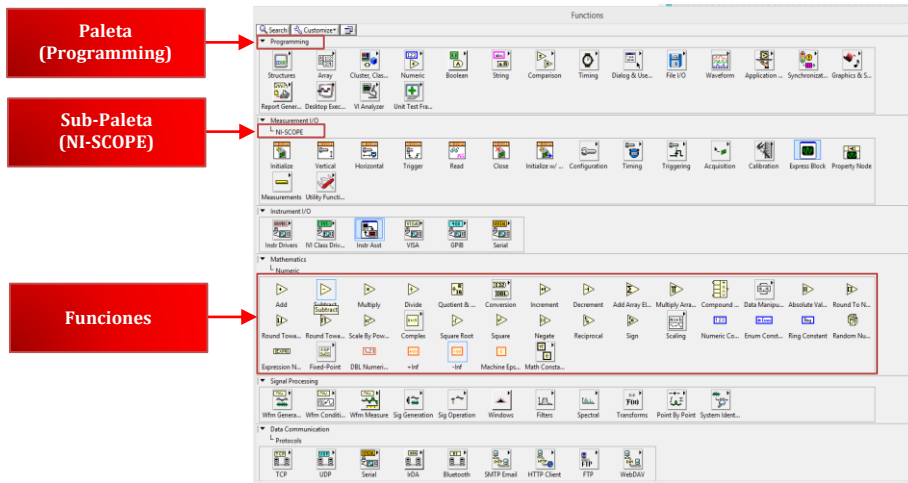
**Figura 2.4.1** Ejemplo del Diagrama de Bloques del Software LabVIEW.

Todas las herramientas y funciones se consiguen a través de las paletas de funciones incorporadas en el Diagrama de Bloques, las diferentes funciones están agrupadas en sub paletas según su utilidad. Existen paletas o herramientas especializadas que son importante en la capacidad de integración en áreas específicas del conocimiento, las cuales permiten crear aplicaciones avanzadas [15], a continuación se describen las paletas básicas y varias de Paletas especializadas que contiene LabVIEW:

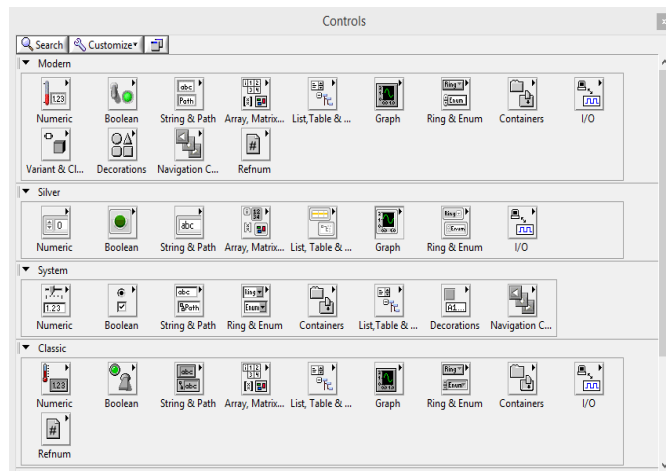
- a) *PROGRAMMING* – Contiene las funciones básicas y complejas usadas en cualquier Lenguaje de Programación (Ciclo For, While Loop, Case Structures, If, arreglos, etc), estas funciones sirven para realizar instrucciones y secuencias de un programa.
- b) *MEASUREMENT I/O* – Permite la comunicación para acceder a controlar las funciones de Instrumentos de Medición, por ejemplo, Osciloscopios, Multímetros, Generador de Funciones, etc., principalmente está enfocado a productos de la compañía de National Instruments.
- c) *INSTRUMENT I/O* – Es muy similar a la anterior, debido a que tiene funciones para controlar Instrumentos de Medición, pero en este caso no necesitan ser exclusivos de National Instruments, puede ser Instrumentos de otras compañías, siempre y cuando, el Instrumento sea capaz de adquirir una medición o lectura de cualquier índole y permita la transferencia para procesar dicha información a través de un protocolo de comunicación.
- d) *MATHEMATICS* – Contiene una rama amplia del área de Matemáticas, las comunes son Cálculo Diferencial, Cálculo Integral, Geometría, Trigonometría, Estadísticas, etc., estas funciones son útiles para hacer cálculos con los datos que adquiere un Instrumento de Medición
- e) *SIGNAL PROCESSING* – Sirve para el tratamiento de señales, tiene una gran variedad de funciones para filtrar, acondicionar, analizar una señal cualquiera.
- f) *DATA COMMUNICATION* – Tiene dos funciones principales, (1) interactuar con los datos que tiene el programa, ayudando en su sincronización y correcta ejecución, (2) y proveer una lista de funciones para establecer comunicación con cualquier dispositivo electrónico que pueda comunicarse a través de los diferentes Protocolos de Comunicación existentes (RS-232, ModBUS, VISA, GPIB, TCP, UDP, etc.).
- g) *REAL TIME* – Son funciones que permiten procesar la información y la comunicación de los dispositivos conectados con LabVIEW en tiempo real, habilitar esta función depende de las características del Instrumento de Medición o Dispositivo.
- h) *FPGA INTERFACE* – Contiene funciones para los dispositivos electrónicos catalogados como FPGA.

i) **ROBOTICS** – Contiene una extensa variedad de funciones para aplicaciones Robóticas.

En la Figura 2.4.2 y 2.4.3 se muestran respectivamente las Paletas, Sub-Paletas y funciones existentes en las pantallas del Diagrama de Bloques y Panel Frontal. Las Paletas, Sub-Paletas y funciones del Panel Frontal hace referencia a las terminales de programación utilizadas en las herramientas del Diagrama de Bloques.



**Figura 2.4.2** Lista de Paletas y Sub-Paletas del Diagrama de Bloques.



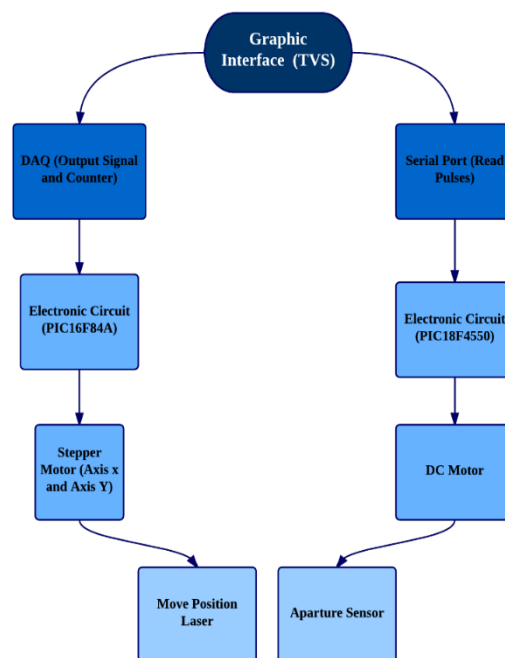
**Figura 2.4.3** Lista de Paletas y Sub-Paletas del Panel Frontal.

De tal forma, LabVIEW es el lenguaje de Programación usado en el desarrollo de Interfaz gráfica del Prototipo TVS por contener herramientas virtuales, simplicidad en la programación, entorno gráfico amigable y accesibilidad en la Instrumentación y Adquisición de Datos de cualquier Instrumentos de Medición o Dispositivo Electrónico. En el apartado 2.5.2 se describe a más detalle el desarrollo de la Interfaz Gráfica.

## 2.5 TECHNICAL VISION SYSTEM (TVS)

El prototipo de Technical Vision System fue un desarrollo académico por dos estudiantes de Doctorado que estaban en el programa de Maestría y Doctorado en Ciencias e Ingeniería (MyDCI) del Instituto de Ingeniería de la Universidad Autónoma de Baja California. Este prototipo consiste en un sistema de escaneo láser a través de un barrido electromecánico y una Interfaz Usuario-Máquina (GUI, por sus siglas en inglés) llamada TVS, este sistema físico fue aplicado experimentalmente en las áreas de Medicina y Navegación Especial, su fabricación principalmente consistió en un Diseño Mecánico, Diseño Electrónico, Diseño Optoelectrónico, Diseño de Software y Diseño Experimental [6] [7]. Este apartado hará énfasis acerca del Diseño Electrónico, Diseño de Software, Diseño Mecánico y descripción del funcionamiento del ángulo de paso de barrido del Prototipo TVS, con la finalidad de conocer y analizar las partes importantes que sirvan para el desarrollo y mejoramiento de este trabajo de investigación. También se comentará recomendaciones necesarias en corregir de manera urgente para mejorar el compartimento y la funcionalidad del Prototipo en caso de seguir con planes de utilizarlo en una aplicación más compleja en las áreas ya insinuadas.

### 2.5.1 DESCRIPCIÓN DEL DISEÑO ELECTRÓNICO DEL TVS



**Figura 2.5.0** Diagrama del funcionamiento electrónico del TVS.

Electrónicamente se divide en dos secciones para realizar su funcionamiento como escaneo láser, en la Figura 2.5.0 se muestra como el Diagrama del funcionamiento del TVS empieza con la Interfaz Gráfica del Usuario (GUI - Graphical User Interface), dónde su función es crear un entorno visual para permitir la comunicación entre el usuario y el sistema de cómputo, de tal forma, el Prototipo puede realizar acciones de ejecución de instrucciones de comandos, control de señales digitales y adquisición de datos (Ver apartado 2.5.2 para más detalle).

En la parte izquierda, muestra el control electrónico del movimiento del Posicionador Láser (PL), dicho control se puede dividir en tres etapas para describir su operación. (1) Está constituido inicialmente de una tarjeta electrónica llamada Adquisición de Datos (DAQ, por sus siglas en inglés) modelo USB-1208SL, fabricada por la compañía MEASUREMENT COMPUTING<sup>TM</sup> [14], su principal función es adquirir y controlar las señales digitales de salida (DIO - Digital Input Output) de cualquier circuito electrónico, siempre y cuando sea compatible con sus características de amplitud de voltaje y corriente, aparte tiene una función integrada que sirve como Contador de Salida (CO - Counter Output), esta función es utilizada por el TVS para enviar digitalmente anchos de pulsos controlados (de forma cuantitativa) al siguiente circuito electrónico. (2) En este circuito inicia la segunda etapa del control mencionado, por lo tanto, el circuito es capaz de recibir los pulsos enviados por el DAQ a través de dos microcontroladores, del modelo PIC16F84A, dichos microcontroladores son genéricos y están físicamente incorporado individualmente dentro de una pastilla electrónica, por ser un circuito programable, sus instrucciones están programadas por sus Desarrolladores [15] para controlar electrónicamente y mecánicamente la última etapa. (3) La última consiste en la instalación de dos motores a pasos incorporados a un sistema mecánico de engranes para mover mecánicamente la sección del Posicionador de Láser, dicha sección está constituido de puntero láser y un lente óptico para ser movido de forma circular dentro de su mismo eje y reflejar el haz de luz del láser mencionado.

La parte derecha muestra el control electrónico del movimiento de la Apertura de Escaneo (SA), este control también se puede dividir en tres etapas para detallar su operación. (1) La primer etapa inicia con el dispositivo del puerto serial que está incorporado dentro del sistema de cómputo del Prototipo, usualmente esta interfaz de comunicación es utilizada para enviar y recibir datos de forma binaria, existen diferentes parámetros y protocolos para llegar a configurar este tipo de puerto [16], en la tabla 2.5.0 se muestra los parámetros y el protocolo que fueron configurados dentro del TVS.

PROTOCOLO	BAUD RATE	DATA BITS	PARITY	FLOW CONTROL
RS-232	9600 Bits/Sec	8	None	None

**Tabla 2.5.0**

La función del puerto serial en el sistema es obtener información de la sección de la Apertura de Escaneo, la información recibida es la adquisición de pulsos digitales que se origina a través del circuito de la segunda etapa. (2) Este circuito consta de un microcontrolador modelo PIC18F4550 que se encarga de realizar el conteo de la detección de pulsos del fototransistor y leer los pulsos del opto-acoplador. El opto-acoplador está montado en el motor DC para conocer cada vuelta o revolución del eje, por lo tanto, funciona como una especie de Encoder [16]. El PIC de esta etapa administra la información de ambos pulsos y después pasa por un proceso de codificación para enviarla a través del puerto serial con la finalidad de ser leída por el Prototipo TVS y realizar automáticamente instrucciones para seguir con el funcionamiento del escaneo. La última etapa (3) es en circuito que se encarga de mandar una Modulación de Ancho de Pulsos (PWM – Pulse-Width Modulation) al motor DC mencionado.

### **2.5.2 DESCRIPCIÓN DEL DISEÑO DE SOFTWARE DEL TVS**

El desarrollo del diseño del Software del Prototipo TVS es esencial para ser utilizado por cualquier usuario u operario capacitado con el fin de aprovechar las funciones para lo que fue diseñado. Este diseño del Software, como se mencionó en el capítulo anterior, consta de un entorno visual conocido como GUI, este término sirve como etiqueta para hacer referencia a todo aquel sistema que se constituyen de una máquina y se desea tener el control electrónicamente a través de una interfaz computacional, es decir, tener un medio de comunicación entre el usuario y la máquina.

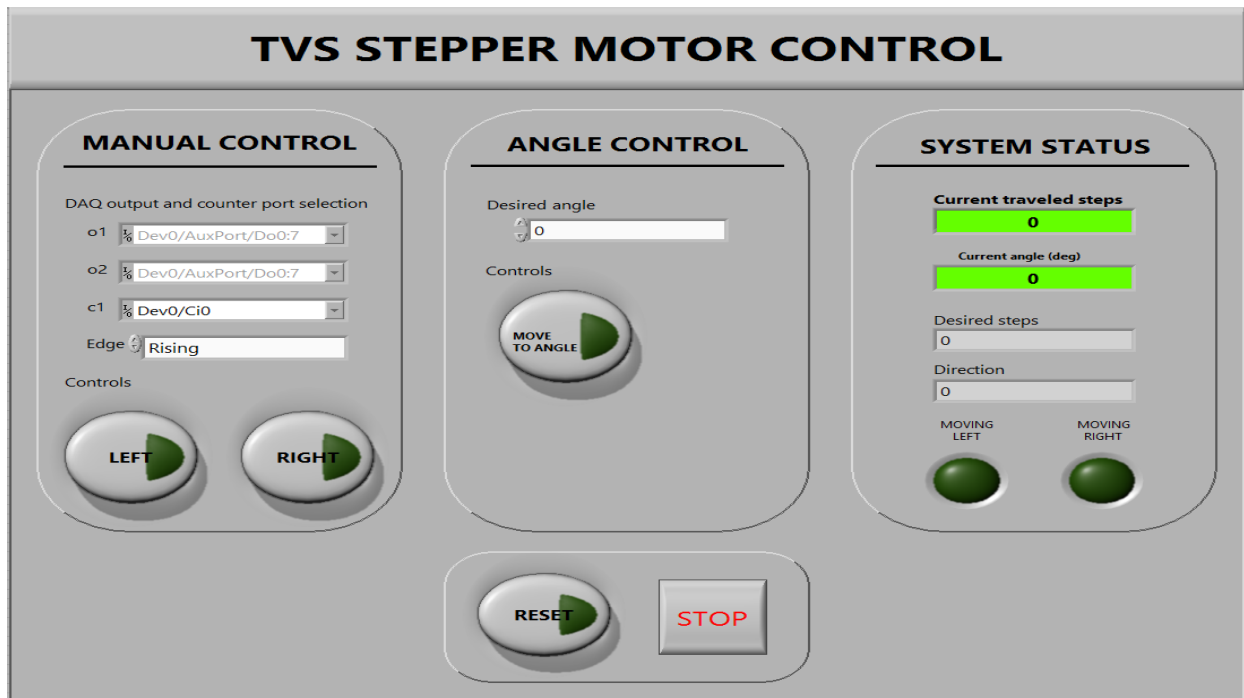
Los creadores del TVS catalogan su diseño de software (entorno visual computacional) como una Interfaz Hombre-Máquina (HMI – Human-Machine Interface) [16] [17], considero importante mencionar diferencias que distingue entre los sistemas HMI y GUI [18], por lo tanto, se debe de usar el término correcto para hacer referencia al Prototipo, entonces a continuación se detallan dos principales diferencias de los términos:

1. Un sistema HMI puede constar de un tablero de control o interruptores para crear un vínculo de comunicación entre el usuario y la máquina, no es obligatorio el uso o la instalación de una

pantalla para que exista tal vínculo. Estos sistemas incluyen funciones de Operación y Mantenimiento, además son generalmente diseñados de forma robusta para ser manejados en sectores industriales [19].

- Un sistema GUI se caracteriza principalmente por el uso de un sistema computacional, comúnmente incluyen teclado, ratón (mouse), y monitor, el diseño del entorno visual consta de ventanas, menús, iconos, botones e indicadores gráficos que representan de forma virtual la máquina. Sus características de Hardware (Memoria RAM, Memoria ROM, Procesador) comúnmente son mayores a los de un sistema HMI, debido a su interfaz está desarrollado en gran medida de contenido de gráficos computacionales que demandan procesamiento y memoria para ejecutarlo con fluidez [20].

Mencionada las diferencias principales entre ambos sistemas, se puede decir con certeza el propósito de los dos sistemas es crear el camino de comunicación entre el usuario y la máquina para ser controlado por el mismo usuario, y con una distinción entre características que los hacen diferentes y únicos. Por consiguiente, es correcto hacer referencia del TVS como un sistema que incorpora tecnología GUI en su Diseño de Software, como se indicó en el primer párrafo del Apartado 2.5.1. El Prototipo TVS cuenta con dos GUI para su funcionamiento, están desarrollados en la plataforma de Programación de LabVIEW versión 2009 para el Sistema Operativo de Windows 7 x86 Bits.



**Figura 2.5.1** Graphical User Interface - v1.6\_con\_angulos.

El primer GUI (Ver Figura 2.5.1) es identificado como “v1.6\_con\_angulos”, esta fue la primera interfaz desarrollada para el Prototipo, se utiliza únicamente como interfaz de rápido acceso para asegurar y verificar el correcto funcionamiento electrónico del Prototipo, principalmente se enfoca en la parte eléctrica de los motores a pasos.

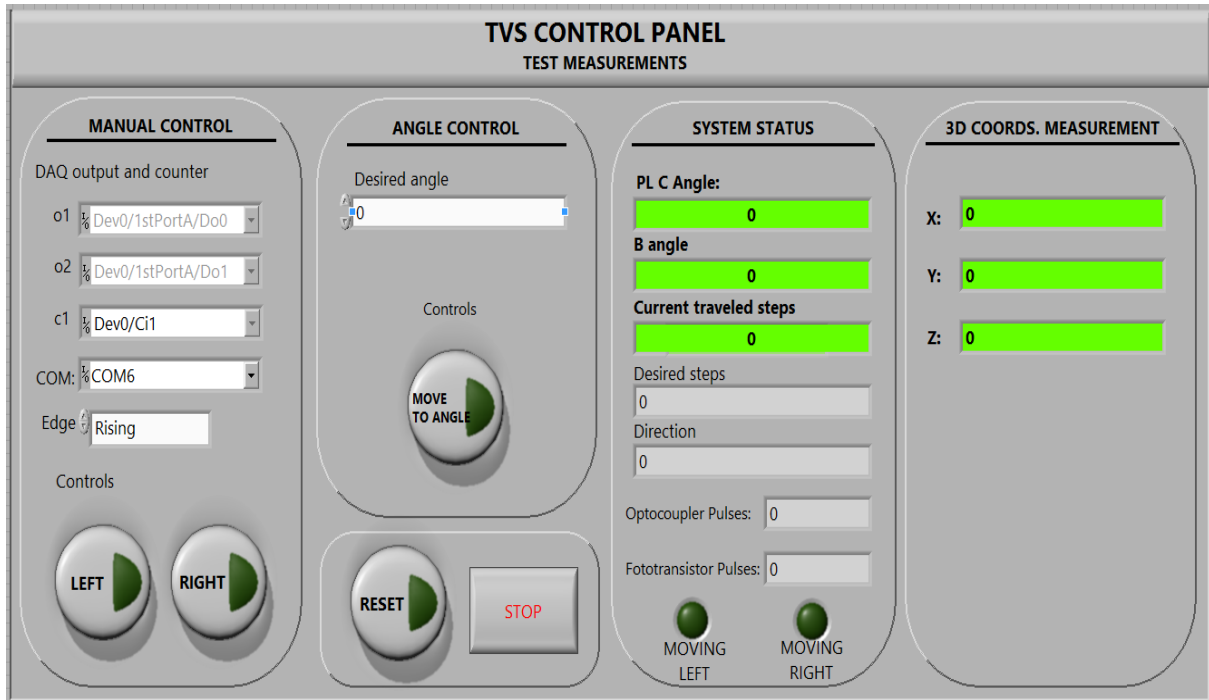
En la Figura 2.5.1 hay cuatro secciones de la interfaz (GUI), de las cuales solo funcionan dos, debido a que es una interfaz ligera, y su única función es asegurar el funcionamiento del Prototipo, a continuación solo se describe las dos secciones funcionales:

1. La Sección “MANUAL CONTROL” muestra 4 controles (parte superior) para acceder a la configuración de comunicación entre la tarjeta DAQ y la computadora, habilitando de tal forma las señales de las salidas digitales y el contador de salida. También hay dos botones que representan el sentido de giro (izquierda y derecha) para controlar la posición del puntero láser (Posicionador Láser).
2. La última sección está ubicada en la parte inferior de la interfaz, no tiene ninguna etiqueta, solo cuenta con dos botones, uno es para reiniciar los parámetros de comunicación de la tarjeta DAQ y el otro detiene el funcionamiento de la interfaz del Prototipo.

El segundo GUI es identificado como “v1.84”, esta versión fue el último desarrollo realizado por los creadores del Prototipo TVS, por lo tanto, se cataloga como mejora del primero e incluye toda las funciones y características que no tiene habilitadas la versión anterior. Las funciones de esta versión son varias y se describen a continuación:

1. Establecer comunicación entre GUI y el sistema electrónico del TVS.
2. Calibrar posición del puntero láser hacia una posición llamada HOME, esta posición es considerada como posición cero o inicial, y desde ese punto el sistema ejecuta sus movimientos o trayectorias para operar.
3. Monitorear en tiempo real los ángulos  $B_{ij}$  y  $C_{ij}$  de la triangulación dinámica del Sistema.
4. Obtener las mediciones de las coordenadas tridimensionales de cualquier objeto detectado dentro del escenario.
5. Monitorear en tiempo real los pulsos digitales del opto-acoplador y fototransistor con la finalidad de ver su comportamiento.

6. Monitorear los pasos de los motores a pasos y saber el sentido de giro actual del punto láser.



**Figura 2.5.2** Graphical User Interface - versión v.1.84.

Como se observa en la Figura 2.5.2, la interfaz cuenta con mayor cantidad de indicadores que sirven para proporcionar información del escaneo láser y obtener los resultados de las mediciones de las coordenadas. Las 4 secciones de la interfaz se describen a continuación:

1. “MANUAL CONTROL”. Su principales funciones son dos. (1) Establecer la comunicación entre la tarjeta DAQ y el sistema de Cómputo como lo hace la interfaz anterior descrita. (2) Dar el sentido de giro para mover la trayectoria del puntero láser a través de los botones “LEFT” y “RIGHT”. También hay dos diferencias entre la sección de esta interfaz y la mencionada con anterioridad, (1) en esta versión se incluye un control de COM para establecer el puerto USB de la computadora al que será conectado la tarjeta DAQ. (2) La otra diferencia es la funcionalidad de esta sección en la posición del Puntero Laser, dicha posición se busca configurar en una posición inicial llamada HOME.
2. “ANGLE CONTROL” muestra dos controles para capturar y posicionar el punto láser hacia un ángulo deseado por el usuario, el primer control etiquetado como “Desired Angle” ubicado en la parte superior, se encarga de leer dicho ángulo a través de un valor numérico para ser

interpretado por el sistema, el segundo control esta etiquetado como “Move to Angle” y se encarga de ejecutar y mover la posición hacia el ángulo deseado.

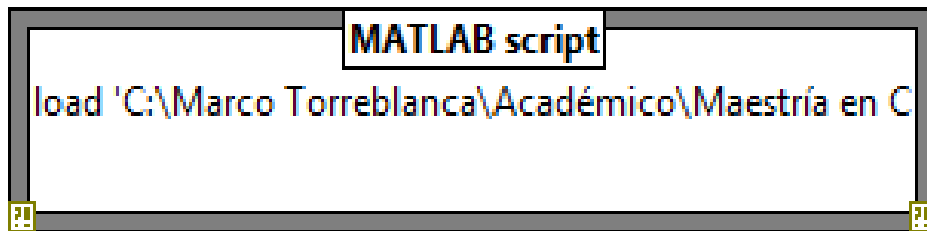
3. “SYSTEM STATUS”. Contiene nueve indicadores que despliegan en su pantalla información y lecturas importantes acerca del sistema. Los dos primeros indicadores de color verde (parte superior) son lecturas de ángulos del Sistema (ángulos  $B_{ij}$  y  $C_{ij}$ ), el tercer indicador esta etiquetado como “Current traveled steps”, despliega la lectura en tiempo real de los pasos generados para mover los motores a pasos. El indicador “Desired Step” muestra el valor configurado para enviar la cantidad de pasos necesarios para que los motores a pasos posicionen el ángulo deseado. “Direction” muestra de forma numérica el sentido de movimiento de los motores, representando con “0” el movimiento hacia la derecha y con “1” el movimiento hacia la izquierda. Los indicadores “Optocoupler Pulses” y “Fototransistor Pulses” muestran la cantidad de pulsos digitales que contiene cada componente electrónico. Los últimos dos indicadores, solo muestran visualmente el sentido de giro de la reflexión del láser, hacia dónde va dirigido, dependiendo del caso, se enciende el indicador “MOVING LEFT” o “MOVING RIGHT”.
4. “3D COORDS. MEASUREMENT”. Contiene la información más importante con respecto a las mediciones del sistema, despliega en sus tres indicadores las coordenadas tridimensionales (X, Y, Z) en donde se encuentra ubicado en el escenario el objeto escaneado.
5. La última sección no está etiquetada, está ubicada en la parte inferior de “ANGLE CONTROL”, solo tiene dos controles que funcionan tal y como se describió en la última sección de la primer interfaz “v1.6\_con\_angulos”.

La versión v1.84 tiene programado en su código un Método Computacional del área de Inteligencia Artificial (AI), el método es conocido como Red Neuronal [21], su objetivo es reducir de manera predictiva los errores que se presentan al momento de realizar las mediciones de las coordenadas tridimensionales, por consiguiente esta mejora ayuda al escáner tener una medición de longitud con mayor precisión y exactitud [2].

El tipo de red neuronal que usan es el *Feedforward Backpropagation* y el Algoritmo usado es el *Levenberg-Marquard*, dicha red neuronal está desarrollada en el software de programación de Matlab versión 2009 [22], esta código se configura de forma especial para ser corrido como un *Script* de Matlab. Un script es un tipo de programa que no tiene entradas ni salidas computacionales, son

usados como comandos que serán trabajados de forma repetida ejecutando una acción programada [23].

El *Script* de la red neuronal se incorpora en el código de la versión v1.84 para ser ejecutado cuando inicia la operación del sistema escáner láser. En la ventana de Diagrama de Bloques (ventana de programación) es donde se encuentra programado el Script, posicionado a partir de toda la configuración inicial del programa. La función utilizada LabVIEW es “MATLAB script” (Ver Figura 2.5.3), este script crea un vínculo entre el programa desarrollado en MATLAB y LabVIEW para proceder a ejecutarlo.

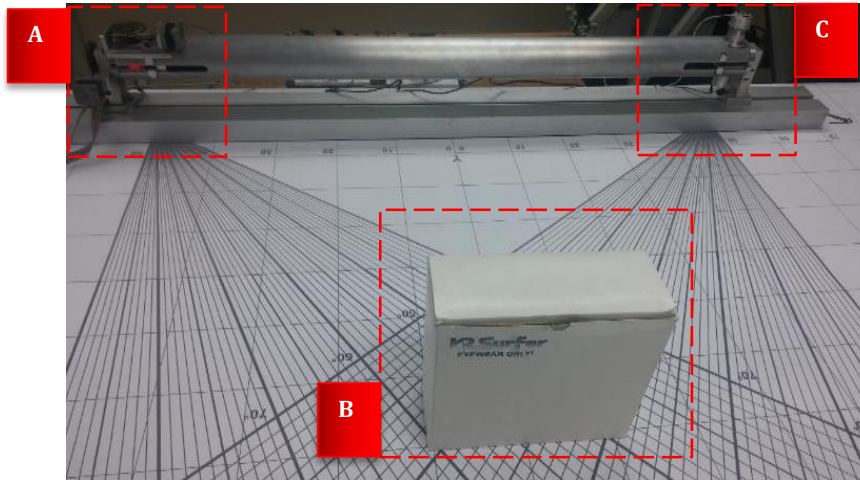


**Figura 2.5.3** MATLAB *Script* y su ruta para acceder a la Red Neuronal.

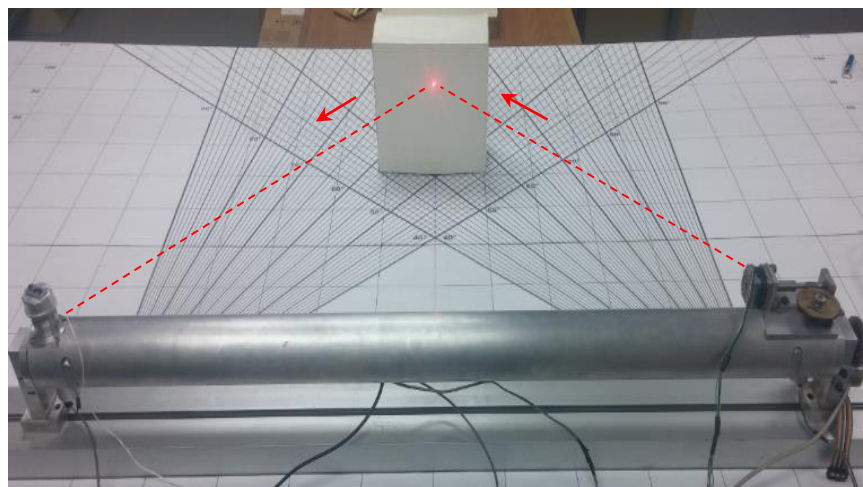
Se analizaron aspectos de programación en el código del programa desarrollado en el diagrama de bloques de la versión v.1.84, se encontraron algunas funciones o instrucciones repetidas, estas pueden originar un programa con ejecución lenta, también se encontraron más detalles que se describen en los siguientes apartados, por lo tanto, existe una gran oportunidad de mejorar el código en cuanto a programación e interfaz para su mejor ejecución.

### **2.5.3 DESCRIPCIÓN ELECTRÓNICA DEL PASO DE BARRIDO DEL TVS**

La parte del Prototipo tiene incorporado un Láser Escáner, consiste en tres secciones importantes para realizar su función, las cuales son: A) Posicionamiento de láser (PL), B) Objeto a mapear, C) Sensor de Apertura (SA, por sus siglas en inglés). En las Figuras 2.5.4 y 2.5.5 se muestran la vista frontal y trasera del Láser Escáner.



**Figura 2.5.4** Vista frontal del láser escáner.

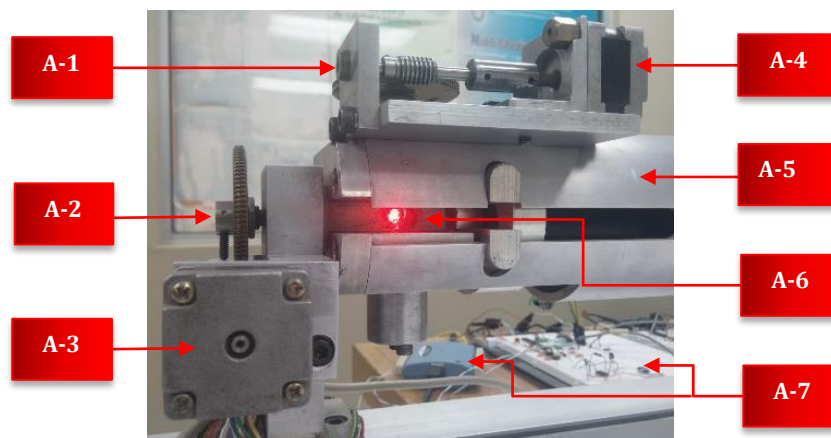


**Figura 2.5.5** Vista trasera del láser escáner.

En la sección “A” de la Figura 2.5.4 está ubicado el Posicionador Láser en donde se realiza el funcionamiento del paso de barrido, este ejecuta el movimiento angular de la reflexión de un rayo láser que va dirigido hacia un espejo óptico, viajando el rayo reflexionado a través del escenario con el objetivo de conocer la distancia de la superficie de cualquier objeto con respecto al láser escáner.

