



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Instrumentación y

Control Automático

**Sistema de posicionamiento autoajustable para encontrar la densidad de campo magnético máximo local entre un arreglo de bobinas de Helmholtz**

TESIS

Que como parte de los requisitos para obtener el grado de  
Maestro en Ciencias en Instrumentación y Control Automático

Presenta:

**Juan Luis Jesús Pérez Ospina**

Dirigido por:

**Dr. Juvenal Rodríguez Reséndiz**

SINODALES

Dr. Juvenal Rodríguez Reséndiz

Presidente

Firma

Dr. Sergio Campos Montiel

Secretario

Firma

Dr. Aleph Hain Pacheco Estrada

Vocal

Firma

Dr. Mario Enrique Rodríguez García

Suplente

Firma

Dr. Salvador Ortíz Santos

Suplente

Firma

Dr. Manuel Toledano Ayala

Director de la Facultad

Dr. Juan Carlos Antonio Jáuregui Correa

Director de Investigación y Postgrado

Centro Universitario

Querétaro, QRO

México.

Mayo 2023



Dirección General de Bibliotecas y Servicios Digitales  
de Información



Sistema de posicionamiento autoajutable para  
encontrar la densidad de campo magnético máximo  
local entre un arreglo de bobinas de Helmholtz

**por**

Juan Luis Jesús Pérez Ospina

se distribuye bajo una [Licencia Creative Commons  
Atribución-NoComercial-SinDerivadas 4.0  
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

**Clave RI:** IGMAC-300635

© 2023 - Juan Luis Jesús Pérez Ospina

All rights reserved.



# Resumen

En el presente trabajo se desarrolla un sistema de medición para mitigar los errores debidos al posicionamiento de un elemento sensor de campo magnético dentro del mismo. Esta mitigación se logra haciendo uso de un sistema de posicionamiento autoajustable guiado por un algoritmo de ascenso de gradiente. El posicionamiento de un elemento sensor dentro de un campo magnético puede llegar a afectar la magnitud del campo magnético medido, eso es debido a que una orientación que no sea perpendicular a las líneas de campo magnético causará una magnitud menor dada por una descomposición trigonométrica del vector de flujo de campo incidente al sensor. Sin embargo, al alterar la posición traslacional del elemento sensor, se puede encontrar un punto dentro de un área definida donde exista el mayor flujo de campo magnético, que corresponde a la medida del sensor perpendicular al campo. Esta alteración de la posición se puede dar de forma automatizada, guiada por un algoritmo de ascenso de gradiente lo que permite la convergencia al punto mencionado de forma consistente. Finalmente, en este proyecto se presenta el diseño, desarrollo, implementación y los resultados experimentales del sistema de medición de campo magnético con un sistema de posicionamiento autoajustable integrado, comparando su rendimiento a mediciones sin este sistema.



# Abstract

In the present work, a measurement system is developed to mitigate the errors due to the positioning of a magnetic field sensor element inside the magnetic field. This mitigation is given by using a self-adjusting positioning system guided by a gradient ascent algorithm. The positioning of a sensing element within a magnetic field can greatly impact the measured magnitude because an orientation not perpendicular to the magnetic field lines will cause a lower magnitude given by a trigonometric decomposition of the field flux vector incident to the sensor.

However, by altering the translational position of the sensing element, a point within a defined area can be found where the largest magnetic field flux occurs, corresponding to the sensor measurement perpendicular to the field. This position alteration can be automated, guided by a gradient ascent algorithm which allows convergence to the mentioned point in a consistent way.

Finally, this project compares the magnetic field measurement system's design, development, implementation, and experimental results with an integrated self-adjusting positioning system, comparing its performance to measurements without this system.





# Agradecimientos

Primeramente a Dios por abrirme las puertas, guiarme y cuidarme en cada paso del camino.

A mis padres, Fariel y Lucelly, que se aseguraron que pudiera tomar la oportunidad que se me presentó, y que junto a mis hermanas, Mónica y Laura, me han dado apoyo y amor incondicional no solo ahora sino durante cada etapa de mi vida.

A mis amigos, Rojas, Marcos, Luz, Oscar, Mariana, Román y Angélica, con quienes he compartido risas, sustos, y en general, mucha alegría, que no dudaron en brindarme su apoyo cuando lo necesité.

Al Dr. Mario Rodríguez que me recibió en este nuevo país listo para guiarme y aconsejarme en este recorrido.

Al Dr. Juvenal Rodríguez, por su constante atención, apoyo y guianza que fue clave en el desarrollo de este proyecto.

A mi sínodo, Dr. Aleph, Dr. Sergio y Dr. Salvador, gracias por su disposición y atención conmigo.

A la Universidad Autónoma de Querétaro (UAQ), por recibirme en la maestría y permitirme hacer uso de los sensores requeridos para el proyecto.

Al laboratorio A4, Biocerámicos del Centro de Física Aplicada y Tecnología Avanzada de la Universidad Nacional Autónoma de México (CFATA - UNAM) por permitirme el uso de sus imanes de neodimio para el proyecto.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo dado para realizar este trabajo de investigación.

# Índice general

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Agradecimientos</b>	<b>V</b>
<b>Contenidos</b>	<b>VII</b>
<b>Índice de figuras</b>	<b>XI</b>
<b>Índice de tablas</b>	<b>XIII</b>
<b>Índice de rutinas de programación</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Justificación . . . . .	2
1.2. Descripción del problema . . . . .	3
1.3. Hipótesis . . . . .	4
1.4. Objetivo general . . . . .	4
1.5. Objetivos específicos . . . . .	4
1.6. Estado del arte . . . . .	5
<b>2. Fundamentación teórica</b>	<b>7</b>
2.1. Campos magnéticos . . . . .	7
2.1.1. Fundamentos de campos magnéticos . . . . .	7
2.1.2. Generación de campos magnéticos . . . . .	9
2.1.3. Medición de campos magnéticos . . . . .	10
2.2. Robots actuados por cables . . . . .	12

2.2.1. Cinemática inversa . . . . .	15
2.3. Sistema de medida . . . . .	16
2.3.1. Tipos de sensores . . . . .	17
2.4. Algoritmos de descenso de gradiente . . . . .	19
<b>3. Metodología</b>	<b>21</b>
3.1. Medición . . . . .	24
3.1.1. Sensores . . . . .	24
3.1.2. Conversor analógico a digital . . . . .	25
3.1.3. Caracterización . . . . .	32
3.2. Movimiento . . . . .	35
3.2.1. Restricciones . . . . .	35
3.2.2. Movimientos . . . . .	35
3.2.3. Diseño . . . . .	40
3.2.4. Motores . . . . .	49
3.3. Posicionamiento . . . . .	51
3.3.1. Uso de los motores . . . . .	51
3.3.2. Cinemática . . . . .	53
3.3.3. Rutina de programación . . . . .	57
3.4. Algoritmo . . . . .	59
3.4.1. Simulación de las condiciones . . . . .	59
3.4.2. Algoritmo . . . . .	60
3.4.3. Rutina de programación . . . . .	61
3.5. Integración . . . . .	63
3.5.1. Integración física . . . . .	63
3.5.2. Integración en software . . . . .	64
<b>4. Resultados</b>	<b>69</b>
4.1. Campo magnético constante . . . . .	69
4.2. Arreglo de bobinas de Helmholtz . . . . .	72
<b>5. Conclusión</b>	<b>79</b>
<b>Bibliografía</b>	<b>84</b>
<b>Apéndice A. Rutinas de programación</b>	<b>85</b>
A.1. Arduino . . . . .	85

A.1.1. Medición . . . . .	85
A.1.2. Posicionamiento . . . . .	86
A.1.3. Algoritmo . . . . .	89
A.2. MatLab . . . . .	92
A.2.1. Simulación del campo . . . . .	92
A.2.2. Ejecución del algoritmo . . . . .	92



# Índice de figuras

2.1. Líneas de campo alrededor de un imán. . . . .	8
2.2. Flujo magnético. . . . .	8
2.3. Arreglo de bobinas de Helmholtz [1]. . . . .	10
2.4. Representación del efecto Hall [2]. . . . .	11
2.5. Instrumentación del sensor de efecto Hall [3]. . . . .	12
2.6. Plataformas paralelas [4]. . . . .	13
2.7. Configuraciones de los robots actuados por cables [4]. . . . .	14
2.8. Variables cinemáticas del actuador, [5]. . . . .	15
3.1. Metodología de trabajo. . . . .	23
3.2. Encapsulado del sensor SS94A1F. . . . .	25
3.3. Módulo ADS1115. . . . .	26
3.4. Circuito implementado para la etapa de acondicionamiento. . . . .	27
3.5. Curvas de medición con el sensor SS94A1F. . . . .	33
3.6. Curvas de medición con el gaussímetro. . . . .	34
3.7. Comparación de los dos sensores. . . . .	34
3.8. Grados de libertad del elemento sensor. . . . .	36
3.9. Influencia del ángulo de incidencia del campo magnético en un plano de trabajo hipotético para el elemento sensor. . . . .	37
3.10. Incidencia de las líneas de campo sobre un plano. . . . .	39
3.11. Robot paralelo actuado por cables en configuración 2T [4]. . . . .	40
3.12. Modificaciones iniciales del esquema 2T. . . . .	41
3.13. Primera iteración del diseño. . . . .	42
3.14. Detalle de montaje frontal para los motores, estandar NEMA 17. . . . .	42
3.15. Decisiones de diseño para la iteración 1. . . . .	43
3.16. Siguiendo iteración del diseño. . . . .	44

3.17. Diseño del carrete. . . . .	45
3.18. Medidas del sensor SS49E dadas por <i>Honeywell</i> , mm/[in]. . . . .	46
3.19. Primera iteración del diseño del <i>holder</i> . . . . .	47
3.20. Siguiete iteración del diseño del <i>holder</i> . . . . .	47
3.21. Medidas del sensor SS94A1F dadas por <i>Honeywell</i> , in. . . . .	48
3.22. Iteración final del diseño del <i>holder</i> . . . . .	48
3.23. Modulo A4988. . . . .	49
3.24. Arduino CNC Shield V3.10 [6]. . . . .	50
3.25. Variables del diseño mecánico. . . . .	54
3.26. Hogar del sistema. . . . .	56
3.27. Campo de pruebas generado. . . . .	60
3.28. Simulación realizada. . . . .	63
3.29. Integración física completa del sistema. . . . .	64
4.1. Posición inicial del experimento. . . . .	70
4.2. Posicionamiento durante el experimento. . . . .	70
4.3. Convergencia al punto de medición deseado. . . . .	71
4.4. Integración física de los sistemas. . . . .	72
4.5. Barrido del experimento a 10 <i>mT</i> . . . . .	73
4.6. Medidas del experimento a 10 <i>mT</i> . . . . .	74
4.7. Barrido del experimento a 1 <i>mT</i> . . . . .	76
4.8. Medidas del experimento a 1 <i>mT</i> . . . . .	77



# Índice de tablas

1.1. Estado del arte. . . . .	5
3.1. Sensores considerados. . . . .	24
3.2. Influencia del PGA en $V_{ref}$ y $r_{ADC}$ . . . . .	26
3.3. Mediciones del sensor SS94A1F, parametrizando la distancia al elemento generador de campo. . . . .	32
3.4. Mediciones del gaussímetro, parametrizando la distancia al elemento generador de campo. . . . .	33
3.5. Mediciones de campo generado por los motores. . . . .	51
3.6. Conexiones a los módulos. . . . .	52
3.7. Valores de las variables del diseño mecánico. . . . .	54
3.8. Comandos del sistema. . . . .	68
4.1. Comparativa de mediciones experimento 10 $mT$ . . . . .	74
4.2. Comparativa de métricas experimento 10 $mT$ . . . . .	75
4.3. Comparativa de mediciones experimento 1 $mT$ . . . . .	77
4.4. Comparativa de métricas experimento 1 $mT$ . . . . .	78



# Índice de rutinas de programación

A.1. Inicialización y lectura del módulo ADC, entorno: Arduino. . . . .	85
A.2. Función para agregar muestras al promedio, entorno: Arduino. . . . .	86
A.3. Función para realizar el promedio, entorno: Arduino. . . . .	86
A.4. Función de ajuste a 0, entorno: Arduino. . . . .	86
A.5. Pruebas inicialización y control de los motores, entorno: Arduino. . . . .	86
A.6. Inicialización del control de posición, entorno: Arduino. . . . .	87
A.7. Variables posicionales, entorno: Arduino. . . . .	88
A.8. Variables cinemáticas, entorno: Arduino. . . . .	88
A.9. Función de cinemática inversa, entorno: Arduino. . . . .	88
A.10. Función de ejecución de la cinemática inversa, entorno: Arduino. . . . .	89
A.11. Función para la desviación estándar de cada medida, entorno: Arduino. . . . .	89
A.12. Variables de operación, entorno: Arduino. . . . .	90
A.13. Funciones de procesamiento serial, signo de la operación gradiente y algoritmo de posicionamiento, entorno: Arduino. . . . .	90
A.14. Función para máximo de un vector y función para realizar un barrido automatizado del área de trabajo, entorno: Arduino. . . . .	91
A.15. Campo magnético simulado, entorno: MatLab. . . . .	92
A.16. Inicialización del campo simulado, entorno: MatLab. . . . .	92
A.17. Inicialización del sensor simulado, entorno: MatLab. . . . .	93
A.18. Algoritmo de posicionamiento en operación por iteraciones, entorno: MatLab. . . . .	93



# Introducción

Las tecnologías basadas en la generación y medición de campos magnéticos han encontrado aplicaciones en diversos sectores. Por ejemplo, en el ámbito de la salud se utilizan para guiar robots dentro del cuerpo humano [7, 8, 9] y en el desarrollo de herramientas de diagnóstico médico como la resonancia magnética [10]. En ciertos procesos industriales, se usa el magnetismo con diferentes objetivos, tales como, calibración de sensores [11], el uso de trampas magnéticas para evitar que partículas de metal se filtren entre productos de consumo humano. También aplica para piezas mecánicas que en su fabricación o procesamiento adquieren un campo magnético, el cual debe ser retirado posteriormente [12, 13].

El campo magnético se puede interpretar como la acción de una fuerza magnética (causada por un objeto, un imán o por una carga eléctrica) sobre el espacio que lo rodea. Este campo posee una densidad de flujo, entendida como la cantidad de campo magnético que pasa por un área definida. Esta densidad cambia dependiendo del punto del campo en el que se mida, y dicha medición puede hacerse por medio de diferentes sensores [14].

En la actualidad existe una amplia gama de sensores de campo magnético, como: dispositivos de interferencia cuántica superconductiva (SQUID, por sus siglas en inglés) [15, 16], compuertas de flujo [17], magnetorresistencias anisotrópicas [18], magnetorresistencias gigantes [19] y sensores de efecto Hall [20]. El sensor de efecto Hall, al ser introducido a un campo magnético reacciona a la densidad de flujo que lo esté atravesando. Como lo indica su nombre, estos sensores usan el efecto Hall. Este efecto utiliza la fuerza que el campo ejerce sobre la corriente que circula dentro del sensor para causar que en este mismo se genere una tensión eléctrica. Esta tensión medida va a variar con la densidad de flujo de campo magnético, el cual se puede adecuar por medio de instrumentación para usarlo en sistemas digitales [3, 20, 21, 22]. Para probar la validez de las medidas realizadas con un sensor de campo magnético es importante tener un campo de pruebas adecuado. En este proyecto de investigación consistirá de un arreglo de bobinas de Helmholtz.

El concepto de bobina se refiere a un cable conductor enrollado en círculos, y por la ley de Biot-Savart, el radio de los círculos, el número de vueltas y la corriente que circula por el cable generan un campo magnético. También se tiene el concepto de las bobinas del Helmholtz, el cual es un arreglo de dos bobinas ubicadas de forma paralela y separadas a una distancia equivalente al radio de cada una. Este permite la generación de un campo magnético uniforme y constante en la región central entre las bobinas [23, 1]. La generación de campo magnético basado en las características de las bobinas y de la corriente eléctrica que fluye a través de estas, permite probar las medidas realizadas, así como el cambio en la densidad de flujo medido según la posición del sensor.

Una forma de controlar la posición del sensor de manera digital es usar un sistema autoajutable, el cual pueda conocer la posición del sensor y cambiarla según una referencia deseada. Esta tecnología ha sido trabajada continua y diversamente en diferentes procesos, por ejemplo, en impresoras 3D [24], o en aplicaciones médicas avanzadas, donde los movimientos de un cirujano se reflejan en el control de posición de un robot teleoperado [25].

El objetivo principal de esta investigación consistió en implementar un sistema de posicionamiento autoajutable para determinar la densidad de campo magnético máximo local. Para lograrlo, se utilizó un sensor de efecto Hall con la capacidad de desplazarse dentro del campo magnético local generado por un arreglo de bobinas de Helmholtz. Se emplearon algoritmos y un control de posición de dos ejes para identificar el punto dentro del campo local donde la densidad de flujo es máxima. Se define como campo local al campo magnético presente en la región central entre las bobinas de Helmholtz, el cual puede ser determinado a partir de la corriente eléctrica aplicada, el número de vueltas y la geometría de las bobinas utilizadas.

## 1.1. Justificación

En el contexto de la medición de campo magnético, la posición del sensor tiene un impacto significativo en la repetibilidad y reproducibilidad de los resultados. El error de medición, atribuible principalmente a factores operativos y no al propio sensor, conduce a lecturas subestimadas del campo magnético real. Este efecto puede resultar en la aprobación de productos en el control de calidad, incluso cuando existe un campo magnético residual que excede el umbral establecido. Un ejemplo de lo problemático que puede ser este error se puede ver en el control de calidad de piezas mecánicas ensambladas en máquinas eléctricas rotatorias, los niveles elevados de campo magnético residual pueden dar lugar a problemas como desgaste mecánico en las piezas, lecturas incorrectas en los sensores y posibles fallos catastróficos que afecten la operatividad de la máquina [12].

## 1.2. Descripción del problema

El campo magnético es una magnitud de gran importancia para muchas industrias, por ejemplo, la medición de campo magnético residual en la industria automotriz en la cual hay piezas que deben ser magnetizadas para su procesamiento, pero deben ser desmagnetizadas para su ensamblaje. En este caso, el posicionamiento erróneo del sensor puede causar que la magnitud de campo medida sea menor a la real por efecto de la descomposición trigonométrica de una línea de campo que atraviese al sensor con un ángulo de incidencia distinto a  $90^\circ$ , causando errores al momento de ejecutar el control de calidad de la pieza, lo que puede culminar en una reducción considerable de la vida útil del vehículo en que se ensamble la pieza, causando pérdidas económicas considerables, o en pérdidas mayores como la de una vida humana [12].

La importancia en el desarrollo de un sistema de medición de campo magnético con un sistema de posicionamiento autoajutable radica en los beneficios de realizar una medición adecuada de esta magnitud, mitigando el error introducido por la ubicación del sensor. Por consecuencia, se espera la disminución de pérdidas comerciales al rechazar o aceptar erróneamente productos durante la evaluación del cumplimiento de los parámetros de calidad cuando se utilizan sensores de campo magnético que cuentan con errores humanos asociados.

### **1.3. Hipótesis**

Mediante la integración de un sistema de medición de campo magnético local y un sistema de posicionamiento autoajutable para identificar la posición del flujo máximo del campo de referencia, es posible reducir el error de posicionamiento del sensor en al menos un 75 % en relación al espacio definido por el campo magnético de referencia

### **1.4. Objetivo general**

Diseñar e implementar un sistema de medición de campo magnético local que integre un sistema de posicionamiento autoajutable para determinar el punto espacial donde el flujo magnético es máximo, permitiendo mitigar los errores de posicionamiento del sensor.

### **1.5. Objetivos específicos**

Los objetivos específicos son los siguientes:

- Entender los fenómenos físicos involucrados en el sistema de medición a desarrollarse, como son: fuerza de Lorentz y efecto Hall.
- Diseñar y construir la instrumentación asociada para la adecuación de la señal entregada por el sensor de campo magnético.
- Diseñar y construir el sistema de ajuste para la manipulación de la posición del sensor para realizar las mediciones de campo magnético local.
- Realizar la experimentación y medición que sustente el correcto funcionamiento del sistema.
- Realizar la documentación correspondiente del proyecto de investigación.



## 1.6. Estado del arte

Como se expone en la Tabla 1.1, la revisión del estado del arte evidencia diversos procesos en el uso del sensor de efecto Hall, así como los sistemas que se aplican o pueden aplicarse para la manipulación de su posición.

<b>Ref.</b>	<b>Aporte</b>
[7]	Arreglo de sensores efecto Hall para rastrear elementos magnéticos en el cuerpo humano.
[3, 21]	Procesos de caracterización e instrumentación para la transducción del campo magnético incidente en una lectura de tensión normalizada.
[26]	Implementación de un arreglo de sensores cuasi-ortogonal para realizar mediciones de flujo de campo magnético en diferentes direcciones espaciales.
[27]	Sistema de mapeo de campo magnético usando sensores efecto Hall y sistemas de control de posición.
[28]	Sistema de mapeo de campo magnético integrando un arreglo de sensores efecto Hall 3D para el mapeo de campos magnéticos.
[29]	Implementación de un sistema actuado por cables con aplicación de conceptos de diseño y nuevos sistemas de manejo de los cables.
[30]	Propuesta de geometría para un robot planar actuado por cables para transporte de suministros sin contacto en un hospital.
[31]	Estudio de los errores de posición del efector final por aproximaciones cinemáticas realizadas comúnmente en sistemas planares actuados por cables (en este caso 2T).
[32]	Propuesta para control del efector final modificando pasivamente los puntos de contacto con el mismo.
[33]	Revisión de sistemas actuados por cables.

Tabla 1.1: Estado del arte.

En el trabajo de Fischer et al. [7], se presenta un sistema enfocado a la medición de campos magnéticos generados por instrumentos quirúrgicos o cápsulas robóticas endoscópicas, esto con el objetivo de poder mantener una percepción de los mismos cuando ingresan al paciente y se pierde la línea de vista. Parte de esta propuesta consiste en una membrana con un arreglo de sensores en diferentes arreglos dimensionales que por su distancia y medición podrían ayudar a ubicar el campo generado por el instrumental dentro del cuerpo del paciente. En este caso se tiene un sistema localizado, limitado por la cantidad de sensores y su ubicación dentro de la membrana, así como la geometría de la misma, lo que se da por el énfasis de supervisar el movimiento del campo en el cuerpo. En el presente trabajo se plantea un sistema que puede operar dentro de un plano de trabajo definido, manipulando la ubicación del sensor junto con un algoritmo de ascenso de gradiente para

converger al punto de máximo campo local.

Otra implementación similar encontrada en la literatura estudia las correcciones posibles sobre una implementación de sensores Hall en configuración 3D. Estos sensores tendrían la característica principal de estar en una formación cuasi ortogonal, la cual permitiría identificar los diferentes vectores direccionales de flujo magnético para identificar los campos que están ejerciendo efecto sobre el punto de este sensor. El objetivo de esta investigación es realizar un sistema de generación y medición de campo magnético para corregir errores de orientación en la configuración 3D del sensor efecto Hall, por medio de una implementación de un sistema de bobinas Helmholtz en 3D que permiten incidir sobre el sensor de formas esperadas, de las cuales puede deducir el error en la orientación de los sensores [26].