El paso de barrido inicia a partir del funcionamiento de la fuente de poder del láser de tipo gas (He-Ne) de 20mW con divergencia de 1mrad, posicionado internamente y horizontalmente en el centro del tubo de aluminio principal del prototipo. El láser emite un rayo o haz de luz hacia un espejo miniatura diseñado a 45° y posicionado verticalmente del lado izquierdo del tubo (viéndolo de enfrente), por consiguiente el espejo refleja el láser hacia el escenario del prototipo, como pausa a la descripción del paso de barrido, nos referimos a un escenario como el área de trabajo del escáner

láser, dentro están posicionado diferentes obstáculos con la finalidad de ser detectados eventualmente con la ayuda de mediciones adquiridas de coordenadas tridimensionales del obstáculo. Continuando con la descripción, el rayo láser va dirigido al escenario para ser proyectada en alguna superficie de un obstáculo (Ver Figura 2.5.5), una vez proyectada, se hace una reflexión (el láser tiene propiedades de reflexión) que sirve para ser captada por la sección de la Sensor de Apertura, donde internamente existe un espejo, lentes ópticos y un fototransistor que captura la reflexión. Este componte electrónico es sensible al espectro de luz láser. El espejo que refleja el láser, gira con ayuda de un juego de engranes movido por un motor a pasos, y dependiendo del tren de pulsos digitales generados del motor, es el ángulo en el que se posicionará el espejo para reflejar el rayo de luz, esta posición se ubica con referencia a los Eje “Y” y Eje “Z” del escenario. Cada posición del espejo se puede denominar paso, por lo tanto, barrido lo podemos describir como el conjunto de pasos donde viaja el láser reflejado por el espejo hacia el escenario. Cada paso representa un movimiento angular, en otras palabras, un paso de barrido tiene un ángulo. Como se mencionó, el espejo refleja el láser hacia el escenario para ser captado por un fototransistor, durante el trayecto del láser hacia la parte del fototransistor, se forma un triángulo con una duración de 4.04ms hasta 8.34ms [16]. Una vez formado el triángulo en ese pequeño intervalo de tiempo, se aplica el método de triangulación dinámica para conocer la distancia reflejada de la superficie de algún objeto (Ver Figura 2.5.5), de tal forma se obtiene la coordenada del Eje “X” que existe entre la distancia del escáner láser y el obstáculo detectado y medido.

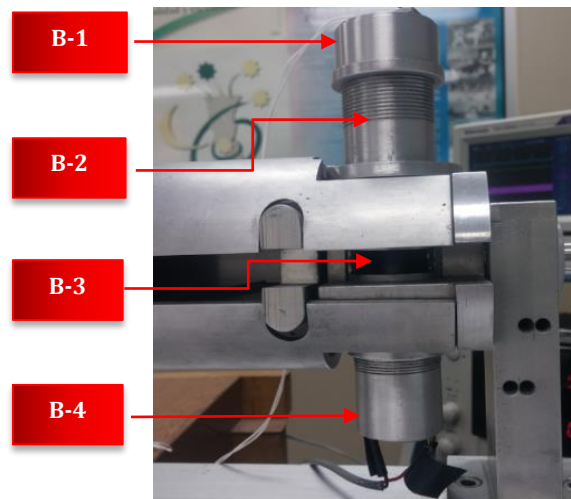


**Figura 2.5.6** Posicionador Láser (PL).

En la Figura 2.5.6 se observa la construcción mecánica y electrónica de la sección del Posicionamiento Láser donde se encuentra la operación del paso de barrido:

- A-1) Engrane mecánico encargado de mover el “Eje Y” de la coordenada tridimensional del escáner láser de la sección del Posicionador Laser.
- A-2) Engrane mecánico encargado de mover el “Eje Z” de la coordenada tridimensional del escáner láser de la sección del Posicionador Láser.
- A-3) Motor a pasos del “Eje Z”.
- A-4) Motor a pasos del “Eje Y”.
- A-5) Tubo de aluminio principal del escáner láser.
- A-6) Espejo óptico diseñado con un corte de 45° para realizar la reflexión del rayo láser hacia el escenario.
- A-7) Circuito electrónico y tarjeta DAQ del escáner láser.

Ahora, en la Figura 2.5.7 se observa la construcción mecánica y electrónica de la sección del Sensor de Apertura, donde se encuentra la detección del rayo láser:



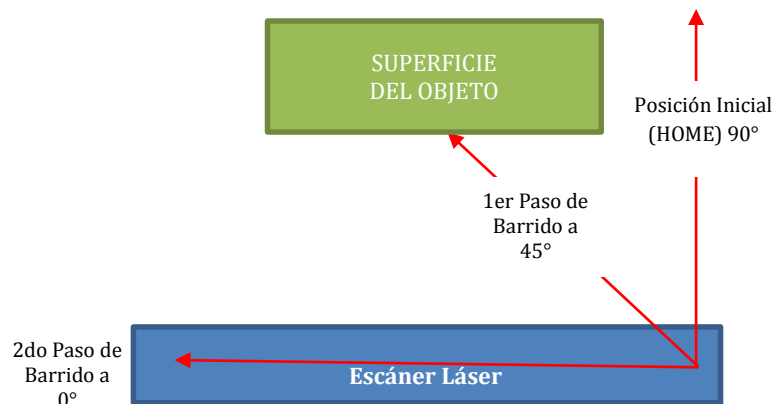
**Figura 2.5.7** Apertura de Escaneo (SA).

- B-1) Base de aluminio donde se encuentra internamente sujetado el componente electrónico llamado fototransistor.
- B-2) Internamente se encuentra montado un juego de lentes ópticas para incrementar la intensidad del rayo láser con la finalidad de mejorar la detección del fototransistor.
- B-3) Área de detección del Sensor de Apertura, donde se encuentra montado un espejo especial óptico diseñado con un corte de 45° para reflexionar el rayo de luz hacia el lente de la parte B-2.

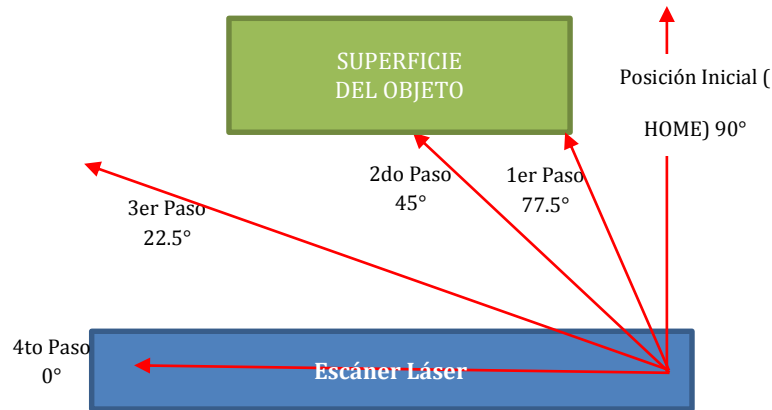
B-4) En esta parte se encuentra montado un motor DC junto con su opto-acoplador para leer las revoluciones del motor, dicho motor mueve el lente óptico que esta constantemente girando 360° de forma vertical, este lente como se mencionó está localizado en la parte B-3

#### **2.5.4 DESCRIPCIÓN FÍSICA DEL PASO DE BARRIDO**

El paso de barrido permite al escáner detectar cualquier superficie dentro de su área de trabajo, el principal objetivo es detectar la mayor cantidad de puntos posibles para describir la forma de una superficie en cualquier tipo de escenario. Como se mencionó en el apartado anterior, en este investigación, la palabra escenario se describe como, el entorno donde existe una superficie de cualquier tipo, ya sea plana, rigurosa o terrestre donde puede existir encima una “n” cantidad de cuerpos con diferentes formas (geométrica o irregular), colores, tamaños o pesos, las cuáles están disponible para ser exploradas por nuestro escaneo láser. Con esta descripción se puede afirmar que existen infinitos escenarios para inspeccionar o explorar. Al seleccionar el ángulo paso de barrido, prácticamente se está seleccionando la resolución del mapeo de puntos de una superficie a escanear. Dependiendo del ángulo del paso, será la distancia en la que se moverá y la cantidad de pasos de barridos que tendrá el barrido total. En las Figuras 2.5.0 y 2.5.1 se muestran dos imágenes muy parecidas para representar la comparación gráfica entre la selección de diferentes valores de ángulo para el paso de barrido del prototipo. Ambas figuras comparten en común una superficie de cualquier objeto, distancia entre el objeto y el láser escáner, referencia del ángulo (tomando la posición perpendicular como 90°). Y Las diferencias son, el ángulo de paso de barrido es de 45° y 22.5° respectivamente, la cantidad de pasos hechos en el paso de barrido total son 2 pasos y 4 pasos respectivamente, y los puntos reflejados de la superficie detectados por el sensor son 1 punto y 2 puntos, también respectivamente.



**Figura 2.5.0** Selección de ángulo de paso a 45°.



**Figura 2.5.1** Selección de paso de 22.5°.

Como se aprecia en la figura anterior, podemos decir, entre menor sea el ángulo de paso barrido mayor será su resolución de escaneo, pero como consecuencia se demorará más en terminar un barrido total por el incremento de la cantidad de pasos. Ahora bien, de caso contrario, entre mayor sea el ángulo de paso de barrido menor será su resolución de escaneo, y como consecuencia beneficia la rapidez del barrido total. Entonces, la selección del paso de barrido es un factor sumamente importante que se debe de mejorar.

## **2.6 ALGORITMO GENÉTICO**

Los Algoritmos Genéticos (AG) son métodos computacionales adaptativos que son utilizados para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de la Teoría de la Evolución postulado por Darwin en el año 1859. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos como comida, agua y refugio. Los individuos que tienen mayor éxito en sobrevivir y en atraer compañeros, cuentan con mayor probabilidad de generar un gran número de descendientes. Por el contrario individuos poco dotados producirán menos descendientes. Por consecuencia, los genes de los individuos mejores adaptados se heredaran a sucesivas generaciones hacia un número de individuos creciente, la combinación de buenas características provenientes de diferentes ancestros

puede a veces producir descendientes “super individuos”, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los Algoritmos Genéticos usan una analogía directa con el comportamiento natural, cuentan con una población de individuos, cada uno de ellos representa una solución viable a un problema dado. A cada individuo se le asigna un valor o puntuación relacionado con las características de dicha solución. En la naturaleza esto correspondería al grado de efectividad de un organismo para competir por unos determinados recursos, cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando de tal forma su material genético con otro individuo seleccionado con mismas o mejores características. Este cruce producirá nuevos individuos, los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

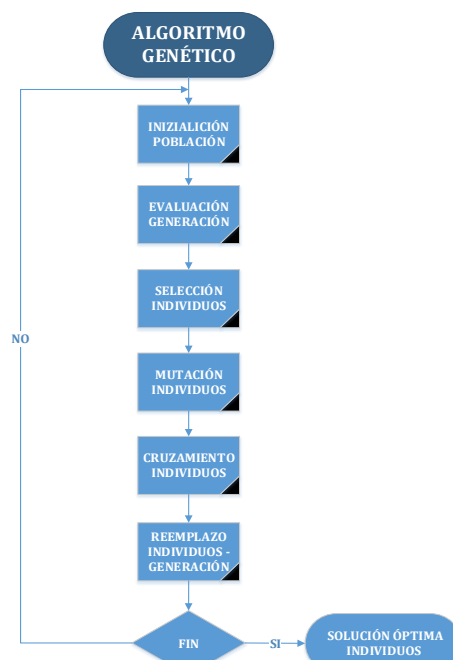
Algoritmos Genéticos es una técnica robusta [24], y pueden lograr con éxito obtener una solución óptima en una gran variedad de problemas provenientes de diferentes áreas [25], incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

### **2.6.1 ESTRUCTURA DEL ALGORITMO GENÉTICO**

Para que el Algoritmo Genético funcione se necesita una codificación o representación del problema, que resulte adecuado al mismo, con la finalidad de obtener una solución posible. Además se requiere una función de ajuste o adaptación al problema, la cual asigna un número real a cada posible solución codificada. Durante la ejecución del algoritmo, los padres deben ser seleccionados para la reproducción, a continuación dichos padres seleccionados se cruzaran generando dos hijos, sobre cada uno de los cuales actuará un operador de mutación. El resultado de la combinación de las anteriores funciones será un conjunto de individuos, los cuales en la evolución del Algoritmo

Genético formarán parte de la siguiente población. La estructura del Algoritmo Genético se divide en tres módulos generales:

1. **Módulo Evolutivo:** Mecanismo de decodificación (interpreta la información de un cromosoma) y función de evaluación (mide la calidad del cromosoma).
2. **Módulo Poblacional:** Tiene una representación población y técnica para manipularla (técnicas de representación, técnica de arranque, criterio de selección y de reemplazo). También se define el tamaño de la población y la condición de terminación.
3. **Módulo Reproductivo:** Contiene los operadores genéticos.



**Figura 2.6.0** Etapas del Algoritmo Genético.

En la Figura 2.6.0 se muestra las etapas del Algoritmo Genético, los módulos mencionados están asociados dentro de las etapas del Algoritmo Genético para interactuar y operar con las posibles soluciones. Las etapas del Algoritmo Genético son diferentes entre sí, su objetivo es ir sometiendo a una población de individuos a diferentes factores de cambio para conseguir una población de individuos óptimos en cada generación. Teniendo una vez identificado el problema y reconocido las posibles soluciones, se puede iniciar con las etapas del Algoritmo Genético para realizar los siguientes pasos que fueron postulados por Holland [24]:

1. **INICIALIZACIÓN.** Es la población inicial de soluciones candidatas, usualmente generada aleatoriamente a través del espacio de búsqueda. Sin embargo, el conocimiento específico de dominio o cualquier otra información se pueden incorporar fácilmente.
2. **EVALUACIÓN.** Una vez inicializada la población o una generación es creada, los valores aptitud de las soluciones candidatas son evaluadas.
3. **SELECCIÓN.** La selección asigna una o más copias de estas soluciones con mayor valor aptitud posible y por tanto impone a la supervivencia del más apto en los soluciones candidatas. La idea principal de la selección es preferir la mejor solución en vez de la peor, muchos procedimientos de selección han sido propuestos para complementar la idea, tales como “roulette-wheel” selección, estocástico universal selección, clasificación de selección y selección por torneo (en el apartado 2.6.2 se describe con más detalle).
4. **CRUZAMIENTO.** Combina partes de dos o más padres de soluciones para crear una nueva, posiblemente se puede generar hijos con mejores soluciones. Hay muchas maneras de lograr esto, el rendimiento de la competencia depende de un mecanismo de cruzamiento correctamente diseñado. La nueva generación bajo cruzamiento no será idéntica a cualquier padre en particular, y en su lugar se combinan rasgos parentales de una manera novedosa.
5. **MUTACIÓN.** Mientras que el cruzamiento opera en dos o más cromosomas de padres, la mutación localmente modifica una solución al azar. Hay demasiadas variaciones de mutación, pero usualmente esto implica uno o más cambios en los rasgos del individuo. En otras palabras, la mutación realiza un paseo aleatorio en la vecindad de una solución candidato.
6. **REEMPLAZO.** La población de nuevas generaciones es creada por selección, cruzamiento y mutación reemplazando el original de la población inicial. Los nuevos individuos o hijos que se dieron por el cruzamiento junto con su mutación, son conocidos como la nueva generación de la población, los mejores de dicha nueva población reemplazará a los peores de la antigua población.
7. **CONTINUAR O PARAR.** Repetir pasos 2 al 6 hasta que su condición sea satisfecha.

### **2.6.2 TÉCNICAS DE SELECCIÓN DEL ALGORITMO GENÉTICO**

El funcionamiento fundamental de un Algoritmo Genético es el proceso de selección de candidatos a reproducirse. Este proceso de selección suele realizarse de forma probabilística (los individuos menos aptos tienen una cierta oportunidad de seguir sobreviviendo), a diferencia de las estrategias evolutivas, en las que la selección es extintiva (los menos aptos tienen cero

probabilidades de sobrevivir). La probabilidad de elegir un determinado individuo es proporcional a su condición física, y puede ser considerado como un experimento aleatorio con la siguiente fórmula:

$$P[b_{j,t}] = \frac{f(b_{j,t})}{\sum_{k=1}^m f(b_{k,t})} \quad (2.6.0)$$

Por supuesto, esta fórmula sólo tiene sentido si todos los valores de la función aptitud son positivos. Si este no es el caso, una transformación no decreciente  $\varphi: \mathbb{R} \rightarrow \mathbb{R}^+$  debe ser aplicada (un cambio en el caso más simple). A continuación, las probabilidades pueden ser expresados como:

$$P[b_{j,t}] = \frac{\varphi(f(b_{j,t}))}{\sum_{k=1}^m \varphi(f(b_{k,t}))} \quad (2.6.1)$$

Las ecuaciones representan la forma aleatoria probabilísticas en cómo la población de individuos puede ser seleccionado dependiendo de su probabilidad y su condición física (función objetivo), una vez teniendo la selección se puede someter hacia las siguientes etapas del Algoritmo Genético. La formulación algorítmica del esquema de selección de la ecuación 2.6.0 y 2.6.1 se puede escribir de la siguiente manera:

```

*****
x := Random[0, 1]; i := 1;
WHILE i < m & x <  $\sum_{j=1}^i f(b_{j,t}) / \sum_{j=1}^m f(b_{j,t})$  DO
    i := i + 1;
    Seleccionar  $b_{j,t}$ 
*****
    
```

Las técnicas de selección usadas en algoritmos genéticos pueden clasificarse en tres grandes grupos [26]: (1) Selección proporcional, (2) Selección mediante torneo, y (3) Selección de estado uniforme.

### 2.6.3 TÉCNICAS DE CRUZAMIENTO DEL ALGORITMO GENÉTICO

Seleccionar una técnica de cruzamiento que se adapte al Algoritmo Genético para resolver cualquier problema del mundo real, significa poseer de forma rápida una cantidad de individuos con

mayor aptitud, considerando de tal forma estos individuos como mejores soluciones para resolver cualquier problema.

Esta técnica es el corazón del Algoritmo Genético debido a que en esta etapa puede existir una cruzamiento bueno o malo, dando como resultado generaciones con individuos aptos o generaciones con individuos no aptos, dependiendo de su parámetros de paro [27], la creación de individuos es muy importante, encontrando una técnica idónea que se adapte al problema de manera satisfactoria, tiene alto índice de probabilidad que existe un cruzamiento bueno para generar individuos aptos que sean la mejor opción. A continuación generación se presenta un la formulación algorítmica del cruzamiento básico.

```

*****
pos := Random{1, . . . , n - 1};
FOR i := 1 TO pos DO
    BEGIN
        Child1[i] := Parent1[i];
        Child2[i] := Parent2[i]
    END
FOR i := pos + 1 TO n DO
    BEGIN
        Child1[i] := Parent2[i];
        Child2[i] := Parent1[i]
    END
*****

```

En el Algoritmo Genético se postulan tres técnicas básicas de Cruzamiento [28], claro está que existe una cantidad mayor de estas técnicas, pero en su mayoría son variantes o mejores de estas [29] [30] Como se mencionó, la formulación algorítmica del cruzamiento es básica, por lo tanto, se puede ajustar a las siguientes tres técnicas:

1. *Cruzamiento de un punto:* El valor del individuo seleccionado se convierte a una cadena de valores binarios, este nuevo valor representativo sirve para combinar las propiedades entre individuos de forma más fácil, comúnmente esta cadena consta de 8 dígitos binarios y la probabilidad de intercambio de información es por lo general a la mitad de la cadena de bits,

por lo tanto, se puede apreciar la combinación de “gen” entre un individuo y otro, dando como resultado uno o dos descendientes, dependiendo de los parámetros de ajuste. El cruzamiento forzosamente debe constar de dos individuos llamados padres para la realización de este proceso como se muestra en la Figura 2.6.1.

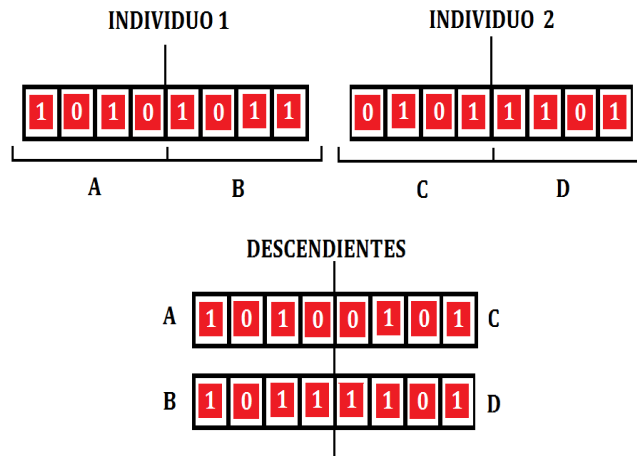


Figura 2.6.1 Ejemplo de Cruzamiento de un Punto o Simple.

2. *Cruzamiento de dos puntos.* Esta técnica es muy similar a la primera, la diferencia principal entre las dos es que ahora hay dos puntos de cruce en cada de las cadenas de bit de cada individuo, por lo tanto, puede existir una mayor cantidad de combinaciones entre sus genes, esta combinación, dependerá de la programación del Algoritmo Genético, existen distintas formas para realizarlo [30], a continuación se muestra en la Figura 2.6.2 un ejemplo del cruzamiento entre dos puntos.

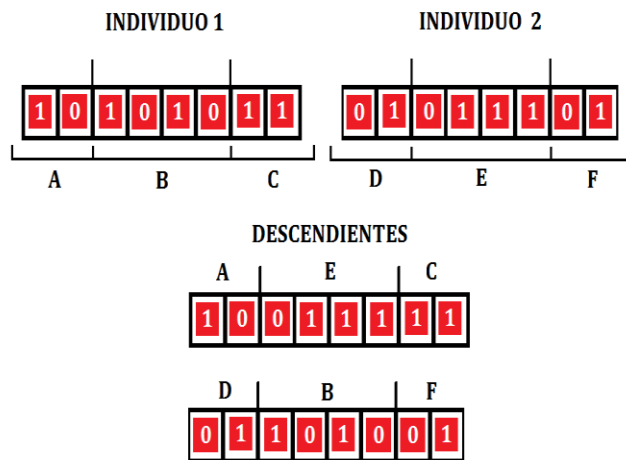


Figura 2.6.2 Ejemplo de Cruzamiento de dos Puntos.

3. *Cruzamiento uniforme.* Esta técnica fue propuesta originalmente por Ackley [31], aunque se le suele atribuir a Syswerda [32]. En este caso, se trata de una cruce de  $n$  puntos, pero en la cual el número de puntos de cruce no se fija previamente. El cruzamiento uniforme tiene un mayor efecto disruptivo que cualquiera de los 2 cruzamientos anteriores. A fin de evitar un efecto excesivamente disruptivo, suele usarse con  $P_c = 0.5$ . Algunos investigadores, sin embargo, sugieren usar valores más pequeños de  $P_c$  [33]. Cuando se usa  $P_c = 0.5$ , hay una alta probabilidad de que todo tipo de cadena binaria de longitud  $L$  sea generada como máscara de copiado de bits. En la Figura 2.6.3 se muestra un ejemplo para su mejor comprensión.

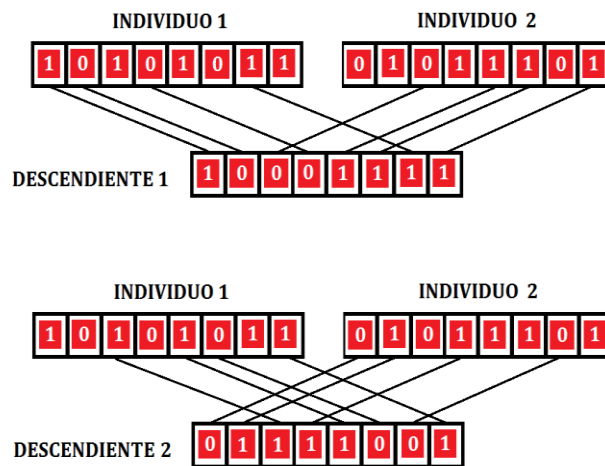


Figura 2.6.3 Ejemplo de Cruzamiento Uniforme.

#### 2.6.4 TÉCNICAS DE MUTACIÓN DEL ALGORITMO GENÉTICO

La Mutación se considera como un operador secundario en los Algoritmos Genéticos. Su uso es en ocasiones menos frecuente que el cruzamiento, en la práctica suelen recomendar porcentajes de mutación entre 0.001% y 0.01% para la representación binaria. Algunos investigadores han sugerido usar porcentajes altos de mutación al inicio de la búsqueda, y luego disminuirlos exponencialmente, de tal forma, favorece el desempeño de un Algoritmo Genético [34].

Otros autores simplemente sugieren ajustar la Probabilidad de Mutación a  $P_m = \frac{1}{L}$  (donde  $L$  es la longitud de la cadena cromosómica). A continuación se presenta la formulación algorítmica de Mutación.

\*\*\*\*\*

```

FOR  $i := 1$  TO  $n$  DO
  IF  $\text{Random}[0, 1] < \rho M$  THEN
    invert  $s[i]$ ;
    
```

\*\*\*\*\*

La mutación es considerada como el último ingrediente de factor de cambio de los individuos de una población en nuestro Algoritmo Genético, la probabilidad de mutación de un gen determinado es casi igual para todos los genes. Una de sus principales características es evitar el estado repetitivo o estancamiento de los individuos del Algoritmo Genético. Cuando una generación suele ser muy repetitiva en sus soluciones, pueden existir dos razones, haber encontrado una solución de máximo local o una solución de mínimo local. Es importante señalar que dependiendo de la construcción del Algoritmo Genético se pueden detectar de forma independiente estas dos posibles soluciones y poder utilizar el operador o la etapa de Mutación para salir del conjunto de soluciones parecidas. Las técnicas dependen solamente de la utilización y el valor asignado a la Probabilidad de Mutación, las cuales se postulan 3 de ellas consideradas como las más básicas de los Algoritmos Genéticos:

1. *La inversión de bits individuales:* Con  $P_m$  probabilidad, un bit es elegido al azar negado.
2. *Bit a bit de inversión:* Toda la secuencia se invierte poco a poco con probabilidad  $P_m$ .
3. *Selección aleatoria:* Con  $P_m$  probabilidad, la cadena está sustituido por uno elegido al azar.

### **2.6.5 TÉCNICAS DE REEMPLAZO DEL ALGORITMO GENÉTICO**

Una vez creadas las nuevas soluciones de la descendencia, el uso de cruce y mutación, hay que introducirlos en la siguiente población de padres, existe diferentes maneras para realizar este proceso. Se debe tener en cuenta los genes de los padres ya hayan sido seleccionados en función de su capacidad, de modo que esperamos los descendientes (que incluye a los padres, que no se sometieron de cruce) se encuentran entre los más aptos en la población y por lo que esperaríamos la población poco a poco, en promedio, aumente su aptitud. Algunos de las técnicas de reemplazo más comunes se describen a continuación.

1. *Eliminar todo:* Esta técnica elimina todos los miembros de la población actual y las reemplaza con el mismo número de cromosomas que se ha creado. Esta es probablemente la técnica más común y será la técnica de elección para la mayoría de la gente debido a su relativa facilidad

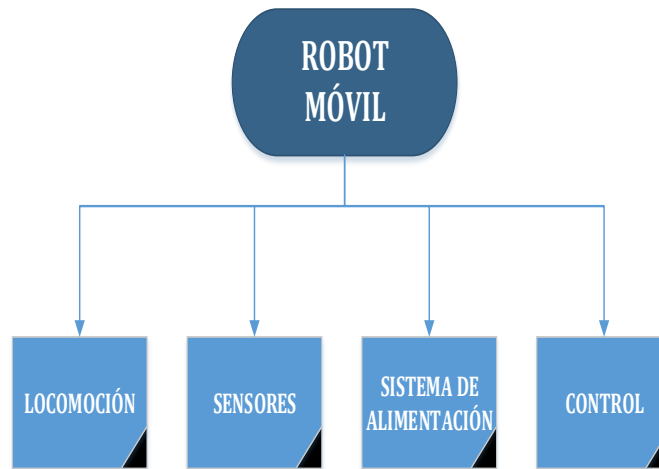
de implementación. También está libre de parámetros, el cual no es el caso para otros métodos o técnicas.

2. *Estado de equilibrio*: Esta técnica elimina  $n$  miembros antiguos y reemplaza con  $n$  nuevos miembros. El número para borrar y sustituir  $n$ , en un momento dado es un parámetro para esta técnica de eliminación. Otra consideración para la técnica es decidir qué miembros debe borrar de la población actual. ¿Cómo se borran los peores individuos, recogerlos al azar o eliminar los cromosomas que utilizó como padres? , por lo tanto, esto es un parámetro para esta técnica que se debe considerar durante la construcción del AG.
3. *Estado de equilibrio-no-duplicados*: Esta es la misma que la técnica de estado estacionario, pero el algoritmo comprueba que no hay cromosomas duplicados se agregan a la población. Esto se suma a la sobrecarga computacional pero puede significar que más del espacio de búsqueda se explora.

## CAPÍTULO 3 – POSTULACIÓN DE PROBLEMAS DEL PROTOTIPO TVS

### 3.1 ¿CÓMO MEJORAR EL PASO DE BARRIDO DEL LÁSER ESCÁNER DEL PROTOTIPO TVS?

Recordando con anterioridad lo expuesto en el apartado 1.1, el Prototipo TVS fue desarrollado con la finalidad de ser utilizado en la integración de un robot móvil capaz de realizar tareas relacionadas en el área de Navegación Autónoma, por consiguiente, es una meta para el equipo de Investigación del Optoelectrónica y Mediciones esto sea posible [35]. Sin duda alguna, el Prototipo TVS es un pilar importante para la integración de un robot móvil autónomo, pero su configuración actual tiene gran oportunidad de mejoras en la parte del Diseño Mecánico, Diseño Electrónico y Diseño de Software. Además, se necesitan desarrollar otros sistemas que complementen la integración de un robot móvil autónomo. La estructura básica de un robot móvil se presenta en el diagrama de la Figura 3.1.0, consta de 4 etapas llamadas, locomoción, sensores, sistema de alimentación y control.



**Figura 3.1.0** Estructura básica de un Robot Móvil.

Cada etapa realiza una tarea específica [36], por lo tanto, el sistema de escaneo láser del Prototipo TVS puede cubrir la etapa de Sensores para la creación de un robot móvil, debido a que el sistema es capaz de reconocer por medio de un láser la presencia de un obstáculo siempre y cuando este dentro de sus límites de trabajo de un escenario (Descrito en el apartado 2.5.3 y 2.5.4). En esta Investigación su principal objetivo es mejorar las condiciones de escaneo del sistema escáner láser por medio de un algoritmo que optimice el ángulo de paso de barrido. Mejorando dichas condiciones de escaneo podemos tener el Prototipo TVS mejor acondicionado en cuestión a velocidad de paso de

barrido y reconocimiento de la información de su entorno, del tal forma, mejorando estos aspectos se puede cubrir la etapa de Sensores para el desarrollo de un Robot Móvil Autónomo.

### **3.2 PROPUESTAS GENERALES PARA LA SOLUCIÓN DE PROBLEMAS DEL TVS**

A continuación se presenta las propuestas para mejorar y optimizar el sistema de escaneo del Prototipo TVS. Es importante mencionar a estudiantes de Posgrado están o estuvieron trabajando en la mejora de la parte del Diseño Electrónico y Diseño Mecánico del Prototipo TVS, por esta razón se mencionará alguna de sus propuestas con sus referencias. Las propuestas del diseño de Software que se mencionan a continuación, es trabajo realizado de esta investigación, debido a que se hizo énfasis a la parte del Diseño del Software como prioridad, las otras propuestas para el Diseño Electrónico o Diseño Mecánico solo se comentan como evidencia del análisis de las condiciones del Prototipo durante la etapa de funcionamiento del Prototipo.

#### 1. DISEÑO ELÉCTRÓNICO.

- a) *Instalar el circuito electrónico en Tarjetas PCB (Printed Circuit Board):* Consiste en mover todos los componentes electrónicos hacia una tarjeta impresa llamada PCB, las ventajas de esta tipo de tarjetas es contar con mayor estabilidad y evitar falsos contactos, como se tienen actualmente en el Prototipo.
- b) *Reemplazar motores a pasos por Servo-Motores:* Esta propuesta se está desarrollando por un estudiante de Doctorado de nuestro equipo de Investigación, la finalidad de este cambio es tener un mejor control en la posición del rayo láser, aumentando su rapidez y precisión en su movimiento puede mejorar electrónicamente el sistema de paso barrido láser [37] [38].
- c) *Mejorar la sección del Sensor de Apertura:* Está sección cuenta con un circuito electrónico para capturar los pulsos digitales del Fototransistor detectados por la reflexión del rayo láser de la sección del Posicionador Láser. Dicho circuito es muy sensible, por lo tanto, la calibración para ajustar la detección de los pulsos es muy complejo y a su vez demora demasiado tiempo lograr la calibración, la propuesta es mejorar la parte electrónica para disminuir la complejidad y el tiempo de calibración por la sensibilidad del circuito, aparte reemplazar el actual fototransistor por otro con mayor características y rendimiento en la detección del rayo láser del Prototipo TVS.

2. DISEÑO MECÁNICO.

- a) *Integración de transmisión para el juego Engranés:* Consiste en un diseño de transmisión para controlar diferentes engranes para modificar en diferentes ángulos el paso de barrido láser, esta propuesta fue realizada por la Dra. Xóchitl García [39].

3. DISEÑO SOFTWARE.

- a) *Depuración de código del TVS:* En el actual código contiene instrucciones y funciones repetitivas, programación espagueti [40], entre otros detalles que afectan las condiciones de ejecución de la interfaz gráfica del TVS. Contar con un programa codificado bajo estas condiciones tiene como consecuencia en perjudicar la ejecución del programa, el tiempo de respuesta de cada instrucción de programación puede ser más lenta de lo normal, y en el peor de los casos se puede llegar a “colgar” (Término usado cuando un programa deja de responder e inhabilita toda función) el programa por algún evento que no reconoció a tiempo en sus líneas de programación. La propuesta es depurar dicho código removiendo instrucciones y funciones repetitivas, agregar funciones para disminuir grandes partes de código, y usar subprogramas para no repetir código, todo esto con el fin de experimentar una interfaz gráfica con mayor estabilidad, rapidez y sencillez a la hora de ejecutar las instrucciones y procesar la adquisición de datos, de tal modo, se cumple un detalle de Software que sirve para mejorar la velocidad del sistema de escaneo láser del TVS.
- b) *Integración de Método de Búsqueda llamado Algoritmo Genético:* Consiste en agregar el Algoritmo Genético en la interfaz gráfica del TVS, como se mencionó en el apartado 2.6, es un método de búsqueda para encontrar la solución óptima de un problema, por lo tanto, nuestro objetivo del uso de este método es encontrar un ángulo óptimo considerado como la solución óptima, el Algoritmo genético debe ser capaz de generar un ángulo óptimo de paso de barrido láser en tiempo real, es decir, tener un ángulo óptimo en cada escenario que el Prototipo se encuentre escaneando.

### **3.3 PROBLEMAS ACTUALES DEL DISEÑO DE SOFTWARE DEL TVS**

Recordando el objetivo principal del trabajo de esta investigación, es introducir el Algoritmo Genético en la programación de la Interfaz Gráfica del Prototipo TVS para optimizar el ángulo de paso de barrido del escaneo láser, este objetivo hace énfasis a las propuestas de mejora del Diseño de

Software que fueron mencionadas anteriormente, por lo tanto, a continuación se hace mención de los problemas presentados durante el desarrollo de esta investigación.

1. El programa de la Interfaz Gráfica de Usuario del TVS no funcionaba cuando el puerto de comunicación serial RS-232 se colgaba, comúnmente este problema se hacía presente cuando el funcionamiento del TVS era mayor a 30 minutos seguidos.
2. Las instrucciones que se activaban en el programa del TVS proporcionaban una sensación de retraso en el tiempo de respuesta de dicha instrucción, por lo general, el retraso podía durar de 1 a 2 segundos en realizar la operación de la instrucción después de ser solicitada, en algunos casos el programa se llegaba a congelar en el momento menos inesperado.
3. Cuando se activaba una función del programa, la instrucción que se deseaba no era la correcta, el programa realizaba otra instrucción diferente a la solicitada o simplemente se colgaba programa.
4. El programa del TVS carece de un barrido automático para la detección de objetos, su funcionamiento es de forma manual, por lo tanto, al momento de realizar pruebas experimentales del Algoritmo Genético, sería muy difícil realizarlas con las condiciones actuales del sistema.
5. El desarrollo del Algoritmo Genético es un problema, debido a que se debe de encontrar s combinación de sus técnicas (mencionados en el apartado 2.6.2) para adaptarlo a nuestro problema de Tesis.
6. La fabricación de diferentes tipos de escenarios es un problema, crear escenarios suficientemente diferentes con respecto a la posición, forma, tamaño y cantidad de objetos de cada escenario creado, puede llegar a demorar tiempo y ser costoso. En pocas palabras, el problema se traduce en tiempo y recursos, los cuales ambos estuvieron muy limitados por las adversidades que se presentaron en el transcurso de esta investigación. Realizando esto, es con la intención de mejorar el funcionamiento y compartimento del Algoritmo Genético en la Etapa de Pruebas Experimentales, por lo tanto, es algo importante que se tenía que hacer.
7. Las actuales condiciones electrónicas de los motores a pasos encargados del funcionamiento del paso de barrido no cuentan con una velocidad aceptable para la realizar las pruebas experimentales de forma rápida, y mucho menos automatizada de los diferentes escenarios. Debido a la necesidad de crear una gran cantidad de escenarios en la Etapa de Pruebas Experimental, es forzoso contar con mayor rapidez en la detección de los escenarios para agilizar el tiempo de duración del entrenamiento del Algoritmo Genético, es decir, si usamos

el actual Prototipo para mejorar nuestro algoritmo Genético con las pruebas mencionadas, su actual rendimiento podría llevar demasiado tiempo en concluir dicha etapa.

### **3.4 PROPUESTAS PARA EL DISEÑO DE SOFTWARE DEL TVS**

Se trabajó analizando los problemas que se describieron anteriormente, con el fin de presentar posibles soluciones para resolver o reducir dichos problemas, y poder continuar con la finalización de este trabajo de Investigación de Tesis. A continuación, se presentan las tres soluciones que abarcan la solución de los problemas del apartado 3.3.

1. *Algoritmo Genético Robusto*: El desarrollo de un Algoritmo Genético Robusto sirve para realizar pruebas a los diferentes métodos expuestos con anterioridad y encontrar la combinación de técnicas que se acoplen a nuestro problema de tesis, y por consiguiente adaptarlo a nuestro programa de la interfaz gráfica mejorada, resolviendo de esta manera el punto número 5 de la lista anterior.
2. *Mejora de la Interfaz Gráfica de Usuario del TVS*: La mejora consiste en depurar todo el código, cambiar la interfaz gráfica por una interfaz más amigable para el usuario, y agregar funciones automatizadas que beneficien al sistema de escaneo láser para el programa de la versión v1.84. Haciendo estas modificaciones al programa, se puede corregir los problemas presentados en los puntos 1, 2 y 3 del apartado 3.3
3. *Simulador de Escenario Virtual*: La creación de un Simulador de escenario, donde virtualmente asemeja las características de un escenario físico real, es una buena opción para evaluar y mejorar nuestro Algoritmo Genético desarrollado e integrado al programa de la interfaz gráfica mejorada del Prototipo TVS, claro está, tener un simulador que genere los escenarios de forma virtual tiene sus limitaciones, pero es prescindible contar con escenarios que se generan de forma rápida y aleatoria para resolver los problemas expuestos de los puntos 6 y 7.

## CAPÍTULO 4 – DESARROLLO DE HERRAMIENTAS COMPUTACIONALES PARA MEJORAR RECONOCIMIENTO DE ENTORNO

En esta Capítulo se describe el desarrollo de las propuestas postuladas del Apartado 3.4, su desarrollo consta en mejorar las condiciones del paso de barrido del escaneo láser del Prototipo TVS para realizar un escaneo o barrido de forma precisa, rápida y automatizada, asumiendo estas características optimizada puede cubrir la Etapa de Sensores que integra un Robot Móvil Autónomo para realizar tareas del área de Navegación Autónoma.

Las tres propuestas fueron desarrolladas en la Plataforma de LabVIEW 2013 con la finalidad de realizar un correcto acoplamiento e integración de las condiciones actuales del Prototipo TVS. Cada propuesta tiene un programa independiente para evaluarlo y usarlo de diferente forma, recordando, la primer propuesta es el desarrollo de un programa de Algoritmo Genético Robusto, dicho programa incluye una gran cantidad de técnicas (Ver apartado 2.6) y uno de sus de sus principales objetivos es poder realizar el análisis, evaluación y experimentación de las técnicas para obtener una mejor comprensión y conocimiento de ¿cómo? y ¿cuál? técnica utilizar para nuestro problema de tesis, por lo tanto, realizando este programa se puede tener un Algoritmo Genético flexible y configurable para ser utilizado en cualquier escenario.

La segunda propuesta consiste en mejorar las condiciones de programación e incluir funciones a la interfaz gráfica del Prototipo TVS para tener un mejor comportamiento en el escaneo láser con las nuevas instrucciones programadas para desenvolver tareas de reconocimiento de escenarios o simplemente tareas relacionadas en el área de Navegación Autónoma.

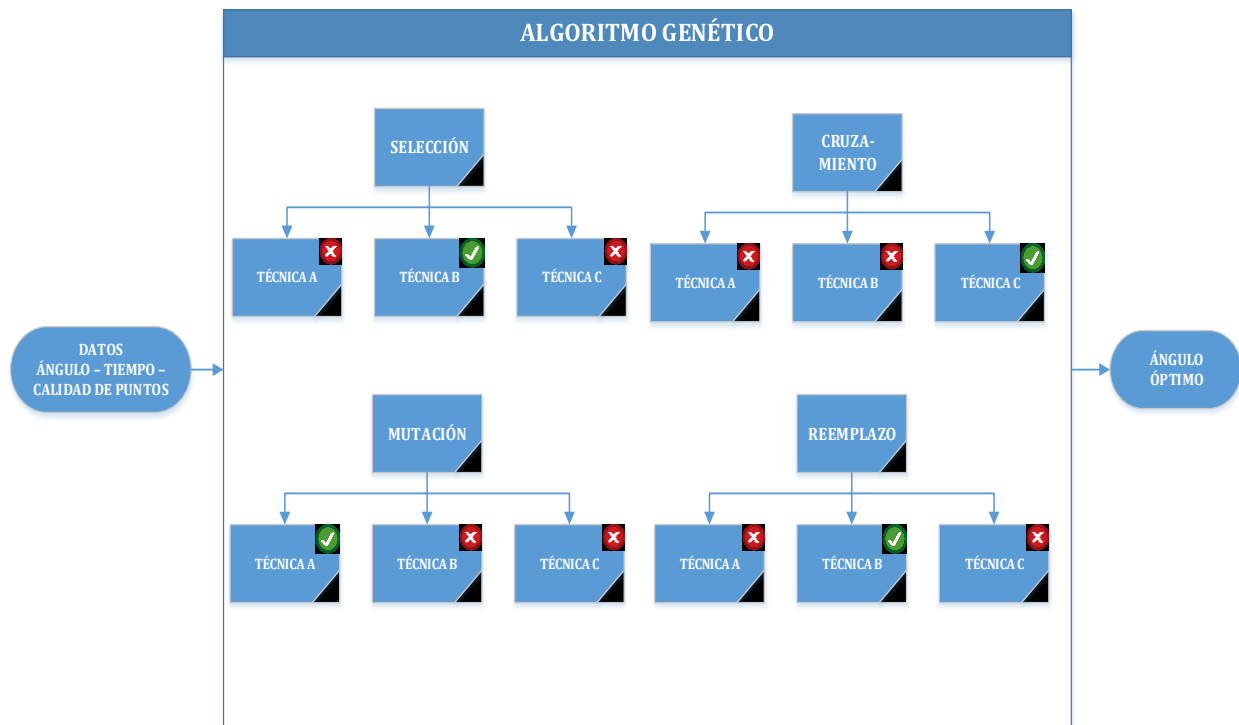
La última propuesta es el desarrollo de un Simulador de Escenarios para analizar y evaluar el comportamiento y funcionalidad del Algoritmo Genético dentro de un escenario virtual que se asemeje a un escenario físico. Cada una de las propuestas aporta cierto porcentaje en mejorar y optimizar el ángulo de paso de barrido.

### **4.1 ALGORITMO GENÉTICO ROBUSTO**

La formulación algorítmica del Método de Búsqueda de Algoritmo Genético se puede llevar acabo en cualquier Lenguaje de Programación (Lenguaje C, Lenguaje R, Lenguaje G, etc.), dicha

formulación es universal para cualquier tipo de lenguaje de programación. Como se mencionó, su codificación del Algoritmo Genético se hizo en la plataforma de LabVIEW para conservar la compatibilidad de la Interfaz Gráfica y la parte electrónica del Prototipo TVS.

Las diferentes técnicas conocidas se van a evaluar en este programa para poder seleccionar la más adecuada con nuestro problema de tesis, esta combinación de técnicas dará como resultado obtener una mayor cantidad de sobresalientes soluciones posibles.



**Figura 4.1.0** Representación del Programa de Algoritmo Genético.

La funcionalidad del programa se puede representar con la Figura 4.1.0, se espera tener una cantidad de datos en la entrada del programa, en nuestro caso, estos datos a considerar son, el valor de un ángulo, el tiempo que toma en realizar un barrido a un escenario y cantidad de puntos detectados en el barrido por el escáner láser. Dentro de este programa se encuentra la estructura básica del Algoritmo Genético con sus diferentes técnicas y parámetros de ajustes [32] [33] [34].

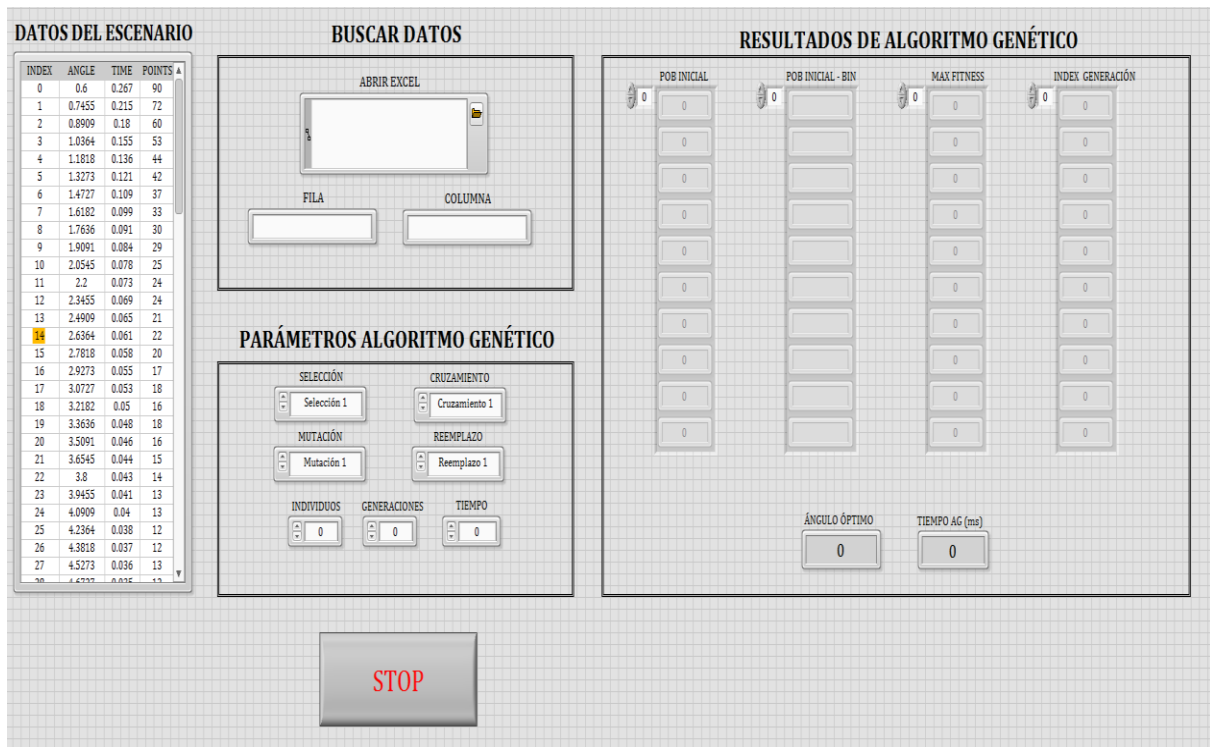
Como se observa, cada etapa del Algoritmo Genético contiene tres técnicas, se muestra así solo como ejemplo para representar a las diferentes técnicas que existen en la construcción del Algoritmo Genético, los iconos verdes y rojos representan un ejemplo de selección de las técnicas por

etapa. Por lo tanto, teniendo estas tres técnicas por etapa incorporadas en el programa se podrá seleccionar la mejor técnica de cada etapa. El resultado o la salida de este programa es proveer la solución óptima, es decir, proveerá ángulos óptimos para ser utilizado en nuestro paso de barrido.

El Algoritmo Genético empieza con una población inicial, recordando existen individuos catalogados como individuos aptos e individuos no aptos. Los individuos aptos son aquellos que se distinguen por ser superiores a los demás conforme a las características establecidas por el diseñador del Algoritmo Genético y los individuos no aptos son los que se distinguen por ser inferiores.

Los individuos de nuestro problema de tesis, están formados por diferentes valores de ángulos (ejemplo:  $\theta_1 = 0.5^\circ$ ,  $\theta_2 = 2^\circ$  hasta  $\theta_n = n^\circ$ ; en caso de mi simulación el valor de  $n$  es igual a  $10^\circ$  como límite, pero en otro caso se puede considerar cualquier otro límite), dichos ángulos simbolizan el ángulo de paso de barrido del láser escáner, donde se desea un ángulo óptimo en cualquier escenario expuesto al láser escáner. Cuando mencionamos la palabra óptimo, nos referimos a la búsqueda del mejor ángulo que interactúe con el escenario.

#### 4.1.1 DESCRIPCIÓN DEL PANEL FRONTAL DEL ALGORITMO GENÉTICO ROBUSTO



**Figura 4.1.1** Panel Frontal del Algoritmo Genético Robusto realizado en la Plataforma de LabVIEW 2013.

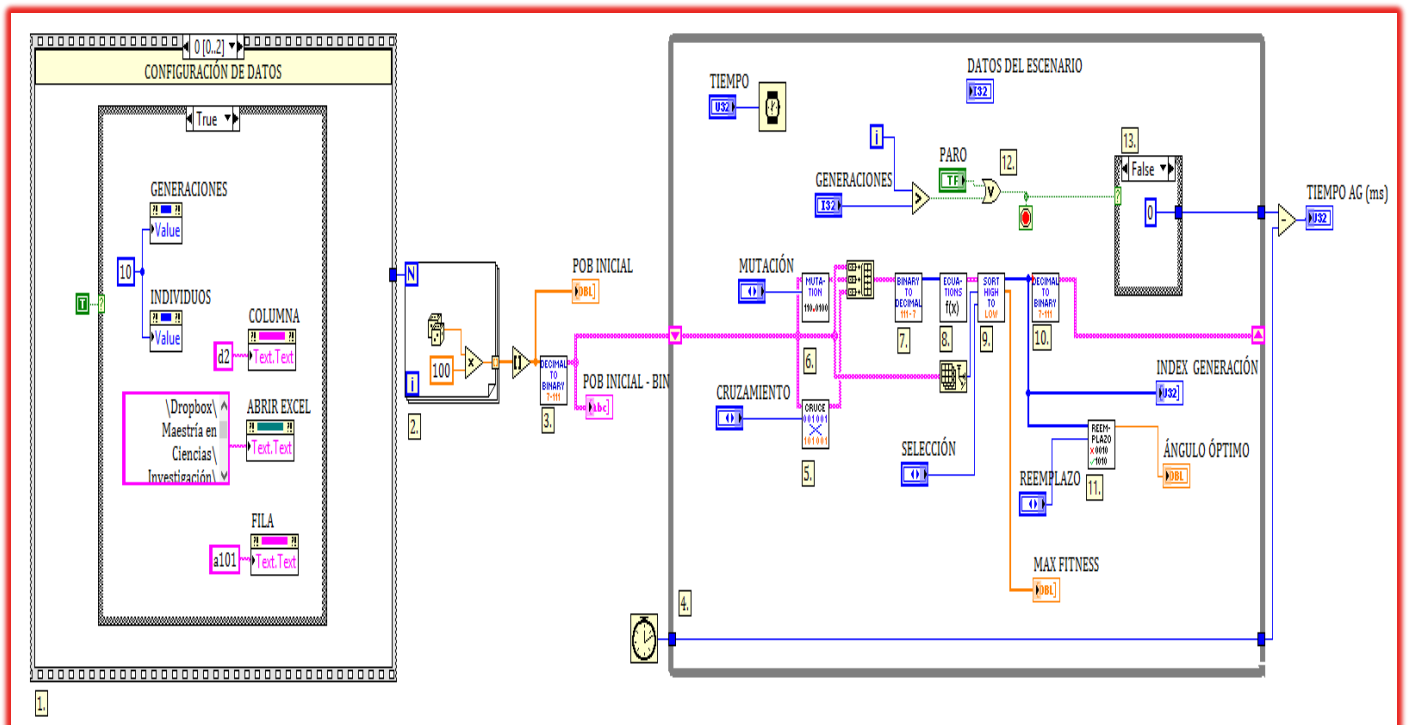
En la Figura 4.1.1 se muestra VI del panel frontal del Algoritmo Genético, diseñado en LabVIEW 2013 específicamente para lograr uno de los objetivos de la presente tesis. Este programa contiene en la parte izquierda los controles para seleccionar los datos a optimizar y los controles para configurar el Algoritmo Genético. En la parte derecha se encuentran los indicadores con información relevante del funcionamiento del programa, aparte muestra el ángulo óptimo obtenido a partir de los datos seleccionados y trabajados. Los controles e indicadores descritos con anterioridad esta agrupados en cuatro secciones, su distribución y descripción de cada sección es la siguiente:

1. *DATOS DEL ESCENARIO*: Despliega una tabla de datos que serán evaluados por el programa del Algoritmo Genético. Esta tabla consta de una columna para guardar el index (Término usado en programación para referir a la posición de un dato ubicado dentro una tabla) de la posición de cada dato, y otras tres columnas de datos, las cuales son, el valor de ángulo, el tiempo y la cantidad de puntos detectados.
2. *BUSCAR DATOS*: Contiene un control para buscar dentro de la memoria física de la computadora los datos que se evaluarán, el reconocimiento de estos datos es a partir de una hoja de Excel, por lo tanto, los datos deben estar contenidos en una hoja de Excel para que el programa pueda reconocer la información y poder continuar con su función. Los otros dos controles sirven para ubicar la posición de los datos en la hoja de Excel (Fila y Columna), el usuario debe de conocer primeramente la posición de los datos en la hoja de Excel, en caso contrario, es posible que el programa no responda de manera adecuada y termine marcando algún error.
3. *PARÁMETROS ALGORITMO GENÉTICO*: Este sección contiene la configuración y los parámetros del Algoritmo Genético. Hay cuatro controles para seleccionar las diferentes técnicas de cada etapa descrita con anterioridad. También cuenta con tres controles para ajustar la cantidad de individuos, cantidad de generaciones y el tiempo de procesamiento entre cada generación.
4. *RESULTADOS DEL ALGORITMO GENÉTICO*: Despliega cuatro listas y dos casillas de indicadores, las cuatro listas que están ubicadas en la parte superior tienen como función mostrar el procesamiento de los datos durante el funcionamiento del programa, por lo tanto, se puede obtener una mayor comprensión del Algoritmo Genético, debido a que muestra cómo va interactuando los datos cargados en cada una de sus etapas. Las dos casillas que están en la parte inferior despliega la información de la solución óptima del ángulo y el tiempo en terminar esa corrida en particular.

La sección 3 es muy importante para el programa, debido a que su configuración marca la pauta entre el comportamiento y duración del Algoritmo Genético. Por lo tanto, se desea que la duración en realizar el procesamiento de datos para obtener una solución óptima, sea en un corto periodo de tiempo. Esto es prescindible para cumplir funciones de tareas de Navegación que demanden velocidad en el reconocimiento de escenarios. La cantidad de individuos es el valor de individuos que contiene cada generación, entre mayor sea la cantidad de individuos, mayor será la información que se procesará dentro del programa, por consiguiente mayor será el tiempo que demore en obtener las soluciones óptimas.

El número de generaciones es otro factor a considerar, funciona de igual manera que los individuos, las generaciones entre mayor sea su cantidad, mayor será el tiempo de procesamiento. Un Algoritmo Genético con un valor de generaciones alto, puede tener sus ventajas y desventajas [41] El Algoritmo Genético se debe ajustar de tal forma que la cantidad de individuos y de generaciones no perjudiquen la duración del procesamiento, pero haciendo énfasis en darle prioridad en obtener soluciones con mayor aptitud.

**4.1.2 PSEUDOCÓDIGO DEL ALGORITMO GENÉTICO ROBUSTO**



**Figura 4.1.2** Pseudocódigo del Algoritmo Genético Robusto.

En el Diagrama de Bloque permite realizar la programación del pseudocódigo del Algoritmo Genético Robusto, la estructura de este pseudocódigo está basada en gran parte en formulación algorítmica postulada en los apartados 2.6.2, 2.6.3, 2.6.4 y 2.6.5. El pseudocódigo se divide en 13 partes con el fin de describir de forma entendible y concisa cada una de sus funciones e instrucciones programadas.

En la Figura 4.1.2 se muestra el pseudocódigo general del Algoritmo Genético Robusto, cada etapa del Algoritmo Genético está constituida por subprogramas llamados SubVI, dentro de ellos se encuentran las técnicas de cada etapa conocidas. A continuación se describen los 13 partes del pseudocódigo:

1. Inicialización de Adquisición de Datos a partir de una hoja de Cálculo en Excel, esta instrucción consta de una función principal llamada “*Stacked Sequence*”, donde se puede programar una serie de instrucciones para ejecutarse de manera secuencial, en esta función se programaron tres secuencias diferentes tal y como se ilustra en la Figura 4.1.3-A.

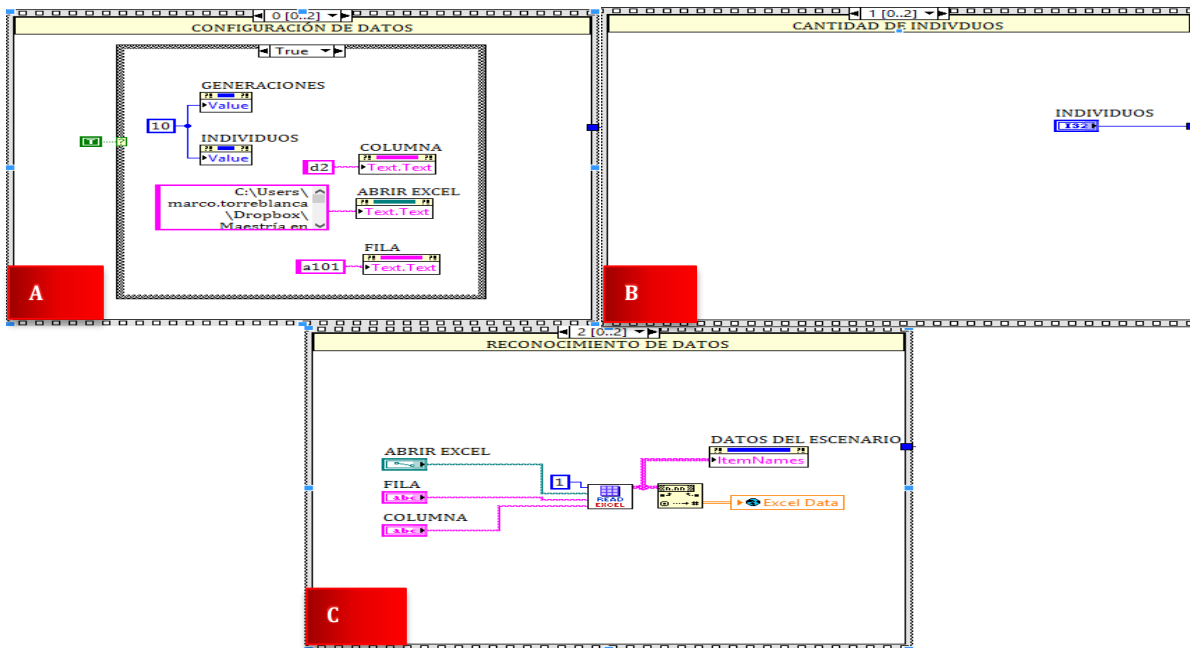


Figura 4.1.3 Función “*Stacked Sequence*” para la Adquisición de Datos.

(A) La primera secuencia realiza la búsqueda de los datos en la computadora. También contiene una función llamada “*Case Structure*”, dicha función se utiliza para dividir entre una búsqueda manual y una búsqueda automatizada. La búsqueda manual hace uso de los

controles que están en el Panel Frontal de la sección de “BUSCA DE DATOS” para localizar el archivo en Excel. Mientras, la búsqueda automatizada se deja las instrucciones programadas para que el programa sea capaz de buscar siempre en la misma dirección, al menos que se programe una dirección diferente. (B) La otra secuencia solo muestra el control de Individuos. (C) Contiene un SubVI desarrollado para poder establecer la comunicación entre LabVIEW y Excel, con el fin de adquirir los datos de la hoja de Cálculo, es decir, los datos son extraídos y enviados a LabVIEW, se programa un indicador con el nombre de “Excel Data” que permite guardar los datos y poder hacer uso de estos datos después. Esta instrucción solo se ejecuta una solo una vez.

- La función “For” contiene una instrucción que genera de forma aleatoria el valor de cada individuo, la cantidad de individuos formados va a depender del valor de la casilla de control “Individuos” (ubicado en el Panel Frontal de la sección “PARÁMETROS ALGORITMO GENÉTICO”), dicha cantidad formará la Población Inicial ( $P_i$ ). Los individuos tienen la posibilidad de tener un valor entre 0.1 a 10, el valor representa el ángulo del paso de barrido, por lo tanto, los valores generados para los individuos son en grados ( $^{\circ}$ ). La función “For” va a estar iterando “n” veces hasta formar los individuos deseados, por consiguiente al terminar su operación, se formará la Población para después ser almacenada en un Arreglo de tipo numérico con la ayuda del Indicador “Excel Data”. En esta sección se realiza la primer etapa de Selección de Individuos, inicialmente en este caso es de forma aleatoria.

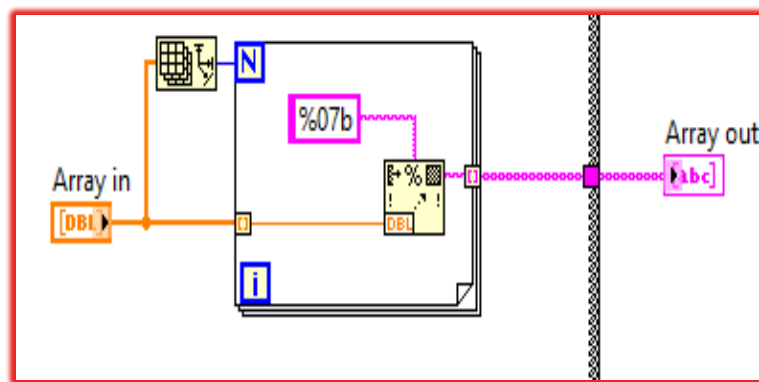
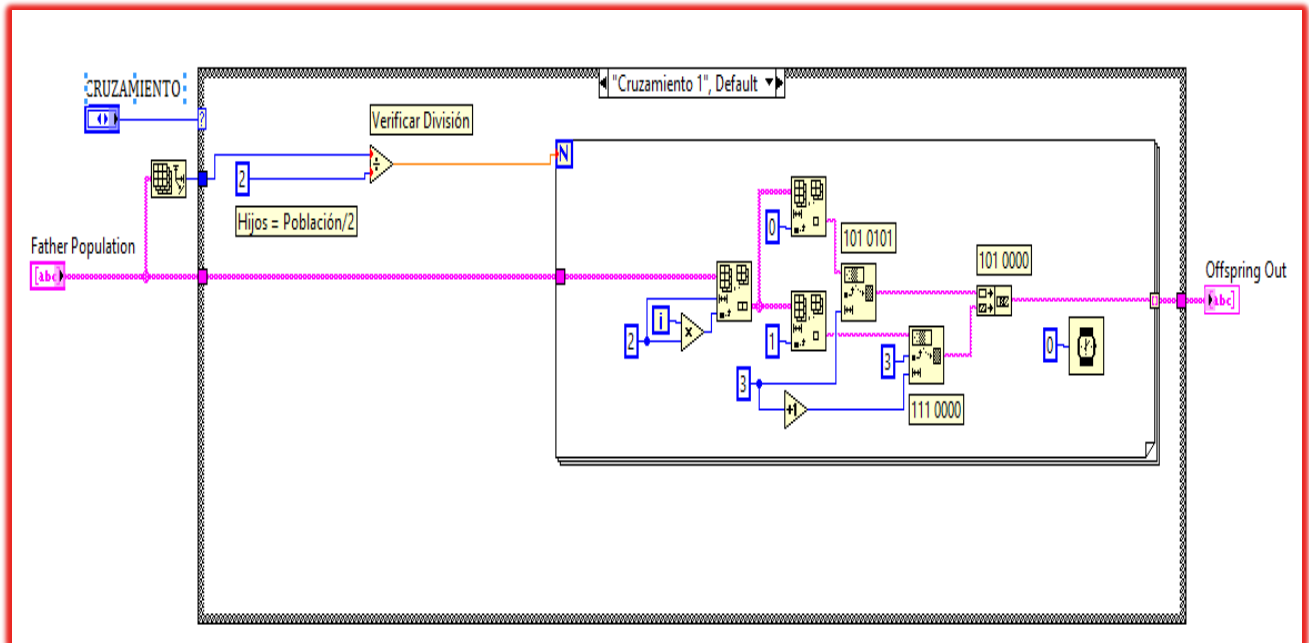


Figura 4.1.4 SubVI para convertir los datos tipo numéricos a tipo cadena.