# Fundamentación teórica

## 2.1. Campos magnéticos

Como se describe en la referencia [14], un campo magnético  $\mathbf{B}$  se puede definir en algún punto del espacio en función de la fuerza magnética que se ejerce sobre una carga que se mueve con cierta velocidad. Con esta información se puede tener una gráfica de las líneas de campo magnético alrededor de un generador de campo, como se aprecia en la Figura 2.1.

El flujo de campo magnético se encuentra definido por la ley de Gauss, la cual se puede interpretar como la magnitud de campo que atraviesa perpendicularmente un área definida, que se ilustra en la Figura 2.2.

En la Figura 2.2(a) se observa que un campo magnético paralelo al área definida, el cual no registrará ningún flujo sobre esta; así mismo, un campo perpendicular al área definida presentará la máxima magnitud posible de flujo para esas variables, como se aprecia en la figura 2.2(b). De aquí se puede concluir que un vector de campo magnético que incida con algún ángulo intermedio presentará un valor de flujo magnético mayor a cero pero menor al presentado si fuera un campo perpendicular.

### 2.1.1. Fundamentos de campos magnéticos

Una vez reconocidos los principios de campo magnético, y flujo de campo magnético, se define el origen de los campos magnéticos como cargas en movimiento [14]. Este fenómeno puede explicarse por medio de las siguientes leyes:

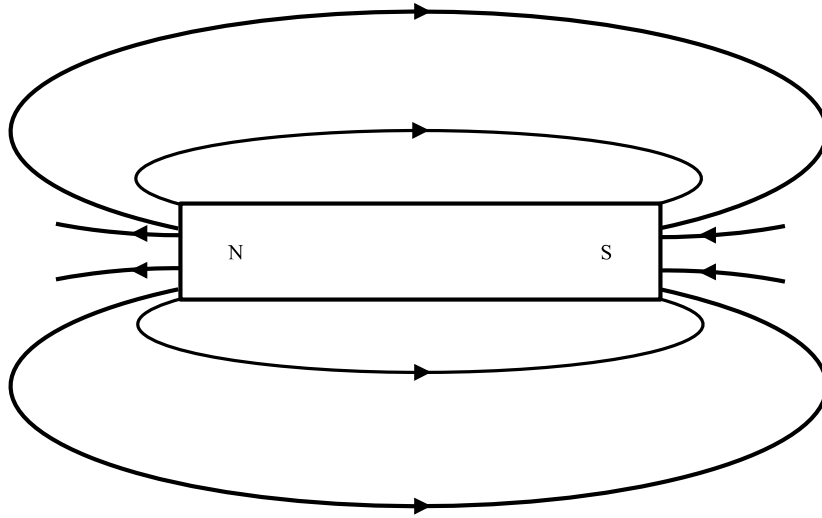
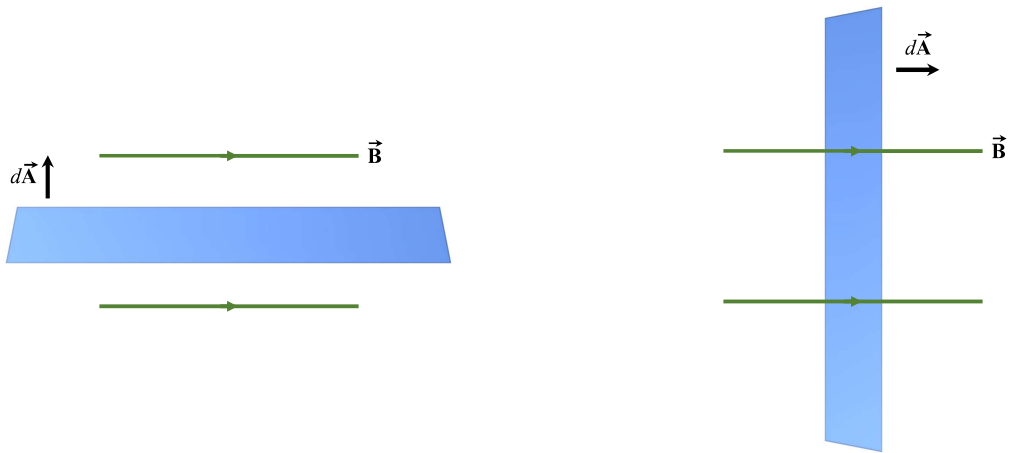


Figura 2.1: Líneas de campo alrededor de un imán.



(a) Campo paralelo al área definida.

(b) Campo perpendicular al área definida

Figura 2.2: Flujo magnético.

## Ley de Biot-Savart

Esta ley asocia un elemento de longitud  $ds$  en el que fluya una corriente eléctrica constante  $I$  con el campo magnético  $d\mathbf{B}$  encontrado en un punto  $P$  en el espacio a una distancia  $r$  del elemento, y un coeficiente de permeabilidad en el vacío  $\mu_0$ , por medio de la siguiente expresión:

$$d\mathbf{B} = \frac{\mu_0}{4\pi} \frac{I d\mathbf{s} \times \hat{\mathbf{r}}}{r^2} \quad (2.1)$$

Como se observa, la expresión realiza el cálculo de diferenciales de campo usando diferenciales de longitud del elemento, por lo tanto un cálculo del campo total  $\mathbf{B}$  requerirá realizar la integral de los diferenciales de longitud para contar todos los aportes al campo magnético realizados por la totalidad del elemento, resultando así de la siguiente manera:

$$\mathbf{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\mathbf{s} \times \hat{\mathbf{r}}}{r^2} \quad (2.2)$$

En la ecuación se encuentra un cociente correspondiente al cuadrado de la magnitud de la distancia  $r$  de cada elemento diferencial, lo que causa que el aporte por cada diferencial de longitud será menor cada que la distancia aumente. También se observa que la corriente eléctrica aparece como un elemento constante en todo el elemento y que la magnitud total del campo aumentará proporcionalmente a la magnitud de esta [14].

## Ley de Ampère

La Ley de Ampère [14] establece una relación entre la circulación del campo magnético a lo largo de una trayectoria cerrada y la corriente eléctrica que atraviesa esa trayectoria. Esta ley se puede interpretar como una simplificación de la Ley de Biot-Savart, en la cual se utiliza una trayectoria cerrada para calcular la circulación del campo magnético en lugar de calcular el campo magnético en cada punto a lo largo de la trayectoria. Lo que se puede evidenciar en la siguiente expresión:

$$\oint \mathbf{B} ds = \mu_0 I \quad (2.3)$$

### 2.1.2. Generación de campos magnéticos

Teniendo en cuenta las leyes descritas anteriormente podemos entender a una bobina de material conductor por la que fluye una corriente eléctrica constante, como un generador de campo magnético, donde una bobina de Helmholtz es un caso específico:

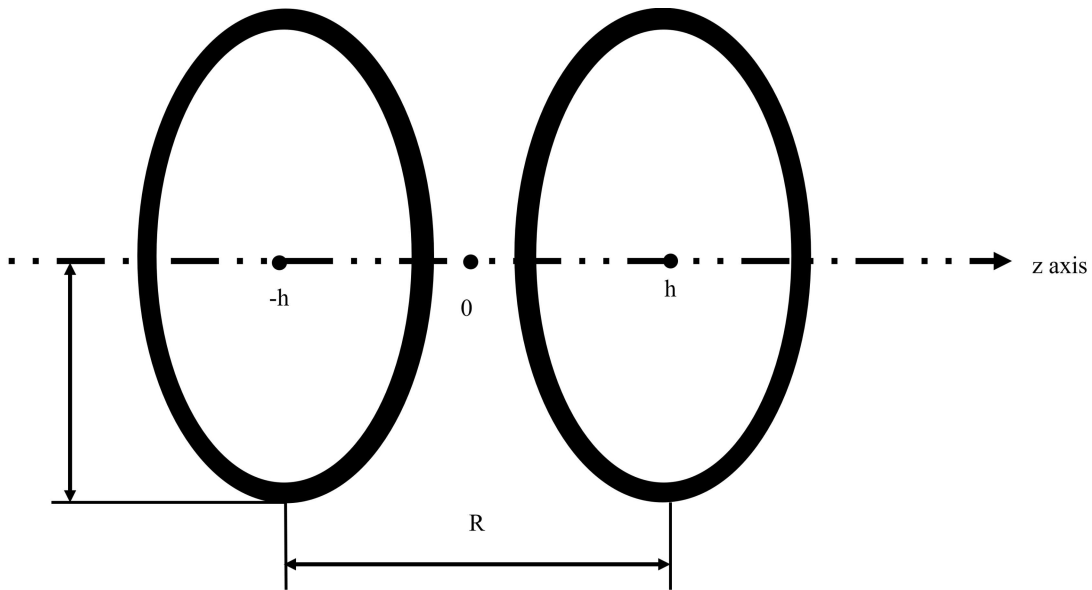


Figura 2.3: Arreglo de bobinas de Helmholtz [1].

## Bobinas de Helmholtz

El arreglo de bobinas de Helmholtz consiste en un par de bobinas idénticas, colocadas paralelamente entre sí y con una distancia entre ellas igual a su radio. Cada bobina está formada por un elemento conductor enrollado de forma concéntrica con un número de vueltas, radio y circulación de corriente eléctrica iguales. Esta configuración se puede observar en la siguiente figura:

Al aplicar una corriente eléctrica a través de estas bobinas, se genera un campo magnético en la región que las rodea, y en la región entre las bobinas se observa un campo magnético uniforme. Debido a que el campo magnético generado por una bobina se suma con el campo magnético generado por la otra bobina, causando una distribución de campo magnético uniforme en la región entre ellas [23, 1].

### 2.1.3. Medición de campos magnéticos

En su cualidad de ser una variable física medible, los campos magnéticos han sido objetivo de diferentes métodos de medición, usados por diversos transductores, como lo son: compuertas de flujo [17], magnetorresistencias anisotrópicas [18], magnetorresistencias gigantes [19] y sensores de efecto Hall [20].

## Sensor de efecto Hall

El efecto Hall expone que una lámina de material conductor o semi conductor, que es sometida a un flujo de corriente eléctrica, simultáneamente la lámina es expuesta a un campo magnético perpendicular. En consecuencia genera va a generar una diferencia de potencial eléctrico en los extremos que son perpendiculares tanto al campo electromagnético como a la corriente eléctrica. Dicho efecto es causado por la fuerza que ejerce el campo magnético sobre los portadores de carga que fluyen por la lámina de material [14, 20, 2].

El principio físico se puede observar en la Figura 2.4, sobre una placa conductora de ancho  $w$  y de alto  $d$ , por la que fluye una corriente eléctrica  $i$  en el eje X e incide un campo magnético  $B$  en el eje Z, lo que genera una diferencia de potencial en el eje Y.

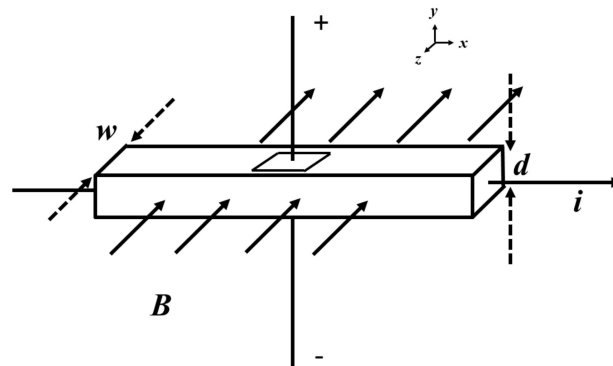


Figura 2.4: Representación del efecto Hall [2].

A partir de la ecuación de fuerza de Lorenz [2], se llega a la siguiente ecuación para representar teóricamente el efecto Hall.

$$V_H = R_H B \frac{i}{w} \quad (2.4)$$

Para la cual, la tensión Hall  $V_H$  corresponde a la tensión generada por la incidencia del campo,  $R_H$  se conoce como el coeficiente Hall, que considera factores como la magnitud de carga, la densidad de portadores de carga y la altura  $d$  de la placa son tomadas en cuenta,  $i$  corresponde a la corriente eléctrica aplicada y se observa que la tensión Hall es inversamente proporcional al ancho  $w$  de la placa.

Para utilizar un sensor de efecto Hall, es necesario caracterizar y normalizar la salida de tensión eléctrica del sensor, esto considerando el cambio de polaridad del diferencial de tensión eléctrica que se da en función de la polaridad de la incidencia del campo magnético. Con esta meta, se implementan circuitos integrados como el que se puede observar en la Figura 2.5:

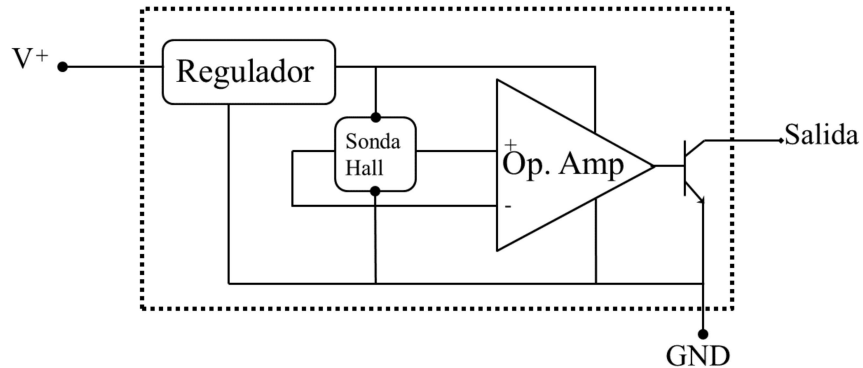


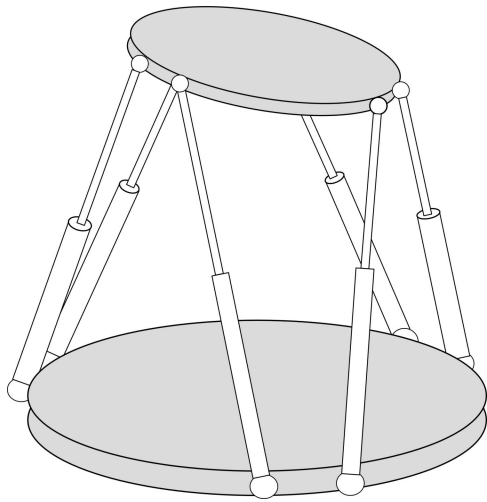
Figura 2.5: Instrumentación del sensor de efecto Hall [3].

Se debe tener en cuenta que la medida del campo magnético obtenida se dará en función del componente del campo que se encuentre perpendicular a la lámina del sensor de efecto Hall por esta razón, una medida correcta del campo dependerá de la perpendicularidad del campo medido.

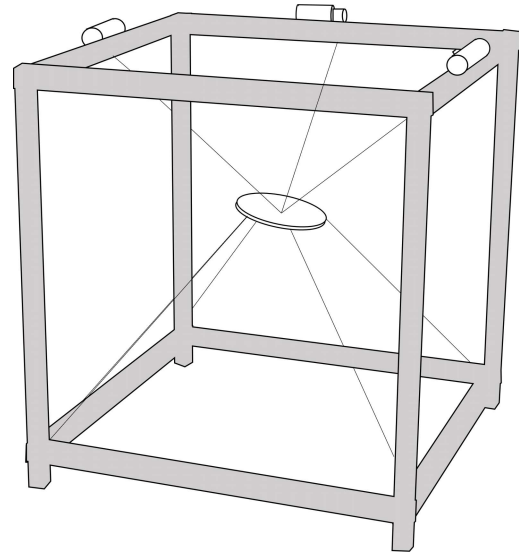
## 2.2. Robots actuados por cables

Los robots actuados por cables, también conocidos como plataformas de Stewart basadas en tendones, son sistemas que toman el concepto de robots paralelos tradicionales, reemplazando el uso de actuadores lineales con desplazamiento limitado por cables, o tendones que por medio de tensión mecánica realizan una forma de actuación muy similar a las plataformas tradicionales.





(a) Plataforma paralela tradicional.



(b) Plataforma paralela actuada por cables

Figura 2.6: Plataformas paralelas [4].

En la Figura 2.6(a) se observa un ejemplo de 6 grados de libertad (*DOF*, Degrees Of Freedom, por sus siglas en inglés) para una plataforma paralela tradicional, mientras que en la Figura 2.6(b) se observa su equivalente en una configuración actuada por cables.

Hay varias diferencias importantes entre los sistemas clásicos y aquellos actuados por cables, entre ellas [4]:

- La flexibilidad de los cables permite cambios de longitud mayores que los posibles con actuadores lineales, gracias a su facilidad de ser enrollados en actuadores rotacionales.
- Gracias a que los cables pueden ser enrollados a altas velocidades, y que la masa móvil del robot suele ser baja, se puede llegar a tener altas velocidades y aceleraciones en el efector final.
- La cantidad de cables puede ser incrementada para aumentar la capacidad de carga o para mejorar la seguridad del sistema por medio de la redundancia del mismo.

Para estos robots, se ha presentado el término redundancia como un criterio de clasificación siendo:

$$r = m - n \tag{2.5}$$

Donde  $r$  es la redundancia,  $m$  la cantidad de cables y  $n$  el número de *DOF*. Dependiendo el valor de  $r$ , se clasifican en las siguientes categorías [34]:

- *CRPM* (Manipulador paralelo completamente contenido, por sus siglas en inglés, Completely Restrained Positioning Mechanism ) donde las variables de posición y orientación del efector final están completamente definidas por los cables en tensión, para esto,  $r > 0$ .
- *IRPM* (Manipulador paralelo contenido incompletamente, por sus siglas en inglés, Incompletely restrained positioning mechanism ) donde además de las definiciones dadas por los cables en tensión se requiere una o más ecuaciones dinámicas para describir las variables de posición y orientación del efector final, para esto  $r \leq 0$ .

Otro criterio de clasificación, se presenta en función del número de *DOF* controlados [35], como se observa en la siguiente figura:

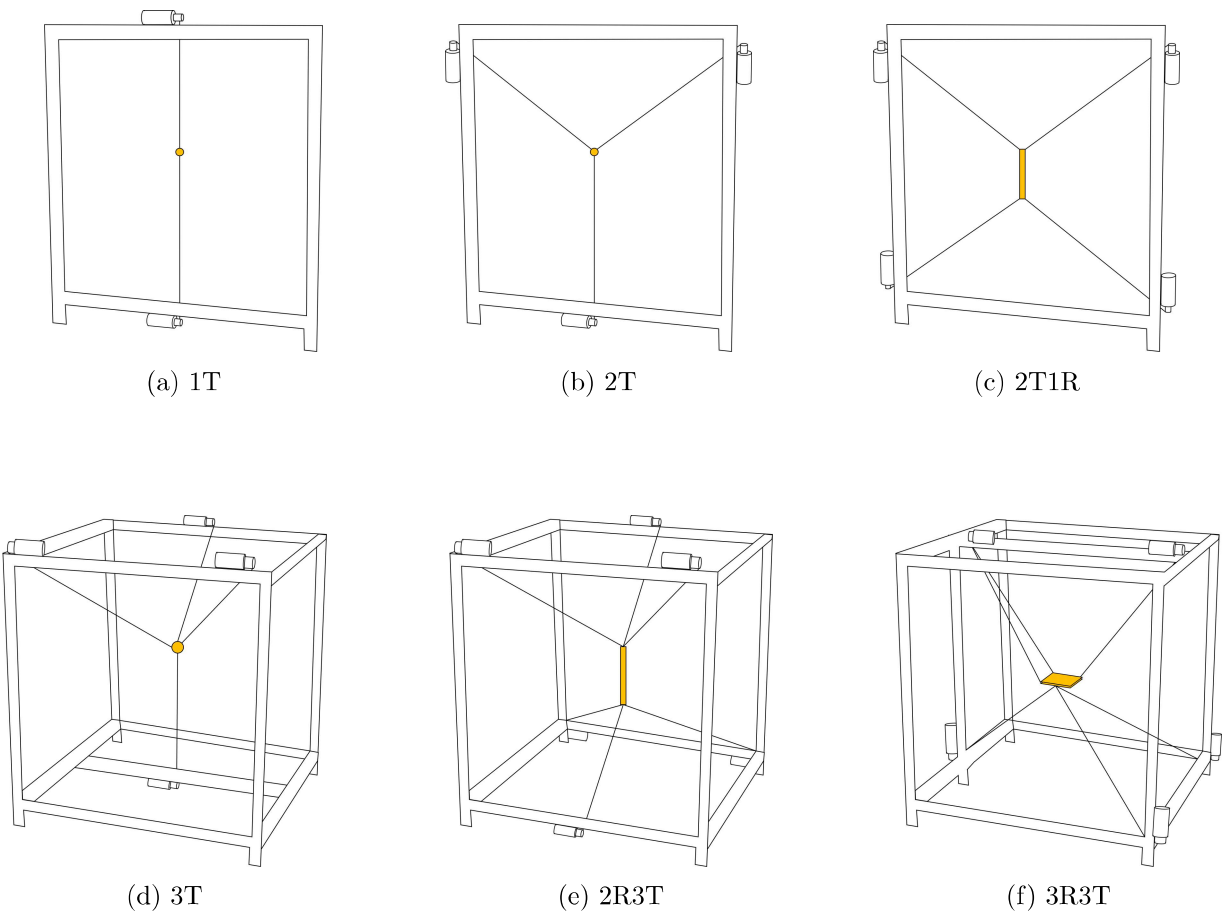


Figura 2.7: Configuraciones de los robots actuados por cables [4].

Donde, T corresponde a los grados de libertad de traslación, y R corresponde a los grados de libertad de rotación. Entonces, en la Figura 2.7(a), se tiene un robot que únicamente manipula una dimensión de traslación del efector final, mientras que en la Figura 2.7(f) se presenta un robot que manipula 3 ejes de rotación y 3 ejes de traslación en el efector final.

### 2.2.1. Cinemática inversa

La cinemática inversa hace referencia a el problema de calcular las variables articulares en función de la posición y orientación deseada del efector final; para los robots paralelos, las variables articulares corresponden a la longitud de actuadores lineales, que en el caso de las plataformas actuadas por cables, será la longitud de los mismos. Mientras que estos se encuentren en tensión mecánica constantemente, el comportamiento de la plataforma se puede equiparar a la estructura cinemática de una plataforma de Stewart [4].

La cinemática de las plataformas actuadas por cables se pueden definir análogamente a las de un manipulador típico.

$$x = C_D(q) \quad (2.6)$$

Donde se tiene  $x$  como el vector de posiciones y orientaciones del sensor;  $C_D$  denota la función vectorial de la cinemática directa y  $q$  es el vector que contiene las longitudes de los cables, de esta forma también se puede plantear la ecuación:

$$q = C_I(x) \quad (2.7)$$

Donde  $C_I$  será la función vectorial de la cinemática inversa.

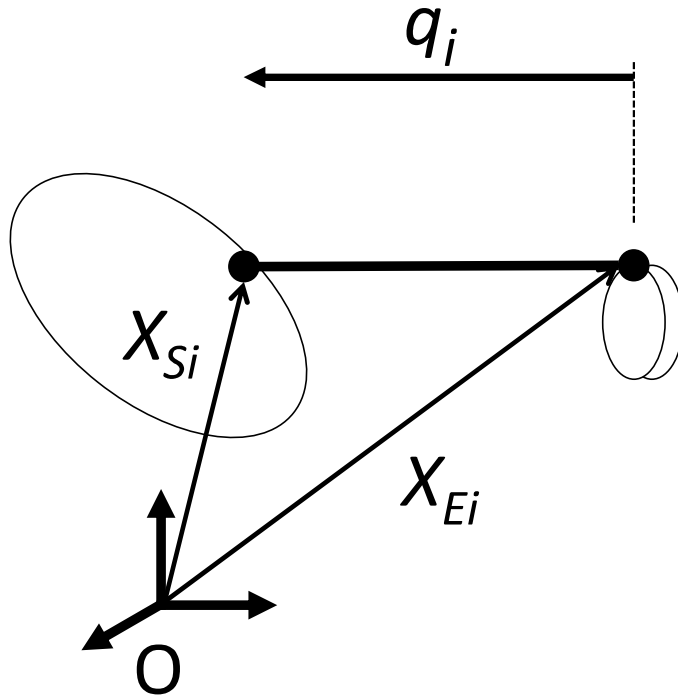


Figura 2.8: Variables cinemáticas del actuador, [5].