- Esta parte contiene un SubVI que se complementa con la etapa de Selección realizada por la segunda parte, dicho SubVI contiene una entrada y una salida de información de datos. En su entrada tiene un control para recibir los datos de la población de individuos que están

almacenados en el arreglo tipo numérico. En salida del SubVI, regresa los individuos ya convertidos en cadenas de BITS (Como se comentó en los apartados 2.6). Dentro del SubVI se encuentra la programación necesaria para realizar dicha conversión de un dato tipo numérico a un dato tipo Cadena. En la Figura 4.1.4 se muestra la programación que hace posible lo mencionado.

4. Las partes del 1 al 3 solo se ejecutan una sola vez, a partir de aquí las demás partes se ejecutan dependiendo del valor de la casilla de "GENERACIONES" o del botón de paro. Esta parte incluye un ciclo "While Loop", con la finalidad de estar iterando "n" cantidad de veces el Algoritmo Genético. Por lo tanto, toda la instrucción contenida dentro de este ciclo se va a estar ejecutando más de una vez, es decir, se ejecutará hasta que la iteración del ciclo sea igual al valor de la casilla de "GENERACIONES" o el usuario active el botón de paro ubicado en la parte inferior del Panel Frontal.
5. En esta parte se encuentra ubicada la Etapa de Cruzamiento dentro de un SubVI. En la Figura 4.1.5 se muestra un "Case Structure", el cual contiene las diferentes técnicas de Cruzamiento.



**Figura 4.1.5** SubVI de la Etapa Cruzamiento del Algoritmo Genético.

6. En esta parte se encuentra ubicada la Etapa de Mutación dentro de un SubVI. En la Figura 4.1.6 se muestra un "Case Structure", el cual contiene las diferentes técnicas de Mutación.

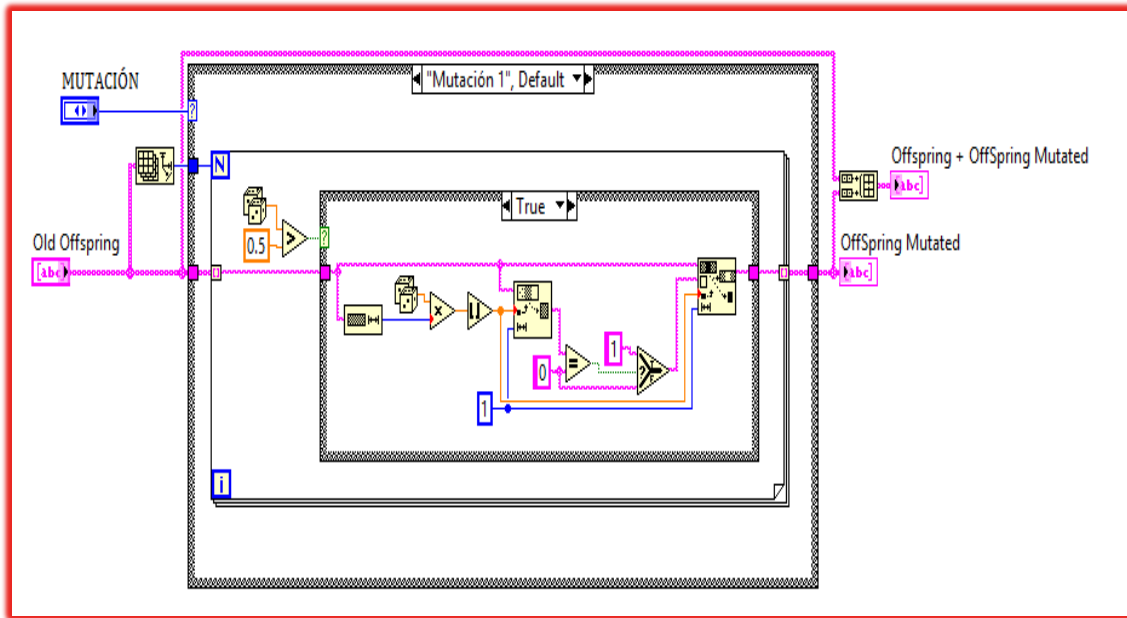


Figura 4.1.6 SubVI de la Etapa Mutación del Algoritmo Genético.

- Hay un SubVI encargado de convertir los datos de tipo cadena a datos numéricos, esta parte es considera el inverso de la conversión de la parte número de 3, se realiza esta operación con la finalidad de evaluar cada uno de los datos con respecto a una función objetivo que se encuentra en la parte número 8. Estos datos solo se pueden evaluar de forma numérica debido a que la función objetivo consta de ecuaciones que solo se pueden ejecutar a través de datos numéricos y no con otro tipo de dato. En la Figura 4.17 muestra la programación que se realizó para poder lograr la conversión de datos.

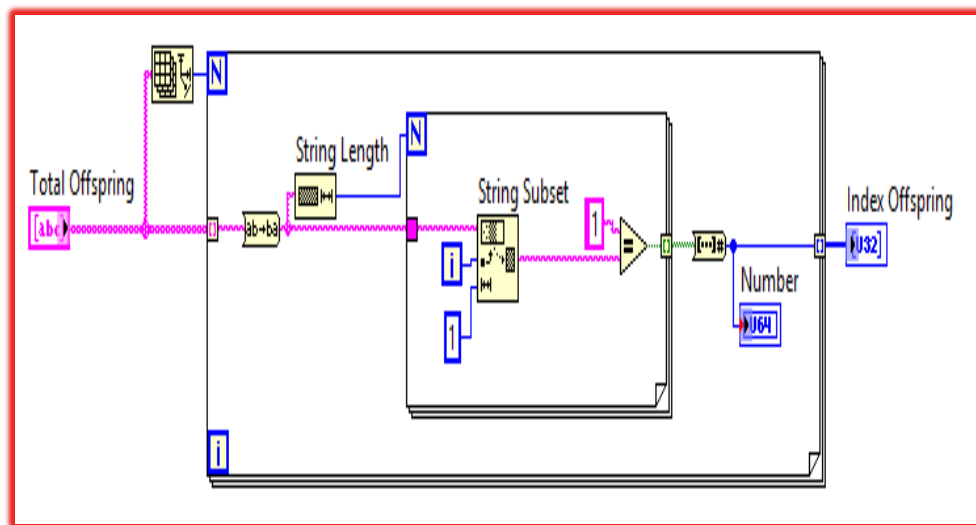


Figura 4.1.7 SubVI para convertir de Cadena a Numérico.

8. Contiene un SubVI con la programación de la Evaluación de los datos, la Función Objetivo se encarga de evaluar las propiedades de los individuos, esta evaluación se realiza en base a ecuaciones para conocer el nivel de aptitud de cada individuo. En la Figura 4.1.8 se muestra la programación del pseudocódigo de la evaluación en base a la Función Objetivo.

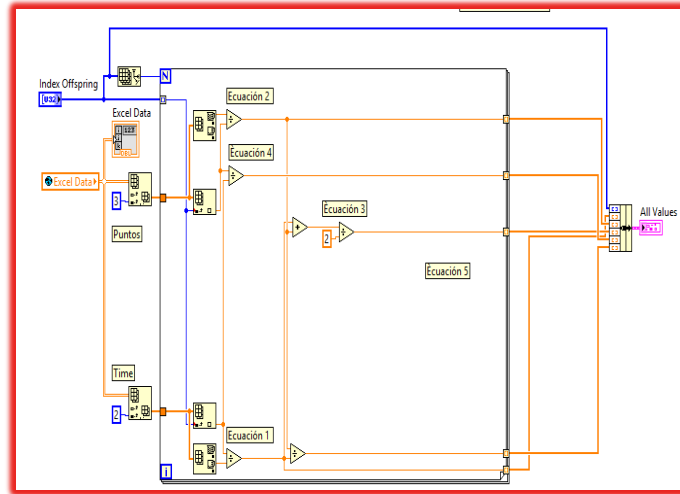


Figura 4.1.8 SubVI de Evaluación.

La función depende del problema que se desee resolver, es decir, su diseño depende directamente del diseñador del algoritmo y las variables del problema que se deseen optimizar. Debemos de hacer énfasis de la importancia de función objetivo, la elaboración de la función es la clave de los resultados del algoritmo genético, es decir, un buen diseño de la función se traduce a resultados confiables y optimizados, de lo contrario, se espera una mala calidad del algoritmo.

En este programa se programaron tres ecuaciones para realizar la función objetivo. La ecuación 4.1.0 describe el error de la Calidad de Información ( $Cler$ ), la ecuación 4.1.1 describe el error de Tiempo de duración del barrido de paso ( $TFer$ ) y la ecuación 4.1.2 describe la función objetivo del Algoritmo Genético

$$Cler = \frac{CImax}{Cicurrent} \quad (4.1.0)$$

$$TFer = \frac{TFcurrent}{Tmin} \quad (4.1.1)$$

$$CITF = \frac{TFer + Cler}{2} \quad (4.1.2)$$

Dónde:

$CI$  = Calidad de Información, es la cantidad de puntos detectados en una superficie de un objeto.

$TF$  = Tiempo de Finalización, es el tiempo para concluir un escaneo o mapeo digital.

$CI_{er}$  = Porcentaje de error de Calidad de Información.

$CI_{max}$  = Valor máximo presente en la población de Calidad de Información.

$CI_{current}$  = Valor actual seleccionada de Calidad de Información.

$TF_{er}$  = Porcentaje de error del tiempo.

$TF_{min}$  = Valor mínimo presente en la población de tiempo.

$TF_{current}$  = Valor actual seleccionado del tiempo.

$CITF$  = Función objetivo.

- En esta parte se encuentra ubicada la Etapa de Selección dentro de un SubVI. En la Figura 4.1.9 se muestra un "Case Structure", el cual contiene las diferentes técnicas de Selección. Cada uno de las técnicas es seleccionada por el control azul "SELECCIÓN", dependiendo de la técnica seleccionada es la selección que ejecutará el programa.

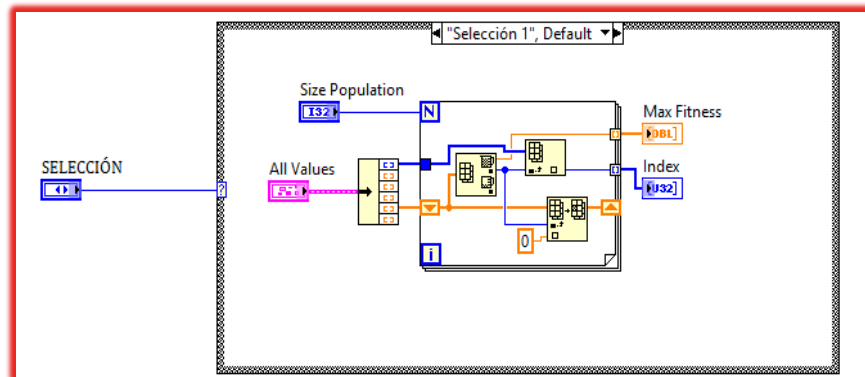


Figura 4.1.9 SubVI de la Etapa Selección del Algoritmo Genético

La diferencia entre el primer tipo de selección ubicado al inicio del código y el de esta parte, este contiene diferentes técnicas para selección, en el otro solo se hace una selección sencilla de la población de individuos generadas aleatoriamente y poder continuar con la iteración de los datos en el ciclo de "While Loop" donde este el Algoritmo Genético estructurado.

- Este SubVI es el mismo que se utiliza en la parte de número 3, pero solamente se usa la parte de conversión de datos tipo numéricos a datos tipo de cadena para seguir iteración del Algoritmo Genético Robusto. Por ser un SubVI se puede reutilizar en cualquier otra parte del pseudocódigo, solo se necesita definir las entradas y las salidas del SubVI.

11. En esta parte se encuentra ubicada la Etapa de Reemplazo dentro de un SubVI. En la Figura 4.1.10 se muestra un “Case Structure”, el cual contiene las diferentes técnicas de Reemplazo. Esta parte solo se ejecuta al final de cada ciclo del “While Loop”, reemplazando de tal manera los individuos más aptos por los menos aptos, dando como resultado el mejor ángulo de la Generación y mostrándose en la casilla de “ÁNGULO ÓPTIMO” en el pseudocódigo principal.

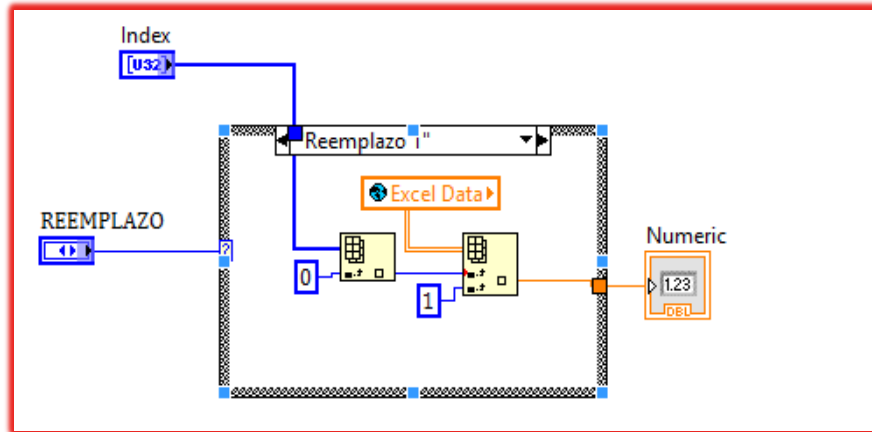


Figura 4.1.10 SubVI de la Etapa Reemplazo del Algoritmo Genético

12. Esta parte se encarga de detener el programa con el paro de salida del ciclo “While Loop”. Hay dos condiciones para poder salir del programa y finalice el Algoritmo Genético, el primero es que se cumpla las generaciones que fueron configuradas en la casilla de “GENERACIONES” y la segunda es que el usuario active el botón de “STOP”.
13. Contiene un “Case Structure” para utilizar una herramienta de tiempo con el fin de obtener la duración que le tome al Algoritmo Genético en dar un ángulo óptimo.

#### 4.2 MEJORA DE LA INTERFAZ GRAFICA DE USUARIO DEL TVS

La Interfaz Gráfica de Usuario establece la comunicación entre el usuario y la máquina, el usuario puede controlar por medio de un entorno visual las instrucciones programadas. La actual interfaz gráfica “v1.84” se encarga de controlar el Prototipo de TVS. Durante la Etapa de Experimentación del Prototipo, se analizó el código y las instrucciones incorporadas, en el apartado 3.4 se hicieron recomendaciones para mejorar la funcionalidad del prototipo. Las recomendaciones principales fueron, depurar el código y agregar una nueva serie de instrucciones que beneficien el comportamiento del paso de barrido del prototipo. Cuando hacemos mención al comportamiento, no nos referimos al incremento de precisión o velocidad de su paso, si no, al movimiento de sus

instrucciones de operación, agregando nuevas funciones es posible realizar movimientos automáticos. A continuación se describe las mejoras presentadas para el Prototipo una vez agregadas las nuevas instrucciones:

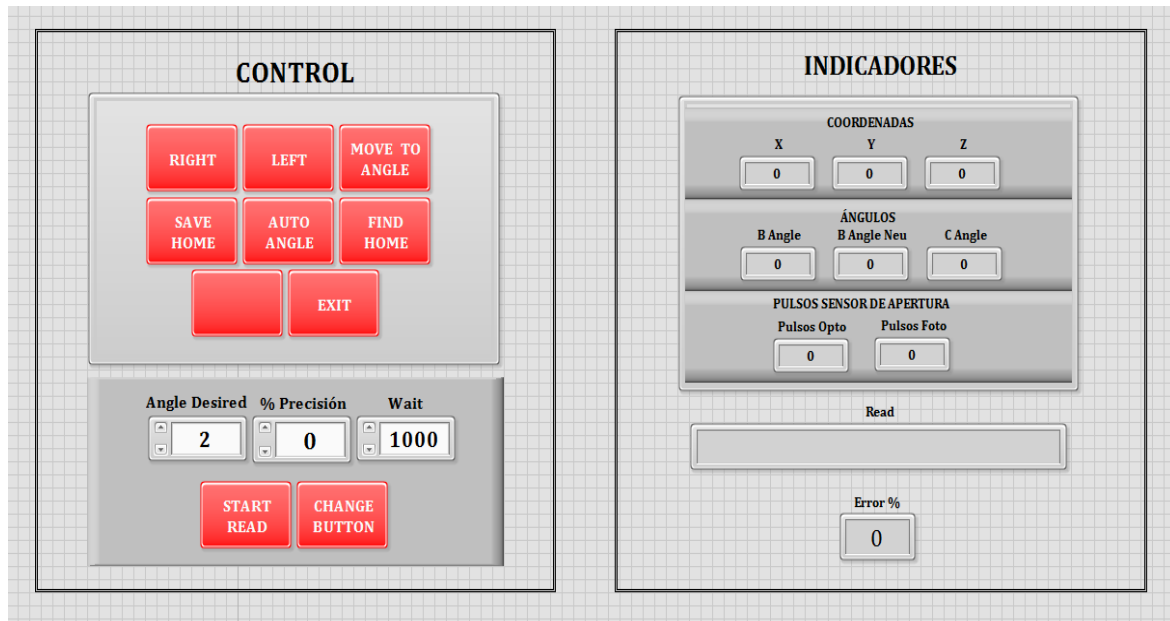
1. Incluir la función de un paso de barrido automático beneficia al escaneo láser para reconocer de forma automática cualquier objeto ubicado dentro del área de trabajo del Prototipo.
2. Corregir las interrupciones del programa v1.84, evita perder la información cuando está en plena operación de adquisición de información para el reconocimiento de algún obstáculo.
3. Una interfaz gráfica con mejores funciones y visualmente más estético hace al usuario comprender y realizar el control de las instrucciones de una forma más rápida y dinámica.
4. Depurar el código actual del programa v1.84 significa corregir las instrucciones que tardan en responder, esto es debido a que el programa cuenta con funciones y código repetitivas, originando una demora en tiempo de procesamiento de la instrucción.
5. Monitor y corrector de Posicionamiento angular, con las condiciones actuales de Hardware el movimiento de Posicionador Láser existe margen de error, este monitor tratará de monitorear y corregir dicho error pero con las condiciones actuales del sistema.

#### **4.2.1 DESCRIPCIÓN DE LA INTERFAZ GRAFICA DEL TVS VERSIÓN 2.0**

Para el desarrollo de este programa se tomó las bases del programa de la versión “v1.84” del TVS, este nuevo programa se etiqueta bajo la “*versión 2.0*”, contiene mejoras tanto en Panel Frontal, como en su programación e incluye nuevas funciones. En la Figura 4.20 se observa los controles e indicadores de este nuevo programa.

Los controles están ubicados en la parte izquierda, mientras que los indicadores se encuentran localizados en la parte derecha. Existen controles e indicadores que se conservan del otro programa, debido a que son muy indispensables para el funcionamiento del Prototipo. Los indicadores que realizan las mediciones también se conservan, por lo tanto, solo se cambiaron aspectos visuales. Las diferencias visuales principales entre ambos programas son tres, los controles e indicadores tienen un mejor aspecto visual, se eliminaron los controles de comunicación entre la

tarjeta DAQ y el programa, y aparte se incluyen nuevos controles para ejecutar nuevas instrucciones. A continuación se describe con detalle las funciones de los controles e indicadores:



**Figura 4.2.0** Panel Frontal de la Interfaz Gráfica del TVS (v2.0).

1. LEFT: Mueve hacia la izquierda el Posicionador Láser.
2. RIGHT: Mueve hacia la derecha el Posicionador Láser
3. MOVE TO ANGLE: Mueve el Posicionador Láser hacia el ángulo deseado digitalizado en el control "Angle Desired".
4. SAVE HOME: Guarda la posición inicial (HOME) del Posicionador Láser, esto es útil debido a que en las pruebas se puede considerar diferentes posiciones iniciales, y con la posibilidad de grabarla hace más sencillo el desarrollo de estas pruebas.
5. AUTO ANGLE: Es uno de los control más importante para realizar funciones de Navegación autónoma por parte del Prototipo, dependiendo del valor del ángulo deseado es el ángulo que se moverá automáticamente el Posicionador Láser, realizando de tal forma un barrido de escaneo automático, su movimiento es de izquierda a derecha de forma continua hasta que el usuario determine detenerlo.
6. FIND HOME: El control activa la instrucción para encontrar la posición inicial guardada, con el fin de calibrar y evaluar las condiciones del Prototipo.
7. EXIT: Detiene la operación del programa.
8. Angle Desired: Ubica angularmente el Posicionador Láser.

9. % Precision: Es la precisión que acepta entre posiciones angulares del Posicionador Láser, en teoría cuando llegue el Posicionador Láser a una posición angular, el programa verificará que el porcentaje de error sea menor o igual al valor contenido en esta casilla, dependiendo del caso, puede seguir buscando la posición o mantenerse en ella
10. Wait: Este control sirve para pausar entre cada posicionamiento angular.
11. START READ: Activa y desactiva la comunicación del puerto serial.
12. CHANGE BUTTTON: Activa y desactiva las propiedades del control "RIGHT" y "LEFT", cuando se activa estos botones, funciona con enclavamiento mecánico y cuando es desactivado el botón también se desactiva el enclavamiento
13. COORDENADAS: Muestra tres indicadores con las coordenadas tridimensionales de la medición de objetos.
14. ÁNGULOS: Muestras los valores de los ángulos  $B_{ij}$  y  $C_{ij}$  de la triangulación dinámica.
15. PULSOS SENSOR DE APERTURA: Muestra los pulsos del fototransistor y opto-acoplador de la sección del Sensor de Apertura.
16. READ: Lee la información del puerto serial:
17. ERROR %: Despliega el porcentaje de error de posicionamiento del Posicionador Láser.

#### 4.2.2 CÓDIGO DE LA INTERFAZ GRAFICA DEL TVS VERSIÓN 2.0

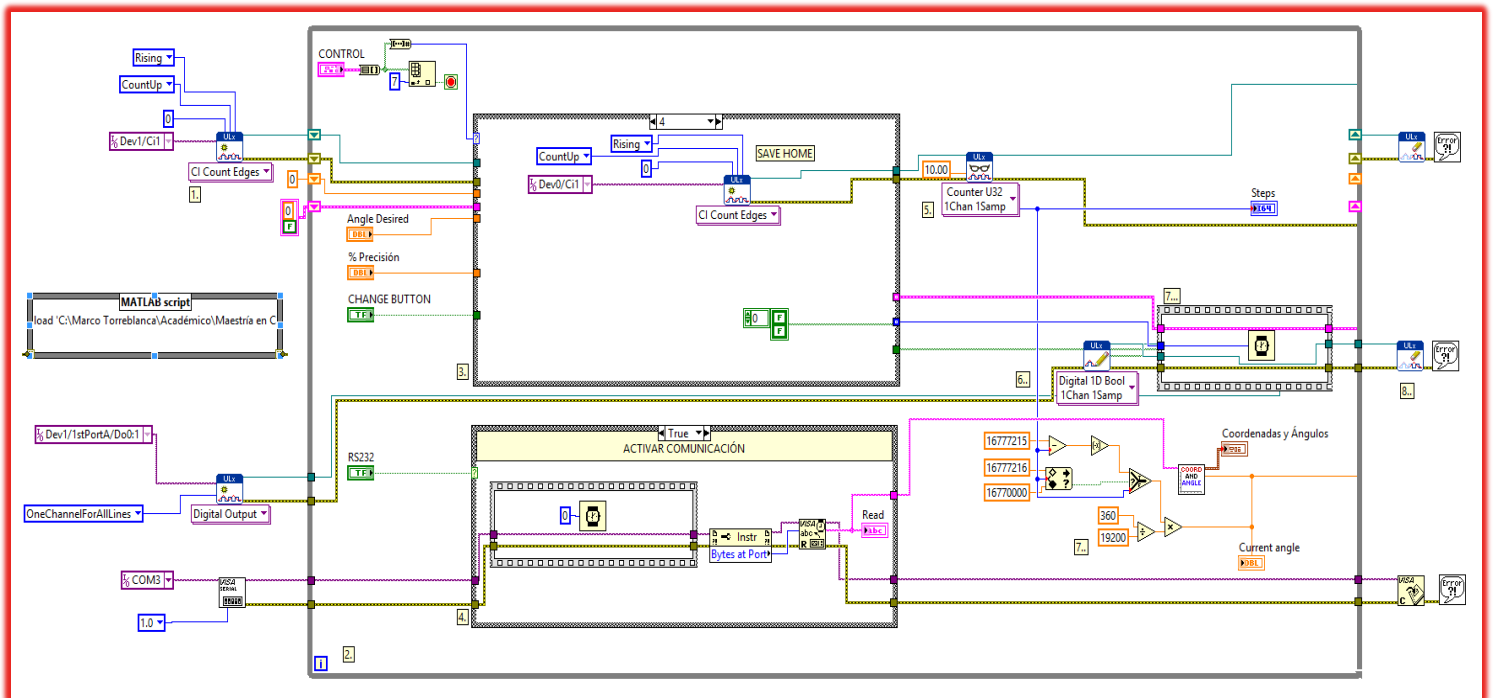
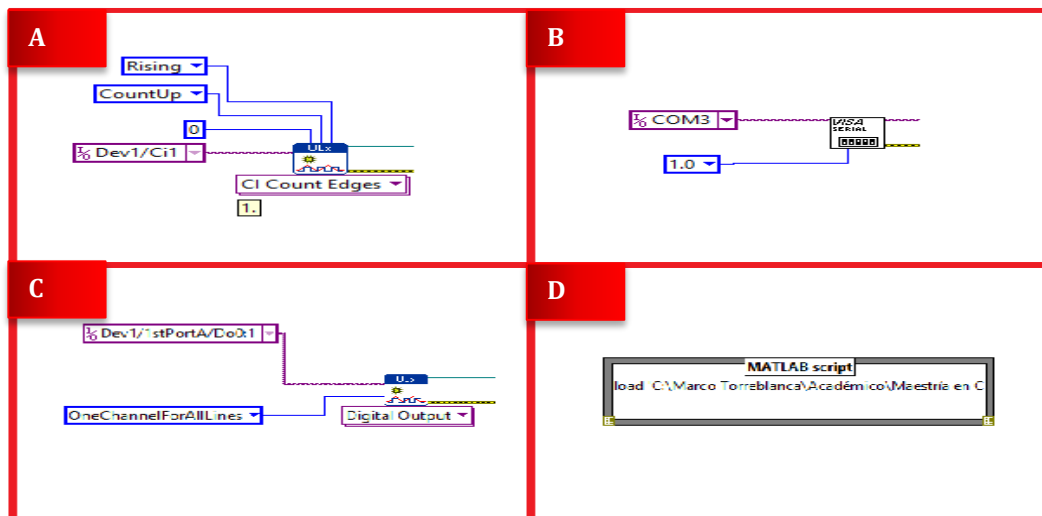


Figura 4.2.1 Código de la Interfaz Gráfica del TVS (v2.0).

En el desarrollo del código se reciclaron algunas instrucciones/funciones del programa “v1.84”, en su mayoría este nuevo código tiene una estructura de programación diferente. Este programa cuenta con mayor uso de funciones de programación de LabVIEW como “SubVI’s”, “Cluster”, “Property Nodes”, “Flat Sequence” que beneficia en la reducción de tener un código espagueti, por lo tanto, se puede tener un código más limpio y simplificado. En la Figura 4.2.1 se muestra Diagrama de Bloques del código.

La estructura de este programa consta de nueve partes (más adelante se describe cada parte para su mejor entendimiento), las cuales hacen posible su programación y ejecución de instrucciones para el Prototipo TVS. Enseguida se presenta la descripción de las nueve partes, haciendo énfasis en las diferencias entre el otro programa “v1.84” y este programa:

1. En la Figura 4.2.2 hay cuatro instrucciones/funciones diferentes. Estas instrucciones solamente se ejecutan una sola vez antes de iniciar el ciclo “While Loop”, por lo general este tipo de instrucciones están al inicio del código para establecer una comunicación con algún dispositivo electrónico.



**Figura 4.2.2.** Funciones de la Primera parte del Código.

Las funciones A y B sirven para establecer comunicación con la Tarjeta DAQ, habilitando de tal forma las señales de entrada, señales de salida y el contador de salida, a diferencia del otro programa, en este se quedan programadas en el código las señales que se utilizan en el circuito electrónico. La función B establece parámetros de comunicación del puerto serial, en esta función solo se configura el “COM” y “Stop Bits”, los otros parámetros están configurados

de forma predeterminada tal y como se muestra en la tabla 2.5.0. La función D realiza la búsqueda en la computadora de la Red Neuronal que utiliza el programa para corregir el error del ángulo [16]. La diferencia principal entre ambos programas, este programa reduce de seis funciones que existía para establecer la comunicación con la Tarjeta DAQ a dos funciones solamente

2. Contiene un Ciclo “While Loop” principal, en este ciclo se ejecuta las instrucciones y tareas del Prototipo TVS. Su funcionamiento se detiene hasta oprimir el botón de paro ubicado en el Panel Frontal.
3. Contiene un “Case Structure” para realizar todas las operaciones programadas para el Posicionador Láser, el control proviene del Cluster que se encuentra en el Panel Frontal ( Ver Figura 4.2.0).

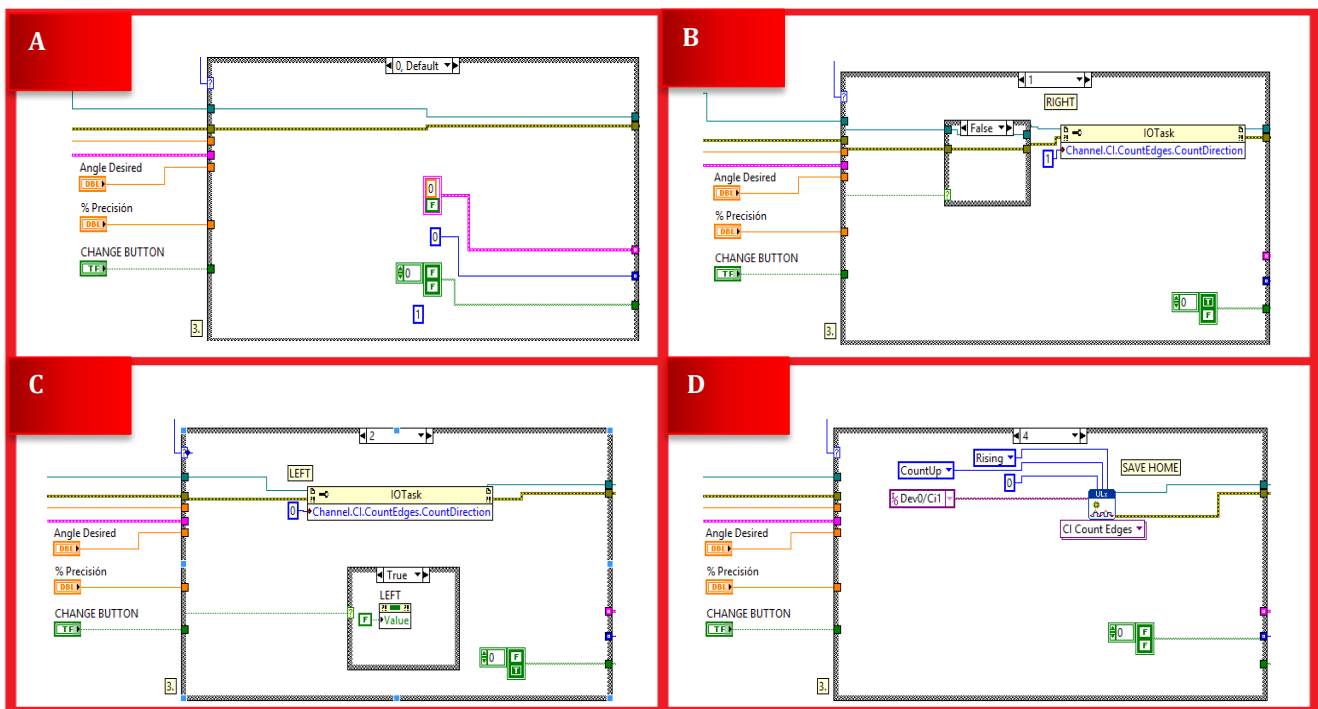


Figura 4.2.3. Casos A, B, C y D del Prototipo TVS.