En la figura 2.8 se pueden observar las diferentes variables cinemáticas y como estas interactúan entre si. Donde se tiene  $q_i$  como la longitud del cable  $i$  de la cadena cinemática,  $X_{S_i}$  como la distancia del marco de referencia  $O$  al punto de contacto del cable en el efector final y  $X_{E_i}$  como la distancia del marco de referencia al punto de salida del cable.

Finalmente se obtiene que la ecuación 2.7 se puede reescribir como:

$$q_i = \|X_{S_i} - X_{E_i}\| = C_I(x) \quad (2.8)$$

Donde la distancia para cada cable  $i$  estará definida por la magnitud euclidiana del vector resultante entre la resta de los vectores al efector final y a la salida del cable, hecho que se puede observar también en la Figura 2.8.

### 2.3. Sistema de medida

El sistema de medida, está compuesto por sensores, transductores y elementos de acondicionamiento, estos trabajan en conjunto para poder tomar una señal de una variable física y permitir su lectura y así el estudio del fenómeno implicado.

Cuando se habla de un elemento sensor, se habla de un dispositivo que usando la energía del medio en el que se encuentra midiendo, otorga una señal que puede ser procesada por un transductor, siendo función de la variable medida. Mientras que por transductor, se debe reconocer a aquellos dispositivos que toman una señal en cierta forma física y la convierte en otra forma, con cierta equivalencia. De forma que es posible referirse a los transductores de entrada con el término sensor, mientras que la señal de salida corresponda a una variable física distinta a la de la señal de entrada [36].

El sentido de esta definición se encuentra en el reconocimiento de los seis tipos de señales que existen [36]: mecánicas, térmicas, magnéticas, eléctricas, ópticas y moleculares. De esta forma, se tiene un dispositivo transductor cuando este convierte, por ejemplo, una señal mecánica en una eléctrica.

Para el uso más general de los transductores se encuentran aquellos que entregan una señal eléctrica, dado a su amplio rango de ventajas, entre ellas: que con el material adecuado se puede realizar un transductor de salida eléctrica para cualquier magnitud física no eléctrica; que se pueden usar pequeñas cantidades de energía a la entrada, para no interferir en el proceso, y posteriormente amplificar la señal eléctrica de salida fácilmente con recursos de acondicionamiento, concepto que se tratará mas adelante, y de la mano de esta ventaja se encuentra la adaptación a sistemas de adquisición de datos para su tratamiento. Estas entre otras ventajas son las que contribuyen a

que el termino transductor se encuentre fuertemente asociado con transductores de salida eléctrica [36]. Por lo anterior, en lo que queda del presente texto, se podrá entender por transductor a los transductores eléctricos.

Al momento de hacer referencia a los acondicionadores de señal, se está hablando de elementos del sistema de medida que a partir de la señal de salida de un transductor electrónico de entrada, pueden presentar una señal apta para registro y posterior procesamiento por medio de sistemas de adquisición de datos. En general son circuitos de amplificación, filtrado, acoplamiento de impedancias, entre otros [36].

Es válido aclarar que también se puede conocer como transductor al conjunto que contiene al sensor, al transductor de salida, y en ciertas ocasiones al circuito de adecuación que este integre.

### **2.3.1. Tipos de sensores**

Hay un gran numero de distinciones para los sensores, pero los criterios que se presentan para este trabajo son los siguientes: aporte de energía, señal de salida y modo de operación [36].

#### **Aporte de energía**

Según este criterio pueden ser moduladores o generadores en función de la fuente de energía para la señal de salida, en el caso de los moduladores, la energía de la señal de salida proviene en su mayor parte de una fuente externa que alimenta el sensor; mientras que en el caso de los generadores, la energía de la señal de salida proviene de la entrada del fenómeno físico.

#### **Señal de salida**

En este criterio, la distinción se da entre salida analógica y salida digital. Donde una salida analógica es aquella continua en el tiempo, mientras que la digital se encuentra en intervalos discretos de valores. En el caso de las salidas analógicas vale la pena mencionar los sistemas que integran una adecuación tal que se tiene una salida radiométrica, en la cual, un sistema de medición del tipo modulador con salida analógica, va a reportar una señal de salida correspondiente a la mitad del voltaje de alimentación cuando la magnitud de la variable estudiada sea cero.

#### **Modo de operación**

Finalmente, modo de operación concierne a sensores de deflexión o de comparación. Donde los sensores de deflexión miden la reacción del fenómeno físico generado sobre el mismo instrumento; mientras que los sensores de comparación, procuran mantener nula esta reacción y la forma en la que se logra da la magnitud de la variable física.

Para el caso de un sensor de efecto Hall, se puede considerar como un modulador debido a su requerimiento de una corriente eléctrica para polarizarse mediante el campo magnético, lo que genera una respuesta en forma de tensión eléctrica. En cuanto a su señal de salida, se clasifica como un sensor analógico, ya que su respuesta es proporcional a la magnitud del campo magnético incidente. Por último, en términos de su modo de operación, se puede describir como un sensor de deflexión, ya que mide el efecto del fenómeno físico generado en el propio sensor.

## 2.4. Algoritmos de descenso de gradiente

El algoritmo de descenso de gradiente corresponde a un proceso de optimización iterativo, útil para encontrar el mínimo local de una función diferenciable. Su fórmula general se puede apreciar en la siguiente ecuación [37]:

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)}) \quad (2.9)$$

Donde se tiene un punto  $x^{(k)}$ , y para encontrar el siguiente punto  $x^{(k+1)}$ , se debe tomar el punto inicial  $x^{(k)}$  y desplazarse una cantidad dada por la segunda parte de la ecuación:  $\alpha_k \nabla f(x^{(k)})$ , donde está la función diferenciable  $f(x)$ , cuya gradiente es evaluada en el punto inicial  $\nabla f(x^{(k)})$ . Sin embargo, la gradiente retorna únicamente la componente de dirección del vector de desplazamiento, la magnitud está dada por el tamaño del paso  $\alpha_k$ , que puede ser recalculada para cada iteración del proceso, resultando en pasos cada vez más pequeños [37].

Es importante mencionar que el algoritmo de descenso de gradiente es ampliamente utilizado en problemas de optimización, ya que permite encontrar mínimos locales de manera eficiente. Sin embargo, existe una variante llamada algoritmo de ascenso de gradiente, que se utiliza para encontrar máximos locales en lugar de mínimos. El algoritmo de ascenso de gradiente se basa en la misma idea que el algoritmo de descenso, pero en lugar de desplazarse en dirección opuesta a la gradiente, se desplaza en línea con el aumento de la gradiente, como se muestra en la ecuación 2.10:

$$x^{(k+1)} = x^{(k)} + \alpha_k \nabla f(x^{(k)}) \quad (2.10)$$

Ambos algoritmos, tanto el de descenso como el de ascenso de gradiente, tienen aplicaciones en diversos campos. Por ejemplo, en el campo del aprendizaje automático, el algoritmo de descenso de gradiente se utiliza ampliamente para entrenar modelos de machine learning ajustando los parámetros del modelo para minimizar la función de costo [38]. Por otro lado, el algoritmo de ascenso de gradiente puede ser utilizado en problemas como el ajuste de modelos generativos, donde el objetivo es encontrar el máximo de una función de probabilidad.





---

# Metodología

Como se observa en la Figura 3.1, la metodología del proceso se puede segmentar en diferentes etapas. Inicialmente se tiene una revisión bibliográfica, la cual incorpora también una revisión del estado del arte. Esta revisión permite definir las temáticas del trabajo, la problemática, las soluciones existentes y como se han aplicado. Posterior a esto, teniendo claridad de las temáticas y soluciones se procede a iniciar las implementaciones del proyecto. En este caso, deriva a la selección del sensor de campo magnético.

Habiendo definido la problemática y habiendo acotado las diferentes variables del problema, se realiza la selección de uno o varios sensores que tengan aplicabilidad al mismo. Se procede a realizar la instrumentación necesaria de los sensores. Cada sensor puede transducir el valor de densidad de flujo de campo magnético en una tensión eléctrica medible de formas muy distintas, para esto se realizó la implementación de operadores electrónicos o digitales que permiten tomar la tensión eléctrica de salida del sensor y adecuarla a los sistemas digitales que son usados en el trabajo, de forma que se aprovechan las cualidades de los mismos. Parte relevante del trabajo consiste en el sistema de posicionamiento autoajutable, continuando con su diseño y construcción. El sistema de posicionamiento consiste en manipular la posición espacial del sensor en dos ejes, los cuales tienen sus propios actuadores, sistemas de medición y de control. Se inicia el proceso de diseño mecánico, asistido por software CAD (Diseño Asistido por Computador, por sus siglas en inglés, Computer Assisted Design), definiendo los ejes a manipular, sus rangos de movimiento y mecanismos, mientras que se seleccionan actuadores adecuados para la tarea. Una vez culminado el proceso de diseño, se dio la construcción del mismo. Una vez implementado físicamente, se probó el correcto funcionamiento de los mecanismos usados, finalmente corroborando que el sistema es adecuado.

Sobre dicho sistema, se procede implementando los actuadores y sensores que se hallan precisado según el diseño para entrar a la fase de caracterización y control, donde tanto el diseño como los

componentes se encuentran una vez mas sujetos a cambios según los requerimientos del sistema.

Cuando el sistema llegó a los criterios de diseño y control definidos tanto por la literatura como por el proceso investigativo, se procedió a integrar los componentes del sistema, el sensor de flujo de campo magnético con instrumentación y el sistema mecánico con control de posición en dos ejes, empezando a realizar mediciones en un campo magnético de control dado por un arreglo de bobinas Helmholtz, en el cual se comenzó el proceso de desarrollo del software que permitiera al sistema encontrar el punto espacial en el rango de trabajo donde el sensor registre mayor flujo de campo magnético de forma automatizada, dándole la característica de autoajustable al sistema de posicionamiento. Finalmente se realizó la experimentación necesaria para comprobar o rechazar la hipótesis planteada, esta misma experimentación permitió identificar oportunidades de mejora para el sistema.

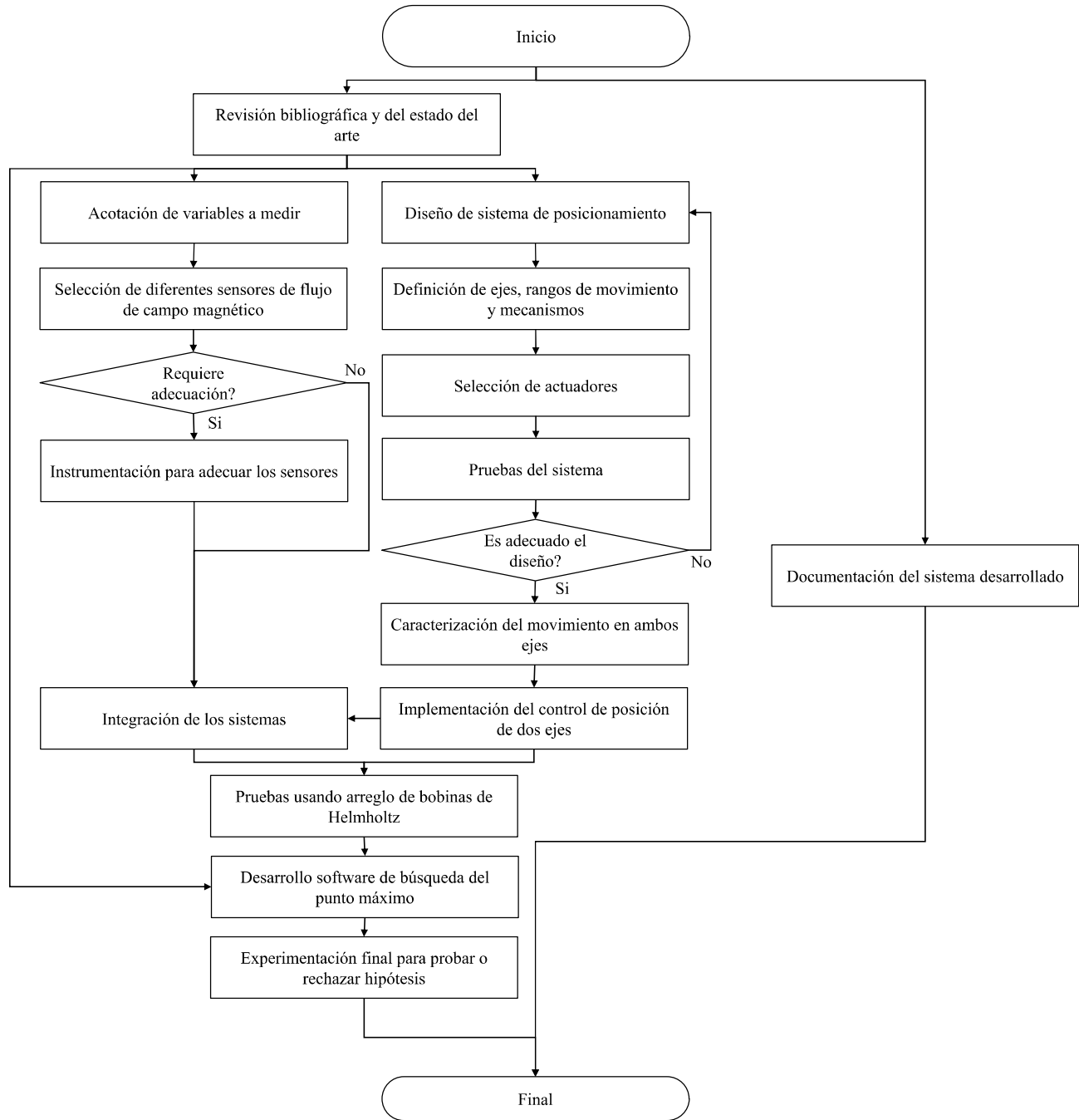


Figura 3.1: Metodología de trabajo.

A continuación se encuentra el desglose de la metodología en el desarrollo del proyecto, dividido en cuatro etapas principales:

- Medición.
- Posicionamiento.
- Algoritmo.
- Integración.

### 3.1. Medición

La sección del sistema de medición concierne a las decisiones y al desarrollo que llevó a la selección del sensor a usar, la electrónica necesaria para caracterizar su salida de voltaje con un sistema micro-controlado y su subsecuente caracterización.

#### 3.1.1. Sensores

En cuanto a la selección del sensor, esta radica en la posible aplicación específica del sistema: la detección de campos magnéticos residuales en piezas de maquinaria u otros ambientes industriales. La literatura menciona un umbral de campo residual[12] de  $3 G$ , equivalente a  $0.3 mT$ . Teniendo en cuenta este criterio se presentan los siguientes sensores de efecto Hall:

Tabla 3.1: Sensores considerados.

Referencia	Sensitividad ( $\frac{mV}{mT}$ )	Rango (mT)
SS495A	31.25	67
SS94A2	50	50
SS495A1	31.25	67
SS496B	25	84
SS49E	14	100
SS94A1F	250	11.5
SS496A	25	84

Se elige como sensor principal al sensor SS94A1F, por su habilidad de trabajo en campos de magnitudes pequeñas con alta sensibilidad, haciéndolo adecuado para el trabajo con magnitudes cercanas al umbral de campo residual.

La referencia a un sensor principal, permite observar la posibilidad de tener otro sensor, como lo puede ser el SS49E, que esté enfocado a un estudio de campos completamente distintos en términos

de magnitud, donde su gran rango y su poca sensibilidad serían ventajosos. De tal forma dentro de las decisiones concernientes a esta sección se tiene en cuenta la posibilidad de intercambiar los sensores en uso o usarlos en conjunto.

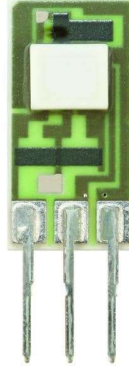


Figura 3.2: Encapsulado del sensor SS94A1F.

La hoja de datos del sensor, proporcionada por la división *Micro Switch* de *Honeywell*, permite identificar que este sensor presenta una salida analógica, radiométrica respecto al voltaje de alimentación  $V_s$  típico de 8 V

Presentando la siguiente ecuación a la salida:

$$V_O = B \cdot S + \frac{V_s}{2} \quad (3.1)$$

Donde  $B$  corresponde al campo medido,  $S$  es la sensibilidad del sensor, y  $V_O$  es la salida de voltaje que reporta el sensor.

Teniendo en cuenta que el sensor reportara la medida por medio de un voltaje de salida, y que este requiere ser adquirido por un sistema digital, se hace uso de un *ADC* (Convertor analógico a digital, por sus siglas en ingles, Anlog to Digital Converter).

### 3.1.2. Convertor analógico a digital

Se presenta el uso del modulo ADC: ADS1115, desarrollado por *Texas Instruments*, y descrito cómo ADC compatible con protocolo de comunicación *I2C*, de bajo consumo energético, de precisión y de 16 bits.

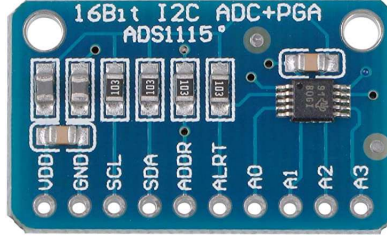


Figura 3.3: Módulo ADS1115.

Este modelo específicamente integra un amplificador de ganancia programable (PGA), de forma que se pueden seleccionar diferentes valores de voltaje de referencia ( $V_{ref}$ ), dependiendo de la ganancia deseada.

Por medio del uso de la PGA, se pueden tener rangos a la entrada de  $\pm 256 \text{ mV}$  con el voltaje de referencia mínimo, o de  $\pm 6.144 \text{ V}$ , permitiendo la medida de señales de pequeñas o grandes amplitudes.

Respecto al voltaje de referencia, vale la pena aclarar que este integrado no soportará un voltaje de entrada mayor a  $V_{DD} + 0.3 \text{ V}$  o menor a  $-0.3 \text{ V}$ , donde  $V_{DD}$  corresponde al voltaje de alimentación del modulo, sino que ese voltaje de referencia será usado para redefinir la resolución  $r_{ADC}$  del ADC, de la siguiente forma:

Tabla 3.2: Influencia del PGA en  $V_{ref}$  y  $r_{ADC}$ .

PGA	$V_{ref}$ (V)	$r_{ADC}$ (mV)
2/3	6.144	0.1875
1	4.096	0.1250
2	2.048	0.0625
4	1.024	0.0312
8	0.512	0.0156
16	0.256	0.0078

Teniendo en cuenta el funcionamiento del sensor seleccionado, se encuentra una discrepancia, ya que el sensor presenta una lectura máxima de  $8 \text{ V}$  mientras que el módulo *ADC* tiene la lectura máxima indicada anteriormente, donde  $V_{DD}$  tiene su valor máximo recomendado en  $7 \text{ V}$ . Esta discrepancia es solucionada con una etapa de acondicionamiento adicional, que consiste de un sistema de atenuación pasivo.

Este sistema de atenuación permite que por medio de un divisor de voltaje se obtenga una atenuación versátil de la salida del sensor.

Finalmente el circuito implementado para esta etapa de acondicionamiento adicional se presenta:

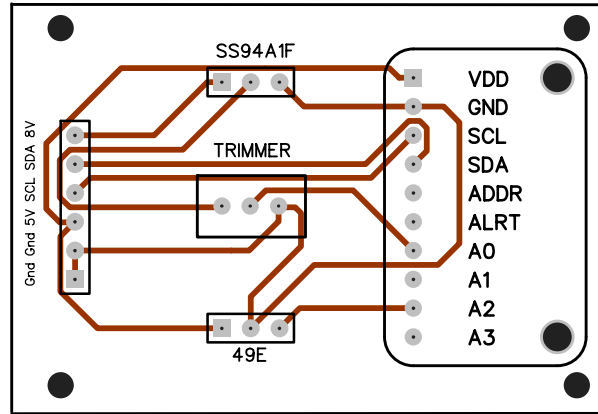


Figura 3.4: Circuito implementado para la etapa de acondicionamiento.

En esta etapa se puede observar una interfaz para acoplar el sensor al sistema micro-controlado, donde además se presentan las alimentaciones del módulo y del sensor seleccionado. Se encuentran interfaces de conexión a sensores, que se han dispuesto específicamente para el encapsulado de los sensores SS94A1F y SS49E. La última interfaz de conexión es la del módulo *ADC*, por medio de la cual se integra este al circuito. El último componente consiste en un potenciómetro *trimmer*, que se utilizará como divisor de voltaje para atenuar la señal del sensor SS94A1F. Al conectar el potenciómetro al sensor, se asegura que la señal de salida de voltaje permanezca en un rango de trabajo adecuado para el módulo.

Con este sistema de atenuación pasivo, se puede llegar a una ganancia de  $5/8$ , asegurando que cuando el sensor esté midiendo un campo magnético  $B = 0 \text{ mT}$ , obtenido con el uso de una cámara de calibración compuesta de  $\mu$ -metal, se presente un voltaje  $V_{DD}/2$ .

Adicional a lo anterior, se debe tener en cuenta que si bien el módulo *ADC* cuenta con 16 bits de resolución, uno de estos es para signo, por lo que en realidad posee 15 bits de resolución. Relacionando lo anterior, se presentan los siguientes cálculos respecto al comportamiento del sistema de medición frente a un campo magnético residual.

$$3 \text{ G} = 0.3 \text{ mT} \quad (3.2)$$

$$0.3 \text{ mT} \cdot 250 \frac{\text{mV}}{\text{mT}} \cdot \frac{5}{8} = 46.875 \text{ mV} \quad (3.3)$$

Donde se observa que un cambio en el campo magnético correspondiente a  $0.3 \text{ mT}$  reflejará un

cambio en  $46.875 \text{ mV}$ , entonces, teniendo un  $ADC$  de 15 Bits a  $V_{ref} = 6.144 \text{ V}$ :

$$\frac{6144 \text{ mB}}{2^{15} - 1 \text{ b}} \approx 0.1875 \frac{\text{mV}}{\text{b}} \quad (3.4)$$

Se comprueba el calculo de la resolución dada por la Tabla 3.2. Dado lo anterior, es posible identificar la cantidad de bits registrados por un cambio en el campo magnético correspondiente a la magnitud mencionada:

$$\frac{250 \frac{\text{mV}}{\text{mT}}}{0.1875 \frac{\text{mV}}{\text{b}}} \cdot \frac{5}{8} \approx 833 \frac{\text{b}}{\text{mT}} \quad (3.5)$$

Presentando aproximadamente  $833 \frac{\text{b}}{\text{mT}}$ , que al ser evaluados con el cambio equivalente a un umbral se producen:

$$833 \frac{\text{b}}{\text{mT}} \cdot 0.3 \text{ mT} \approx 250 \quad (3.6)$$

Donde podemos observar que teóricamente el sistema reportará una cantidad significativa de bits por cada variación del campo magnético en el que se encuentre el sensor.

Posterior a esto, procedemos con la implementación de código de lectura del sensor. Para esto se emplea un Arduino UNO con un puerto serie a  $9600 \text{ Bd}$ , el cual establecerá una comunicación  $I2C$  con el módulo  $ADS1115$  y usará su lectura en modo continuo.

Usando la librería correspondiente a este modulo, desarrollada por *Adafruit* (*Adafruit\_ADS1X15.h*), se inicia configurando el voltaje de referencia, por medio del PGA, a un valor  $V_{ref} = 6.144 \text{ V}$ , que según la tabla 3.2 resulta en una resolución  $r_{ADC} = 0.1875 \text{ mV}$ . Una vez realizado esto se procede a realizar la lectura del puerto 0 del módulo ADC, esta medida, se obtiene en un valor entero de bits registrados, el cual se pasa a campo leído por medio de la siguiente ecuación:

$$B = (b * r_{ADC} \cdot \frac{8}{5} - Offset) \cdot \frac{1}{S} \quad (3.7)$$

Que, realizando un análisis dimensional se observa de la siguiente forma:

$$\text{mT} = (\text{bits} * \text{mV} \cdot \frac{\text{V}}{\text{V}} - \text{mV}) \cdot \frac{\text{mT}}{\text{mV}} \quad (3.8)$$

Donde  $B$  es el campo magnético medido en  $\text{mT}$ ,  $b$  corresponde a los Bits medidos por el ADC,  $\frac{8}{5}$  es el inverso de la ganancia de atenuación del sensor para su lectura,  $S$  corresponde a la sensibilidad del sensor SS94A1F y offset será la resta de campo necesaria para compensar la radiometría del sensor.



Con la operación de estos elementos se encuentra que se obtiene el voltaje medido, se tiene en cuenta la atenuación a la entrada, y se realiza la conversión de voltaje entregado por el sensor, a mT. Vale la pena mencionar que en el marco de programación de lenguaje C, es necesario utilizar el operador (*float*) de forma que el resultado de esta ecuación sea entregado en un valor de punto flotante, en lugar de en un valor entero.

El procedimiento mencionado resulta en la siguiente rutina:

---

Inicialización y lectura del módulo ADC, entorno: Arduino.

---

**Require:**

- 1: Include Wire library
  - 2: Include Adafruit\_ADS1X15 library
  
  - 3: Declare vol0 as float
  - 4: Declare ads as Adafruit\_ADS1115
  
  - 5: **procedure** SETUP
  - 6:     Begin Serial communication at 9600 baud rate
  - 7:     Set the gain of ads to GAIN\_TWOTHIRDS
  - 8:     Begin ads
  - 9: **end procedure**
  
  - 10: **procedure** LOOP
  - 11:     Declare adc0 as short
  - 12:     Read the ADC value from channel 0 using ads and assign it to adc0
  - 13:     Calculate vol0 as  $(adc0 \times 0.1875 \times 8/5 - 2500)/250$
  - 14:     Print vol0 to the Serial monitor
  - 15: **end procedure**
- 

Donde se observa la inclusión de la librería de manejo de protocolo *I2C* (*Wire*), y la librería del módulo, mencionada anteriormente. Se declara la variable flotante *vol0* que almacenará el resultado de campo magnético y se crea la instancia del modulo ADC. Se inicia la comunicación serial y a continuación se configura la PGA como se mencionó, se inicia el funcionamiento del módulo y se procede a realizar la medición del mismo. Esta se almacena en una variable de tipo *short*, indicada para valores de 16 bits. Finalmente se realiza la conversión a mili Tesla y se presenta por el puerto serie.

Para este sistema se decidió implementar un filtro promedio de  $N = 10$  valores, que permita obtener una lectura instantánea del sensor más confiable y repetible, para ello se hizo uso de dos funciones aparte.

La primera función consiste en agregar las muestras, esto es, teniendo un vector de 10 valores que almacena las medidas más recientes, se toma la nueva lectura, desplazando el vector para

acomodarla y actualizando el valor que totaliza el resultado. Es una función tipo *void*, por ende no retorna un resultado, sino que altera las variables de entorno global del programa.

---

Función para agregar muestras al promedio y ejecutarlo, entorno: Arduino.

---

**Require:**

```
1: Declare 'history' as an array of floats
2: Declare 'index' as an integer
3: Declare 'total' as a float
4: Declare 'N' as an integer

5: procedure ADD_SAMPLE(sample)
6:   Declare 'previous' as a float
7:   previous  $\leftarrow$  history[index]
8:   history[index]  $\leftarrow$  sample
9:   index  $\leftarrow$  index + 1
10:  if index  $\geq$  N then
11:    index  $\leftarrow$  0
12:  end if
13:  total  $\leftarrow$  total - previous
14:  total  $\leftarrow$  total + sample
15: end procedure

16: function LATEST_MEAN
17:  return float(total)/N
18: end function
```

---

Y la ultima función toma el valor de la suma total del vector hasta el momento y la divide por el número de muestras, obteniendo así el promedio de las 10 lecturas anteriores a la llamada de esta función.

De forma que cada que se realiza una medida del módulo ADC, esta se transforma a una medida de campo magnético, la cual se almacena en el vector de promedio, cuando el código ha de enviar los resultados de una operación de medición se procede a llamar a la función de promedio para retornar el valor deseado.

Se halló la necesidad de integrar otra función al sistema, aquella que ajusta el sensor para que inicie sus lecturas desde un valor 0. Esta función toma 50 muestras del sensor y finalmente obtiene el promedio de las ultimas 10 para calcular el desfase ( $a$ ) en la medida causado por el ambiente, restando este desfase a todas las medidas subsecuentes que realice el sensor.

---

**Función de ajuste a 0, entorno: Arduino.**

---

**Require:**

```
1: Declare 'ads' as an object of type Adafruit_ADS1115
2: Declare 'N' as an integer
3: Declare 'a' as a float

4: procedure ZERO
5:   for  $i \leftarrow 0$  to  $5 \times N$  do
6:     adc0  $\leftarrow$  ads.readADC_SingleEnded(0)
7:     a  $\leftarrow$  float(adc0)  $\times$  0.1875  $\times$   $\frac{8}{5}$ /250
8:     add_sample(a)
9:   end for
10:  a  $\leftarrow$  latest_mean()
11: end procedure
```

---

Esta función se ejecuta principalmente cuando se asegura una lectura de 0  $mT$ , lo que se logró con una jaula de campo cero, compuesta de capas de  $\mu$ -metal.

De esta forma se obtuvo el código para realizar medidas con el sensor, y así caracterizarlo en la siguiente sección.

### 3.1.3. Caracterización

Para la etapa de caracterización del sistema se realizaron pruebas en contraste con un gaussímetro de mano referencia 4048 de *Bell Technologies*, usando este como patrón de medición. Las pruebas realizadas fueron respecto a un generador de campo permanente, parametrizando su distancia mientras que se mantenía constante su orientación.

Para el experimento se realizaron varias medidas, inicialmente con el sensor SS94A1F, tomando las mediciones de campo a distintas distancias:

Tabla 3.3: Mediciones del sensor SS94A1F, parametrizando la distancia al elemento generador de campo.

Distancia (mm)	Mediciones (mT)		
	1	2	3
1	7.62	8.42	8.35
2	2.17	1.92	2.01
3	0.74	0.75	0.77
4	0.33	0.35	0.35
5	0.18	0.19	0.2
6	0.11	0.11	0.11
7	0.07	0.07	0.07
8	0.05	0.05	0.05
9	0.03	0.03	0.03
10	0.02	0.02	0.02
11	0.018	0.015	0.019
12	0.012	0.012	0.012
13	0.01	0.01	0.01
14	0	0	0
15	0	0	0

De donde se obtiene la siguiente gráfica:

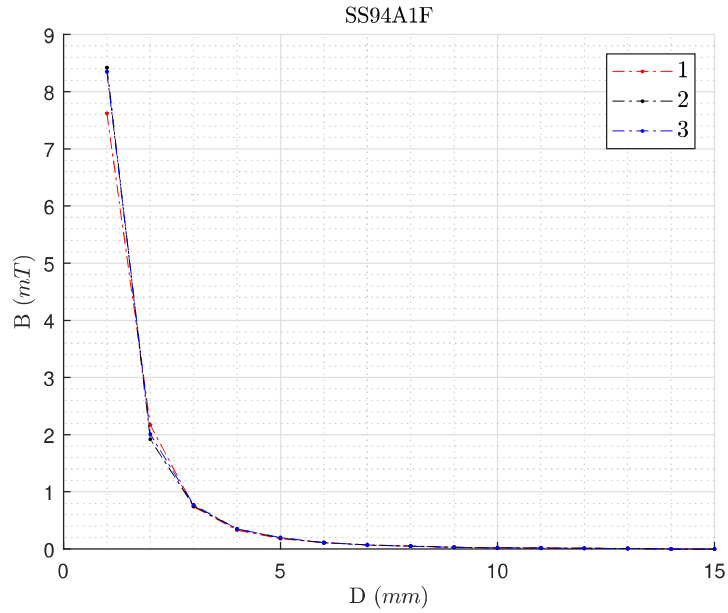


Figura 3.5: Curvas de medición con el sensor SS94A1F.

Posterior a esta medición se realizó el mismo proceso para el gaussímetro, resultando en los siguientes datos:

Tabla 3.4: Mediciones del gaussímetro, parametrizando la distancia al elemento generador de campo.

Distancia (mm)	Mediciones (mT)		
	1	2	3
1	8.2	7.73	7.42
2	1.81	1.77	1.77
3	0.66	0.71	0.69
4	0.32	0.36	0.34
5	0.16	0.22	0.22
6	0.09	0.14	0.16
7	0.03	0.11	0.1
8	0.02	0.09	0.08
9	0.01	0.07	0.06
10	0	0.05	0.05
11	0	0.04	0.04
12	0	0.03	0.03
13	0	0.01	0.01
14	0	0	0
15	0	0	0

Presentando la siguiente gráfica:

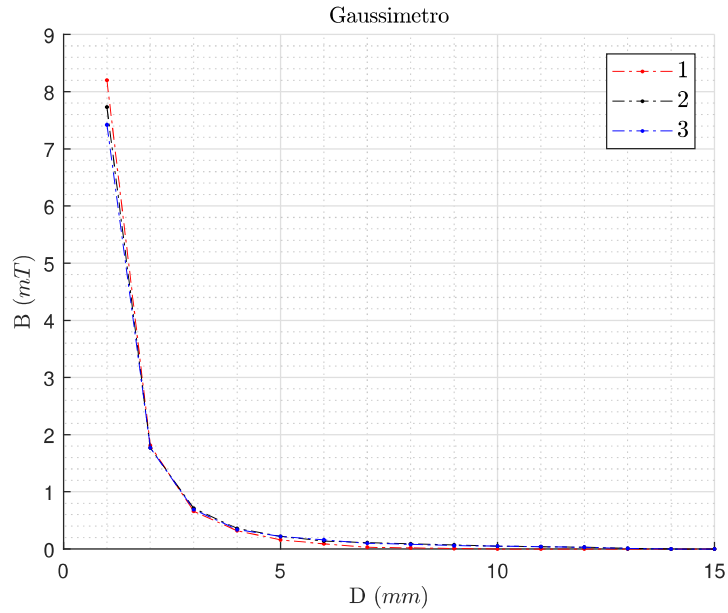


Figura 3.6: Curvas de medición con el gaussimetro.

Finalmente se realiza la comparación de los dos dispositivos con sus promedios y desviaciones:

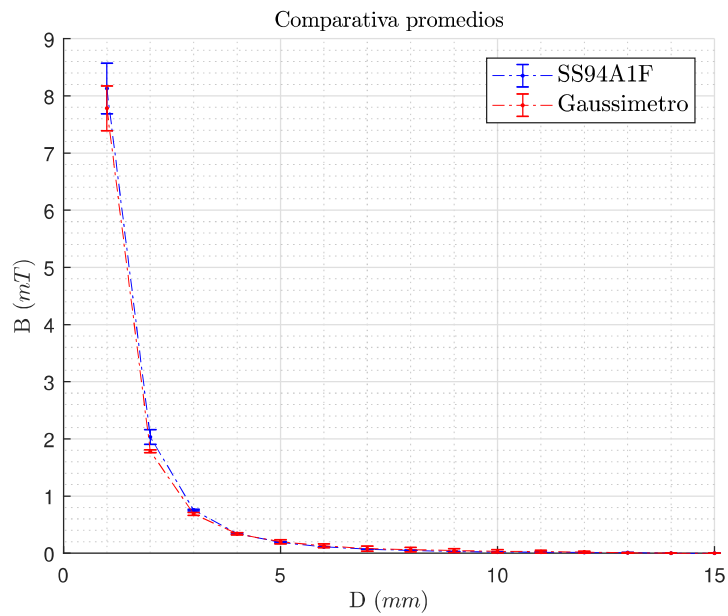


Figura 3.7: Comparación de los dos sensores.

Donde se concluye que el sensor seleccionado realiza la lectura adecuada del sistema respecto al patrón usado.

## 3.2. Movimiento

La sección de movimiento se concierne con el proceso de desarrollo para que el sistema incorpore los motores y mueva el *Holder*, iniciando por el planteamiento y terminando en un sistema listo para ser codificado.

### 3.2.1. Restricciones

Inicialmente se deben de tener en cuenta las características deseadas del movimiento del sistema, las que serán dadas por diferentes restricciones que encierran el proyecto.

La principal restricción se encuentra en la necesidad de ubicar el efector final dentro del campo magnético, sin afectar el mismo, esto es, teniendo los actuadores apartados del campo, y evitando mecanismos que incluyan partes metálicas cercanas al campo.

Para solucionar este problema se propone el uso de robots paralelos actuados por cables, que tienen la capacidad de separar el efector final de los actuadores a usar, y la posibilidad de usar mecanismos compuestos únicamente por poleas y rieles que pueden ser fabricados sin el uso de elementos metálicos.

Teniendo en cuenta la solución propuesta, se lleva a cabo un proceso de toma de decisiones para identificar los movimientos, o grados de libertad, que se consideran necesarios para el proyecto, seguido de una revisión en la literatura sobre el tipo de sistemas propuesto en esta sección, buscando principalmente su teoría básica, casos de aplicación y algunos casos específicos.

### 3.2.2. Movimientos

Esta sección ilustrará las decisiones tomadas respecto a los criterios de movimiento que debe tener el sensor dentro de un campo magnético.

Generalmente hablando, el sensor tendrá 6 grados de libertad para ubicarse dentro de un campo magnético  $B$ , que serían:  $X$ ,  $Y$ ,  $Z$ ,  $\phi$ ,  $\theta$  y  $\psi$ , refiriéndose a 3 traslaciones espaciales y a 3 rotaciones espaciales, como se ve en la figura:

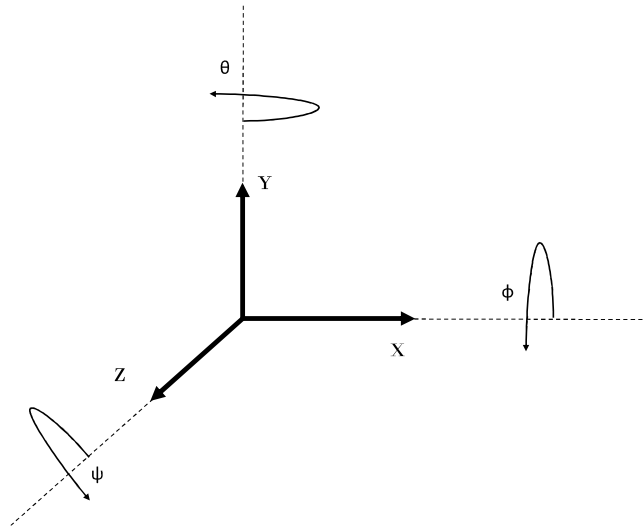


Figura 3.8: Grados de libertad del elemento sensor.

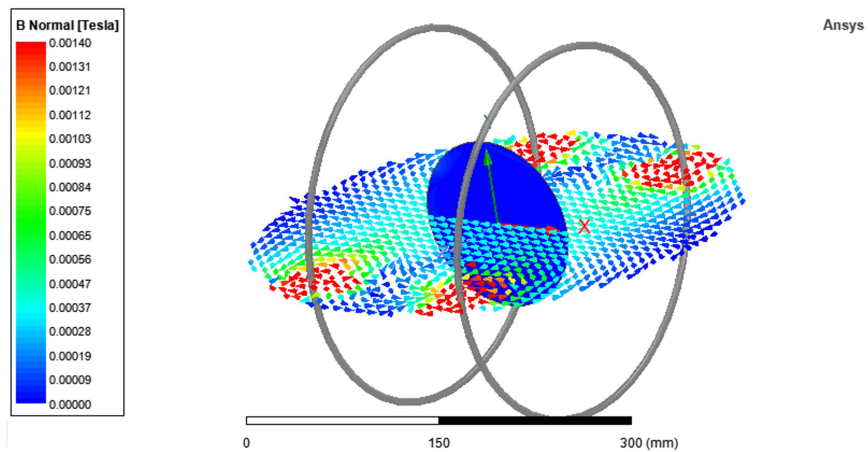
Existe la posibilidad de realizar un control de todos los grados de libertad, sin embargo, en el marco de esta investigación, se encuentra que la hipótesis puede ser comprobada, y se puede obtener un sistema adecuado realizando únicamente el control de dos grados de libertad:  $X$  y  $Y$ .

Para llegar a esta conclusión se realizó la siguiente simulación en *ANSYS*, que consistió de tres elementos principales, un generador de campo magnético permanente en configuración análoga a las bobinas de Helmholtz, un plano de trabajo de medición y un plano de observación de las líneas de campo magnético.

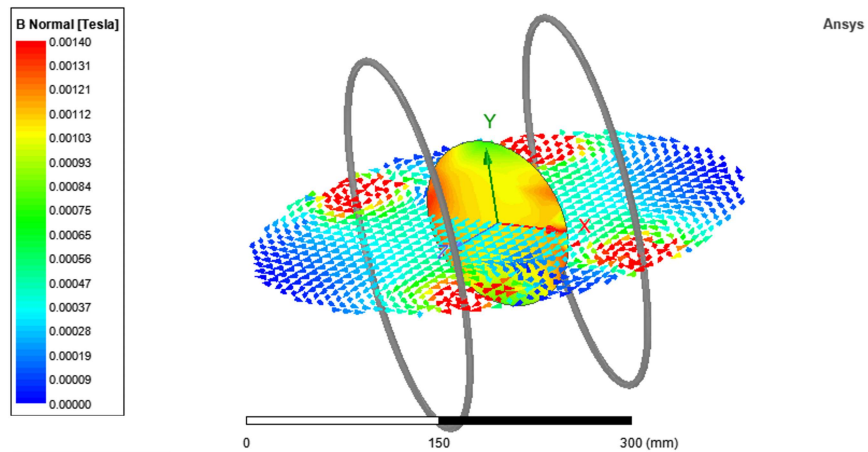
El plano de trabajo simula la presencia de sensores efecto Hall en diferentes puntos espaciales del mismo plano, permitiendo así observar la lectura hipotética del sensor como si se estuviese realizando un barrido de medidas en toda el área del plano.

El elemento generador de campo magnético presentará diferentes ángulos de rotación respecto a su propio eje, simulando así la perturbación por el ángulo de incidencia del campo magnético en el sensor. Este elemento generador presentará un campo magnético constante de  $1.4 \text{ mT}$ .

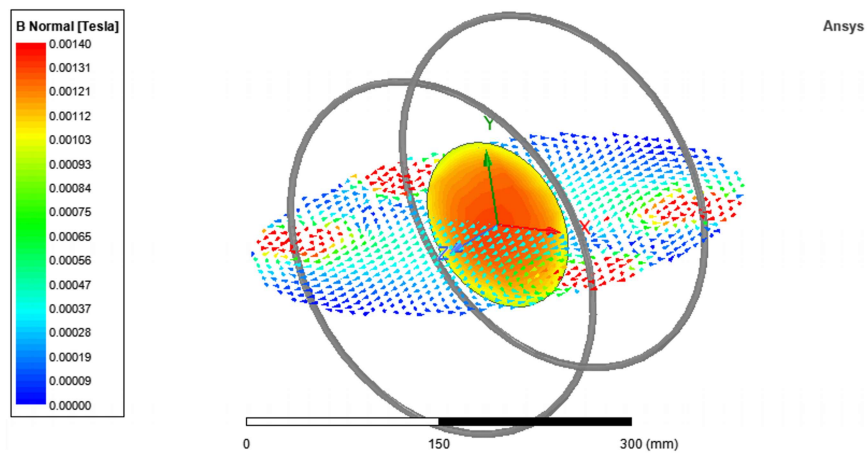




(a)



(b)



(c)

Figura 3.9: Influencia del ángulo de incidencia del campo magnético en un plano de trabajo hipotético para el elemento sensor.

En la Figura 3.9 se pueden encontrar tres etapas del experimento realizado, donde el cambio principal radica en el ángulo del campo magnético generado. La información de relevancia en esta simulación se encuentra en la magnitud del campo magnético medido perpendicular al elemento sensor.

Inicialmente, en la Figura 3.9 (a) se da el caso de líneas de campo completamente paralelas al elemento sensor, donde se observa que la magnitud medida es nula, lo que es erróneo ya que el elemento sigue estando dentro del campo magnético y que la magnitud de este es distinta a 0.

En la Figura 3.9 (b) se observan líneas de campo incidiendo sobre el sensor con un ángulo de 50 deg, dicho ángulo genera una descomposición trigonométrica del vector, reflejando una magnitud menor a la del campo. Otro aspecto relevante de la misma figura es la existencia de puntos de mayor magnitud en el área del sensor, estos son puntos que se encuentran en zonas donde las líneas de campo magnético divergen causando que haya algunas líneas que atraviesen al sensor de forma perpendicular, reflejando así la magnitud del campo magnético.