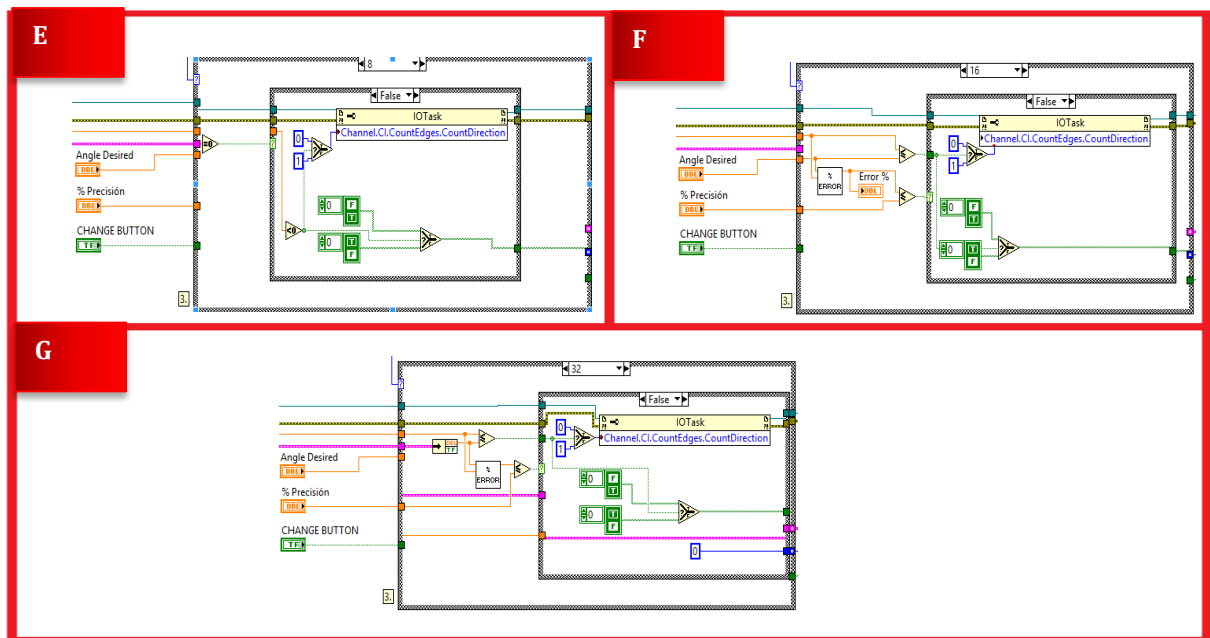
En las Figuras 4.2.3 y 4.2.4 se presentan la programación de las 7 operaciones representadas en Casos. La ejecución de cada operación se hace a través de un clúster que contiene siete controles que fueron descritos en el apartado 4.2.1. Por ser controles booleanos, y ordenados en el clúster tienen un valor binario, por lo tanto, puede ser introducido en el “Case Structure” para seleccionar cada operación.

*Caso A:* En este caso se encuentra en espera de alguna instrucción para poder realizar algún movimiento al Posicionador Láser, en otras palabras, cuando el “*Case Structure*” se encuentra en este caso no se ejecuta ninguna instrucción.

*Caso B:* Se ejecuta la instrucción de mover el Posicionador Láser hacia la derecha, esta instrucción modifica la señal de una función que utiliza la tarjeta DAQ para el sentido de giro, y se realiza mediante la función “*Property Node*”

*Caso C:* Se ejecuta la instrucción de mover el Posicionador Láser hacia la izquierda y al igual que el caso B, también hace la misma modificación para el sentido de giro.

*Caso D:* Se almacena la posición inicial (HOME) del Posicionador Láser mediante el reinicio del contador de la tarjeta DAQ.



**Figura 4.2.4.** Casos E, F, y G del Prototipo TVS.

*Caso E:* Realiza la búsqueda de la posición inicial, dependiendo de qué lado este orientado el Posicionador Láser es hacia donde se va a modificar la función “*Property Node*” para dar el sentido de orientación y trasladar de esta manera a la posición inicial.

*Caso F:* Ejecuta la posición angular del Posicionador Láser. Dentro del caso hay un SubVI encargado de monitorear el margen de error del posicionamiento angular. La ecuación 4.2.0 representa este margen de error.

$$\theta_{Error\%} = \left( \frac{\theta_{actual}}{\theta_{deseado}} - 1 \right) * 100 \quad (4.2.0)$$

Dónde:

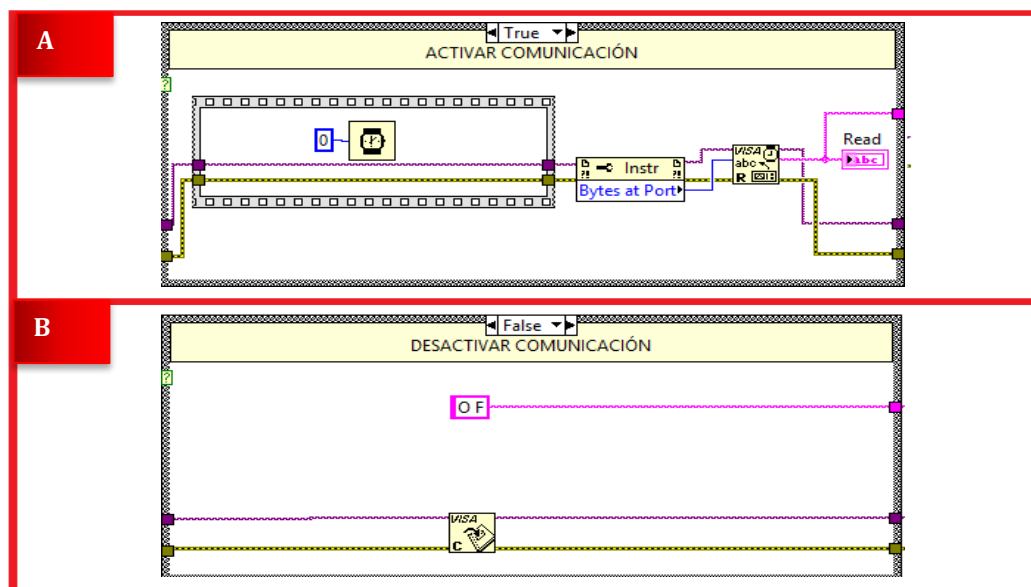
$\theta_{Error\%}$ : Porcentaje de error del ángulo deseado.

$\theta_{actual}$ : Valor del ángulo actual recorrido.

$\theta_{deseado}$ : Valor del ángulo deseado de la ubicación del Posicionador Láser.

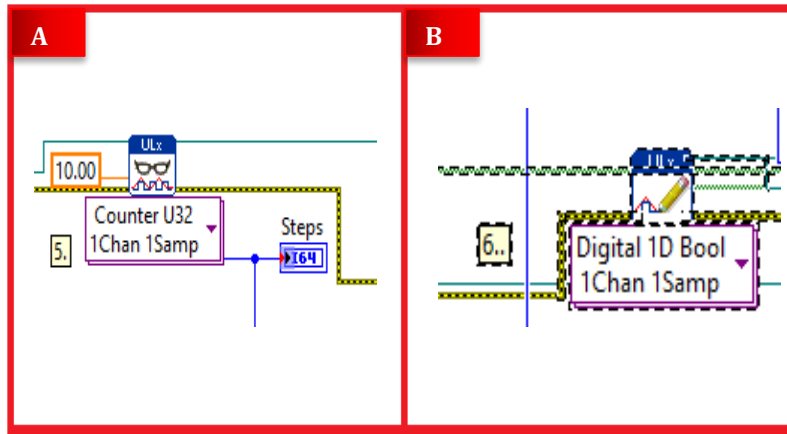
Caso G: En este caso se encuentra la programación para realizar el paso de barrido de forma automática. También se encuentra programado el SubVI que contiene el porcentaje de error del ángulo, pero la diferencia es que en este caso toma el valor del  $\theta_{Error\%}$  para tratar de corregir el ángulo hasta que el porcentaje de error sea menor al 0.5%.

- En esta parte se ejecuta la comunicación del puerto serial para adquirir los datos provenientes del circuito electrónico. En la Figura 4.2.5 se muestra la programación para activar y desactivar la lectura de datos. La Figura A activa el puerto serial mientras que la Figura B lo desactiva.



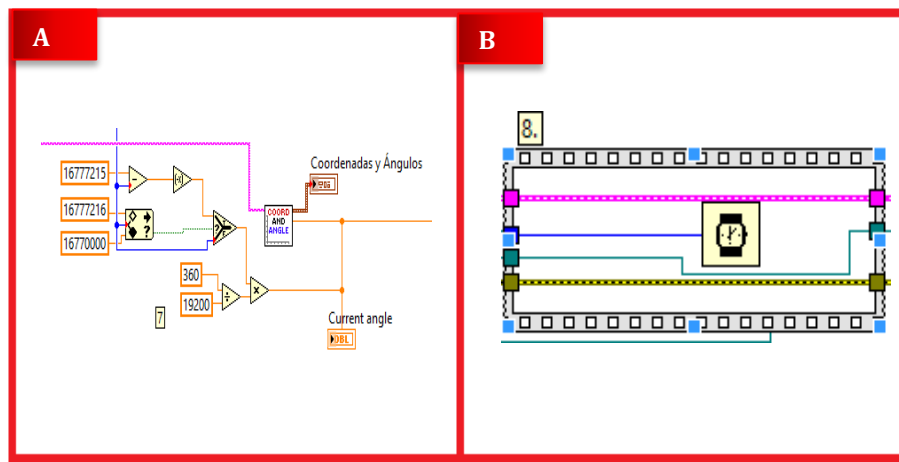
**Figura 4.2.5.** Código para activar y desactivar comunicación del Puerto Serial.

- En esta función realiza el conteo de pulsos que se envían a los motores a pasos, la cantidad de pulsos es necesaria para obtener el ángulo que está ubicado el Posicionador Laser. Esta función es originaria de la tarjeta DAQ, por lo tanto, siempre se está monitoreando la cantidad de pulsos y la información se guarda en la casilla "Steps" (Ver Figura 4.2.6-A).



**Figura 4.2.6.** Contador de Pulsos y Activador de Señales de Salida.

6. Esta función también es originaria de la Tarjeta DAQ, se encarga del sentido de giro del Posicionador Láser a través de su señal de salida. Dependiendo de la instrucción del “*Case Structure*” de la parte número 3, es el sentido del Posicionador Láser. Para realizar este movimiento es con la ayuda de dos señales digitales que están almacenadas en un arreglo booleano (Ver Figura 4.2.6-B).
7. Esta parte del código se reutilizo tal y como se usa en el programa “v1.84” debido a que todo el desarrollo para la adquisición del ángulo ya está elaborado [16] [17], por lo tanto, lo única mejora realizada fue poner lo más importante de este código en un SubVI para simplificar las programación tal y como se muestra en la Figura 4.2.7-A.

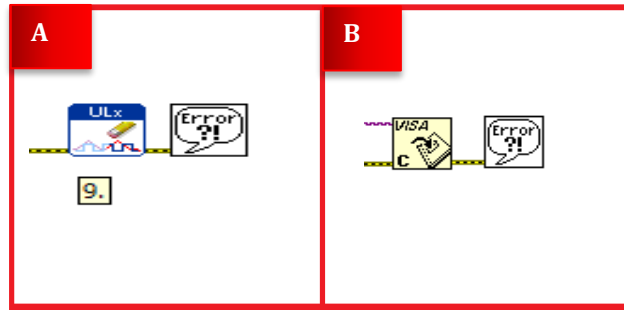


**Figura 4.2.7.** Código de la Red Neuronal y Pausado de tiempo.

8. Contiene la función “*Flat Sequence*”, toda la información de los controles principales y los parámetros de comunicación de la tarjeta de DAQ entran en esta función con la finalidad de

hacer una pausa entre cada iteración del ciclo “*While Loop*”. Su aplicación es utilizada para generar un tiempo muerto entre cada movimiento angular (Ver Figura 4.2.4-B).

9. La última parte cierra los puertos de comunicación de la Tarjeta DAQ y del puerto serial para evitar errores. Se utilizan las funciones al finalizar el ciclo “*While Loop*” principal (Ver Figura 4.2.8).



**Figura 4.2.8.** Funciones para cerrar comunicación de Tarjeta DAQ y Puerto Serial. TVS.

### 4.3 SIMULADOR DE ESCENARIOS VIRTUALES

Se debe realizar una simulación por computadora que beneficie el avance y desarrollo del Algoritmo Genético, para ser implementado en la etapa de evaluación, experimentación, entrenamiento, y validación de dicho algoritmo. La simulación bajo el nombre de Simulador de Escenarios Virtuales (SEV) realiza el funcionamiento del sistema láser real de manera virtual (con ciertas restricciones), su comportamiento se interpreta mediante programación.

Una de las principales ventajas de la simulación, es reducir el tiempo de preparación de cada una de las pruebas a la que será expuesta el Algoritmo Genético, se contemplan miles de pruebas con diferentes condiciones en el escenario, de igual modo, sí las pruebas se llegan a preparar manualmente, el tiempo en concluir cada una de las pruebas de cada etapa se elevaría drásticamente a comparación de utilizar la simulación, dichas pruebas las puede generar automáticamente, asimismo, mejora la eficiencia y velocidad de las pruebas para conocer con mayor rapidez los resultados que serán el cimiento de un buen Algoritmo Genético para la optimización del paso de barrido.

Se debe considerar también el error entre el escenario real y virtual, nos referimos como error, a la discrepancia que hay entre el comportamiento de la simulación con respecto al sistema

real, lógicamente este error debe de ser el menor para no afectar los resultados en las pruebas (¿Cuál es el menor error permisible?). Aparte, la simulación cuenta con restricciones que disminuyen la complejidad de la simulación al momento de ser programada en el código (detalladas más adelante).

La simulación se llevará a través de LabVIEW, en él se integrarán las etapas del sistema láser para ser programadas en la simulación, por lo tanto, se podrá visualizar el mapeo de los obstáculos tal y como lo hace el sistema real pero en forma 2D. Todas estas características se evaluarán con el sistema láser real para conocer el valor real del error con una corrida de 100 pruebas. Las cuales se someterá a condiciones “iguales” con respecto a la dimensión del obstáculo para ver su comportamiento de cada uno.

Recordando las secciones del sistema láser (especificadas en el apartado 2.5.3) cuenta con tres principales secciones para su funcionamiento, los cuales se vuelven a mencionar: Posicionamiento de Láser (A), Objeto a mapear (B), y Sensor de Apertura (C), si alguna de estas tres secciones deja de funcionar, se daña o no está, el sistema láser no funciona. En la simulación se deben de programar dichas secciones y otras consideraciones, podemos adelantar que no es necesario en la simulación integrar la sección (c), debido a que existen herramientas y funciones en el software que pueden reemplazar dicha sección.

La simulación SEV está programada por tres etapas principales, como el sistema láser real (Se describirá a fondo en los siguientes apartados), por lo pronto las etapas son: (1) Simulación del Posicionamiento de Láser, (2) Simulación de escenario, (3) Simulación de Mapeo de objetos. Es importante resaltar que la simulación tiene como fin en ayudar al Algoritmo Genético para su mejoramiento, por lo tanto, es válido descartar o considerar idóneas algunos detalles específicos del sistema láser. La simulación contará con restricciones y parámetros que den como resultado un comportamiento similar al sistema real.

#### **4.3.1 DESCRIPCIÓN DE SIMULACIÓN DEL POSICIONAMIENTO LÁSER**

La Simulación del Posicionador Laser (PL) es fundamental en el diseño del Simulador de Escenarios Virtuales. Recordando, el Posicionador Láser apunta hacia la superficie de un objeto para reconocer las dimensiones del objeto y la distancia entre el objeto y el láser escáner del Prototipo (Ver Figura 4.3.0).

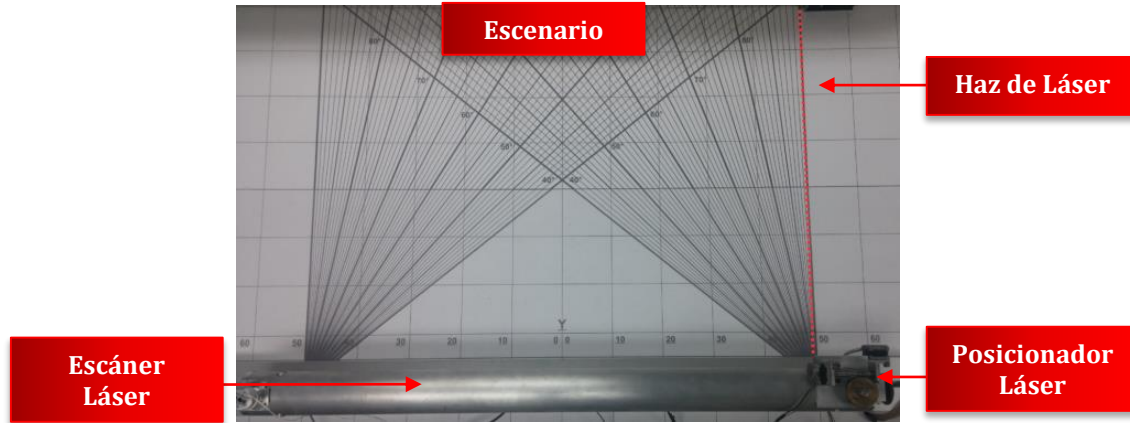


Figura 4.3.0. Vista panorámica del Sistema Láser del TVS.

La simulación de Posicionador Láser debe tener el mismo comportamiento, se debe de mover hacia la izquierda y derecha de manera angular. El rayo de láser se debe de representar mediante la construcción de varios puntos hasta formar una recta. Por medio de programación se puede usar una ecuación matemática que describa y realice la construcción de la recta (Ver ecuación 4.3.0).

$$m_i = \frac{Y_{Ti} - Y_{PL}}{X_{Ti} - X_{PL}} \quad (4.3.0)$$

Dónde

$X_{PL}$ : Coordenada en el Eje X para el Posicionador Láser.

$X_{Ti}$ : Conjunto de puntos de coordenadas en el Eje X para la posición final de la recta.

$Y_{PL}$ : Coordenada en el Eje Y para el Posicionador Láser.

$Y_{Ti}$ : Conjunto de puntos de coordenadas en el Eje Y para la posición final de la recta.

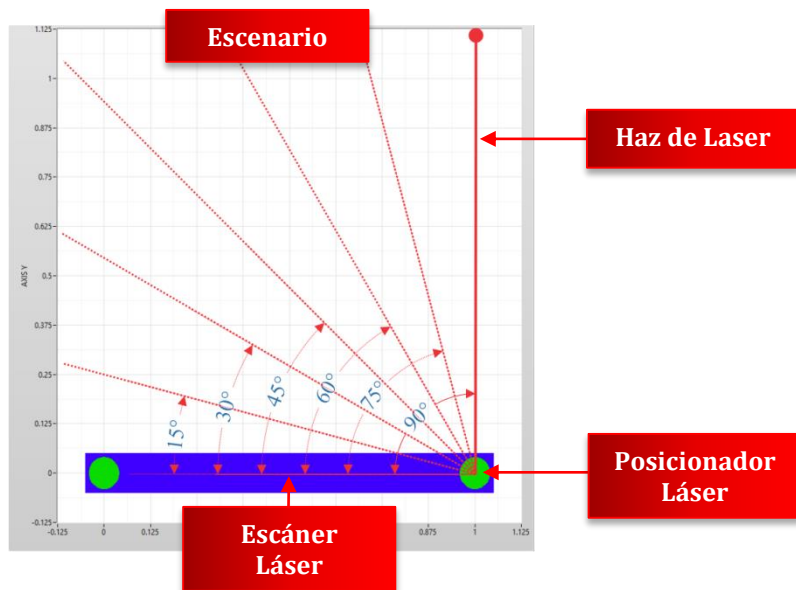
$m_i$ : Conjunto de valores de ángulos para la representación de inclinación de la recta o (Simulando el haz de luz del Posicionador Láser).

Las variables de la ecuación de la recta son modificadas para distinguir los puntos de coordenadas correspondientes a la colocación del Posicionador Láser y la colocación del final de la recta en el Simulador. Las coordenadas  $X_{PL}$ ,  $Y_{PL}$  y  $Y_{Ti}$  tienen valores constantes, es decir, siempre serán los mismos valores en la simulación porque el Posicionador Láser siempre mantendrá la misma colocación, a excepción de la variable  $Y_{Ti}$ , está puede cambiar dependiendo de qué tan largo se desee la longitud del haz de láser simulado. Las primeras dos variables representan los puntos de coordenadas del Posicionador Láser. La variable  $Y_{Ti}$  representa el otro punto de la coordenada para

formar la longitud de la recta. La variable,  $X_{Ti}$  tendrá un conjunto de puntos de coordenadas diferentes para estar realizando el movimiento angular del Posicionador Láser en la simulación, pero antes de eso se debe generar manualmente o automáticamente el valor de inclinación de la recta. La variable  $m_i$  realiza esta función, dependiendo del valor de la variable, es la inclinación que tendrá la recta. Toda esta descripción se puede representar con el despeje de la variable  $X_{Ti}$  de la ecuación 4.3.0, como se muestra en la ecuación 4.3.1.

$$X_{Ti} = \frac{m_i}{Y_{Ti} - Y_{PL}} + X_{PL} \quad (4.3.1)$$

La ecuación 4.3.1 representa en la simulación el comportamiento del movimiento del haz de láser del Posicionador Láser, su origen siempre será donde mismo, su movimiento angular se podrá mover gráficamente hacia cualquier ángulo. La simulación del Posicionador Láser cumplirá con un movimiento manual y automático. El movimiento manual nos referimos en seleccionar un valor de ángulo, por lo tanto, la simulación debe proceder a ejecutar la instrucción y posicionarse al ángulo deseado, tal y como lo hace físicamente el prototipo TVS. La modalidad automática se refiere en seleccionar un valor de ángulo, pero en este caso se estará moviendo de forma continua el ángulo seleccionado, por lo tanto, se va a trasladar en ambas direcciones, tal como se presenta en la Figura 4.3.1.



**Figura 4.3.1** Simulación del Posicionador Láser.

En la misma Figura se hace una representación del movimiento automático dentro del escenario simulado, se aprecia en la simulación del haz de laser está en la posición de 90° con

respecto al escáner láser. Como se comentó en los apartados anteriores esta es la posición inicial considerada como HOME, por lo tanto, a partir de esa posición nuestra simulación empieza a funcionar, por consiguiente empieza a trasladarse de izquierda a derecha, en la figura se aprecia con línea punteada de color rojo el traslado de 15° por cada movimiento angular.

#### **4.3.2 DESCRIPCIÓN DE SIMULACIÓN DE ESCENARIOS**

Recordando, existe una cantidad infinita de escenarios para realizar un reconocimiento digital con nuestro Prototipo TVS. Cada escenario tiene condiciones distintas y únicas en donde el Prototipo puede utilizarse. Se considera el ángulo de paso de barrido se debe ajustar a condiciones y características del escenario para tener como resultado una mejor precisión y velocidad en la captura del reconocimiento digital del escenario. En la Tabla 4.3.0 se muestra diferentes factores con sus aspectos, el cual considero práctico para distinguir diferencias entre escenarios.

<b>FACTORES</b>	<b>ASPECTOS</b>
Forma del obstáculo	Formas Geométricas (Círculo, Cuadrado, Triángulo, etc.)
	Formas Irregulares
	Mezclado de Formas (Geométricas e Irregulares)
	Tamaño (Dimensiones)
Dimensiones del escenario	Color
	Dimensión 2D (Largo y Ancho)
Posición del Obstáculo	Dimensión 3D (Largo, Ancho y Alto)
	Obstáculos en diferente posición angular
Distribución del Obstáculo	Obstáculos a diferente distancia
	Cantidad de obstáculos
	Distribución centralizada de los obstáculos
	Distribución descentralizada
Escenario	Mezcla de Distribuciones
	Plano
	Perfil Complejo

**Tabla 4.3.0** Factores de las Escenarios.

Todos los aspectos mostrados en la tabla anterior, se consideran para un escenario real, en nuestra simulación se debe descartar varios factores. Esto es debido a su complejidad al momento de realizar la programación, también podemos considerar que simplemente no es necesario simularlos. El objetivo de la simulación es generar tipos de escenarios con obstáculos diferentes en cuestión de

forma, cantidad y posición de forma aleatoria. En la tabla 4.3.1 se muestra los factores programados en la simulación, estos factores son los más importantes a considerar como útil y convenientes

FACTORES	ASPECTOS
Forma del obstáculo	Formas Geométricas (Círculo, Cuadrado, Triángulo, etc.)
	Tamaño
Dimensiones del escenario	Dimensión 2D (Largo y Ancho)
Posición del Obstáculo	Obstáculos en diferente posición angular
	Obstáculos a diferente distancia
Distribución del Obstáculo	Cantidad de obstáculos

**Tabla 4.3.1** Factores de las Escenarios a considerar.

Los obstáculos son una parte fundamental en la simulación, sus principales características se basan en la Tabla 4.3.1. Las propiedades de los obstáculos son: forma geométrica, posición dentro del escenario, cantidad de obstáculos, dimensiones de su longitud, y posición angular. Para generar los obstáculos dentro del escenario existen dos maneras programadas. La primera es la manera manual, en esta se ajustan manualmente sus propiedades con la ayuda de controles, en la segunda se generan de forma aleatoria sus propiedades.

La construcción de obstáculos simulados está representada por figuras geométricas en 2D, para su formación por lo menos se necesitan dos puntos de coordenadas, requerimiento mínimo para formar una sola recta, por consiguiente se puede iniciar con el funcionamiento de la simulación. El primer punto de coordenada es el origen de la figura. Definiendo su primera coordenada se puede formalizar su posición dentro del plano (considerado como el escenario virtual), de tal forma, se continúa con la construcción de la figura geométrica y el tamaño. Recordando, hay dos maneras de configurar las propiedades, por esta razón los valores de la primera coordenada de la figura puede construirse de manera manual o automática. Las siguientes ecuaciones expresan matemáticamente la construcción de las figuras presentadas en el plano simulado:

$$CX_i = PX + SX \cdot \sin\left(\frac{i \cdot 2 \cdot \pi}{R-1}\right) \quad (4.3.2)$$

$$CY_i = PY + SY \cdot \cos\left(\frac{i \cdot 2 \cdot \pi}{R-1}\right) \quad (4.3.3)$$

Dónde

$CX_i$ : Conjunto de puntos de Coordenada en el Eje X

$CY_i$ : Conjunto de puntos de Coordenada en el Eje Y

$PX$ : Factor para modificar la posición de la figura en el Eje X.

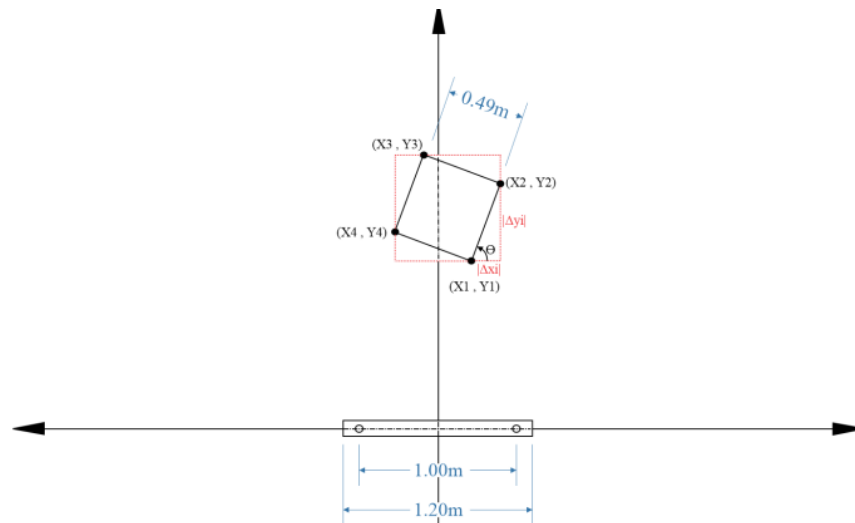
$PY$ : Factor para modificar la posición de la figura en el Eje Y.

$SX$ : Factor de tamaño de longitud de la figura en el Eje X

$SY$ : Factor de tamaño de longitud de la figura en el Eje Y

$R$ : Resolución de la figura, es el número de iteraciones para formar cierta cantidad de puntos de coordenadas.

Se considera la variable  $R$  como la más importante de las ecuaciones presentadas, esta define la forma de la figura (triángulo, cuadrado, pentágono, etc.) que será construida. Entre mayor sea el valor de la resolución mayor será el número de puntos de coordenadas que tendrá. Si el número de Resolución es grande, la figura puede ser considerada como un polígono, chiliágono (1000 rectas), o hasta un megágono (1 000 000 rectas). Por lo tanto, la figura con muchos puntos de coordenadas puede llegar a ser considerado como círculo. En la Figura 4.3.2 se muestra la construcción de un cuadrado, donde la variable  $R$  tiene el valor de 5, debido al inicio de la coordenada y el final de la coordenada son los mismos, por lo tanto, se consideran 5 coordenadas. Toda figura construida siempre tendrá uno más que el número de rectas formadas en la figura, ejemplo, el cuadrado tiene cuatro rectas, entonces el valor de la variable  $R$  será de 5.

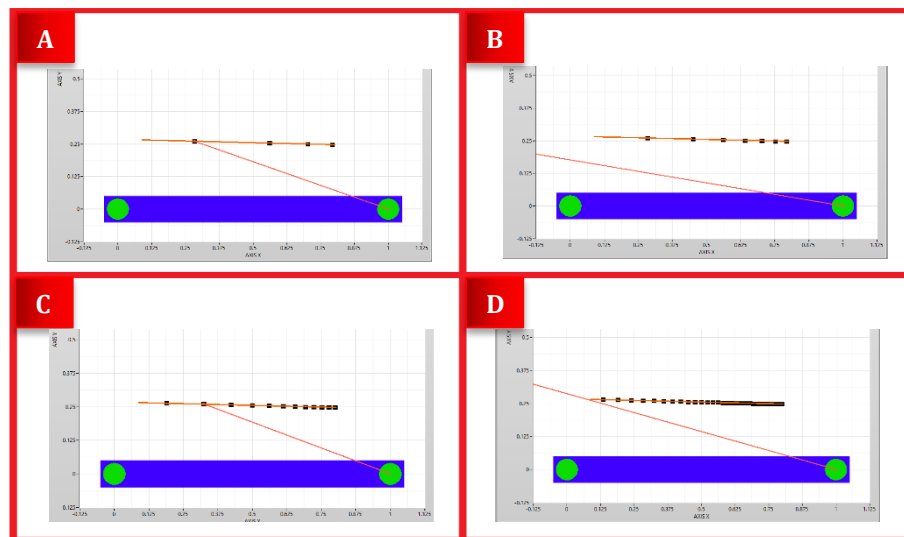


**Figura 4.3.2** Construcción de una figura geométrica en el Simulador.

### 4.3.3 DESCRIPCIÓN DE SIMULACIÓN DE MAPEO DE FIGURAS

La simulación del Posicionador de Laser permite realizar el ángulo de paso de barrido y la simulación del escenario permite crear figuras que representan los obstáculos, por lo tanto, teniendo estas dos simulaciones se procede a realizar el mapeo digital de las figuras ubicadas dentro del plano. La acción de realizar una detección en la superficie de un obstáculo lo podemos catalogar como mapeo digital. La cantidad de puntos detectados en la misma superficie durante el mapeo digital la podemos llamar como Calidad de Información (*CI*).

Lógicamente, entre mayor sea la Calidad de Información, mayor es el reconocimiento del objeto. Una Calidad de Información considerada como aceptable, significa un mayor tiempo en finalizar el mapeo digital, debido al ángulo de paso de barrido es menor y los puntos detectados aumentan. En la Figura 4.3.3 se presenta cuatro figuras, cada una tiene diferencias en el ángulo de paso de barrido, entonces su comportamiento de su mapeo digital será distinto en cada figura (A, B, C y D). Además se simula con una recta la superficie de un objeto. La Figura A cuenta con un valor de ángulo mayor con respecto a las otras figuras, este ángulo tiene un valor de  $10^\circ$ . Significativamente hay un número de puntos detectados menor que en las otras figuras.



**Figura 4.3.3** Simulación de escaneo digital con diferentes ángulos.

En la Tabla 4.3.3 se muestran los datos del comportamiento de cada figura, de tal forma, se aprecia tiempo de duración del mapeo digital, calidad de información de barrido y el ángulo de paso de barrido.

FIGURA	ÁNGULO DE PASO DE BARRIDO	PUNTOS DETECTADOS	DURACIÓN (seg)
A	10°	4	2
B	5°	7	3
C	3°	12	4
D	1°	35	9

**Tabla 4.3.2** Datos de Comportamiento de la Figura 4.3.3.

Es importante mencionar los resultados de duración del mapeo digital entre el escenario real y el escenario virtual son sumamente diferentes por las condiciones electrónicas y mecánicas del Prototipo. El simulador actúa más rápido, debido a que no hay ningún movimiento mecánico y todo se realiza por medio de programación. La configuración de pasos (Half-Step, Full Step) de los motores a pasos es el mismo sin importar el valor del ángulo sea mayor o menor, por lo tanto, el traslado será el mismo.

El Sensor de Apertura para el Prototipo TVS es fundamental para realizar la detección del obstáculo y a través de triangulación dinámica obtener sus coordenadas. Para la simulación no es necesario simular el Sensor de Apertura, basta con solo simular el Posicionador Laser y la figura geométrica (en este caso una es una recta) para obtener sus coordenadas y distancia entre el objeto y el sistema. El Sensor de Apertura tiene una duración para adquirir y enviar la información hacia la Interfaz Gráfica, por lo tanto, no teniendo esta sección programa, se reduce ese tiempo en la Simulación de Escenarios Virtuales.

#### **4.3.4 RESTRICCIONES DEL SIMULADOR DE ESCENARIOS VIRTUALES**

En este apartado se describe las restricciones y límites del Simulador de Escenarios Virtuales con respecto al funcionamiento del Prototipo TVS y las características de un Escenario Real. El funcionamiento y las características de ambos no se pueden programar debido a su complejidad. Recordando, uno de los objetivos del Simulador es obtener el funcionamiento fundamental del Prototipo TVS sin importar ¿con que elementos hace posible su función?, si no ¿cómo realiza su función?, por lo tanto, se puede omitir partes del Prototipo, siempre y cuando cumpla con el funcionamiento integro. Las restricciones principales del simulador son las siguientes:

- a) Simulación diseñada en 2D.** Se realizó la simulación en 2D para disminuir la complejidad en la programación y coincidir con las condiciones actuales del Prototipo. Como se comentó,

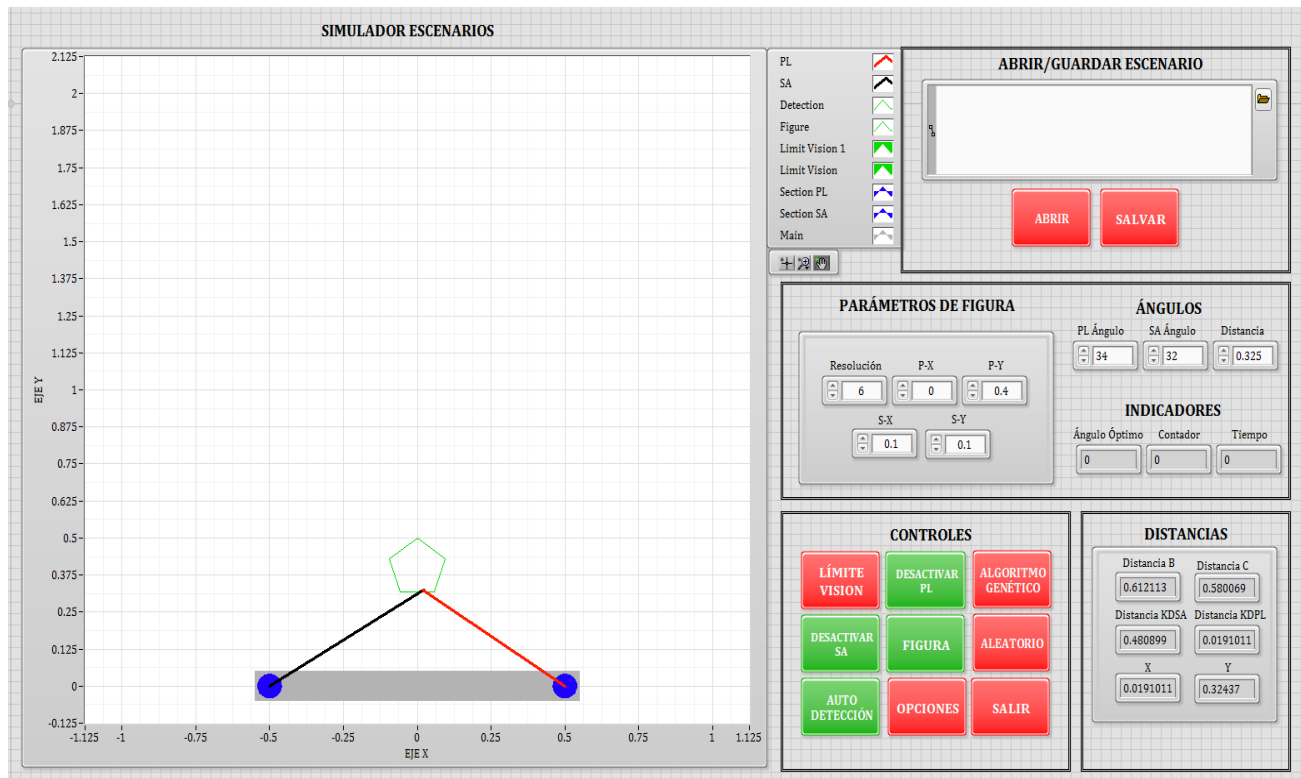
el Prototipo fue diseñado para realizar un escaneo tridimensional, pero en las actuales condiciones de barrido por tercer eje se repite en plano ortogonal todas las acciones sucesivas, por lo tanto son realizadas en dos dimensiones. Por esta razón se decide desarrollar la simulación en 2D para tener las mismas condiciones. La simulación en 2D es más ligera que una programación en 3D con respecto a programación. Programar una simulación en 3D significa un aumento considerable de instrucciones y funciones.

- b) Simulación sin el soporte del Sensor de Apertura.** Como se comentó en el apartado anterior, en la simulación se puede omitir toda la sección del Sensor de Apertura debido que se puede utilizar herramientas y funciones programables capaces de replicar su funcionamiento en otra etapa. Este funcionamiento es a través de obtener la intersección de la recta que representa el objeto y la recta que representa el Posicionador Láser. Teniendo las coordenadas de la intersección, podemos saber la distancia que existe entre la figura y el escáner láser simulado. No obstante, se programa el Sensor de Apertura para representarlo en la simulación y no perder esencia del escáner láser. La única diferencia es que carece de su funcionalidad.
- c) Simulación sin movimiento mecánico y fricción.** El movimiento mecánico y la fricción que contiene el sistema se hace presente en la parte mecánica del juego de engranes y en la rotación de los motores a pasos. Estas características no son necesarias en programar porque virtualmente se descartan toda cuestión mecánica para realizar el movimiento del Posicionador Láser.
- d) Simulación sin reflexión:** Al no contar con la funcionalidad del Sensor de Apertura, no es necesario programar las propiedades de reflexión que tienen las superficies de cualquier objeto. Por lo tanto, el nivel de complejidad en la programación de la simulación se reduce.
- e) Escenario y Figuras Geométrica estáticas:** Recordando, el Prototipo TVS está orientado a ser utilizado en robot móviles autónomos, por lo tanto, cuando este en operación, el prototipo se estará desarrollando en escenarios y figuras que van cambiando su posición, es decir, mientras esté realizando su trayectoria de un punto hacia otro, el prototipo va a experimentar diferencias en la posición de las figuras dentro del escenario que está reconociendo. En la simulación, las figuras y el escenario se mantendrán de manera estática, en otras palabras carecen de características dinámicas.
- f) Dimensiones del Escenario:** Las dimensiones del Escenario son a escala, por lo tanto, simuladamente están restringidas a 2.25m de largo y 2.25m de ancho. Dentro del inicio del escenario están ubicado el Posicionador Láser.

Todas las restricciones mencionadas están incluidas en la simulación y no afecta el funcionamiento del Posicionador Láser y al ángulo de paso de barrido simulado. En el sistema real es importante tener estas características para su correcto funcionamiento.

#### 4.3.5 DESCRIPCIÓN DEL PANEL FRONTAL DEL SIMULADOR DE ESCENARIOS VIRTUALES

El Simulador de Escenarios Virtuales es diseñado en el programa de LabVIEW 2013, su interfaz tiene una gráfica principal donde se despliega visualmente la simulación del Escenario y del Escáner Láser. Respectivamente se encuentran simulados los obstáculos y el Posicionador Láser. En la Figura 4.3.3 se presenta el Panel Frontal de la simulación.



**Figura 4.3.4** Panel Frontal del Simulador de Escenarios Virtuales.

Como se observa en la Figura anterior, el simulador consta de controles e indicadores para su funcionamiento. En la parte izquierda se encuentra la gráfica mencionada anteriormente y en la parte derecha los controles e indicadores. Las funciones programadas en esta simulación son las siguientes:

1. Simulación de la funcionalidad del Posicionador Láser.

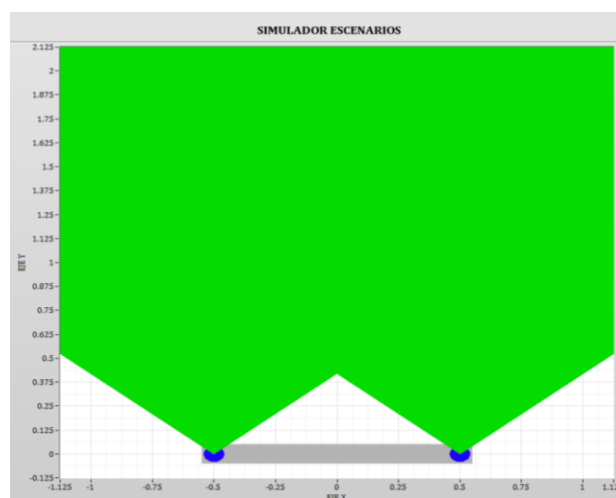
2. Simulación del Sensor de Apertura solo como referencia.
3. Activador del Algoritmo Genético para optimizar ángulo de paso de Barrido.
4. Modificación de la dimensión y posición de la Figura dentro del escenario virtual.
5. Generación aleatoria manual o automática de figuras geométricas (Línea recta, triángulo, cuadrado, polígonos).
6. Opciones para guardar y abrir los Escenarios generados en la simulación.
7. Contador de puntos detectados de la Figura o línea.
8. Indicador de duración del paso de barrido.
9. Paso de Barrido automático o manual.
10. Simulación de la visión de trabajo del escáner láser.
11. Indicador de las coordenadas de la Figura.

Como se puede observar, la simulación se divide en siete secciones donde están programadas las once funciones descritas (Fig. 4.3.4):

1. *SIMULADOR DE ESCENARIOS*: En esta sección está contenida una gráfica para simular el Posicionador Láser, Sensor de Apertura, escenario y figura geométrica. En la parte inferior esta un recuadro gris que representa el tubo del escáner láser. El Sensor de Apertura y el Posicionador Láser, cada uno está representado por un círculo azul. El haz del láser se representa por medio de una recta roja y la reflexión del haz del láser se representa con una recta de color negro. La ubicación en coordenadas dentro de la gráfica para el Posicionador Láser es  $x = 0$ ;  $y = 0.5$ , para el Sensor de Apertura es  $x = 0$ ;  $y = -0.5$ . La Figura puede posicionarse en cualquier parte de la gráfica y sus dimensiones se controlan a través de la siguiente sección.
2. *ABRIR Y GUARDAR ESCENARIO*: En esta sección permite abrir y guardar cualquier escenario generado por esta simulación. Todas las características del escenario se pueden guardar con la finalidad de volverse a usar.
3. *PARÁMETROS DE LA FIGURA*: En esta sección se encuentran los controles de los parámetros para la construcción de la figura. Como se describió en el Apartado 4.3.2 con las ecuaciones 4.3.2 y 4.3.3 se puede formar una figura geométrica. La sección consta de cinco controles basados en las ecuaciones, estos controles están etiquetados como, “Resolución”, “P-X”, “P-Y”, “S-X” y “S-Y”. La función de cada control corresponde a la descripción del apartado mencionado, por lo tanto, los controles configuran la forma de la figura geométrica, la

posición y la dimensión. Estos controles son usados de forma manual. Cuando la construcción de la figura es de forma aleatoria, los parámetros de estos controles son omitidos.

4. **ÁNGULOS:** Existe tres controles para mover la referencia angular y la distancia de la recta del Posicionador Láser y del Sensor de Apertura. Cada control tiene su nombre característico, por lo tanto, el control "*PL Ángulo*" controla el ángulo de inclinación de la recta del Posicionador Láser, "*SA Ángulo*" controla el ángulo de inclinación de la recta del Sensor de Apertura y el control "*Distancia*" controla la distancia proyectada de ambas rectas
5. **INDICADORES:** Hay tres indicadores útiles para conocer los resultados de cada mapeo digital cuando se usa el Algoritmo Genético. El indicador "*Ángulo Óptimo*" provee el ángulo óptimo del mapeo digital realizado, el indicador "*Tiempo*" provee la información de la duración y el indicador "*Contador*" muestra la cantidad de puntos detectados de la figura.
6. **CONTROLES:** Existe nueve controles para activar y desactivar diferentes funciones. El primer control es "*LÍMITE VISIÓN*", tiene como función realizar la simulación de visión del área de trabajo del Prototipo, es decir, el área en donde el Prototipo es capaz de realizar una detección sin problema alguno, si se encuentra fuera de estos límites no será capaz de realizar una detección. En la Figura 4.3.5 se muestra la simulación del área de trabajo. Es importante señalar que el Posicionador Láser y Sensor de Apertura tiene su área de trabajo y está representado con el color verde. Mientras la intersección entre las dos rectas estén ubicadas dentro de zona verde se puede realizar la detección sin ningún problema. La función se programó con el objetivo de asegurarnos en la simulación de las figuras estén ubicadas dentro del área de trabajo.



**Figura 4.3.5** Simulación de visión del área de trabajo del Prototipo.

El segundo control se llama “*ACTIVAR PL*”, activa y desactiva la simulación del Posicionador Láser. El control cuando se activa o desactiva (dependiendo sea el caso), muestra una leyenda para indicar su estado, esta leyenda se muestra como “*ACTIVAR PL*” o “*DESACTIVAR PL*”. El tercer control. “*ALGORITMO GENÉTICO*” activa y desactiva la opción del Algoritmo Genético para obtener un ángulo óptimo en la simulación. El cuatro control “*ACTIVAR SA*” activa y desactiva la simulación del Sensor de Apertura, funciona al igual que el segundo control, sus leyendas son “*ACTIVAR SA*” y “*DESACTIVAR SA*”. El quinto control “*FIGURA*” activa y desactiva la opción de simular alguna figura geométrica. El sexto control “*ALEATORIO*” activa y desactiva generar la posición, dimensión y forma de las figuras geométricas de manera aleatoria. El séptimo control “*DETECCIÓN*” se encarga de activar y desactivar la detección automática del Posicionador Láser. El octavo control “*OPCIONES*” se deja con la posibilidad de agregar más opciones a futuro. El noveno control “*SALIR*” es el paro de salida de la simulación, presionando este control detiene toda ejecución

7. *DISTANCIAS*: Despliega seis indicadores para mostrar las diferentes distancias programadas para las rectas del Posicionador Láser, Figura y Sensor de Apertura. Las dos distancias más importantes de esta sección son los indicadores “*X*” y “*Y*”, porque son las coordenadas de la intersección de las rectas del Posicionador y la recta de la Figura. EL indicador “*Y*” muestra la distancia que hay entre la figura y el sistema.

#### **4.3.6 DESCRIPCIÓN DEL CÓDIGO DEL SIMULADOR DE ESCENARIOS VIRTUALES**

En esta sección se describe el código del Simulador de Escenarios Virtuales. Al igual que los otros programas, la codificación se realiza en el Diagrama de Bloques del Software LabVIEW 2013. En la Figura 4.3.6 se muestra el código que consta de ocho partes importantes. La mayoría del código contiene las funciones del SubVI para simplificar el código.

1. *INICIALIZACIÓN*: En estas parte se tienen cuatro diferentes instrucciones, tal y como se muestras en la Figura 4.3.7. La Figura A representa la programación inicial, esta realiza un ajuste en dos controles, el primero ajusta los límites del área de trabajo del Posicionador Láser y el Sensor de Apertura (Punto 6 del Apartado 4.3.5). El segundo ajusta los valores predeterminados de los controles de “*PARÁMETROS DE FIGURA*” (Punto 3 del Apartado 4.3.5). La Figura B grafica dos círculos de color azul para simular el Posicionador Láser y

elSensor de Apertura respectivamente. La Figura C grafica un rectángulo gris para representar el Escáner Láser. La Figura D ajusta las dimensiones de la gráfica.

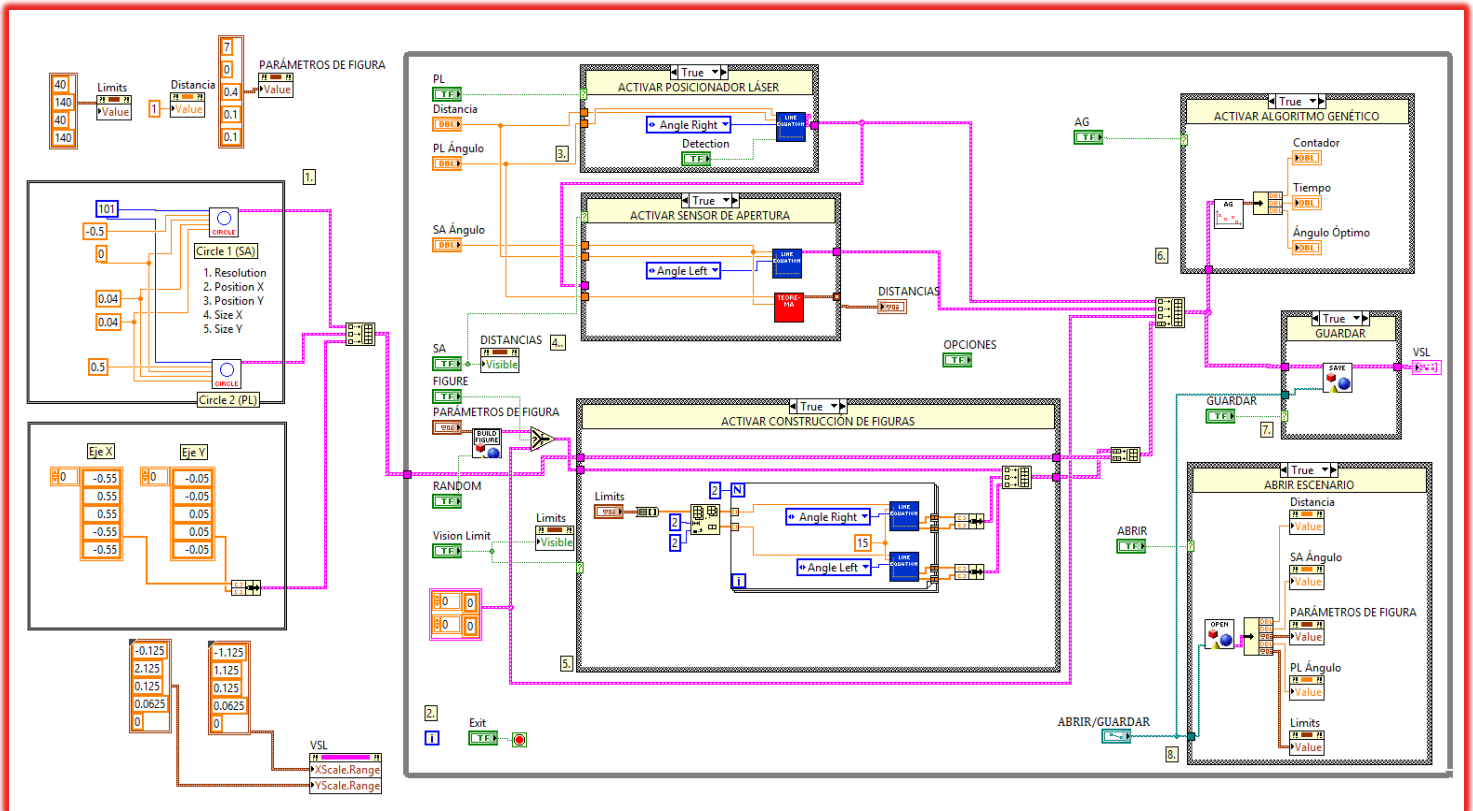


Figura 4.3.6 Diagrama de Bloques del Simulador de Escenarios Virtuales.

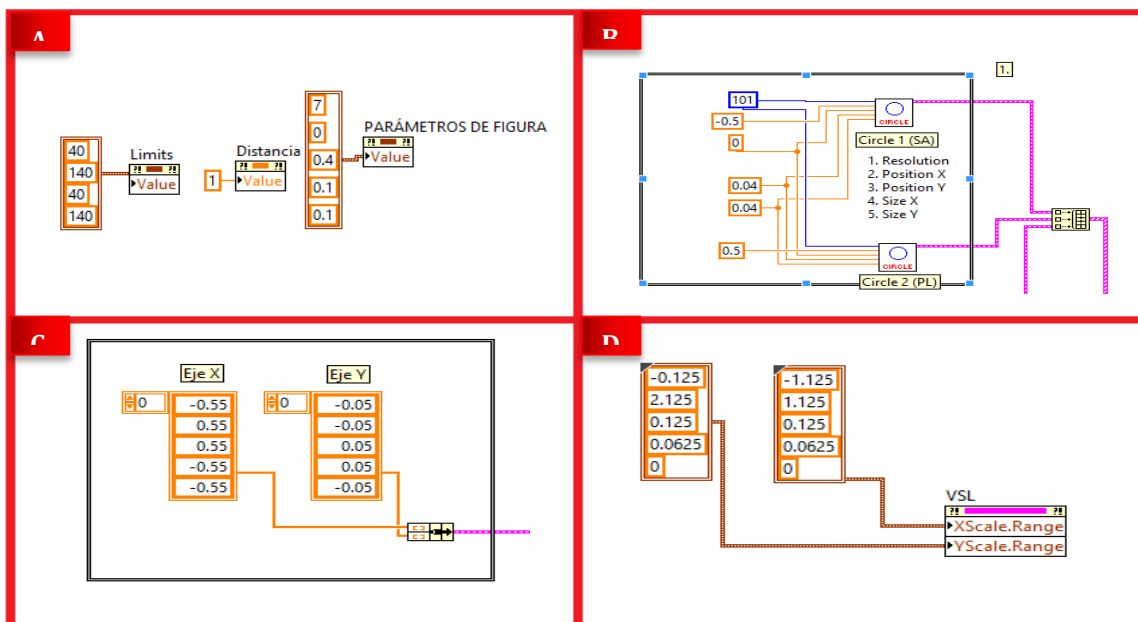


Figura 4.3.7 Código de la inicialización de los parámetros de la gráfica.

1. **CICLO PRINCIPAL "WHILE LOOP":** Realiza la ejecución de todas las instrucciones y simulaciones del programa a través de un ciclo "While Loop". Este ciclo se ejecutará indefinidamente hasta que se oprima el control "SALIR" para terminar con su ejecución.
2. **SIMULACIÓN DEL POSICIONADOR LÁSER:** Esta regida por el control "PL" para activar y desactivar la simulación del Posionador Láser (Ver Figura 4.3.8). Dependiendo del estado booleano del control se podrá habilitar o no la función "Case Structure" mediante el SubVI "Line Equation". El SubVi recibe cuatro datos para realizar su operación. Los dos primero son datos numéricos, la distancia y el ángulo que se desea configurar para simulación. El otro da dos opciones de selección, las cuales son la simulación del Posicionador Láser o del Sensor de Apertura. Esta opción es requerido porque el SubVI se utiliza para ambos simulaciones, la diferencia esta en la selección deseada. Otra diferencia es la detección automática del Posicionador Láser. Esta opción solo se puede habilitar en esta simulacion.

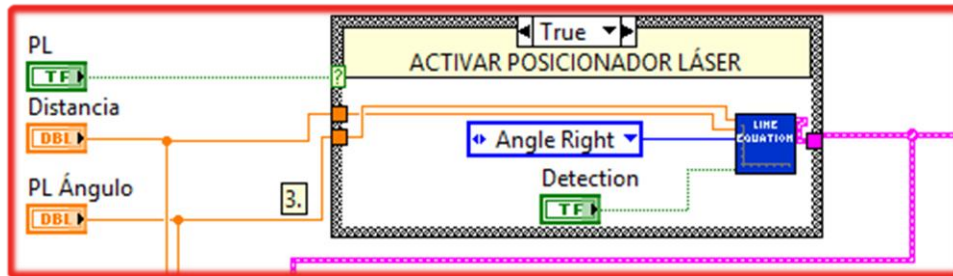


Figura 4.3.8 Código de la Simulación del Posicionador Láser.

3. **SIMULACIÓN DEL SENSOR DE APERTURA:** Esta regida por el control "SA" para activar y desactivar la simulación del Sensor de Apertura (Ver Figura 4.3.9). Dependiendo del estado booleano del control se podrá habilitar o no la función "Case Structure" mediante el SubVI. En este solo hay dos controles.

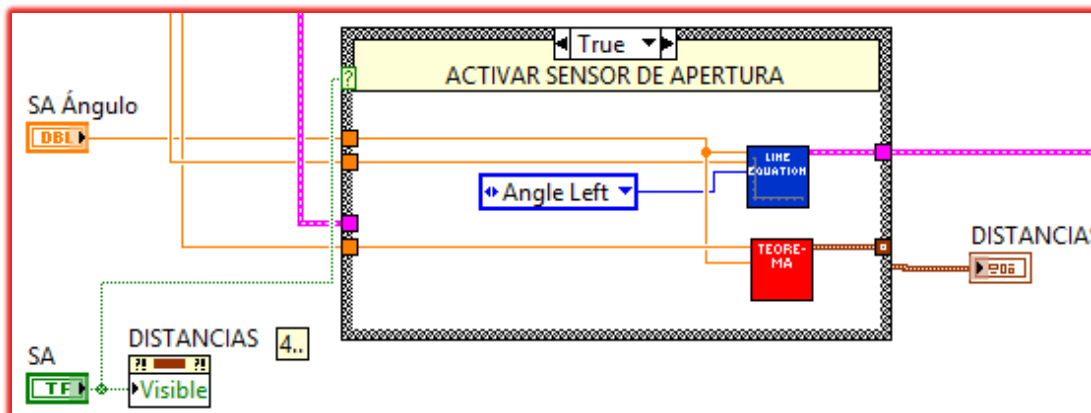


Figura 4.3.9 Código de la Simulación del Sensor de Apertura.

4. *SIMULACIÓN DE FIGURAS GEOMÉTRICAS*: En esta parte permite realizar la construcción de figuras geométricas (Ver Figura 4.3.10). El Botón “FIGURE” habilita la simulación. En esta parte están programadas todos los parámetros del escenario. También se realiza la simulación de límites de trabajo del Posicionador Láser y Sensor de Apertura. Aparte se habilita la opción de generar aleatoriamente las figuras con el control de “RANDOM”.

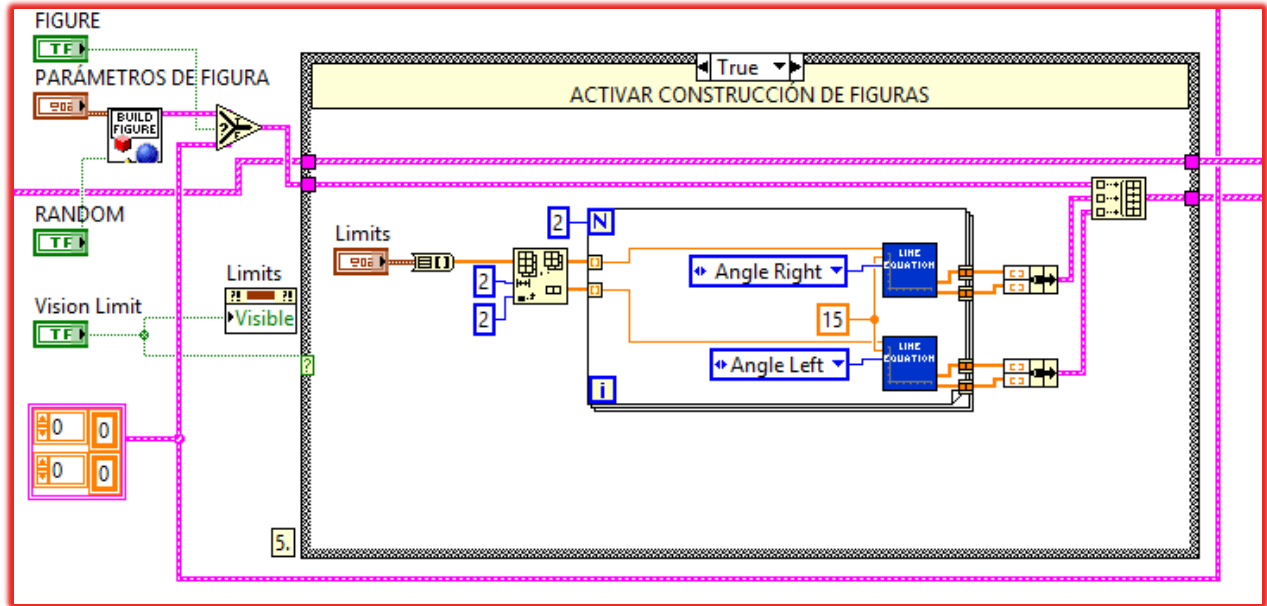


Figura 4.3.10 Código de la Simulación de Figuras.

5. *ALGORITMO GENÉTICO*: Es considerado la parte fundamental del Simulador de Escenarios Virtuales. El control “AG” habilita el Algoritmo Genético mediante la función “Case Structure”. El SubVI contiene una entrada y una salida. La entrada necesita toda la información de la simulación, la cual está almacenada en el conector rosa para ser procesado por el Algoritmo Genético. La salida provee el resultado del procesamiento del Algoritmo Genético, dando como resultado tres indicadores “Contador” “Tiempo” y el más importante “Ángulo Óptimo”

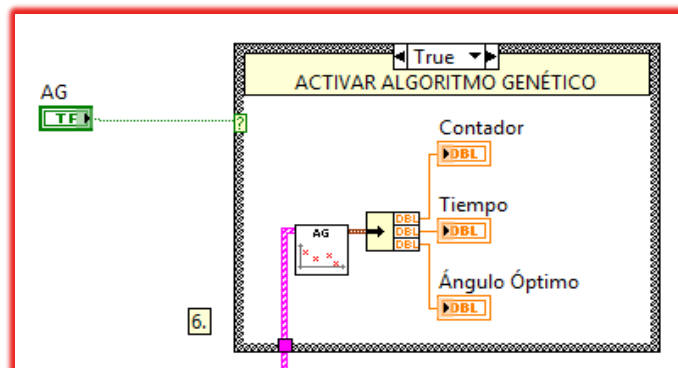


Figura 4.3.11 Código de la Activación del Algoritmo Genético.

- GUARDAR ESCENARIO: El control "GUARDAR" permite activar la función "Case Structure" para almacenar todas las características del escenario en una dirección de memoria (Ver Figura 4.3.12).

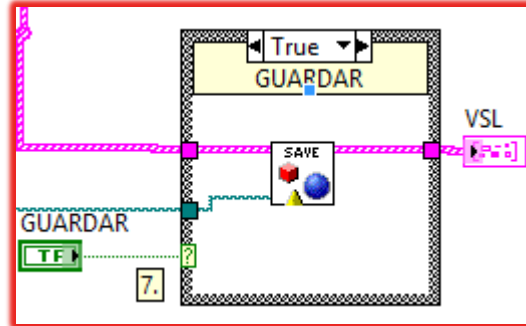


Figura 4.3.12 Código de la función guardar un Escenario

- ABRIR ESCENARIO: El control "ABRIR" permite activar la función "Case Structure" para leer todas las características de un escenario guardado en una dirección de memoria (Ver Figura 4.1.13).