Finalmente, en la Figura 3.9 (c) las líneas de campo en el centro del arreglo son perpendiculares al sensor, causando que se refleje la lectura de magnitud de forma adecuada. En dirección a la periferia del plano de medición se da una disminución de la magnitud medida, esto se da por que a medida que se alejan del centro, las líneas de campo magnético divergen tomando ángulos de incidencia distintos de 90 deg

Este comportamiento lleva a analizar el comportamiento de las líneas de campo respecto a un elemento completamente planar:

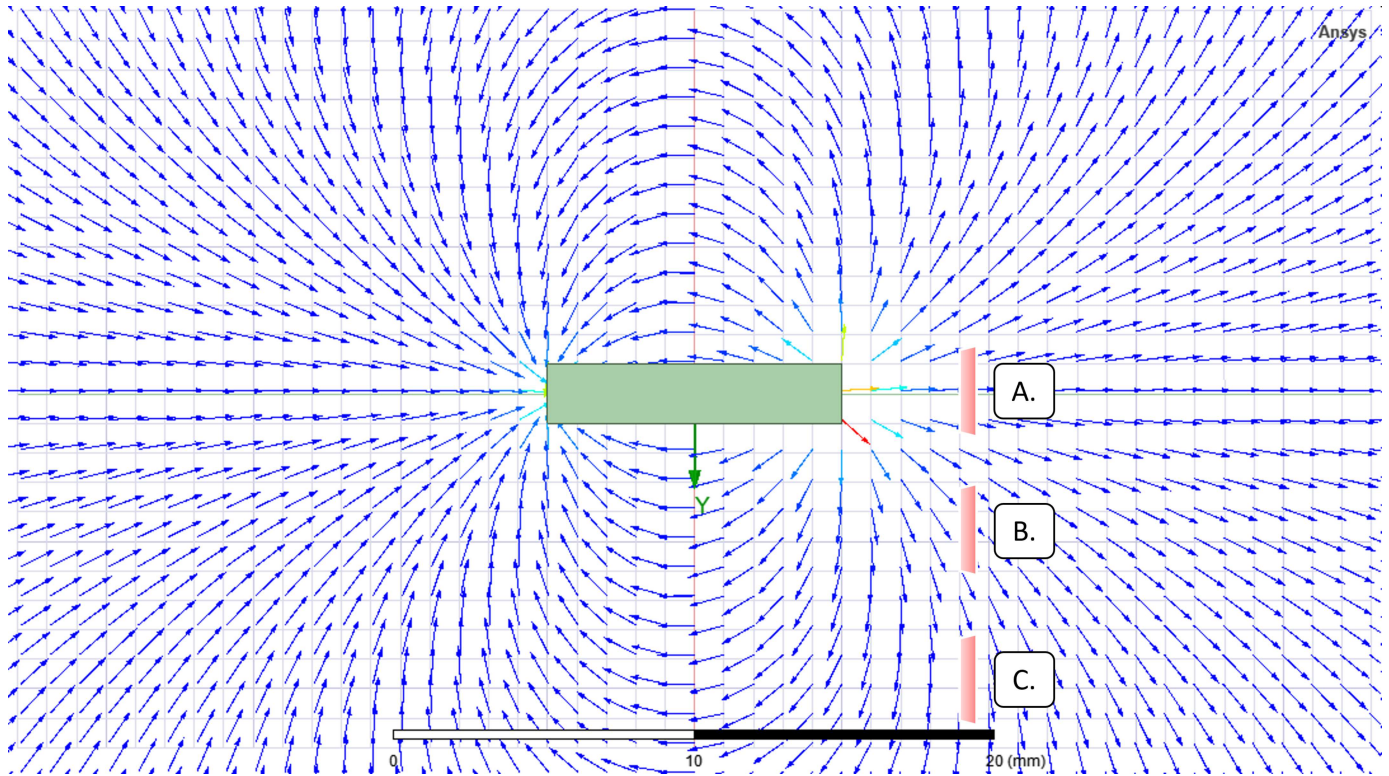


Figura 3.10: Incidencia de las líneas de campo sobre un plano.

En la figura 3.10, Se tiene un único elemento generador, sus líneas de campo y tres elementos A, B y C que comparten el mismo plano. Estos tres elementos solo difieren en un desplazamiento traslacional. De forma que, la divergencia de las líneas de campo magnético va a causar una lectura de magnitud distinta en cada uno. Para el elemento A se tiene que las líneas de campo lo están atravesando de forma perpendicular, causando una lectura máxima, para el caso de B las líneas no están paralelas o perpendiculares, por lo que se tendrá una lectura intermedia y el elemento C está paralelo a las líneas, por lo que presentará una lectura nula, análogo a lo ilustrado en la figura 3.9.

De esta forma se puede observar que teniendo un plano donde se pueden realizar mediciones con un sensor efecto Hall, existirá en este un punto donde por la divergencia natural de las líneas del campo magnético a medir, habrá una línea perpendicular al campo, exhibiendo la máxima magnitud del campo. Concluyendo que con dos grados de libertad:  $X$  y  $Y$ , que ubiquen el sensor dentro de un plano de trabajo, se podrá llegar al punto de campo magnético máximo independientemente de la orientación del mismo.

Según la solución propuesta en la sección previa, se puede hacer uso de robots paralelos actuados por cables para realizar el posicionamiento del sensor. Dada la simulación presentada, se identifica el uso de la configuración 2T como una meta deseada para llegar al sistema de posicionamiento necesario para el proyecto.

Para llegar a dicha meta de diseño se tienen diversas fuentes, entre las cuales se encuentra un libro de la teoría general de los sistemas actuados por cables [4], que incluye diferentes aspectos a trabajar en estos sistemas, como configuraciones, usos, cálculos de cinemática, entre otros. Adicional a este se encuentra una revisión de estos sistemas donde se mencionan varias aplicaciones ya conocidas tanto como aplicaciones novedosas de estos sistemas [33].

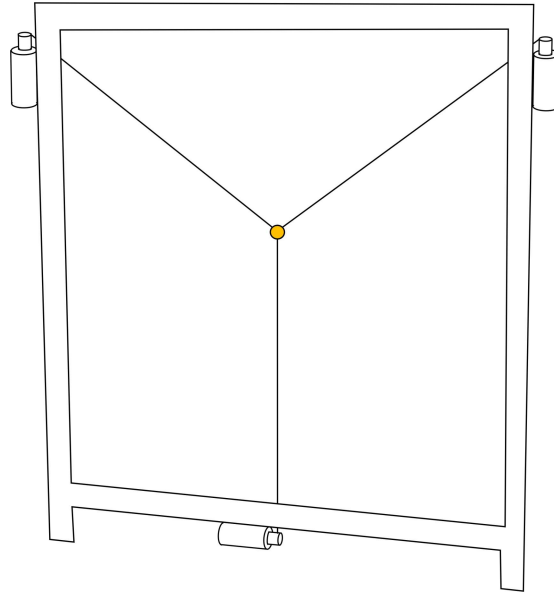


Figura 3.11: Robot paralelo actuado por cables en configuración 2T [4].

La clasificación 2T, proviene de los grados de libertad que controla el sistema, en este caso, 2T se refiere a dos grados de libertad de traslación. Se puede observar en la figura 3.11 que esta configuración cuenta con tres motores, que ejercen fuerzas en tres direcciones distintas del efector final, pero actuándolo sobre el mismo plano de trabajo, permitiendo así ejercer componentes de fuerza y desplazamiento en este plano.

### 3.2.3. Diseño

La etapa de diseño plantea una estructura similar para tres componentes principales del sistema: el cuerpo, los carretes y el elemento que asegurará el sensor durante el proceso, referido como *holder* en lo que resta del documento.

Una vez identificado el sistema que se quiere llevar a cabo, se toman algunas decisiones de diseño, para asegurar que se tenga un prototipo adecuado para las restricciones del problema:

## Cuerpo

Algunas de estas decisiones de diseño recaen en la estructura del cuerpo, que se encarga de proporcionar soporte a los motores, crear una trayectoria para las cuerdas y definir el área de trabajo.

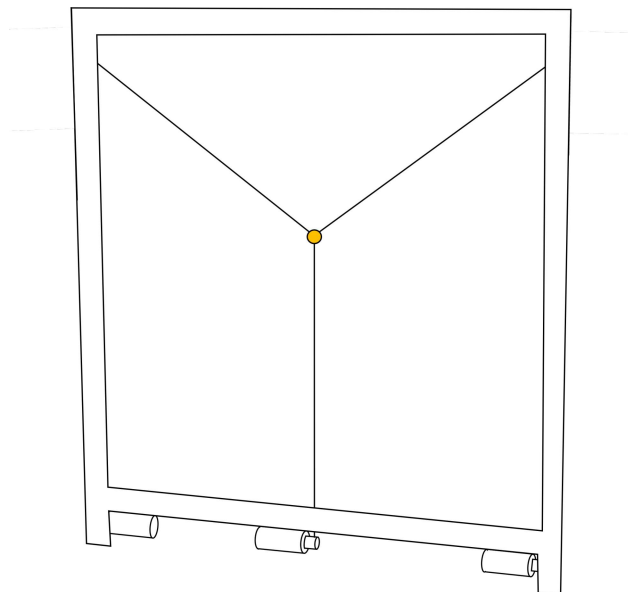


Figura 3.12: Modificaciones iniciales del esquema 2T.

La primera de estas decisiones de diseño requiere alejar los elementos actuadores del campo magnético, lo que se puede realizar por medio de elementos tipo polea que guíen los cables hacia los puntos que estaría si no se diera este movimiento se planea tener un área de trabajo de  $15\text{ cm}$  por  $15\text{ cm}$ . De aquí se inicia el proceso de diseño asistido por computadora.

El primer diseño realizado se muestra en la Figura 3.13. Este diseño tiene varios elementos clave para su funcionamiento específico: como lo es la base de montaje para los motores, la zona de trabajo y los canales de los cables. El elemento principal que se puede observar en es la distancia de separación entre el montaje para los motores y la zona de trabajo, esta decisión de diseño se toma para evitar la posibilidad de la influencia de los motores sobre el campo magnético, hecho que se estudiará más adelante en las iteraciones finales de diseño.

Otro de los elementos de diseño importantes es la base de montaje para los motores que se han planteado genéricamente hacia el uso de motores NEMA 17 (Asociación Nacional de Fabricantes Electrónicos, por sus siglas en inglés, National Electrical Manufacturers Association), que tienen un estándar de medidas tanto para el montaje frontal de sus motores como para el tamaño de el eje del mismo.

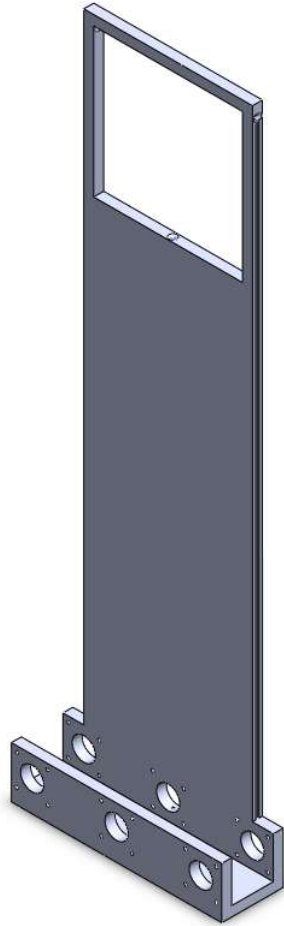


Figura 3.13: Primera iteración del diseño.

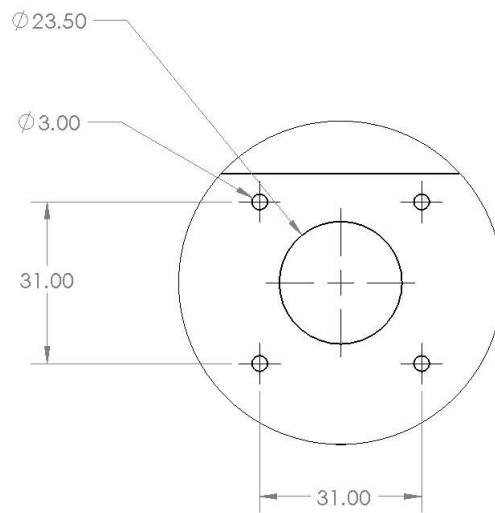


Figura 3.14: Detalle de montaje frontal para los motores, estandar NEMA 17.

En esta figura se pueden encontrar 3 medidas principales, la correspondiente a los hoyos para tornillos M3, con su medida correspondiente de  $3\text{ mm}$ , la distancia entre centros de estos tornillos, correspondiente a  $31\text{ mm}$  y finalmente la medida de  $23.5\text{ mm}$ , correspondiente al relieve que rodea al eje del motor, cuya medida es de  $22\text{ mm}$ , se le deja esta diferencia para acomodar mejor el montaje de los motores y para tener en cuenta la posibilidad de un acople que requiera estar cercano al contacto con el motor.

Otras tres decisiones importantes para el diseño se pueden encontrar en la siguiente figura:

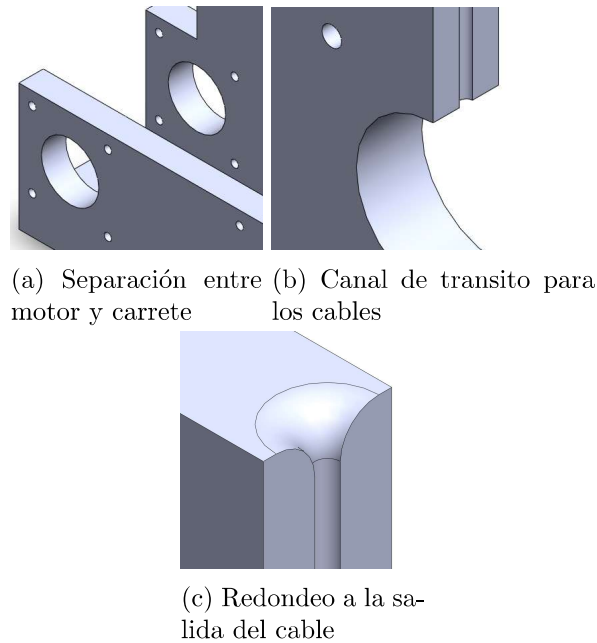


Figura 3.15: Decisiones de diseño para la iteración 1.

En la Figura 3.15(a) Se muestra el montaje frontal para los motores, que se encuentra separado cierta distancia de el cuerpo principal del diseño, donde hay un espejo del mismo montaje. Esta separación se da para dar espacio a un acople del eje al carrete que enrollará el cable a usar.

En el la Figura (b) se halla un corte transversal al modelo, permitiendo observar el canal interno del motor del medio, donde se observa el espacio por el cual este transitará, permitiendo que esté en línea directa con el carrete y con el punto de salida.

Finalmente, el detalle (c) presenta el punto de salida del canal del medio, donde se puede observar una salida redondeada, este radio a la salida permitirá un mejor movimiento del cable a usar.

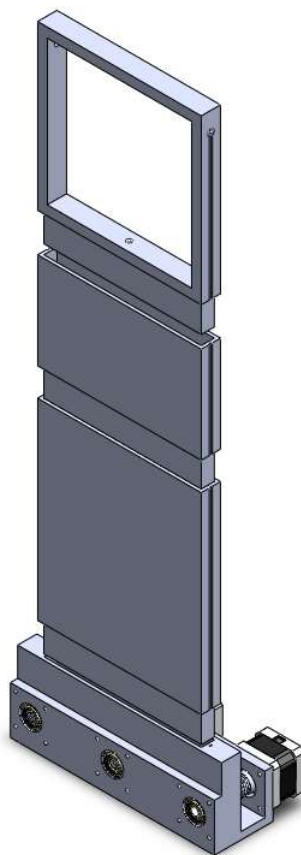


Figura 3.16: Siguiete iteración del diseño.

Posterior a este diseño, siguiendo el diseño iterativo, se tomaron nuevas decisiones a partir de las anteriores. La principal fue realizar el diseño de forma modular, esto es, que cada sección importante correspondiera a su propia parte, y que estas se puedan ensamblar después del proceso de manufactura y según su necesidad.

Para realizar el proceso de manufactura en impresora 3D, el primer diseño muestra dimensiones excesivas para el volumen de trabajo de las impresoras disponibles, por lo que la decisión de la modularidad permite mayor accesibilidad al diseño final.

En la Figura 3.16 se puede observar la implementación de la modularidad, junto con otras dos decisiones de diseño concernientes al módulo de separación entre motores y campo, así como del soporte para los carretes de los cables.

Respecto al módulo de separación entre motores y campo, se tomó la decisión de incorporar variadas longitudes, de forma que si el sistema requiere una mayor distancia esta se pueda acomodar, así como si se observa que el sistema no requiere esta distancia, también se pueda acoplar directamente al marco del plano de trabajo, obviando la necesidad de este módulo.



Finalmente, el soporte para los carretes se complementó, de forma que tengan menos espacio para variar su posición, y que esta posición sea a su vez soportada por un rodamiento rotacional.

Con las decisiones actuales, se considera que el diseño es suficiente para probar su funcionalidad en un prototipo físico.

## Carretes

Los carretes son las partes del diseño que se encargarán de almacenar el cable a usar, y tienen dos partes principales, que son el acople y el embobinado.

El acople se hará con una geometría estándar tipo D de 5 *mm*, permitiendo usarlo con el acople de 8 a 5 *mm* mencionado anteriormente.

El embobinado se plantea de forma que permita almacenar una cantidad mayor a la necesaria de cable, de evitando que se opere en los extremos del mismo, y que este se pueda alinear con los canales implementados en la sección de cuerpo.

El diseño para los carretes es como se observa a continuación:



Figura 3.17: Diseño del carrete.

En esta figura se pueden observar las partes mencionadas anteriormente, el embobinado de diseño de tal forma que pudiese almacenar 10 vueltas de cable, que, a su diámetro de 8 *mm*, resulta en  $\approx 251$  *mm* de cable, según la siguiente ecuación:

$$l = \pi \cdot D \cdot N \quad (3.9)$$

Donde  $l$  es la longitud de cable almacenado,  $D$  es el diámetro del carrete, y  $N$  el número de vueltas del embobinado.

Teniendo en cuenta que para un área de trabajo de  $150\text{ mm}$  por  $150\text{ mm}$ , la longitud máxima será, por teorema de pitágoras,  $212\text{ mm}$ , comprobando que se da la longitud de cable deseada.

Se observa también en el diseño un orificio por el cual pasará el cable para ser asegurado al carrete.

Con las decisiones actuales, se considera que el diseño es suficiente para probar su funcionalidad en un prototipo físico.

### Holder

Para el *holder* se tienen dos problemas principales a resolver, el primero consta en asegurar el elemento sensor y el segundo consta en adherir este a los cables del sistema de trabajo.

Inicialmente se tienen en cuenta las medidas para el sensor SS49E de acuerdo a su hoja de datos:

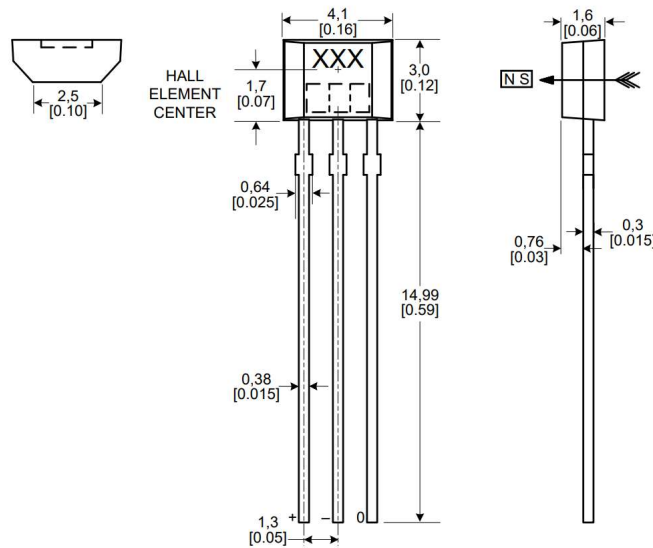


Figura 3.18: Medidas del sensor SS49E dadas por *Honeywell*, mm/[in].

A partir de las cuales se genera la primera iteración de diseño para el *holder* del sensor, observado en la Figura 3.19. En esta primera iteración se usaron tres soportes para los cables, en la esquina superior izquierda y derecha, y en el fondo del diseño, estos soportes cuentan con un radio interno y un orificio de entrada y de salida, que permite asegurar el cable adecuadamente.

Adicional a esto, este diseño cuenta con una sección que permite asegurar el sensor dentro del mismo, así como una salida para sus terminales. La principal desventaja de este diseño es que no permite asegurar adecuadamente al sensor en sus grados de libertad, por lo que se decide iterar nuevamente.

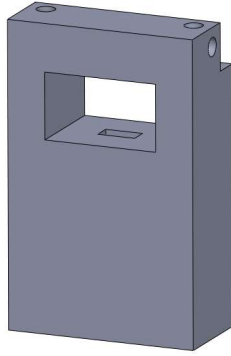


Figura 3.19: Primera iteración del diseño del *holder*.

Teniendo en cuenta la anterior iteración se decide hacer un diseño modular, que por medio de tolerancias en la impresión fuera capaz de encerrar al sensor de efecto Hall de forma que sus variables de posición y orientación quedaran fijas, mientras que mantenía las otras características deseadas del anterior diseño.

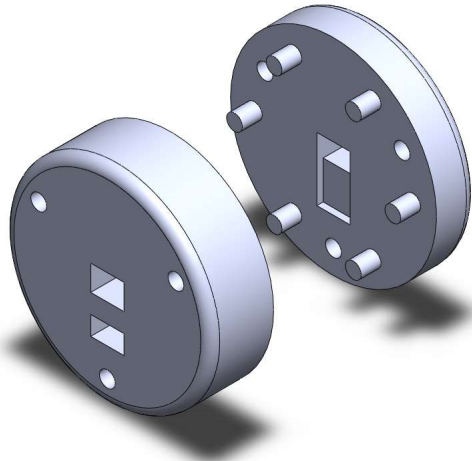


Figura 3.20: Siguiete iteración del diseño del *holder*.

En esta iteración se observan mecanismos de acople, junto con los orificios que permitirán asegurar los cables, este modelo encierra al sensor y permite que sus conexiones salgan por la parte anterior que se observa en la Figura 3.22.

Esta presenta dos problemas principales, el primero es que las conexiones no se logran asegurar de una forma adecuada ya que mayor parte del contacto queda dentro del *holder*, la segunda es que aún no contempla las medidas del sensor SS94A1F.

En la siguiente iteración se decidió tener en cuenta también las mediciones del empaquetado del sensor SS94A1F segun su hoja de datos:

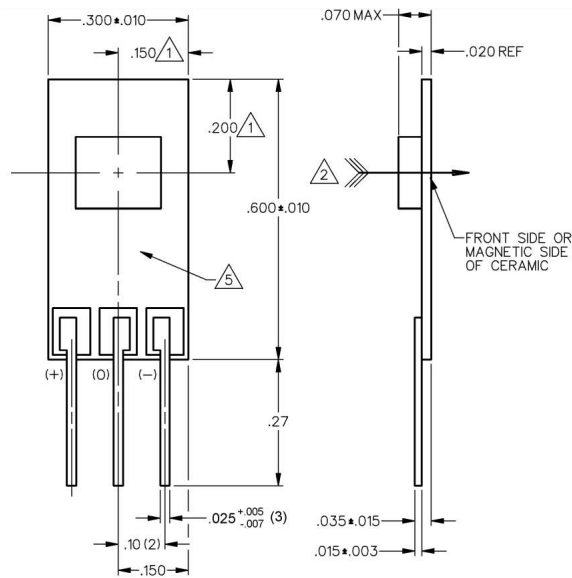


Figura 3.21: Medidas del sensor SS94A1F dadas por *Honeywell*, in.

Con estas medidas, las anteriores y nuevas decisiones de diseño, se realiza la siguiente iteración.

Finalmente se decide realizar un sistema compatible con las medidas de ambos sensores, de forma que con un sistema modular como el anterior, este puede contener a los sensores asegurando sus posiciones y orientaciones, adicional a esto, el problema de los cables se ha planteado de forma que se tiene una modularidad asimétrica, donde uno de los elementos tiene la salida de los contactos y otro de los elementos tiene la sujeción de los cables del sistema.

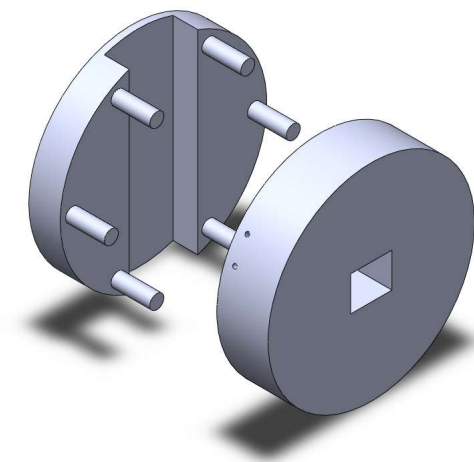


Figura 3.22: Iteración final del diseño del *holder*.

Con las decisiones actuales, se considera que el diseño es suficiente para probar su funcionalidad en un prototipo físico.

### 3.2.4. Motores

Teniendo en cuenta las restricciones de diseño, se observa que es necesario plantear un sistema de control de posición, donde se plantean algunas opciones, entre lazo cerrado y lazo abierto.

- Realimentación de posición del efector final por sistemas de visión de máquina.
- Realimentación de posición de los motores por medio de un elemento sensor acoplado al eje.
- Uso de motores que integren control de posición en lazo abierto.

Las primeras opciones agregarían un factor de complejidad no deseado para el marco de este proyecto, por lo que se toma el tercer planteamiento.

Los principales motores que integran control de posición de lazo abierto son los motores a pasos, que en general permiten rotaciones discretas de  $1.8 \text{ deg/paso}$  y son ampliamente usadas para sistemas de posicionamiento de precisión como lo son CNC (Control numérico por computadora) o impresoras 3D.

Por lo que se considera este tipo de sensores ampliamente suficientes.

Usando los motores seleccionados, NEMA 17HS3401, se procede a realizar la interfaz de control, la que consiste principalmente del módulo A4988:

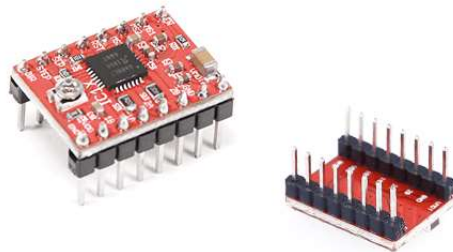
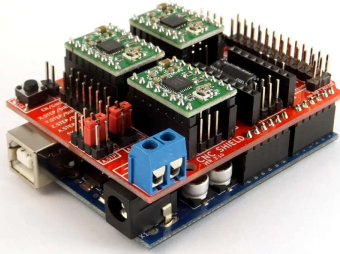
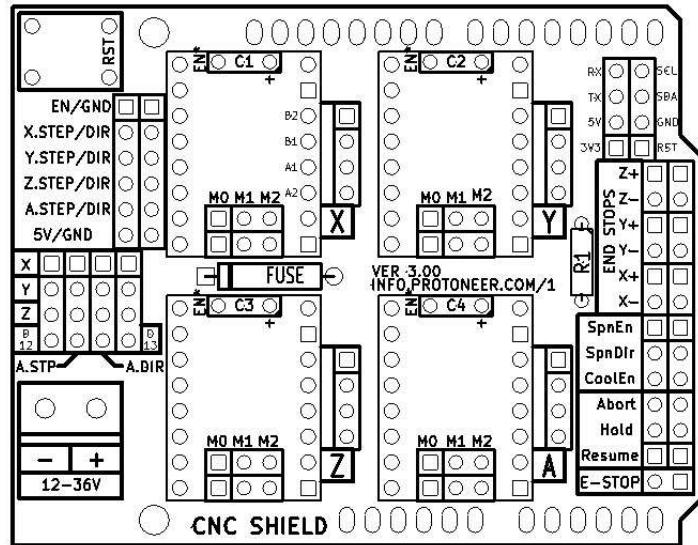


Figura 3.23: Modulo A4988.

Este módulo permite realizar el control del motor principalmente por pulsos entregados y por el estado del pin de dirección, así mismo también permite realizar micro-pasos que habilitarán un funcionamiento más fluido en el sistema. Estos módulos pueden encontrarse acoplados a un circuito CNC desarrollado para la placa Arduino[6], que compacta las conexiones necesarias en el sistema.



(a) Montaje



(b) Conexiones

Figura 3.24: Arduino CNC Shield V3.10 [6].

En esta etapa se hace necesario únicamente realizar el control de los motores para llegar al control de posición del efector final.

Para el uso de los motores se hace necesario inicialmente configurar la salida del módulo A4988, este permite controlar la corriente eléctrica por bobina, de forma que se pueda manejar el motor a su corriente eléctrica nominal o a una fracción de esta. Generalmente, está recomendado usar los motores a un 70 % de su corriente eléctrica nominal, de forma que no sean propensos a daños por temperatura y que tengan una vida útil más larga.

En el caso presente, teniendo una corriente eléctrica nominal de 1.3 A se considera adecuado bajar el margen de trabajo hasta 0.7 A, correspondiendo a un margen de  $\approx 53\%$  ya que el requerimiento de fuerza es mínimo y esto permite un trabajo adecuado de los motores.

Este ajuste de corriente eléctrica se realiza por medio de un potenciómetro integrado para este expreso propósito. El método de realizar esta configuración basta con energizar los motores por

medio del módulo, de forma que no se muevan pero que si mantengan su posición, se conecta un multímetro en serie a una de sus bobinas y se ajusta la perilla a la corriente eléctrica deseada, esto se realiza para los tres motores dejándolos configurados de forma satisfactoria.

Una vez ajustados los motores y las conexiones de los mismos, se realiza el código para sus movimientos y llegar a implementar la cinemática del sistema con los mismos.

Algo a tener en cuenta al ahora de trabajar con los motores es el hecho de que su funcionamiento está basado en el uso de campos magnéticos, por lo cual en esta sección se decide hacer un pequeño estudio correspondiente a la posible incidencia de estos campos magnéticos en el área de trabajo del sensor, este estudio se realiza tomando medidas a diferentes distancias del motor usando el gaussímetro de la sección de caracterización.

Este estudio presenta los siguientes resultados:

Tabla 3.5: Mediciones de campo generado por los motores.

Distancia (mm)	Mediciones (mT)		
	En movimiento	Mantiene posición	Sin energizar
1	0.32	0.2	0.1
2	0.09	0.06	0.03
3	0.03	0.02	0.01
4	0.01	0.01	0
5	0.01	0	0

Estas mediciones se realizaron variando la distancia del sensor a los motores en diferentes estados, primero ejecutando un movimiento, luego manteniendo una posición y finalmente sin energizar, podemos observar que el campo magnético resultante deja de ser influyente a una distancia mayor a 5 *mm*, que es mucho menor a la distancia de separación de los motores a la base del marco de trabajo, siendo esta de 40 *mm*.

### 3.3. Posicionamiento

Para la sección de posicionamiento, se toma el diseño con los motores integrados y se realizan los pasos necesarios para ubicar el elemento sensor en una posición deseada.

#### 3.3.1. Uso de los motores

Para el uso de los motores, se realiza el siguiente el código, para el cual se debe tener en cuenta que el Shield CNC ya tiene conectados los terminales necesarios para el control de pulsos, dirección y el pin de habilitación de movimiento:

Tabla 3.6: Conexiones a los módulos.

Terminal	Número
Paso X	2
Dirección X	5
Paso Y	3
Dirección Y	6
Paso Z	4
Dirección Z	7
Habilitación	8

Donde X, Y y Z corresponden a los motores izquierdo, central y derecho.

Para su uso simultaneo y con control de velocidades se integra la librería *AccelStepper* [39] y su complemento *MultiStepper*, las cuales construyen sobre la librería interna de Arduino para permitir comportamiento simultaneo, fluido y controlado de los motores.

Para iniciar pruebas se realiza la siguiente rutina:

---

**Pruebas inicialización y control de los motores, entorno: Arduino.**

---

**Require:**

- 1: Include AccelStepper library
  - 2: Include MultiStepper library
  
  - 3: Define constants *x\_paso*, *x\_dire*, *x\_habi*, *y\_paso*, *y\_dire*, *y\_habi*, *z\_paso*, *z\_dire*, *z\_habi*
  - 4: Define constant *motorInterfaceType* 1
  
  - 5: Declare array *positions* with initial values {0, 0, 0}
  - 6: Declare objects *X*, *Y*, *Z* as instances of the AccelStepper class
  - 7: Declare object *Mot* as an instance of the MultiStepper class
  
  - 8: **procedure** SETUP
  - 9:     Set the maximum speed for motor *X* to 1000
  - 10:    Set the maximum speed for motor *Y* to 1000
  - 11:    Set the maximum speed for motor *Z* to 1000
  - 12:    Add motor *X* to the MultiStepper object *Mot*
  - 13:    Add motor *Y* to the MultiStepper object *Mot*
  - 14:    Add motor *Z* to the MultiStepper object *Mot*
  - 15: **end procedure**
  
  - 16: **procedure** LOOP
  - 17:     Set the target positions of motors in *Mot* to *positions*
  - 18:     Move all motors in *Mot* to their respective target positions with speed control
  - 19: **end procedure**
- 

El cual incluye las librerías necesarias, posteriormente declara los terminales a usar.



Después, se define el tipo de interfaz de conexión de los motores a usar, esta interfaz de conexión se entiende desde la documentación de la librería, la que establece que esta variable debe estar en 1 para usar el tipo de conexión por medio de un modulo como el que se está usando.

Se procede a inicializar la variable de posiciones, esta variable va a almacenar la cantidad de pasos que se desea que avance cada uno de los motores, el vector se puede entender de la siguiente manera:

$$positions[] = [desplazamiento_X, desplazamiento_Y, desplazamiento_Z]$$

Se inicializan las instancias de cada motor, indicando su interfaz de conexión y su interfaz. Posterior a esto se inicializa la instancia de múltiples motores, nombrada para este caso *Mot*, a la cual se deben agregar las instancias de los motores.

Después de esto, en la configuración, se procede a definir una velocidad máxima de movimiento para cada motor, la que está dada en pasos por segundo. Acto seguido, se agregan las instancias de los motores individuales a la instancia de múltiples motores, vale notar que el orden en el que se agreguen estas instancias será el orden en el que se den los desplazamientos dados por la variable de posiciones.

Finalmente en el bucle del código se le ordena a la instancia de los motores realizar el movimiento por medio de las dos funciones que se observan, es notable que cuando se ejecute el movimiento una vez, no lo repetirá, debido a que ya estará en la posición deseada.

### 3.3.2. Cinemática

En la sección de cinemática, se presentará el procedimiento necesario para llegar a las funciones vectoriales que permitirán identificar las longitudes de cables necesarias para puntos deseados de el *Holder*.

#### Variables

Para realizar adecuadamente la cinemática, es necesario identificar ciertas variables del diseño mecánico:

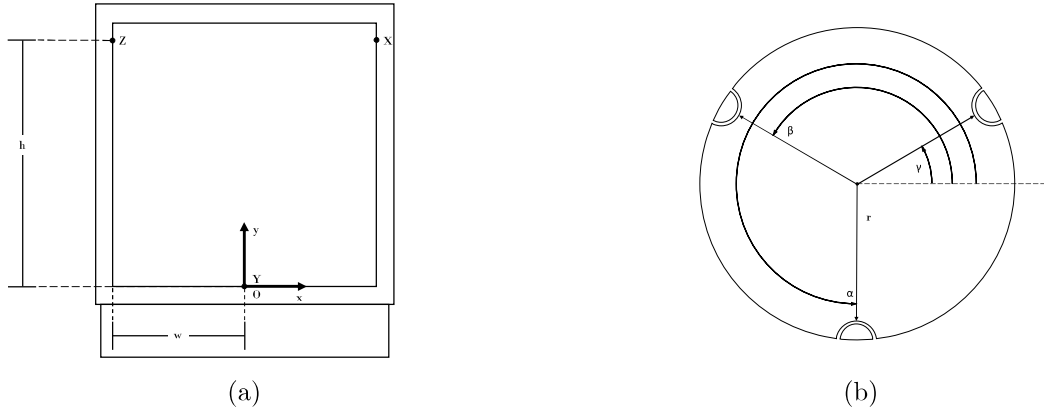


Figura 3.25: Variables del diseño mecánico.

En la Figura 3.25(a) se encuentran las posiciones de las salidas de los cables, marcados como X, Y y Z, también se encuentra el origen y el eje coordenado del sistema, que inicia en el punto de contacto Y, el eje 'y' positivo es en dirección a los otros puntos, y el eje 'x' positivo es en dirección al punto X. Las medidas más relevantes para la cinemática en este punto son:  $w$  y  $h$ , que corresponden a la distancia en 'x' hacia cada punto de contacto, y la distancia en 'y' a la que se encuentran estos.

En la Figura 3.25(b) se hallan las medidas relevantes del *Holder*, que corresponden al radio de contacto de los cables y al ángulo al que se encuentra el punto de contacto para cada uno:  $r$ ,  $\alpha$ ,  $\beta$  y  $\gamma$ .

Para el diseño actual, las medidas son las siguientes:

Tabla 3.7: Valores de las variables del diseño mecánico.

Variable	Medida
$w$	$75 \text{ mm}$
$h$	$140 \text{ mm}$
$r$	$11 \text{ mm}$
$\alpha$	$3\pi/2 \text{ rad}$
$\beta$	$5\pi/6 \text{ rad}$
$\gamma$	$\pi/6 \text{ rad}$

## Ecuaciones

Usando estos valores se puede avanzar con el problema de cinemática inversa del sistema, usando la ecuación 2.8.

Donde los vectores  $X_S$  y  $X_E$  se replantearán, recordando que los vectores  $X_S$  corresponden a la distancia del origen al punto de contacto del cable en el efector final, tenemos lo siguiente:

$$X_S = \begin{bmatrix} X_{SX} \\ X_{SY} \\ X_{SZ} \end{bmatrix} = \begin{bmatrix} x_d + r \cdot \cos(\gamma) & y_d + r \cdot \sin(\gamma) \\ x_d + r \cdot \cos(\alpha) & y_d + r \cdot \sin(\alpha) \\ x_d + r \cdot \cos(\beta) & y_d + r \cdot \sin(\beta) \end{bmatrix} \quad (3.10)$$

En estas ecuaciones se encuentran los términos  $x_d$  y  $y_d$ , que corresponden a la posición deseada del sensor, serán los valores que se ingresen a la función vectorial para retornar la longitud de los cables.

Posteriormente tenemos el planteamiento de los vectores  $X_E$ , correspondientes a la posición de salida de los cables:

$$X_E = \begin{bmatrix} X_{EX} \\ X_{EY} \\ X_{EZ} \end{bmatrix} = \begin{bmatrix} w & h \\ 0 & 0 \\ -w & h \end{bmatrix} \quad (3.11)$$

Reemplazando en la ecuación 2.8:

$$q = \begin{bmatrix} l_X \\ l_Y \\ l_Z \end{bmatrix} = \left\| \left\| \begin{bmatrix} x_d + r \cdot \cos(\gamma) & y_d + r \cdot \sin(\gamma) \\ x_d + r \cdot \cos(\alpha) & y_d + r \cdot \sin(\alpha) \\ x_d + r \cdot \cos(\beta) & y_d + r \cdot \sin(\beta) \end{bmatrix} - \begin{bmatrix} w & h \\ 0 & 0 \\ -w & h \end{bmatrix} \right\| \right\| \quad (3.12)$$

Recordando que el operador  $\|\cdot\|$  corresponde a la magnitud del vector resultante.

## Parámetros iniciales

Una vez confirmado el correcto funcionamiento de los motores a usar, se procede a ensamblarlos dentro del sistema realizado, usando como cable, nailon de  $0.3 \text{ mm}$ , el cual se conecta a los motores, por medio del carrete que a su vez se encuentra acoplado al eje de los mismos, este cable se lleva por los conductos del marco y se asegura al *Holder*, al cual aún no se le asegura el elemento sensor.

Inicialmente se realiza el cálculo de cuantos milímetros de cuerda corresponden a cada paso del motor.

Se conoce que cada paso del motor corresponde a  $1.8 \text{ deg}$ , de forma que aplicando la siguiente ecuación se calcula la cantidad de cable desplazado por paso:

$$\Delta l = 1.8 \text{ deg} \cdot \frac{\pi}{180} \cdot r \quad (3.13)$$

donde  $\Delta l$  corresponde a la longitud desplazada en  $\text{mm}$ , y  $r$  corresponde a el radio del carrete, en este caso:  $4 \text{ mm}$ , evaluar esta ecuación resulta en  $\Delta l \approx 0.126 \text{ mm}$ .

Para simplificación del código, sin embargo, se usará el inverso de este valor que resulta en

$\approx 7.96 \frac{p}{mm}$ . Teniendo como ultima aseveración, se tiene en cuenta que los módulos de control tienen la facilidad de micro-pasos, lo que debe ser tenido en cuenta si se va a usar, en este caso se usó su configuración de  $\frac{1}{16}$  de paso, por lo que el valor obtenido se debe multiplicar por 16, obteniendo finalmente  $\approx 127 \frac{p}{mm}$ , que luego por producto de experimentación se observó como 129.

## Hogar

Debido a que el sistema funciona sin una retroalimentación de posición, este requiere que al inicio de su funcionamiento se encuentre ubicado en una posición familiar, la que se conocerá como hogar en este trabajo.

Para continuar con el procedimiento de implementar la cinemática, se debe identificar el punto que será llamado hogar por el sistema de trabajo.

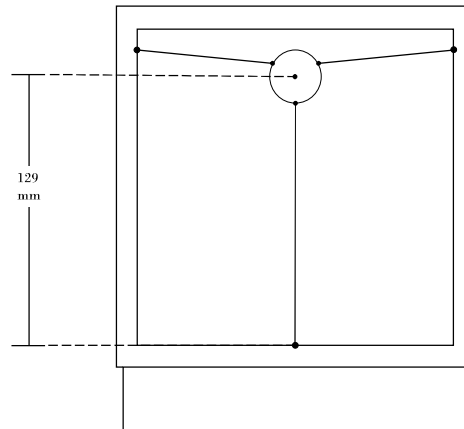


Figura 3.26: Hogar del sistema.

En la Figura 3.26 se puede observar La ubicación en la que se ubicará el elemento sensor como parte de la calibración de su movimiento, las coordenadas de esta ubicación son  $(0, 129)$  ubicando el marco de referencia en el punto de salida de la cuerda Y.

Para ubicarlo en esa posición se hace necesario usar las ecuaciones de cinemática inversa que resultarán en las longitudes de cuerda necesarias:

$$q \approx \begin{bmatrix} 65.7 \\ 118.65 \\ 65.7 \end{bmatrix} [mm] \quad (3.14)$$

Donde las medidas se dan en  $mm$ . Conociendo estas variables, y habiendo realizado el estudio de la cinemática para este problema, se procede a realizar el código.

### 3.3.3. Rutina de programación

Inicialmente se incluyen las librerías necesarias, se declaran los pines de trabajo, la interfaz de trabajo de los motores y se inician las instancias correspondientes como se mostró anteriormente:

---

**Inicialización del control de posición y variables posicionales, entorno: Arduino.**

---

**Require:**

- 1: Include AccelStepper library
  - 2: Include MultiStepper library
  
  - 3: Define constants  $x\_paso$ ,  $x\_dire$ ,  $x\_habi$ ,  $y\_paso$ ,  $y\_dire$ ,  $y\_habi$ ,  $z\_paso$ ,  $z\_dire$ ,  $z\_habi$
  - 4: Define constant *motorInterfaceType*
  
  - 5: Declare objects  $X$ ,  $Y$ ,  $Z$  as instances of the AccelStepper class
  - 6: Declare object  $Mot$  as an instance of the MultiStepper class
  
  - 7: Declare integer  $ppm$  with a value of 129
  - 8: Declare array  $pd$  with initial values  $\{0, 129\}$
  - 9: Declare array  $positions$  with initial values  $\{0, 0, 0\}$
  - 10: Declare array  $positionsmm$  with initial values  $\{0, 0, 0\}$
  - 11: Declare array  $IniPositions$  with initial values  $\{65.7, 118, 65.7\}$
  
  - 12: Declare variables  $h$ ,  $w$ ,  $r$  as floats with initial values 140, 75, 11
  - 13: Declare variables  $alph$ ,  $bet$ ,  $gamm$  as floats with initial values  $M\_PI*3/2$ ,  $M\_PI*5/6$ ,  $M\_PI/6$
  - 14: Declare arrays  $Zanch$ ,  $Yanch$ ,  $Xanch$  with initial values  $\{r*\cos(bet), r*\sin(bet)\}$ ,  $\{r*\cos(alph), r*\sin(alph)\}$ ,  $\{r*\cos(gamm), r*\sin(gamm)\}$
  - 15: Declare arrays  $Zpos$ ,  $Ypos$ ,  $Xpos$  with initial values  $\{-w, h\}$ ,  $\{0, 0\}$ ,  $\{w, h\}$
- 

Posterior a esto, se declaran las variables requeridas donde  $ppm$  corresponde al numero de pasos por milímetro de cable,  $positionsmm$  es el mismo vector de posiciones pero en milímetros,  $IniPositions$  hace referencia a las longitudes de cable para el Hogar del sistema, punto donde iniciará y terminará sus procedimientos, el vector  $pd$  corresponde a la posición en 'x' y 'y' del efector final en el sistema, y será el vector que se modifique para cambiar esta posición.

Despues se encuentran las variables de la tabla 3.7, el conjunto de vectores  $Zanch$ ,  $Yanch$  y  $Xanch$  que calculan la distancia entre el centro del elemento sensor a los puntos de contacto, recordando que las coordenadas del centro del sensor corresponden a las variables de posición deseadas  $x_d$  y  $y_d$ . Finalmente tambien se encuentran las variables correspondientes a  $X_E$ , en forma de  $Zpos$ ,  $Ypos$  y  $Xpos$ .

Se procede a la sección de configuración del código, donde se establecen las velocidades y se asignan las instancias como se realizó anteriormente.

---

Inicialización de las instancias de los motores, entorno: Arduino.

---

```
1: procedure SETUP
2:   Set the maximum speed of motor  $X$  to 1000
3:   Set the maximum speed of motor  $Y$  to 1000
4:   Set the maximum speed of motor  $Z$  to 1000
5:   Add motor  $X$  to the MultiStepper  $Mot$ 
6:   Add motor  $Y$  to the MultiStepper  $Mot$ 
7:   Add motor  $Z$  to the MultiStepper  $Mot$ 
8: end procedure
```

---

Teniendo en cuenta la ecuación 3.12 se realiza la siguiente función, para que el sistema se encargue de los cálculos de cinemática inversa:

---

Función de cinemática inversa, entorno: Arduino.

---

```
1: procedure IKINEMATICS
2:   Calculate  $Xsz$  as  $[pd[0] + Zanch[0], pd[1] + Zanch[1]]$ 
3:   Calculate  $Xsy$  as  $[pd[0] + Yanch[0], pd[1] + Yanch[1]]$ 
4:   Calculate  $Xsx$  as  $[pd[0] + Xanch[0], pd[1] + Xanch[1]]$ 
5:   Calculate  $Lz$  as  $\sqrt{(Xsz[0] - Zpos[0])^2 + (Xsz[1] - Zpos[1])^2}$ 
6:   Calculate  $Ly$  as  $\sqrt{(Xsy[0] - Ypos[0])^2 + (Xsy[1] - Ypos[1])^2}$ 
7:   Calculate  $Lx$  as  $\sqrt{(Xsx[0] - Xpos[0])^2 + (Xsx[1] - Xpos[1])^2}$ 
8:   Set  $positionsmm[0]$  to  $Lz - IniPositions[0]$ 
9:   Set  $positionsmm[1]$  to  $Ly - IniPositions[1]$ 
10:  Set  $positionsmm[2]$  to  $Lx - IniPositions[2]$ 
11:  Set  $positions[0]$  to  $positionsmm[0] \times ppm$ 
12:  Set  $positions[1]$  to  $positionsmm[1] \times ppm$ 
13:  Set  $positions[2]$  to  $positionsmm[2] \times ppm$ 
14: end procedure
```

---

Esta función obtiene los valores de posición deseada que estén establecidos en el sistema y realiza los cálculos correspondientes, obteniendo las longitudes de cable deseadas en  $mm$ , sin embargo en este punto se restan las posiciones iniciales de los mismos, para que el sistema pueda realizar los movimientos incrementales según su punto de inicio. Finalmente se escribe al vector de posiciones el valor de desplazamiento calculado, expresado en pasos.