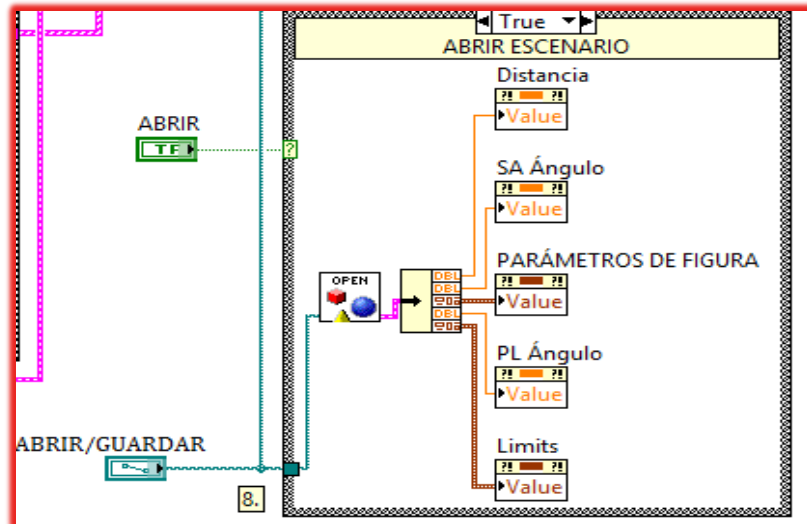


Figura 4.3.13 Código de la función abrir un Escenario Guardado.

## CAPÍTULO 5 - RESULTADOS DE APLICACIÓN DE HERRAMIENTAS COMPUTACIONALES PARA OPTIMIZAR EL PASO DE BARRIDO.

Los programas mencionados en el Capítulo anterior fueron diseñados por tres razones. La primera es para analizar, experimentar y aprender acerca del comportamiento de las distintas técnicas del Algoritmo Genético. La segunda razón y más importante es para optimizar el ángulo de paso de barrido del Prototipo TVS.

Durante el análisis y experimentación del Prototipo TVS concluí que el sistema no tiene las condiciones adecuadas para que el Algoritmo Genético se puede acoplar de manera satisfactoria. Las condiciones electrónicas y mecánicas no permiten al Posicionador Láser cambiar de velocidad dependiendo del ángulo seleccionado. Lo ideal para el movimiento del Posicionador Láser es tener una velocidad mayor para un valor de ángulo de paso de barrido considera mayor y una velocidad menor para un ángulo de paso de barrido considerado menor. La velocidad actual del Prototipo lo realiza a una velocidad constante, no importa que ángulo se seleccione, la velocidad del traslado del Posicionador Láser siempre será la misma. Cuando el Algoritmo Genético se someta a pruebas, difícilmente optimizará la velocidad de barrido por las condiciones actuales del Prototipo. Entonces, es factible evaluar nuestro Algoritmo Genético para hacer cambios en este sistema, con el objetivo de optimizar su modo de trabajo.

Por esa razón, decidí diseñar una simulación llamada Simulador de Escenarios Virtuales, la cual simula la funcionalidad del paso de barrido del Prototipo junto con las condiciones comunes de un escenario real. La idea principal de este desarrollo es crear una serie de pruebas para evaluar y analizar las soluciones óptimas del Algoritmo Genético. Con el comportamiento del Algoritmo Genético finalizado, puede ser capaz de desenvolverse en cualquier escenario real.

Las recomendaciones electrónicas del Prototipo TVS mencionadas en el Capítulo 3, está siendo atendidas por el Estudiante de Doctorado Lars Lindner, su desarrollo consta de un nuevo Prototipo para mejorar las condiciones mecánicas y electrónicas del actual sistema [37]. Por consiguiente, el Pseudocódigo y las técnicas seleccionadas del Algoritmo Genético se pueden integrar a su nuevo prototipo sin importar que tipo de Lenguaje de Programación use al momento de programar su funcionamiento. En caso de seleccionar otro lenguaje de programación, se debe programar en base al pseudocódigo desarrollado en esta Investigación de Tesis.

## 5.1 RESULTADOS DE ALGORITMO GENÉTICO ROBUSTO

El Algoritmo Genético proporciona soluciones óptimas para cualquier problema real. Las soluciones para este problema de tesis es obtener un conjunto de valores de ángulos considerados óptimos para realizar el paso de barrido (mapeo digital) de manera balanceada con respecto a las variables de Calidad de Información (*CI*) y el tiempo de Finalización (*TF*) durante el proceso del mapeo digital para cualquier escenario.

Para realizar el funcionamiento del Algoritmo Genético, lo primero que se debe tener son datos que procesar. Los datos para este algoritmo consisten de tres variables obtenidas al momento de realizar un mapeo digital de un escenario.

La primera variable siempre se propone su valor de ángulo para realizar el movimiento angular del Posicionador Láser y continuar con en el ángulo de paso de barrido del mapeo digital. En la finalización del mapeo digital se obtiene datos de dos variables, las cuales son Calidad de Información y Tiempo de Finalización (Como se describió en el Capítulo 4). La Figura 5.1.0 representa el proceso del mapeo digital.



**Figura 5.1.0** Variables del Mapeo Digital.

Cada mapeo digital realizado por el Prototipo TVS o la Simulación de Escenarios Virtuales, dan como resultado tres variables que son consideradas como datos. Los valores de las dos variables de salida (*CI* y *TF*) dependen del valor del ángulo seleccionado y las características del escenario. La función objetivo (Ver ecuaciones 4.3.0, 4.3.1 y 4.3.2) se apoya de las tres variables mencionadas. Por lo tanto, las variables son sumamente necesarios para el funcionamiento del Algoritmo Genético.

El Algoritmo Genético empieza con una serie de individuos, en este caso, son valores de ángulos para formalizar una población, la cual será sometida a varias generaciones hasta encontrar el individuo o el ángulo óptimo. Los individuos se irán generando de forma aleatoria, por lo tanto, cada individuo generado debe de buscar su complemento de las dos variables de salida para continuar con su proceso.

En la Tabla 5.1.0 se muestran como ejemplo 10 ángulos diferentes con sus variables de salida. Supongamos que el Algoritmo Genético generó el ángulo 1.2°, el programa será capaz de acceder a los datos del mapeo digital y buscar su posición en la tabla de las variables que necesita para continuar, la fila subrayado con amarillo muestra el ángulo que coincide con el ángulo supuesto. Entonces todo los datos que están en el posición 5 (index), se van a someter a las etapas del Algoritmo Genético para ir evaluando los datos con la mejor variable,

INDEX	ÁNGULO (°)	TIEMPO DE FINALIZACIÓN (seg)	CALIDAD DE INFORMACIÓN(cantidad de puntos detectados)
0	0.6	0.267	90
1	0.7	0.215	72
2	0.8	0.18	60
3	1.0	0.155	53
4	1.1	0.136	44
5	1.3	0.121	42
6	1.4	0.109	37
7	1.6	0.099	33
8	1.7	0.091	30
9	1.9	0.084	29

**Tabla 5.1.0** Ejemplo de Datos de un Mapeo Digital.

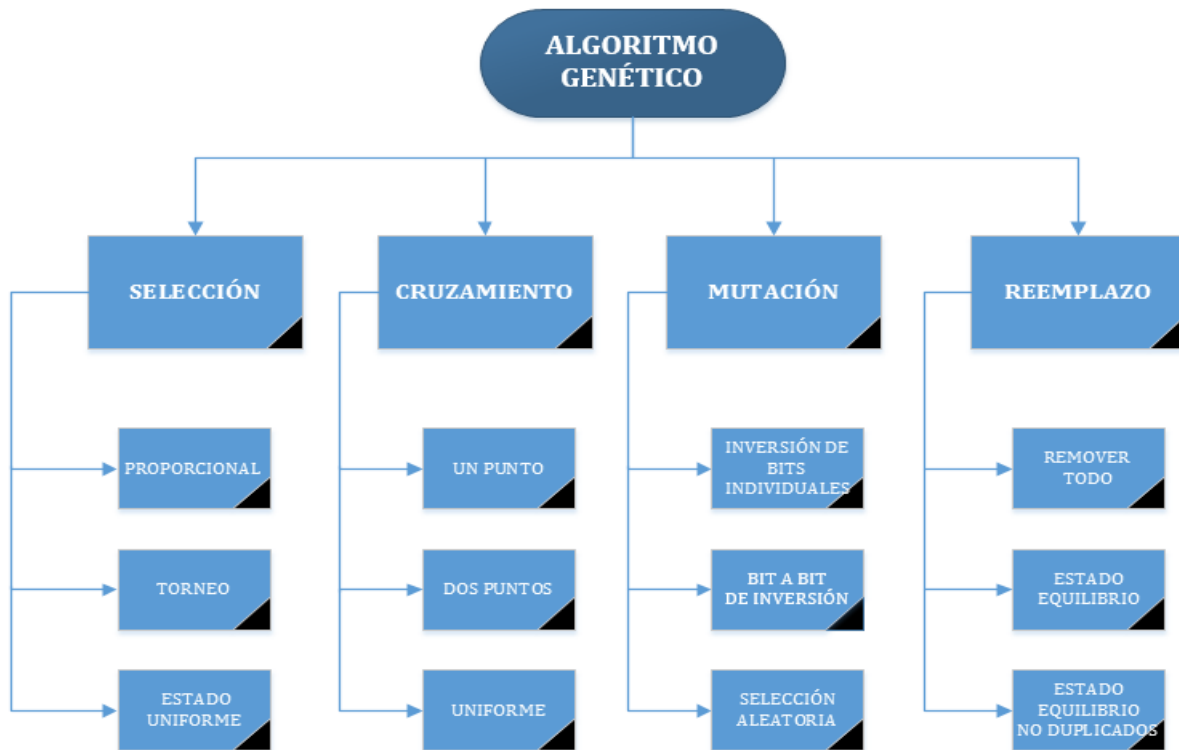
Los datos del mapeo digital proceden del Simulador de Escenarios Virtuales, por lo tanto, con la estructura de los datos completa podemos continuar con la ejecución del Algoritmo Genético y la etapa de evaluación de las diferentes técnicas presentadas en el apartado 2.6. Recordando, las técnicas del Algoritmo Genético pueden llegar a presentar un comportamiento distinto durante la ejecución del programa y también en las soluciones óptimas que proporciona.

En la construcción de mi Algoritmo Genético incluí las técnicas más comunes y recomendadas por expertos en el tema [24] [29]. Como se describió en el apartado 4.1.2, las técnicas de cada etapa

del Algoritmo Genético fueron incluidas en una función “*Case Structure*” para diferenciarlas y tener acceso a ellas cuando fuera necesario.

La selección de técnicas se realiza a través de los controles principales del Panel Frontal del programa, esta selección genera una combinación de técnicas que se van a evaluar. En la Figura 5.1.1 se presentan 12 técnicas que están programadas dentro del Algoritmo Genético.

Las 12 técnicas están divididas en cada etapa del Algoritmo Genético, por lo tanto, cada etapa contiene tres técnicas que son exclusivamente utilizadas en su etapa correspondiente. Existen 81 combinaciones para evaluar y encontrar las mejores cinco combinaciones que presenten las mejores soluciones óptimas.



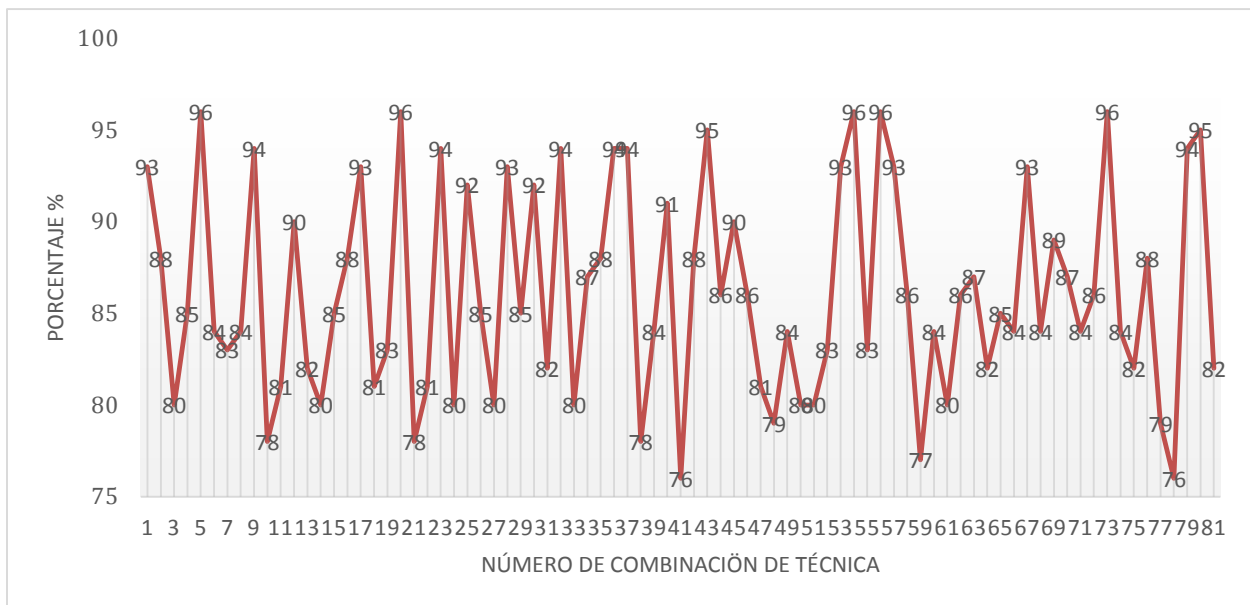
**Figura 5.1.1** Diagrama de las diferentes técnicas del Algoritmo Genético.

Las 81 combinaciones están presentadas en Tabla A-2 de la sección de Anexos. La evaluación de las combinaciones de las técnicas consistió en someter a cada combinación a 100 pruebas para analizar el comportamiento de las soluciones óptimas con respecto a la función objetivo y las ecuaciones 4.1.0, 4.1.1 y 4.1.2 presentadas en el apartado 4.1.2.

Cada prueba contiene un escenario con 100 ángulos diferentes para realizar el escaneo virtual. Como sabemos, en un escaneo virtual se propone valores de ángulos diferentes entre sí para iniciar con el ángulo de paso de barrido del Posicionador Láser, con el fin de obtener los valores de dos variables (Calidad de Información y Tiempo de Finalización) que definan el número de puntos detectados y la duración del escaneo. Por lo tanto, cada escenario tuvo un rango de valor de ángulo de  $0.1^\circ$  a  $10^\circ$ , su resolución del ángulo fue de  $0.1^\circ$ . Por esta razón se hicieron 100 ángulos diferentes en cada escenario.

La Figura 5.1.2 muestra el contenido de los resultados del porcentaje de eficiencia (*PF*) de las 81 combinaciones. El porcentaje de Eficiencia, en este caso se define como el porcentaje en acercarse a la solución óptima de los datos procesados. Claro está, los resultados provenientes del Algoritmo Genético se consideran soluciones óptimas, pero no significa en realidad lo sea, dependiendo de la combinación de técnicas puede encontrar la verdadera solución o no.

En caso de no encontrarlo, la solución óptima presentada, es muy posible sea la más cercana a la solución óptima verdadera. Entonces, la gráfica muestra como las diferentes combinaciones en su mayoría encontraron la verdadera solución óptima de las 100 pruebas realizadas mediante un valor de porcentaje



**Figura 5.1.2** Gráfica del comportamiento de las 81 combinaciones de Técnicas del Algoritmo Genético.

En La Tabla 5.1.1 se presentan los resultados de las cinco mejores combinaciones de técnicas realizadas durante la Evaluación. La combinaciones 72, 78, 73, 35 y 9 tienen un porcentaje de eficiencia mayor a 95% a comparación de las otras combinaciones de técnicas. La combinación con mayor porcentaje es la 72 con un valor de 96.8%. Esto significa lo siguiente, de las 100 pruebas sometidas en las diferentes combinaciones, la número 72 tuvo mayor porcentaje en proporcionar la solución óptima verdadera.

COMBINACIÓN	SELECCIÓN	CRUZAMIENTO	MUTACIÓN	REEMPLAZO	PORCENTAJE
72	UNIFORME	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS	96.9%
78	UNIFORME	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS	95.8%
73	UNIFORME	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO	95.7%
35	TORNEO	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO	95.5%
9	PROPORCIONAL	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS	95.1%

**Tabla 5.1.1** Ranking de las mejoras 5 combinaciones evaluadas.

Teniendo la combinación de técnica ya definida, la cual proporciona mayor precisión en la búsqueda de solución óptima de ángulos, podemos continuar ahora en la búsqueda de un valor de ángulo óptimo que pueda ser adaptado para cualquier escenario que cumpla con las siguientes características descritas a continuación:

1. La cantidad de objeto serán 1, 2 o 3 dentro del escenario.
2. La Dimensión de un objeto se catalogará con respecto al porcentaje de la dimensión de longitud del escáner láser. Su dimensión es de 1m, por lo tanto, en base a esa medición los porcentajes catalogados serán 25%, 50%, 75% y 100%.
3. La distancia entre el Objeto y el Escáner Láser se catalogará en 0.5m, 1m y 1.5m.

Cualquier escenario que cumpla con alguna combinación de los puntos mencionados, se podrá tener acceso a utilizar un ángulo ya definido para poder iniciar con el escaneo. Las pruebas para conseguir esté ángulo se hicieron bajo las características mencionadas y usando la combinación de técnicas ganadora. Los resultados de los ángulos óptimos son presentados en la Tabla 5.1.2

DISTANCIA (m)	DIMENSIÓN (%)	CANTIDAD DE. OBJETOS	ÁNGULO ÓPTIMO (°)
0.5	25	1	1.5
		2	1.2
		3	1.2
	50	1	1.0
		2	1.8
		3	1.8
	75	1	2.1
		2	1.9
		3	2.0
	100	1	2.2
		2	2.5
		3	2.4
1	25	1	4.1
		2	3.9
		3	4.3
	50	1	4.1
		2	4.2
		3	4.4
	75	1	5.1
		2	4.9
		3	5.2
	100	1	5.4
		2	5.4
		3	5.4
1.5	25	1	5.9
		2	6.1
		3	6.1
	50	1	6.4
		2	6.7
		3	6.8
	75	1	8.0
		2	8.7
		3	8.6
	100	1	8.5
		2	9.1
		3	9

**Tabla 5.1.2** Ángulos Óptimos propuestos para los diferentes escenarios comunes.

De lo datos presentados de la tabla anterior, propongo tres ángulos para utilizar y adaptar en el escaneo láser de cualquier escenario que cumpla con las características comunes postuladas en el apartado 5.1 En la Tabla 5.1.3 se observa los tres ángulos óptimos..

DISTANCIA (m)	DIMENSIÓN (%)	CANTIDAD DE. OBJETOS	ÁNGULO ÓPTIMO (°)
0.5	50	3	1.8
1	50	3	4.4
1.5	50	3	6.8

**Tabla 5.1.3** Tabla de los principales ángulos óptimos propuestos.

Dichos ángulos son propuestos y seleccionados en base a su distancia entre el objeto y sistema láser simulado, el porcentaje de la dimensión con respecto al sistema láser y la cantidad de objetos. Los ángulos propuestos tendrán un buen comportamiento en el escaneo virtual debido a que sus valores de ángulo son cercanos a las otras características.

## 5.2 RESULTADOS DE LA MEJORA DE LA INTERFAZ DEL PROTOTIPO TVS

En la mejora de la Interfaz Gráfica del Prototipo agregue nuevas funciones para desempeñar tareas relacionadas con el área de Navegación Autónoma. La función principal programada fue el control del Posicionador Láser para ejecutar un barrido o escaneo de forma automática en todo el escenario, tal y como lo ejecuta el Simulador de Escenarios Virtuales (Ver Figura 5.3.0). También corregí en el código cuestiones de programación. El programa de la Interfaz tenía cuatro problemas principales al momento de ejecutarlo:

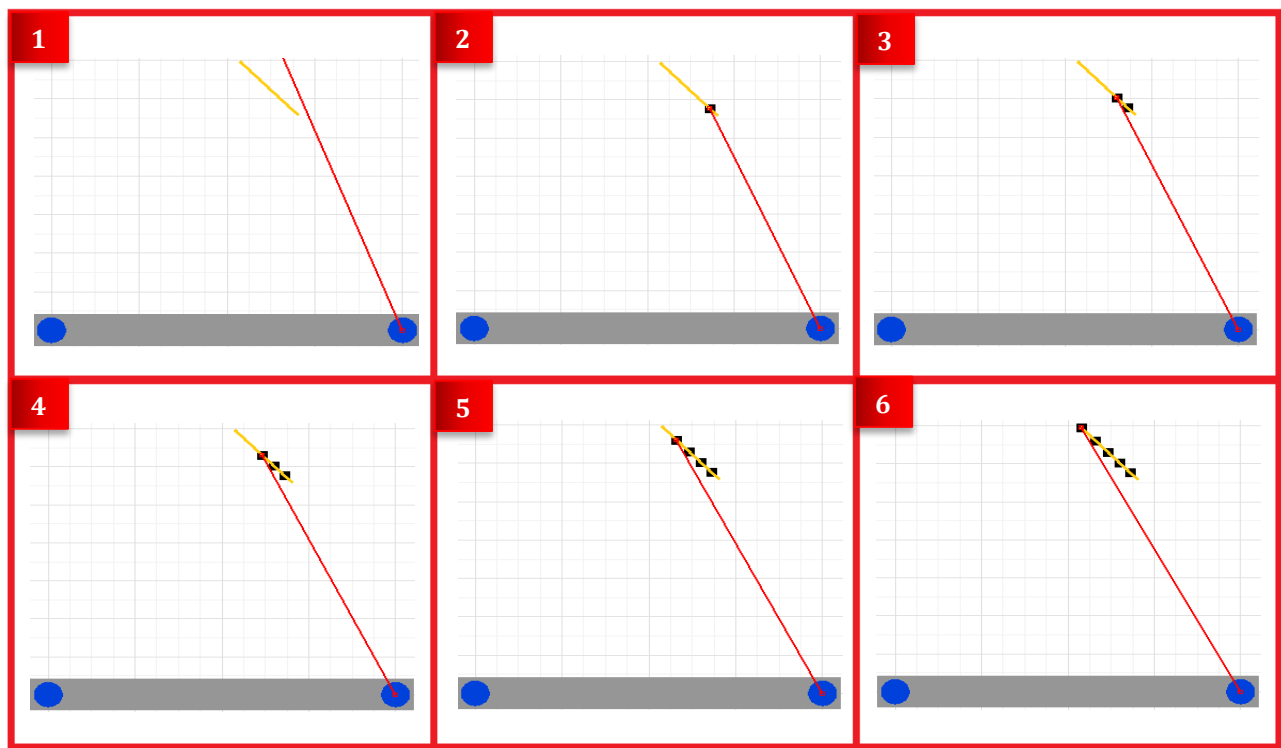
1. El programa dejaba de funcionar sin razón alguna.
2. El programa ejecutaba una instrucción diferente a la deseada. Ejemplo, se activaba el botón hacia la derecha, en vez de eso, se ejecutaba la instrucción que activaba el sentido contrario al deseado.
3. Algunas instrucciones cuando se activaban se demoraban en ejecutarse (Aproximadamente 1 segundo de retraso).
4. El puerto serial se inhabilita sin razón alguna, y el programa dejaba de funcionar.

Para comprobar la efectividad de la mejora acerca de los tres primeros puntos, se puso a prueba el funcionamiento del programa de la interfaz “*versión 2.0*” durante 4 horas seguidas por 5 días. El programa no presentó ningún error durante su ejecución y sus instrucciones cuando se activaban se ejecutaban al instante.

El Punto 3 su falla se presentaba principalmente con problemas de compatibilidad entre el Sistema Operativo de Windows 8.1 y el driver del cable convertidor RS-232 a USB. La solución fue cambiar el cable convertidor por el modelo U209-000-R y mejorar aspectos de programación de las funciones de VISA, tal y como se muestra en la Figura 4.2.5 del apartado 4.2.2. Las mejoras están relacionadas en la parte del Diseño de Software del Prototipo TVS, mencionados en el Capítulo 3 y explicados en el Capítulo 4. El Programa tiene mejor comportamiento en su ejecución. La codificación de este programa simplifica instrucciones con la ayuda de funciones (SubVI, Property, Flat Sequence) que no estaban implementadas en el antiguo código. La funcionalidad del programa se puede asegurar que es confiable, la ejecución de instrucciones es rápida al momento de realizar y el control de TVS lo realiza de manera decente con respecto a las condiciones mecánicas y electrónicas actuales.

### 5.3 RESULTADOS DE SIMULADOR DE ESCENARIOS VIRTUALES

El simulador de escenarios virtuales consiste en representar de modo virtual el comportamiento fundamental del Prototipo TVS y del escenario físico. El Posicionador Láser es el que realiza el escaneo virtual en esta simulación. En la Figura 5.3.0 podemos observar el funcionamiento del escaneo virtual. Recordando, el rectángulo gris representa el tubo principal del escáner láser. Los dos círculos azules representan el Sensor de Apertura (de izquierda a derecha) y el Posicionador Láser. La línea anaranjada simula la representación de la superficie de un objeto u obstáculo que se desea escanear. La línea roja representa el haz del láser proyectado hacia el plano.



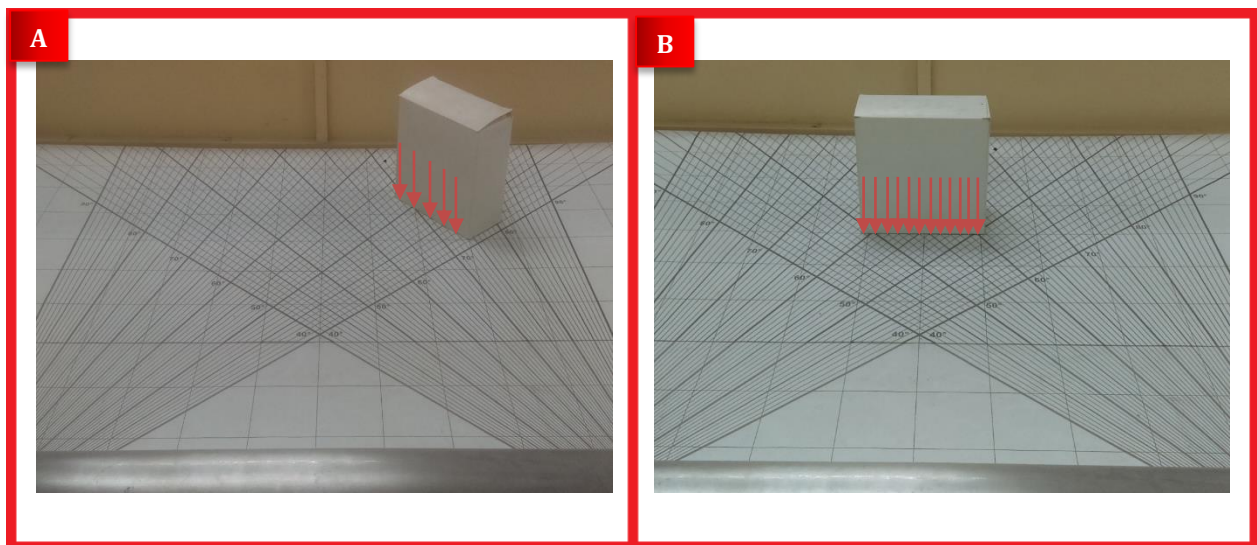
**Figura 5.3.0** Funcionamiento del Escaneo Virtual.

La Figura consta de 6 imágenes que representan cada una, el proceso de cómo va ejecutando el paso de barrido en el escenario, hasta conseguir la detección de todos los puntos posibles del objeto. Como ya sabemos, el número de puntos detectados en el objeto dependerá del ángulo seleccionado. La Figura muestra cinco puntos detectados en el objeto simulado, el cual es el valor de la variable de Calidad de Información (CI) postulada en el Capítulo 3.

Se realizaron pruebas para determinar el error de la variable de Calidad de Información, es decir, el error de detección entre en el escaneo simulado con respecto al escaneo real. La prueba más importante consistió en realizar 100 detecciones con el escáner real y otras 100 detecciones con el escáner virtual. El objeto a escanear tiene las mismas características con respecto a los dos tipos de escaneo, las cuales son:

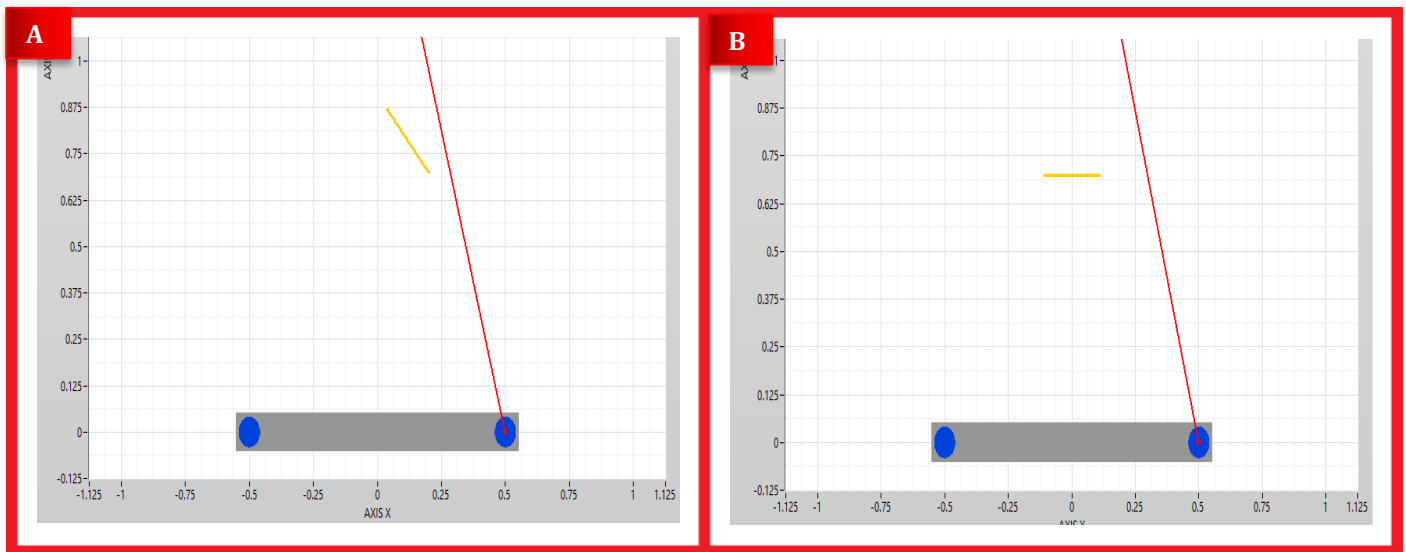
1. Coordenada de referencia de posición del objeto.
2. Posición angular dentro del Escenario
3. Tamaño del objeto. Para el escaneo real las dimensiones del objeto son Largo = 22cm; Ancho = 22cm; Altura = 10cm. Para el escaneo simulado solo se necesita una línea que represente la superficie del escenario, esta dimensión es el Largo de 22cm.

La prueba inicio con 100 detecciones con el Prototipo TVS. En la Figura 5.3.1 se muestra dos ejemplos de las 100 pruebas que se realizaron. Esta Figura contiene dos imágenes con posiciones diferentes del objeto dentro del escenario real. En la Figura A se muestra el objeto con una coordenada inicial en uno de sus esquinas de  $X = 0.2m$ ;  $Y = 0.7m$  (dónde  $X$  y  $Y$  son las coordenadas en 2D de la posición del objeto y  $m$  es la distancia en metros), y con una ángulo de inclinación de  $45^\circ$  con respecto al láser escáner. Las flechas rojas representan los cinco puntos detectados en el objeto. En la Figura B se muestra la misma figura, ahora la coordenada inicial es de  $X = 0.11m$ ;  $Y = 0.7m$  con una ángulo de inclinación de  $0^\circ$  con respecto al láser escáner. La cantidad de puntos detectados para esta prueba son de 12 puntos.