Posterior a esto, se implementó otra función que se encarga de ejecutar los cálculos realizados por la anterior:

---

Función de ejecución de la cinemática inversa, entorno: Arduino.

---

```
1: procedure EXECUTE( $x, y$ )
2:   Set  $pd[0]$  to  $x$ 
3:   Set  $pd[1]$  to  $y$ 
4:   Call IKINEMATICS()
5:   Set the target positions of the motors to positions
6:   Move the motors to the target positions with controlled speed
7: end procedure
```

---

Esta función toma como argumento la posición deseada del efector final, actualiza el vector, ejecuta la anterior función y ejecuta el movimiento en la instancia de los motores.

Se encuentra que se realizó satisfactoriamente el sistema de posicionamiento de un robot paralelo 2T para el efector final presentado, teniendo esto, se procede a la sección de algoritmos de búsqueda de punto máximo.

### 3.4. Algoritmo

El algoritmo desarrollado presenta la necesidad de converger al punto máximo local, inicialmente se presenta un acercamiento a su implementación en el software *MatLab*.

En esta implementación se tendrán en cuenta dos cosas principalmente, la simulación de las condiciones reales del problema, y la implementación del algoritmo en si.

#### 3.4.1. Simulación de las condiciones

La simulación de las condiciones reales se observara por medio de algunas decisiones en el código de *MatLab*.

La primera de estas decisiones corresponde a la simulación del campo magnético a medir, en este caso se considera que visto directamente, el campo magnético se verá de una forma similar a una distribución de probabilidad de gauss, donde el punto de mayor magnitud es aquel donde las líneas de campo son completamente perpendiculares al sensor, mientras que los puntos de menor magnitud serán aquellos donde las líneas sean paralelas al sensor.

Por este motivo se realiza una función para generar una imagen de distribución de probabilidad de gauss de tamaño cuadrado y con sigma ( $\sigma$ ) variable:

Como se observa, el resultado de esta función será una imagen de tamaño  $m$  con una distribución gaussiana:

```
1: procedure GAUSS( $m, \sigma$ )
2:   Create meshgrid  $x$  and  $y$  from  $-m$  to  $m$ 
3:   Compute  $Exp\_comp$  as  $-(x^2 + y^2)/(2 \times \sigma^2)$ 
4:   Compute  $I$  as  $1 - (Exp\_comp / \min(\min(Exp\_comp)))$ 
5:   Return  $I$ 
6: end procedure
```

---

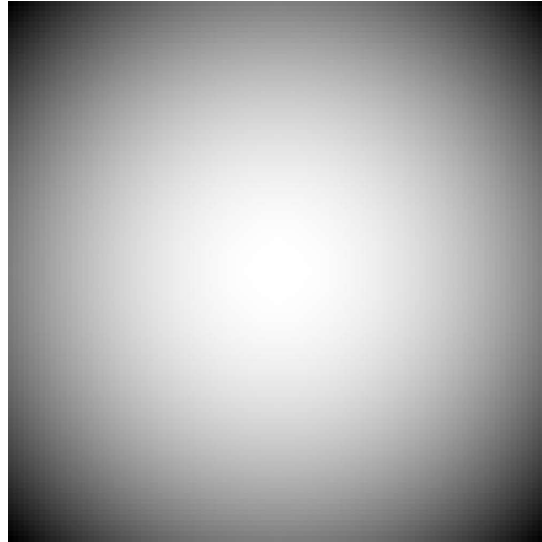


Figura 3.27: Campo de pruebas generado.

Donde la claridad del color indica la intensidad del flujo de campo magnético que registraría el sensor para cada posición.

Otra condición a ser simulada será la ubicación del sensor, para esto se asegurará que el algoritmo calcule una posición inicial aleatoria para el sensor cada vez que se ejecute.

Finalmente se presentara una imagen en formato RGB (Red, Green, Blue, por sus siglas en inglés), de forma que se pueda identificar el punto de partida del sensor, los puntos que visita durante su recorrido y el punto final.

### 3.4.2. Algoritmo

Se realiza un algoritmo de ascenso de gradiente simple, el cual va a calcular la dirección de desplazamiento del sensor en 'x' y 'y' en función del cambio en la medición que registre, de forma que si el movimiento en una dirección causa un aumento en la medida, el algoritmo calcula el siguiente desplazamiento en esa dirección:



$$\Delta P = a \cdot \text{sign}\left(\frac{M_1 - M_2}{P_1 - P_2}\right) \quad (3.15)$$

Se puede observar que esta ecuación es análoga a la segunda parte de la ecuación 2.9, donde la variable  $a$  corresponde a el tamaño de paso, que en este caso se trabaja como una constante, definiendo cuanto avanzará el sensor en la dirección seleccionada. La función  $\text{sign}\left(\frac{M_1 - M_2}{P_1 - P_2}\right)$  corresponde a la gradiente de la función de campo magnético que se encuentra evaluada en función de su desplazamiento en  $P$ . En esta ecuación se tiene  $\Delta P$  como el cambio en la posición para variaciones en 'x' o 'y', y  $M_1 - M_2$  como el cambio entre la medición actual y la anterior.

### 3.4.3. Rutina de programación

La implementación del algoritmo inicia realizando la limpieza de variables y ventanas del sistema, para posteriormente generar el campo de prueba, usando un tamaño de imagen de  $s = 101$ :

---

Inicialización del campo y el sensor simulado, entorno: MatLab

---

- 1: Clear the command window, workspace, and close all figures
  - 2: Create a figure with index 1
  - 3: Set the size  $s$  to 101
  - 4: Compute  $m$  as  $\lceil (s - 1)/2 \rceil$
  - 5: Initialize  $I$  using `GAUSS(m, 100)`
  - 6: Create an RGB image  $L$  with all channels equal to  $I$
  - 7: Display the image  $L$  after resizing it by a factor of 10 using nearest-neighbor interpolation
  
  - 8: Generate a random integer  $x$  in the range  $[1, s]$
  - 9: Generate a random integer  $y$  in the range  $[1, s]$
  - 10: Set  $med$  as the pixel value of  $I$  at coordinates  $(x, y)$
  - 11: Update the pixel at coordinates  $(x, y)$  in  $L$  with the value of  $med$
  - 12: Set the RGB channels at coordinates  $(x, y)$  in  $L$  to  $[1;0;0]$
  - 13: Display the updated image  $L$  after resizing it by a factor of 10 using nearest-neighbor interpolation
  - 14: Set  $xgrad$  to -1
  - 15: Set  $ygrad$  to -1
  - 16: Set  $a$  to 1
  - 17: Set  $err$  to 1
  - 18: Set  $i$  to 1
- 

Se inicializa la posición del sensor, se realiza la primera medida y se inicializan las variables correspondientes. La variable  $med$  llevará la medida tomada por el sensor, se puede observar el uso de las variables de valor aleatorio para iniciar las coordenadas del sensor. Posteriormente se realiza la medición del campo en esas posiciones, y se hace que en el punto de inicio se presente un píxel rojo para identificarlo. Se inicializan las variables de gradiente en 'x' y 'y' que serán los movimientos

iniciales que realizara el sensor, la tasa de aprendizaje se inicializa en 1 píxel de desplazamiento, el error se inicializa para que inicie el bucle y se inicia el conteo de iteraciones con la variable  $i$ .

Se procede al bucle *while*:

---

Algoritmo de posicionamiento en operación por iteraciones, entorno: MatLab

---

```

1: Initialize the variables:  $err \leftarrow 1$ ,  $i \leftarrow 1$ 
2: while  $|err| > 0$  and  $i < 50$  do
3:   Update  $y1$  as  $y + ygrad$ 
4:   Compute  $med1$  as the pixel value of  $I$  at coordinates  $(x, y1)$ 
5:   Update  $err$  as  $med - med1$ 
6:   Update  $med$  with  $med1$ 
7:   Update  $ygrad$  as  $a \cdot \text{sign}\left(\frac{err}{y-y1}\right)$ 
8:   Update  $y$  with  $y1$ 
9:   Update the RGB channels at coordinates  $(x, y)$  in  $L$  to  $[0;0;1]$ 
10:  Display the updated image  $L$  after resizing it by a factor of 5 using nearest-neighbor interpolation
11:  Update the display
12:  Display the current values of  $x$  and  $y$ 

13:  Update  $x1$  as  $x + xgrad$ 
14:  Compute  $med1$  as the pixel value of  $I$  at coordinates  $(x1, y)$ 
15:  Update  $err$  as  $med - med1$ 
16:  Update  $med$  with  $med1$ 
17:  Update  $xgrad$  as  $a \cdot \text{sign}\left(\frac{err}{x-x1}\right)$ 
18:  Update  $x$  with  $x1$ 
19:  Update the RGB channels at coordinates  $(x, y)$  in  $L$  to  $[0;0;1]$ 
20:  Display the updated image  $L$  after resizing it by a factor of 5 using nearest-neighbor interpolation
21:  Update the display
22:  Display the current values of  $x$  and  $y$ 

23:  Increment  $i$  by 1
24: end while

25: Update the RGB channels at coordinates  $(x, y)$  in  $L$  to  $[0;1;0]$ 
26: Display the final image  $L$  after resizing it by a factor of 5 using nearest-neighbor interpolation
27: Update the display

```

---

Donde se iterará siempre que el número de iteraciones sea menor a 50 y que el valor absoluto del error sea mayor a 0.

Se inicia recalculando la nueva posición vertical según la gradiente anterior, se mide en esta nueva posición y se calcula la nueva gradiente para la siguiente iteración, esto mientras se marca el camino en el que el sistema ha medido, se realiza el mismo procedimiento para el eje vertical y se

realiza el bucle. Eventualmente cuando el sistema converge, se procede a poner un píxel verde en el punto final.

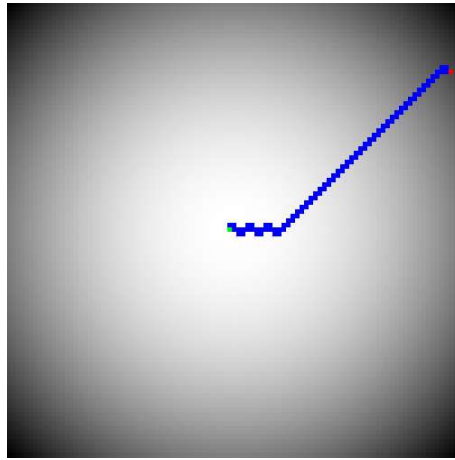


Figura 3.28: Simulación realizada.

En la Figura 3.28 se presenta una prueba del algoritmo presentado, identificando que es satisfactorio para proceder a la etapa de integración de los tres sistemas principales

## 3.5. Integración

La integración de los sistemas consiste principalmente en sujetar el elemento sensor al *holder*, e integrar las funciones del algoritmo al código de posicionamiento ya implementado.

### 3.5.1. Integración física

Se presenta el sistema completamente ensamblado, integrando el circuito de adecuación del elemento sensor así como el acoplamiento del mismo al elemento *holder* del sistema.

Teniendo en cuenta que los cables de conexión al elemento sensor pueden llegar a generar un elemento de torsión, se asegura la posición del sensor ejerciendo tensión mecánica extra en los cables, que será dada por medio del código.

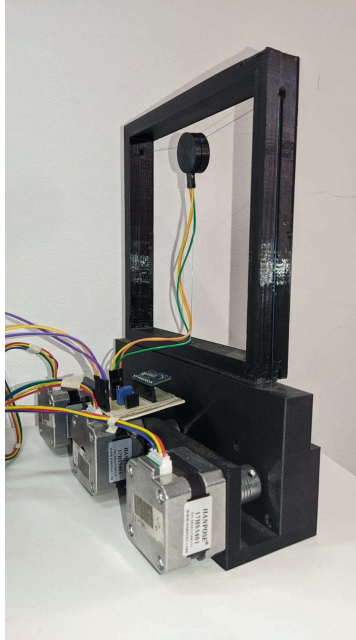


Figura 3.29: Integración física completa del sistema.

### 3.5.2. Integración en software

Para la parte de integración de software, se decide usar las funciones ya establecidas, así como nuevas funciones para distintos procesos del sistema. Principalmente integrando un sistema de selección de modo de operación, el cual recibirá un *Byte* del puerto serie y decidirá la acción del sistema según este.

Se inicia incluyendo las librerías correspondientes a movimiento y a medición como ya se ha mostrado en sus correspondientes secciones.

---

Función para la desviación estándar de cada medida, entorno: Arduino

---

```

1: function SD
2:   Initialize variables:  $sum \leftarrow 0$ ,  $med \leftarrow \text{latest\_mean}()$ 
3:   for  $i \leftarrow 0$  to  $N - 1$  do
4:     Update  $sum$  as  $sum + (\text{history}[i] - med)^2$ 
5:   end for
6:   Calculate  $sd$  as  $\sqrt{\frac{sum}{N-1}}$ 
7:   return  $sd$ 
8: end function

```

---

Se procede declarando las funciones principales del sistema de medición, así como las variables de funcionamiento de esta sección, adicionando una nueva función para el cálculo de la desviación estándar de cada medida realizada, dato que se considera relevante para la etapa experimental

subsecuente.

---

Variables de operación, entorno: Arduino

---

- 1: Declare and initialize variables:
  - 2:  $ft \leftarrow 1$
  - 3:  $Operation \leftarrow 'S'$
  - 4:  $xgrad \leftarrow -1$
  - 5:  $ygrad \leftarrow -1$
  - 6:  $grad \leftarrow 1$
  - 7:  $err \leftarrow 1$
- 

Se procede a declarar las variables e instancias necesarias para los motores, y se encuentra una variable nueva, el factor de tensión ( $ft$ ), que se encargará de hacer que para cada posición los cables tengan un milímetro menos que la distancia calculada, permitiendo que la tensión ejercida mantenga el *holder* en la orientación deseada. Posterior a esto se agregan las variables geométricas y las inicializaciones necesarias para el cálculo de la cinemática inversa, todo esto según se desglosó en su sección correspondiente. Posterior a esto se declara la variable *Operation* que almacenará la operación que el sistema debe realizar, inicializándose en 'S'. Se inician las variables para el algoritmo de posicionamiento, siendo estas la gradiente en cada eje, la gradiente general que gobernará cuanto se desplaza el sensor por cada iteración, en este caso 1 *mm*, y el error inicial.

La etapa de configuración se realiza de forma similar a las anteriores, se configuran las velocidades máximas, se agregan los motores a las instancias, se inicializa el puerto serie, así como el módulo ADS con su PGA, y se realiza una operación de tare a la medida del sensor por medio de la función *Zero* discutida anteriormente.

Se generan nuevas funciones, como la que se encargará de leer el byte enviado a puerto serie, haciendo uso de retorno y salto de línea, así como la función signo y la función del algoritmo de posicionamiento:

---

Funciones de procesamiento serial, signo de la operación gradiente y algoritmo de posicionamiento, entorno Arduino

---

```
1: procedure PROCESSINCOMINGBYTE2(const byte inByte)
2:   if inByte is ' n' or inByte is ' r' then
3:     break
4:   else
5:     Operation  $\leftarrow$  inByte
6:   end if
7: end procedure
8: function SIGN(float num):
9:   float s  $\leftarrow$  0
10:  if num  $\neq$  0 then
11:    s  $\leftarrow$  1
12:  end if
13:  if num == 0 then
14:    s  $\leftarrow$  0
15:  end if
16:  if num  $\neq$  0 then
17:    s  $\leftarrow$  -1
18:  end if
19:  return s
20: end function
21: procedure ALG2
22:  float med  $\leftarrow$  abs(latest_mean())
23:  float x  $\leftarrow$  pd[0]
24:  float y  $\leftarrow$  pd[1]
25:  Execute(x, y + ygrad)
26:  Measure()
27:  err  $\leftarrow$  med - abs(latest_mean())
28:  ygrad  $\leftarrow$  grad * sign(err / (y - pd[1]))
29:  med  $\leftarrow$  abs(latest_mean())
30:  x  $\leftarrow$  pd[0]
31:  y  $\leftarrow$  pd[1]
32:  Execute(x + xgrad, y)
33:  Measure()
34:  err  $\leftarrow$  med - abs(latest_mean())
35:  xgrad  $\leftarrow$  grad * sign(err / (x - pd[0]))
36: end procedure
```

---

La primera asigna a *Operation* el byte que se haya recibido por serial, posteriormente, *sign* retorna el signo del número que toma como argumento, con la característica de retornar 0 cuando el número es 0. Finalmente la implementación del algoritmo sigue el propuesto en el software *MatLab*, de forma que toma una medición en el punto en el que se encuentra, actualiza la posición vertical del sensor y toma otra medida, después de calcular el error y la nueva gradiente procede a

realizar lo mismo en el eje horizontal.

Se agregan dos nuevas funciones para el funcionamiento del sistema, valor máximo de un vector y barrido:

---

Función para máximo de un vector y función para realizar un barrido automatizado del área de trabajo, entorno: Arduino

---

```
1: function MAXARR(float M[])
2:   int res ← 0
3:   for  $i \leftarrow 1$  to 3 do
4:     if  $|M[i - 1]| < |M[i]|$  then
5:       res ←  $i$ 
6:     end if
7:   end for
8:   return res
9: end function
10: procedure MAP(float x1, float x2, float y1, float y2)
11:   Serial.println("Inicio Mapeo—————")
12:   for  $x \leftarrow x1$  to  $x2$  do
13:     for  $y \leftarrow y1$  to  $y2$  do
14:       Execute(x, y)
15:       Serial.print("X: ")
16:       Serial.print(x)
17:       Serial.print("Y: ")
18:       Serial.print(y)
19:       for  $i \leftarrow 0$  to 2 do
20:         Measure()
21:       end for
22:       Serial.print("B[mT]: ")
23:       Serial.print(latest_mean(), 10)
24:       Serial.print("T: ")
25:       Serial.println(millis())
26:       while Serial.available()  $\neq 0$  do
27:         Operation ← 'S'
28:         break
29:       end while
30:     end for
31:   end for
32:   Serial.println("Finaliza Mapeo—————")
33: end procedure
```

---

El valor máximo de un vector será utilizado para inicializar la posición del sensor, de forma que se ubique en cuatro puntos específicos, realice las mediciones y se ubique en el punto que haya registrado la medida más alta, para iniciar la iteración del algoritmo ahí. La función de mapeo toma un rango de valores en el eje horizontal y en el eje vertical, haciendo que el sensor se posicione en

estos puntos y reporte la medida en el puerto serial, de forma que se pueda caracterizar por medio de un barrido, el campo magnético sobre el que se está trabajando, lo que será de gran utilidad para los experimentos.

Finalmente, en el bucle, se tiene la sentencia *switch* que evaluará el modo de operación del sistema, ubicándolo en uno de los siguientes estados:

<b>Modo de operación</b>	<b>Descripción</b>
S	Detiene todos los procesos del sistema.
H	Envía el sensor a la posición de hogar, posteriormente ejecuta 'S'.
M	Realiza la medición del campo magnético, devolviendo su magnitud y la desviación de los datos tomados, permanece iterando hasta que haya un cambio de operación.
T	Realiza una prueba de movimiento, ubicando el sensor en diferentes puntos específicos del área de trabajo, posteriormente ejecuta 'H'.
A	Inicia el algoritmo de posicionamiento automático, permanece iterando hasta que haya un cambio de operación.
Z	Realiza la operación de tare al sensor, posteriormente ejecuta 'M'.
C	Envía el sensor a la posición en el centro del área de trabajo, posteriormente ejecuta 'S'.
E	Realiza la evaluación del campo, ubicando el sensor en cuatro puntos específicos, para finalmente ubicarlo en el punto que haya registrado el mayor campo, posteriormente ejecuta 'M'.
P	Realiza la operación de mapeo, posteriormente ejecuta 'H'.

Tabla 3.8: Comandos del sistema.

Dando por finalizada la etapa de implementación del sistema, obteniéndose un sistema de posicionamiento autoajutable que permite ejecutar un algoritmo para hallar el punto donde el campo magnético es máximo.



# Resultados

Se realizan dos tipos distintos de experimentos, inicialmente compuestos por el sistema y por un generador de campo magnético constante, y posteriormente integrando el sistema a una bobina de Helmholtz.

## 4.1. Campo magnético constante

Para los experimentos con campo magnético constante, se deja el marco del sistema fijo, presentando el generador de campo en diferentes puntos y con diferentes orientaciones respecto al sensor en su posición original.

El objetivo de estos experimentos es identificar el correcto funcionamiento del sistema, de forma que se observe un cambio en la posición, la convergencia a una posición específica y una mejora en la lectura del campo magnético, en la forma de un aumento en la magnitud.

El primer experimento ubica el sensor en el centro del sistema y se posiciona el generador de campo de la siguiente forma:

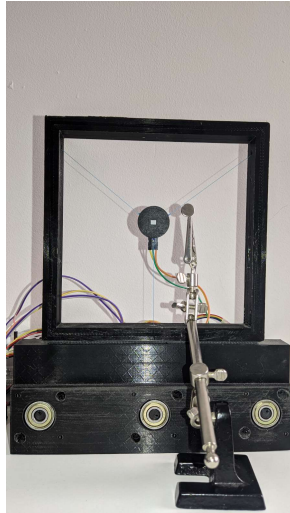
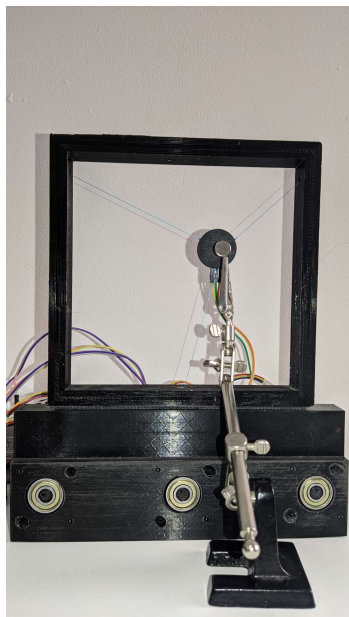
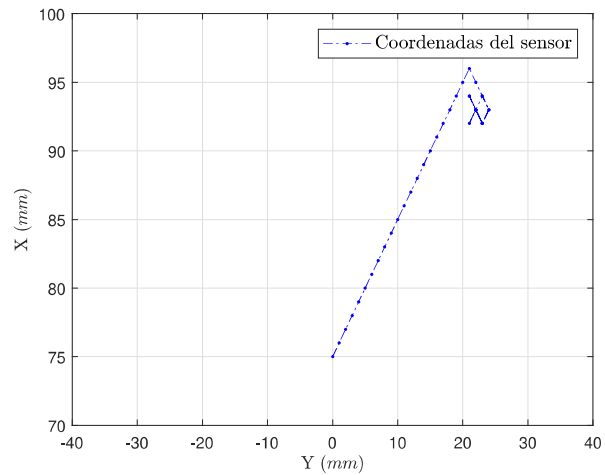


Figura 4.1: Posición inicial del experimento.

Donde se procede a iniciar el algoritmo de posicionamiento, que converge en posición de la siguiente manera:



(a)



(b)

Figura 4.2: Posicionamiento durante el experimento.