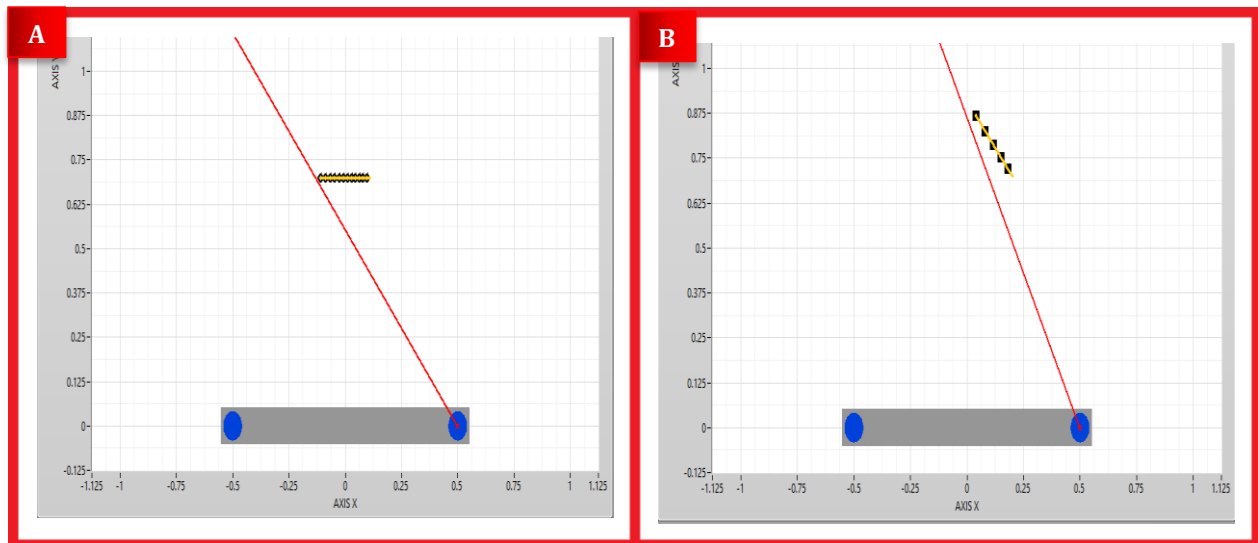


**Figura 5.3.1** Posición y Tamaño del Objeto en el Escenario Real.

Una vez que las 100 pruebas se finalizaron en la parte del escaneo del Prototipo TVS. Las pruebas siguientes fueron para el escaneo del Simulador de Escenarios Virtuales. Como se mencionó, las dimensiones y posiciones del objeto en cada prueba son igual para ambos escaneos. Por lo tanto, se presenta en la Figura 5.3.2 dos imágenes con las mismas características que la Figura 5.3.1. Es importante resaltar que ambas imágenes tiene su escala métrica, por lo tanto, en ambas imágenes se encuentra en la parte inferior su escala del Eje X y en la parte izquierda su escala del Eje Y. Representando de esta forma las mismas dimensiones del Escenario real con respecto al escenario virtual.



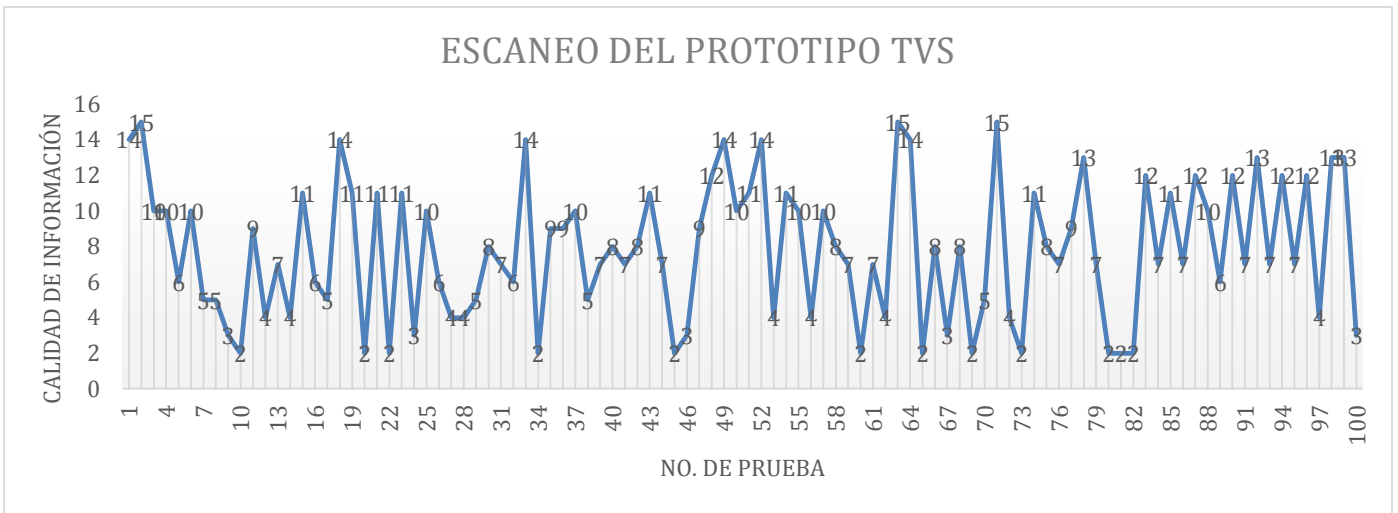
**Figura 5.3.2** Posición y Tamaño del Objeto en el Escenario Virtual.



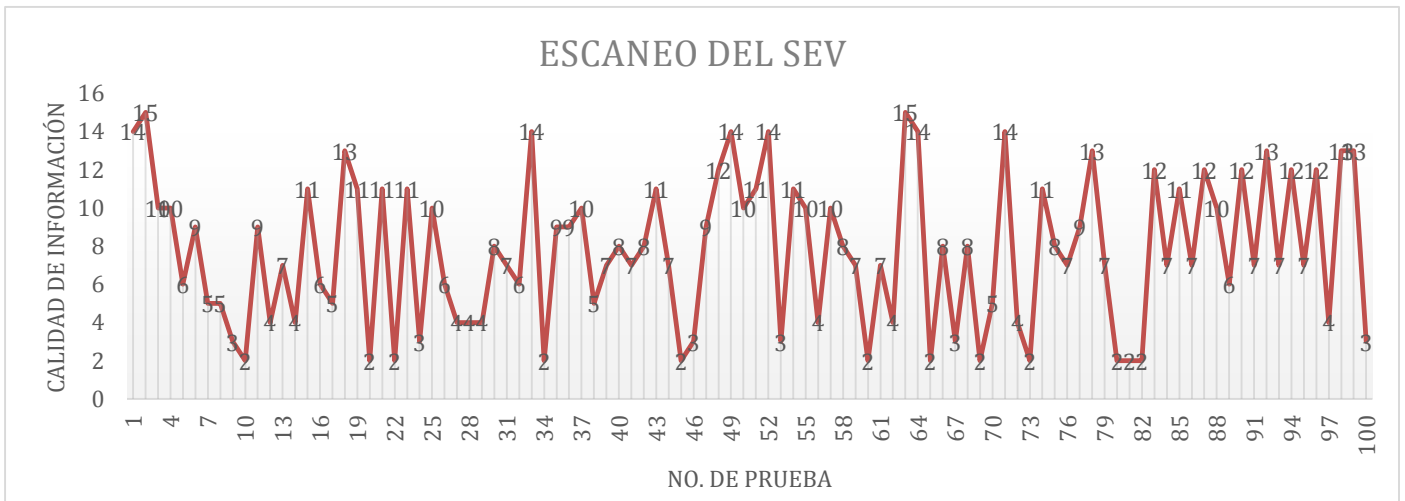
**Figura 5.3.3** Escaneo Virtual de objeto en el Simulador de Escenarios Virtuales..

En la Figura 5.3.3 se muestra el escaneo virtual del objeto. La Figura A muestra los 12 puntos detectados por el escáner virtual y la Figura B muestra los 5 puntos detectados. Los resultados de los puntos detectados en las pruebas presentadas del escaneo del Prototipo TVS coinciden con los resultados de esta simulación. Por lo tanto, en estas dos pruebas el margen de error fue 0% en ambos escaneos.

Los resultados de las 100 pruebas para cada escaneo se presentan en las siguientes figuras. La Figura 5.3.4 muestra una gráfica de los resultados de puntos detectados (Calidad de Información) en cada prueba del escaneo del Prototipo TVS y la Figura 5.3.5 los resultados del Simulador de Escenarios Virtuales (SEV).



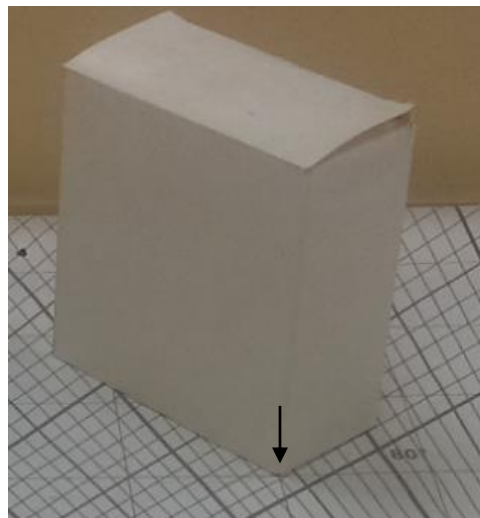
**Figura 5.3.4** Gráfica del Escaneo del Prototipo TVS.



**Figura 5.3.5** Gráfica del Escaneo del Simulador de Escenarios Virtuales.

Ambas gráficas contienen en la parte izquierda el Eje X para representar la cantidad de puntos detectados de cada objeto y en la parte inferior el Eje Y para representar el número de prueba realizada. De las 100 pruebas realizadas durante escaneo del simulador solo 6 pruebas (Prueba 6, 18,29, 53 y 71) no coincidieron con las pruebas del escaneo del Prototipo TVS. La Diferencia de puntos detectados de las cinco pruebas fue solo de un punto detectado.

Con los datos presentados en las gráficas, el margen de error es del 1% en el escaneo del Simulador de Escenarios Virtuales con respecto al escaneo del Prototipo TVS. No obstante, en las cinco pruebas que no coincidieron, su diferencia entre puntos detectados fue solamente de uno. Se analizó las condiciones de las cinco pruebas durante la ejecución del escaneo real y el simulador. Se observó en los puntos que no detectó el Simulador, son puntos localizados casi en el borde del objeto durante el escaneo del Prototipo TVS. Se puede apreciar en la Figura 5.3.6 el caso mencionado, la flecha negra indica una detección por parte del TVS, pero el simulador no la registró, esto es ocasionado al momento de posicionar físicamente el objeto, porque hubo un error entre la posición física y la posición virtual, siendo más exacta la posición virtual. Por ende, Prototipo TVS si detectó este mala colocación del TVS y el simulador no lo hizo porque su posición fue tal como se programó. Por lo tanto, podemos concluir en: si la posición física del objeto hubiera sido exacta, el margen de error hubiera sido 0% entre ambos sistemas. Dando como resultado una simulación de la confiable y precisa.



**Figura 5.3.6** Objeto Detectado por el Prototipo TVS.

## CAPÍTULO 6 – CONCLUSIONES

El objetivo principal de la investigación de tesis es optimizar el ángulo de paso de barrido del prototipo TVS mediante una herramienta de búsqueda llamada Algoritmo Genético. La herramienta proporciona una solución óptima para cualquier problema real, en este caso, mi problema se define en encontrar un valor de ángulo de paso de barrido con la capacidad de interactuar en cualquier escenario, con el fin de obtener un comportamiento equilibrado entre la cantidad de detección de puntos de una superficie de cualquier obstáculo y el tiempo que realiza en hacer el recorrido del barrido. Como los resultados de presente trabajo fue logrado lo siguiente:

1. Realizado la construcción de un Algoritmo Genético Robusto, con el fin de analizar y evaluar las diferentes técnicas explicadas en el apartado 2.6.2, 2.6.3, 2.6.4 y 2.6.5, El análisis y evaluación de las combinaciones de técnicas del Algoritmo Genético se hizo con respecto al Porcentaje de Eficiencia, como se mencionó, este porcentaje consiste en el valor de obtener con mayor cantidad la solución óptima o las mejores soluciones. De las 81 combinaciones generadas, la combinación número 72 tuvo mejor compartimento con el 96.9% de eficiencia. Esta combinación contiene las técnicas de Selección Uniforme, Cruzamiento a dos puntos, Mutación Aleatoria, y Reemplazo de Estado Equilibrio No Duplicados.
2. Los datos usados para evaluar las combinaciones fueron hechas a través del Simulador de Escenarios Virtuales, el cual es el producto original de esta tesis. Este desarrollo fue hecho con el fin de crear una herramienta de soporte y mejora para el Algoritmo Genético Robusto. El simulador consistió en realizar la simulación de tres secciones del Prototipo TVS, las cuales son: Simulación del Posicionamiento de Láser, Simulación de escenario, Simulación de Mapeo de objetos. La unión de las tres secciones hace posible la simulación general del escaneo de un objeto. Los resultados de cada escaneo es proporcionar los datos del ángulo de paso de barrido ( $A$ ), Calidad de Información ( $CI$ ) y Tiempo de Finalización ( $TF$ ) para ser procesados por el Algoritmo Genético Robusto mediante sus ecuaciones (Ver ecuaciones 4.1.0, 4.1.1 y 4.1.2) y funciones programadas (Ver apartado 4.1.2). La evaluación del Simulador de Escenarios Virtuales con respecto al comportamiento del Prototipo TVS y su escenario físico se realizó mediante 100 pruebas de posiciones diferentes de un objeto. Estas posiciones fueron iguales tanto en el escaneo virtual como el escaneo del Prototipo TVS. Los resultados de esta evaluación fueron del

- 1% error entre ambos escaneos. Como se describió en el apartado 5.3, este error fue originado por el incorrecto acomodo del objeto cuando se posiciono físicamente para ser evaluado por el Prototipo TVS. Sí el acomodo del objeto no hubiera sido incorrecto, es muy posible que el porcentaje hubiera sido menor o nulo.
3. El desarrollo de la mejora de la interfaz gráfica del Programa “*versión 2.0*” consiste principalmente en mejorar la codificación del programa “*v1.84*” y a su vez incluir en su código nuevas instrucciones. Durante el desarrollo de este programa se comprendió el funcionamiento electrónico, mecánico y el diseño del software del Prototipo TVS. Por lo tanto, en el Software se hicieron las siguientes mejoras:
    - a) Depuré el código para evitar eventos inesperados en la activación de instrucciones y detenciones durante su ejecución.
    - b) Evité el congelamiento del programa cuando se realiza la comunicación entre el puerto serial y la computadora, con el fin de adquirir los pulsos digitales del Foto-Transistor y el Opto-acoplador. Esto se hizo cambiando funciones en la programación y reemplazando el convertidor RS-232 a USB por el modelo U209-000-R.
    - c) Se incluyeron funciones, la principal es en el movimiento automático del Posicionamiento Láser. Con esta función el paso de barrido del escaneo láser puede realizar un reconocimiento del escenario de forma automatizada.
    - d) En el entorno visual del programa fue mejorado en su aspecto y simplificación de controles e indicadores.
  
  4. Con las condiciones electrónicas y mecánicas actuales (detalladas en el Capítulo 3) concluí en mejorar el Algoritmo Genético con mi diseño del Simulador de Escenarios Virtuales, en vez de utilizar el Prototipo TVS para cual fue pensando desde un principio. Como demostré en el Apartado 5.3, mi simulación tiene un gran porcentaje de coincidencia entre el comportamiento real y el virtual, por lo tanto, utilizar mi Algoritmo Genético Robusto en mi Simulador, podemos continuar con la tarea de mejorar y encontrar soluciones óptimas de valores de ángulo para un paso de barrido que se pueda adaptar a cualquier escenario. En el apartado 5.3 se mostró la Tabla 5.1.2, en ella se presentan los ángulos óptimos evaluados por el Algoritmo Genético Robusto bajo las características comunes de un escenario (explicadas en ese mismo apartado). En la Tabla 5.1.3 se proponen tres valores de ángulos, los cuales son 1.8°, 4.4° y 6.8°. Dichos ángulos son propuestos y seleccionados en base a su distancia entre el objeto y sistema láser

- simulado, el porcentaje de la dimensión con respecto al sistema láser y la cantidad de objetos. Los ángulos propuestos tendrán un buen comportamiento en el escaneo virtual debido a que sus valores de ángulo son cercanos a las otras características.
5. Recomiendo el uso del Algoritmo Genético en el Prototipo TVS cuando se realice las siguientes mejoras pendientes:
    - a) Mejorar el control de la velocidad del Posicionador Láser. Teniendo un control electrónico (no mecánico) que permita cambiar el ritmo de velocidad del Algoritmo Genético, se traduce en mejor comportamiento de la variable  $TF$  (Tiempo de Finalización) dentro del escaneo con el fin de tener mejores soluciones.
    - b) Utilizar tarjetas PCB para la instalación electrónica con el fin de evitar falsos contactos en el sistema.
    - c) Mejorar la funcionalidad y Calibración del Sistema de Apertura. Esto puede ser posible con el reemplazo de un fototransistor con mejores características electrónicas.
    - d) Reemplazar los juegos de engranes y motores a pasos por servomotores con la finalidad de evitar pérdida de pasos en los motores a pasos.
    - e) Integrar los dispositivos Encoder en el Posicionador Láser para tener una mejor retroalimentación de su posición angular.
  6. Una vez implementado las mejoras, considero factible integrar el Algoritmo Genético al programa de la interfaz "*versión 2.0*" para cumplir con dos cosas. Primero, comparar los resultados del Simulador de Escenarios Virtuales con respecto al Prototipo TVS. Segundo, continuar con el escaneo de objetos de forma dinámica, con el fin de evaluar los resultados de los ángulos propuestos por el Algoritmo Genético con este nuevo nivel de dificultad.
  7. Por último, se deja abierto la posibilidad de utilizarse el Algoritmo Genético para cualquier tipo de Lenguaje de Programación mediante el pseudocódigo presentado en el Capítulo 2 y en el apartado del 4.1.2. También se postuló la configuración de la selección de la combinación de técnicas ganadora Todo esto, con el fin de utilizarse en cualquier sistema de escaneo láser. Principalmente en el desarrollo del trabajo del estudiante de Doctorado Lars Lindner [37] [38].

## REFERENCIAS

- [1] Oleg Sergiyenko, Vira Tyrsa, Luis Carlos Basaca Preciado, Julio Cesar Rodriguez Quinonez, Wilmar Hernandez, Juan Ivan Nieto Hipolito, et al., Optoelectronic devices and properties. Electromechanical 3D optoelectronic scanners: resolution constraints and possible ways of improvement. Austria: IN-TECH; 2010 (p. 549–82).
- [2] Rodriguez-Quiñones JC, Sergiyenko O, González-Navarro FF, Basaca-Preciado LC, Tyrsa V. Surface recognition improvement in 3D medical laser scanner using Levenberg–Marquardt method. Signal Process (Elsevier) February 2013; 93(2):378–86 (ISSN 0165-1684).
- [3] Luis C. Básaca, Julio C. Rodríguez, Oleg Sergiyenko, Vera V. Tyrsa, Wilmar Hernández, Juan I. Nieto Hipólito, et al., Resolution improvement of dynamic triangulation method for 3D vision system in robot navigation task. In: Proceedings of IEEE—36th annual conference of IEEE industrial electronics IECON'10; 2010: p. 2886–91.
- [4] Moisés Rivas, Oleg Sergiyenko, Mario Aguirre, Luis Devia, Vera Tyrsa, Ismael Rendón, Spatial data acquisition by laser scanning for robot or SHM task. In: IEEE–IES Proceedings of the international symposium on industrial electronics ISIE-2008, Cambridge, United Kingdom; 2008: p. 1458–63.
- [5] Básaca-Preciado LC, et al. Optical 3D laser measurement system for navigation of autonomous mobile robot. Opt Laser Eng (2013), <http://dx.doi.org/10.1016/j.optlaseng.2013.08.005i>.
- [6] Luis C. Básaca, Julio Rodríguez<sup>1</sup>, Oleg Yu. Sergiyenko, Vera V. Tyrsa, Wilmar Hernández, Juan I. Nieto Hipólito, Oleg Starostenko, 3D Laser Scanning Vision System for Autonomous Robot Navigation, 978-1-4244-6392-3©2010 IEEE: p. 1773-78.
- [7] J. Rodriguez-Quinonez, O. Sergiyenko, V. Tyrsa, L. Basaca, J. Hipolito, Continuous monitoring of rehabilitation in patients with scoliosis using automatic laser, in: Pan American Health Care Exchanges— PAHCE, Rio de Janeiro, Brazil, 2011, pp. 410–414.
- [8] Luis C. Basaca-Preciado\*; Oleg Yu. Sergiyenko; Julio C. Rodriguez-Quinonez; Moises Rivas-Lopez Optoelectronics and Automated Measurement Laboratory, 8-1-4577-0733-9©2012 IEEE p. 648-49
- [9] José Luis Lerma, Miriam Cabrelles, Santiago Navarro Ana Elena Seguí, Photorealistic 3D modelling by photogrammetry: laser scanning versus digital image, ISSN 1699-0889 <http://cuadernosdearterupestre.es>
- [10] Lerma García, J.L.; Biosca Tarongers, J.M. (Septiembre 2008): “3D RiskMapping, Teoría y práctica del Escaneado Láser Terrestre.
- [11] Federico Cristina, Sebastián H. Dapoto, Javier Vegas, Verónica Artola, Claudia Russo, Abásolo

María José, Desarrollo de un escáner 3D mediante cámaras estereoscópicas e iluminación láser, XIII Congreso Argentino de Ciencias de la Computación p. 689-97.

[12] Sahba, K. , Alameh, K. , & Smith, C. L. (2007). Triangulation Based Static Wide Angle Laser Scanning For Obstacle Detection. Proceedings of The Sixth International Workshop for Applied PKC. (pp. 138-148).

[13] Jeffrey Travis, Jim Kring, LabVIEW for Everyone: Graphical Programming Made Easy and Fun (3rd Edition) (National Instruments Virtual Instrumentation Series), Prentice Hall PTR Upper Saddle River, NJ, USA ©2006, ISBN:0131856723

[14] La Evolución de LabVIEW, Instrumentation, Décadas de Desarrollo, La Publicación Mundial Sobre Medición y Automatización | Segundo Trimestre del 2009.

[15] José Rafael Lajara Vizcaino, José Pelegri Sebastia, LabVIEW. Entorno gráfico de programación, Segunda edición, 2011, ISBB: 978-84-267-1696-5.

[16] Julio Cesar Rodríguez Quiñonez. Tesis: Medición de parámetros biométricos por medio de Barrido Láser dinámico. Universidad Autónoma de Baja California, 2013.

[17] Luis Carlos Básaca Preciado, Tesis: Sistema De Visión En 3d Mediante Barrido Laser Para Navegación Autónoma De Robots Móviles, Universidad Autónoma de Baja California, 2013.

[18] Difference between Human Machine Interface and Graphical User Interface, <http://www.anaheimautomation.com/manuals/forms/hmi-guide.php#sthash.aPkDQEGe.pN02EVvI.dpbs>.

[19] Texas Instruments, Human Machine Interface (HMI) Guide, [www.ti.com/automation](http://www.ti.com/automation) 2014.

[20] Chapter 12 Graphical User Interface Concepts: Part 1, Editorial: Deitel and Associates, Inc. and Pearson Education, 2004.

[21]Raúl Rojas, Neural Networks A Systematic Introduction, Springer-Verlag, Berlin, 1996.

[22]B.M. Wilamowski, H. Yu, Improved computation for Levenberg– Marquardt training, IEEE Transactions on Neural Networks 21 (6) (2010) 930–937.

[23]David Houcque, Introduction To Matlab For Engineering Students, Northwestern University, (version 1.2, August 2005).

[24] Graham Kendall, Kumara Sastry, David Goldberg, Chapter 4 GENETIC ALGORITHMS, SEARCH METHODOLOGIES Introductory Tutorials in Optimization and Decision Support Techniques, Library of Congress Control Number: 2005051623.

[25] Garrett M. Morris, David S. Goodsell, Robert S. Halliday, Ruth Huey, William E. Hart, Richard K. Belew, Arthur J. Olson, Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical

Binding Free Energy Function, *Journal of Computational Chemistry*, Vol. 19, No. 14, 1639-1662 (1998).

[26] David E. Goldberg and Kalyanmoy Deb. A comparison of selection schemes used in genetic algorithms. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, California, 1991.

[27] John J. Grefenstette. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16(1):122–128, January/February 1986.

[28] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.

[29] Larry J. Eshelman, Richard A. Caruana, and J. David Schaffer. Biases in the Crossover Landscape. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 10–19, San Mateo, California, 1989. Morgan Kaufmann Publishers.

[30] Kenneth A. De Jong. Genetic Algorithms are NOT Function Optimizers. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 5–17. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[31] David H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic Publishers, Boston, Massachusetts, 1987.

[32] Gilbert Syswerda. Uniform Crossover in Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9, San Mateo, California, 1989. Morgan Kaufmann Publishers.

[33] William M. Spears and Kenneth A. De Jong. An Analysis of Multi-Point Crossover. In Gregory E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 301–315. Morgan Kaufmann Publishers, San Mateo, California, 1991.

[34] Terence C. Fogarty. Varying the Probability of Mutation in the Genetic Algorithm. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 104–109, San Mateo, California, 1989. Morgan Kaufmann Publishers.

[35] O.Yu. Sergiyenko, *Optoelectronics, Instrumentation and Data Processing*, ISSN 8756-6990, 2010, Vol. 46, No. 5, pp. 414–428. Allerton Press, Inc., 2010. Original Russian Text c, 2010, published in *Avtometriya*, 2010, Vol. 46, No. 5, pp. 18–35.

[36] Siegart, Roland. *Introduction to autonomous mobile robots*. ISBN 0-262-19502-X (hc : alk. paper) 1. Mobile robots. 2. Autonomous robots. 2004 Massachusetts Institute of Technology.

- [37] Lars Lindner, Oleg Sergiyenko, Vera Tyrsa, and Paolo Mercorelli, An approach for dynamic triangulation using servomotors in Industrial Electronics (ISIE), 2014 IEEE 23<sup>rd</sup> International Symposium on, Istanbul, June 2014, pp. 1926-1931.
- [38] Lars Lindner, Oleg Sergiyenko, Julio Rodriguez-Quinonez, and Wendy Flores-Fuentes, Sustitución de motores de pasos con servomotores en triangulación dinámica de láser para escaneo continuo de coordenadas espaciales in ARGOS 2014, Mexicali, 2014, pp. 159-172.
- [39] X.M. García-Cruz, O. Yu. Sergiyenko, Vera Tyrsa, M. RivasLópez, D. Hernández-Balbuena, J.C. Rodríguez-Quinonez, L.C. Basaca-Preciado, P. Mercorelli, Optimization of 3D laser scanning speed by use of combined variable step, Optics and Lasers in Engineering, vol. 54, Elsevier, March 2014, pp. 141- 151.
- [40] David Cram and Paul Hedley, Pronouns and procedural meaning: The relevance of spaghetti code and paranoid delusion, Oxford University Working Papers in Linguistics, Philology and Phonetics (2005), vol. 10, pages 187-210.
- [41] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 191–197, George Mason University, 1989. Morgan Kaufmann Publishers.
- [42] <http://www.deskeng.com/de/3d-scanning-101/>
- [43] <http://www.automationtechnology.de/cms/en/cx-technologie>.

## ANEXOS

Tabla A-1 Diferentes tipos de Tecnología Láser [43]

Name of Method	Contact/ Non-contact Method	Incremental or Absolute Distance Measurement (ADM)	Detection Method	What is Measured or Algorithmically Extracted	Type of Tracking Along Surface	Portability
Traditional Fixed CMM	Contact	Absolute	Center of Spherical Tip Contacts Surface	Probe tip's center (x,y,z) position is recorded, true surface calculated based on a compensation vector, displacement in time	Fixed multi-axis coordinate measurement machine (CMM)	Immobile technique requiring part be brought to the measurement tool
Portable CMM: Articulated Arm	Contact	Absolute	Center of Spherical Tip Contacts Surface	Probe tip's center position is recorded, true surface calculated based on a compensation vector, displacement in time	Six-axis or a seven-axis articulated (jointed) arm	Arm with many flexible joints, brought near the part to be measured
Portable CMM: Laser Tracker	Contact	Either or both, Interferometer based or ADM	Center of a spherically mounted retro reflector (SMR) contacts surface	Laser beam is locked on center of SMR; azimuth and elevation angles of laser beam are measured, distance between	Laser beam tracks the 3D coordinates of SMR's center	Portable; tracker is brought near the part to measure

				laser tracker origin and SMR is measured		
Portable CMM: Optical Tracker	Contact	Absolute	Center of a spherical tip mounted with light- emitting or passive targets contacts surface	Center of all visible light- emitting or passive targets affixed on contact probe	Tracks reflective or emitting targets	Portable; tracker is brought near the part to measure
Photogrammetry	Non- contact (but targets do contact object)	Absolute	Optical: Digital camera photographs targets on a surface	Center of light- emitting or passive targets affixed or mounted on object's surface	Tracks targets whose position are determined by photogrammetry	Portable; cameras are brought to the part to be photographed

Tabla A-2. 81 combinaciones posibles de Técnicas del Algoritmo Genético.

COMBINACIONES	TÉCNICAS			
	SELECCIÓN	CRUZAMIENTO	MUTACIÓN	REEMPLAZO
1	PROPORCIONAL	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
2	PROPORCIONAL	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
3	PROPORCIONAL	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
4	PROPORCIONAL	UN PUNTO	BIT A BIT DE INVERSIÓN	REMOVER TODO
5	PROPORCIONAL	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
6	PROPORCIONAL	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
7	PROPORCIONAL	UN PUNTO	SELECCIÓN ALEATORIA	REMOVER TODO
8	PROPORCIONAL	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
9	PROPORCIONAL	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
10	PROPORCIONAL	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
11	PROPORCIONAL	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
12	PROPORCIONAL	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
13	PROPORCIONAL	DOS PUNTOS	BIT A BIT DE INVERSIÓN	REMOVER TODO
14	PROPORCIONAL	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
15	PROPORCIONAL	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
16	PROPORCIONAL	DOS PUNTOS	SELECCIÓN ALEATORIA	REMOVER TODO
17	PROPORCIONAL	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
18	PROPORCIONAL	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
19	PROPORCIONAL	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
20	PROPORCIONAL	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
21	PROPORCIONAL	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
22	PROPORCIONAL	UNIFORME	BIT A BIT DE INVERSIÓN	REMOVER TODO
23	PROPORCIONAL	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO

**OPTIMIZACIÓN DE PASO DE BARRIDO TRIDIMENSIONAL A TRAVÉS DE ALGORITMO GENÉTICO**

24	PROPORCIONAL	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
25	PROPORCIONAL	UNIFORME	SELECCIÓN ALEATORIA	REMOVER TODO
26	PROPORCIONAL	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
27	PROPORCIONAL	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
28	TORNEO	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
29	TORNEO	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
30	TORNEO	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
31	TORNEO	UN PUNTO	BIT A BIT DE INVERSIÓN	REMOVER TODO
32	TORNEO	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
33	TORNEO	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
34	TORNEO	UN PUNTO	SELECCIÓN ALEATORIA	REMOVER TODO
35	TORNEO	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
36	TORNEO	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
37	TORNEO	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
38	TORNEO	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
39	TORNEO	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
40	TORNEO	DOS PUNTOS	BIT A BIT DE INVERSIÓN	REMOVER TODO
41	TORNEO	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
42	TORNEO	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
43	TORNEO	DOS PUNTOS	SELECCIÓN ALEATORIA	REMOVER TODO
44	TORNEO	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
45	TORNEO	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
46	TORNEO	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
47	TORNEO	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
48	TORNEO	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
49	TORNEO	UNIFORME	BIT A BIT DE INVERSIÓN	REMOVER TODO
50	TORNEO	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
51	TORNEO	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
52	TORNEO	UNIFORME	SELECCIÓN ALEATORIA	REMOVER TODO
53	TORNEO	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
54	TORNEO	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
55	UNIFORME	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
56	UNIFORME	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
57	UNIFORME	UN PUNTO	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
58	UNIFORME	UN PUNTO	BIT A BIT DE INVERSIÓN	REMOVER TODO
59	UNIFORME	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
60	UNIFORME	UN PUNTO	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
61	UNIFORME	UN PUNTO	SELECCIÓN ALEATORIA	REMOVER TODO
62	UNIFORME	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
63	UNIFORME	UN PUNTO	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
64	UNIFORME	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
65	UNIFORME	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
66	UNIFORME	DOS PUNTOS	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
67	UNIFORME	DOS PUNTOS	BIT A BIT DE INVERSIÓN	REMOVER TODO
68	UNIFORME	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
69	UNIFORME	DOS PUNTOS	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
70	UNIFORME	DOS PUNTOS	SELECCIÓN ALEATORIA	REMOVER TODO
71	UNIFORME	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
72	UNIFORME	DOS PUNTOS	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS
73	UNIFORME	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	REMOVER TODO
74	UNIFORME	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO
75	UNIFORME	UNIFORME	INVERSIÓN DE BITS INDIVIDUALES	ESTADO EQUILIBRIO NO DUPLICADOS
76	UNIFORME	UNIFORME	BIT A BIT DE INVERSIÓN	REMOVER TODO
77	UNIFORME	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO
78	UNIFORME	UNIFORME	BIT A BIT DE INVERSIÓN	ESTADO EQUILIBRIO NO DUPLICADOS
79	UNIFORME	UNIFORME	SELECCIÓN ALEATORIA	REMOVER TODO
80	UNIFORME	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO
81	UNIFORME	UNIFORME	SELECCIÓN ALEATORIA	ESTADO EQUILIBRIO NO DUPLICADOS