En la Figura 4.2(a) se observa la posición final a la que llega el sistema, mientras que en el apartado (b) se puede ver los datos de posicionamiento generados durante el experimento, los cuales corroboran su trayectoria y punto de convergencia.

Los resultados de la medición del sistema se pueden observar en la siguiente gráfica:

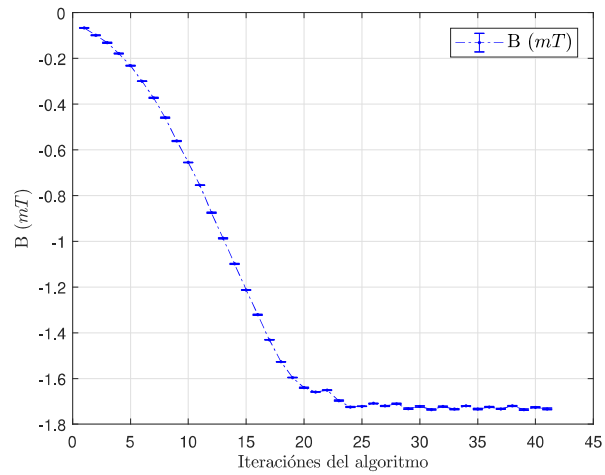


Figura 4.3: Convergencia al punto de medición deseado.

Donde se observa un aumento en la magnitud de campo magnético medido, identificando que se llega a un punto de medición deseado.

Sin embargo, se identifica la necesidad de realizar experimentos en un campo que se pueda caracterizar, que tenga un valor de campo magnético deseado y una forma de campo magnético identificable, por lo que se procede a la etapa de experimentación usando un arreglo de bobinas de Helmholtz.

## 4.2. Arreglo de bobinas de Helmholtz

El proceso de integración del sistema al arreglo de bobinas se lleva a cabo de la siguiente manera:

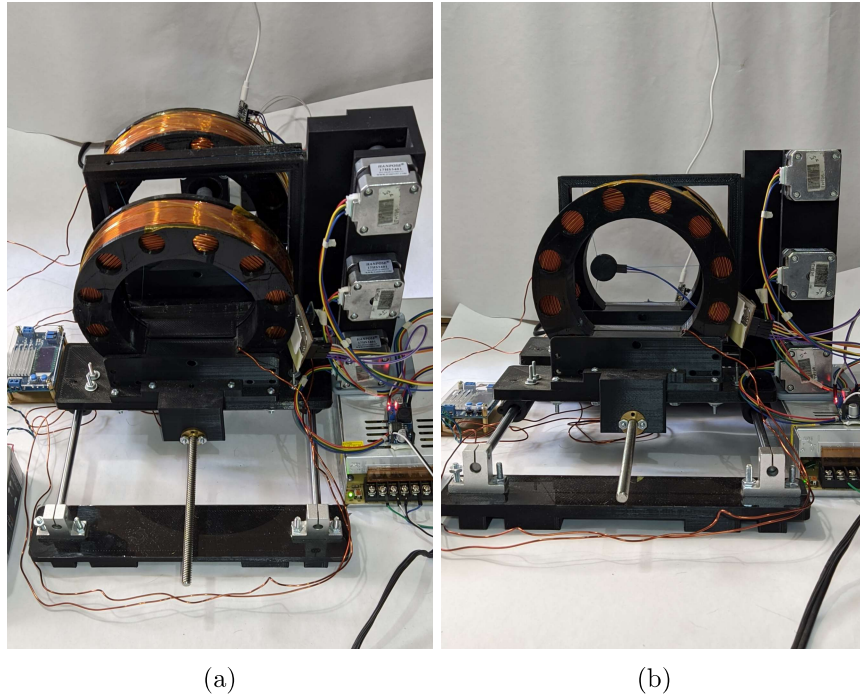


Figura 4.4: Integración física de los sistemas.

Donde el sistema de posicionamiento se puede ubicar de forma horizontal para realizar las mediciones entre el arreglo de bobinas.

Los experimentos se realizarán con dos variaciones del parámetro de magnitud de campo magnético, primero midiendo un campo de  $10\text{ mT}$  y posteriormente un campo de  $1\text{ mT}$ , se realiza el barrido del campo magnético para caracterizarlo como zona de trabajo, y posterior a esto, se realizarán 10 mediciones de campo magnético con el algoritmo de posicionamiento y se contrastarán con otras 10 mediciones realizadas ubicando el sensor de forma tradicional dentro del campo magnético, procurando centrarlo en el punto donde se conoce que se expresa el campo magnético máximo.

El primer experimento se da de la siguiente forma:

## Experimento 10 $mT$

Inicialmente se realiza el barrido del área de trabajo, este es realizado desde  $-20\text{ mm}$  hasta  $20\text{ mm}$  en el eje 'X' y desde  $60\text{ mm}$  hasta  $124\text{ mm}$  en el eje 'Y' según el marco de referencia del sistema de posicionamiento, el resultado de este proceso de barrido se puede observar en la siguiente figura:

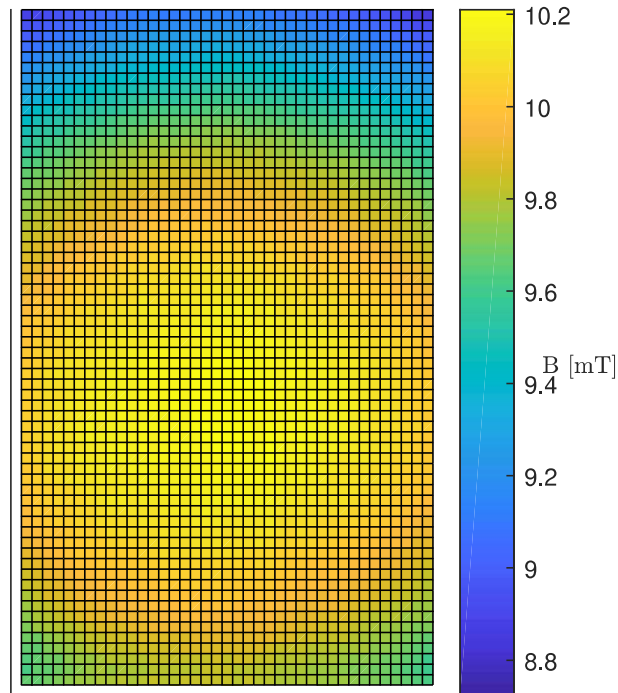


Figura 4.5: Barrido del experimento a  $10\text{ mT}$ .

Donde se puede observar las escala de magnitudes y una distribución similar a la distribución de probabilidad de gauss simulada en la Figura 3.27.

Las bobinas de Helmholtz usadas para este experimento tienen un radio de  $12\text{ cm}$  a 400 vueltas cada una, siendo de cobre calibre 22 AWG. El arreglo se encuentra encendido con una corriente de  $3.3\text{ A}$ , lo que resulta en un campo de  $10\text{ mT}$ , en el barrido, el valor máximo observado en el área de trabajo es de  $10.21\text{ mT}$ .

Posteriormente se realizan 10 mediciones de campo magnético usando el algoritmo y el método tradicional para así contrastarlos. A continuación se observan las curvas de convergencia para el algoritmo en sus 10 iteraciones:

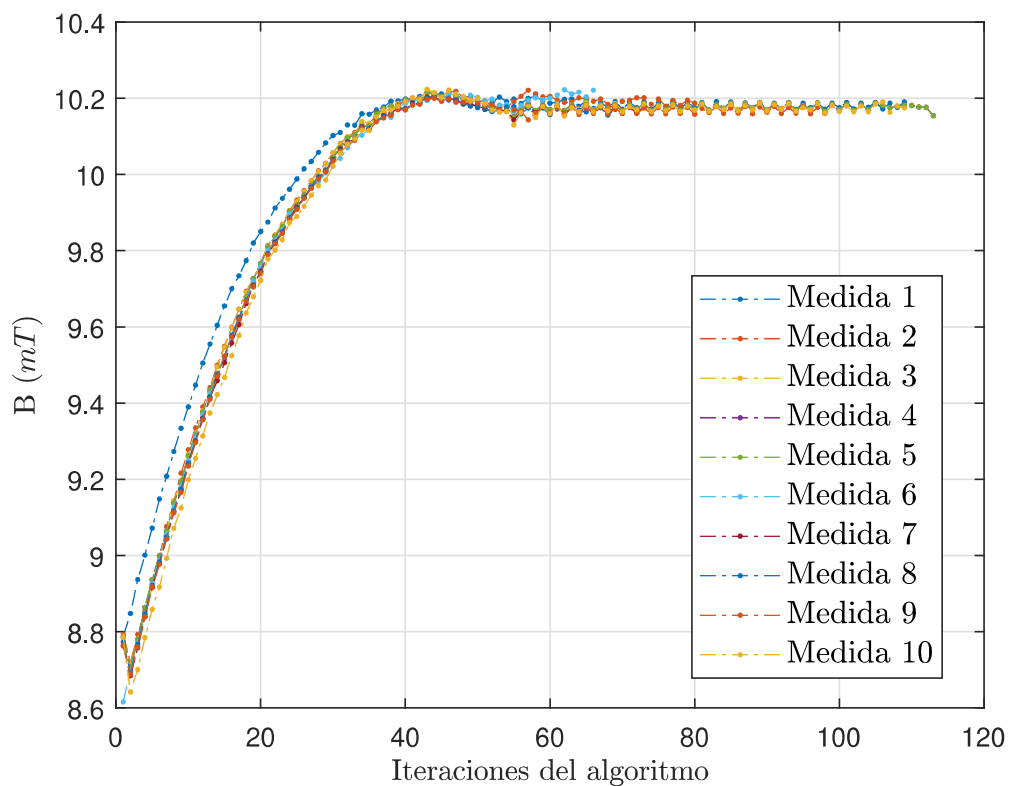


Figura 4.6: Medidas del experimento a 10  $mT$ .

Corroborando la repetibilidad de las mediciones realizadas con el sistema. Posteriormente se hace el contraste con las mismas medidas tomadas de forma tradicional:

Tabla 4.1: Comparativa de mediciones experimento 10  $mT$ .

Número de medición	Algoritmo (mT)	Método tradicional (mT)
1	10.211	9.211
2	10.221	9.834
3	10.214	8.310
4	10.215	9.452
5	10.209	9.267
6	10.222	8.600
7	10.203	9.052
8	10.203	8.853
9	10.200	8.129
10	10.200	9.919

Observando superficialmente los datos es posible identificar que el algoritmo presenta mayor

exactitud y precisión que el método tradicional. Lo que se corrobora con las siguientes métricas estadísticas:

Tabla 4.2: Comparativa de métricas experimento 10  $mT$ .

	<b>Algoritmo (mT)</b>	<b>Método tradicional (mT)</b>
<b>Promedio</b>	10.210	9.062
<b>Desviación estándar</b>	0.008	0.599

Donde el promedio confirma la exactitud de las medidas, mientras que la desviación estándar confirma la precisión de las mismas. Resultando en una mejora general de la desviación de la medida correspondiente a un 98.63 %.

## Experimento 1 $mT$

Inicialmente se realiza el barrido del área de trabajo, este es realizado desde  $-20\text{ mm}$  hasta  $20\text{ mm}$  en el eje 'X' y desde  $60\text{ mm}$  hasta  $124\text{ mm}$  en el eje 'Y' según el marco de referencia del sistema de posicionamiento, el resultado de este proceso de barrido se puede observar en la siguiente figura:

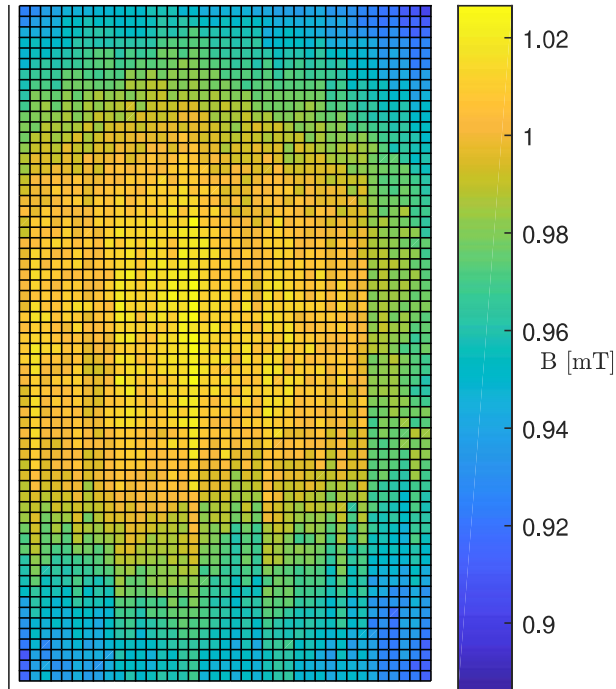


Figura 4.7: Barrido del experimento a  $1\text{ mT}$ .

Donde se puede observar la escala de magnitudes y una distribución similar a la distribución de probabilidad de gauss simulada en la Figura 3.27, también se puede observar que esta distribución ya no es tan uniforme como la del experimento anterior, lo que puede ser atribuido a la disminución, en un orden de magnitud, del campo magnético a medir.

La bobina de Helmholtz usada para este experimento se encuentra encendida con una corriente de  $0.3\text{ A}$ , lo que resulta en un campo de  $1\text{ mT}$ , según el barrido, el valor máximo observado en el área de trabajo es de  $1.027\text{ mT}$ .

Posteriormente se realizan 10 mediciones de campo magnético usando el algoritmo y el método tradicional para así contrastarlos. A continuación se observan las curvas de convergencia para el



algoritmo en sus 10 iteraciones:

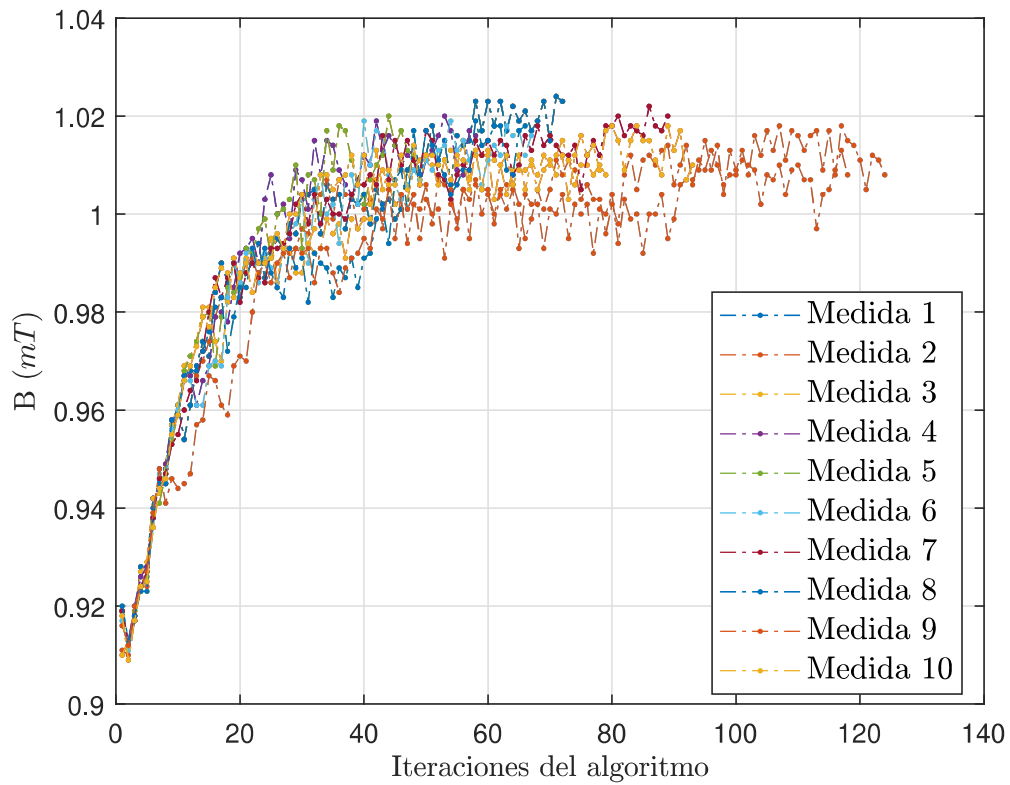


Figura 4.8: Medidas del experimento a 1  $mT$ .

Corroborando la convergencia de las mediciones realizadas con el sistema. Posteriormente se hace el contraste con las mismas medidas tomadas de forma tradicional:

Tabla 4.3: Comparativa de mediciones experimento 1  $mT$ .

Número de medición	Algoritmo (mT)	Método tradicional (mT)
1	1.024	1.062
2	1.014	1.064
3	1.015	1.007
4	1.02	1.006
5	1.02	0.989
6	1.019	1.004
7	1.022	1.008
8	1.019	1.014
9	1.018	0.962
10	1.018	1.030

Observando superficialmente los datos es posible identificar que el algoritmo presenta mayor exactitud y precisión que el método tradicional. Lo que se corrobora con las siguientes métricas estadísticas:

Tabla 4.4: Comparativa de métricas experimento 1  $mT$ .

	<b>Algoritmo (mT)</b>	<b>Método tradicional (mT)</b>
<b>Promedio</b>	1.018	1.013
<b>Desviación estándar</b>	0.003	0.032

Donde el promedio confirma la exactitud de las medidas, mientras que la desviación estándar confirma la precisión de las mismas. Resultando en una mejora general de la desviación de la medida correspondiente a un 90.79 %.

## Conclusión

En esta investigación, se logró el cumplimiento de los objetivos propuestos, lo que ha permitido obtener resultados coherentes y relevantes. A través del desarrollo de un sistema de medición de campo magnético con un sistema de posicionamiento autoajustable integrado, se logró mitigar los errores de medición atribuidos al posicionamiento del sensor en al menos un 90%, como se puede comprobar en las Tablas 4.3 y 4.4.

Los resultados obtenidos demuestran que el sistema propuesto presenta una mejora significativa en la exactitud y precisión de las mediciones de flujo de campo magnético en comparación con los métodos tradicionales. Esta mejora se justifica gracias a la comprensión desarrollada de los fenómenos físicos involucrados en el sistema de medición, que guiaron tanto el diseño del sistema de medición e instrumentación como el diseño y construcción del sistema de ajuste automático. La metodología experimental empleada permitió contrastar el método tradicional con el método desarrollado en este proyecto, validando así la superioridad del sistema propuesto.

Además, este sistema no solo representa una mejora tecnológica en los procesos de medición de campo magnético, sino que también se puede adaptar y aplicar en diferentes contextos y necesidades específicas. Aporta mediciones más precisas y confiables del campo magnético dentro del rango de trabajo requerido.

Es importante destacar que este sistema permite ubicar el sensor de forma automática siempre en el mismo punto espacial donde el flujo magnético es máximo a pesar de la orientación que pueda tener, lo que mejora la consistencia y confiabilidad de las mediciones. Sin embargo, se reconoce que existen oportunidades para ampliar los alcances de la implementación. Por ejemplo, se puede buscar hacer el sistema más compacto y portátil, sin generar conflictos entre los campos magnéticos de los motores y el campo medido por el sensor. Además, se puede considerar la implementación de una batería para independizarse de un sistema de alimentación externo. Estas estrategias de mejora permitirían una mayor versatilidad y autonomía en la operación del sistema.

El desarrollo de este sistema permite presentar una mejora tecnológica a los procesos de medición de flujo de campo magnético, pues esta metodología puede ser aplicada y adaptada para diferentes procesos o necesidades, aportando a mediciones más acertadas de campo magnético dentro del rango de trabajo que sea necesario.

---

# Bibliografía

- [1] A. Mukhamedgali and Z. B. Rakisheva, “Design and simulation of uniform magnetic field,” *Balkan journal of electrical & computer engineering*, vol. 6, no. 4, pp. 232–236, 2018.
- [2] R. G. Jackson, *Novel Sensors and Sensing*. IOP Publishing, 2004.
- [3] A. Velazquez, J. Baena, and F. Gutiérrez, “Diseño y construcción de un teslámetro digital embebido para la caracterización magnética de materiales,” *Ingeniería y Ciencia*, vol. 11, no. 21, pp. 99–113, 2015.
- [4] T. Bruckmann, L. Mikelsons, T. Brandt, M. Hiller, and D. Schramm, “Wire robots part i: Kinematics, analysis & design,” in *Parallel Manipulators, New Developments*, I-Tech Education and Publishing, Apr. 2008.
- [5] H. Kino, Fukuoka Institute of Technology, S. Kawamura, and Ritsumeikan University, “Mechanism and control of parallel-wire driven system,” *J. Robot. Mechatron.*, vol. 27, pp. 599–607, Dec. 2015.
- [6] “Arduino cnc shield – 100 % grbl compatible,” Jul 2018.
- [7] N. Fischer, J. Kriechbaum, D. Berwanger, and F. Mathis-Ullrich, “Compliant hall-effect sensor array for passive magnetic instrument tracking,” *IEEE Sensors Letters*, vol. 7, pp. 1–4, Mar. 2023.
- [8] C. Hyunchul, J. Semi, G. Gwangjun, L. Cheong, Z. Jin, Y. K. Seong, P. Jong-Oh, and S. Park, “Equitranslational and axially rotational microrobot using electromagnetic actuation system,” *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1342–1350, 2017.
- [9] U. Riaz, F. A. Razzaq, A. Areces-Gonzalez, M. C. Piastra, M. L. B. Vega, D. Paz-Linares, and P. A. Valdés-Sosa, “Automatic quality control of the numerical accuracy of eeg lead fields,” *NeuroImage*, vol. 273, 2023.

- [10] L. Pan, N. Shang, J. Shangguan, M. Figini, W. Xing, B. Wang, C. Sun, J. Yang, Y. Zhang, S. Hu, Q. Ma, J. Wang, Y. Velichko, V. Yaghamai, A. Benson, and Z. Zhang, “Magnetic resonance imaging monitoring therapeutic response to dendritic cell vaccine in murine orthotopic pancreatic cancer models,” *American journal of cancer research*, vol. 9, pp. 562–573, 2019.
- [11] P. Baranov, V. Baranova, S. Uchaikin, and Y. Pisarenko, “Creating a uniform magnetic field using axial coils system for calibration of magnetometers,” pp. 1–5, 2016.
- [12] E. Gónzales and M. Apanco, *Campo magnético residual en máquinas rotatorias*.
- [13] L. Tomkow, A. Smara, and B. A. Glowacki, “Application of hot press bending for shaping a stack of hts tapes operating as a trapped flux magnet,” *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 8, pp. 1–5, 2020.
- [14] R. A. Serway and J. W. Jewett, *Física para ciencias e ingeniería con física moderna*, vol. 2. Cengage Learning, 7 ed., 2009.
- [15] W. G. Jenks, S. S. H. Sadeghi, and J. P. Wikswow, “SQUIDS for nondestructive evaluation,” *Journal of Physics D: Applied Physics*, vol. 30, no. 3, pp. 293–323, 1997.
- [16] M. J. Caruso, T. Bratland, C. H. Smith, and R. Schneider, “A new perspective on magnetic field sensing,” *Sensors (Peterborough, NH)*, vol. 15, no. 12, pp. 34–46, 1998.
- [17] P. Ripka, “Advances in fluxgate sensors,” *Sensors and Actuators A: Physical*, vol. 106, no. 1-3, pp. 8–14, 2003.
- [18] D. He and M. Shiwa, “Deep defect detection using eddy current testing with amr sensor,” *Progress in Electromagnetics Research Symposium*, pp. 493–495, 2013.
- [19] A. Jander, C. Smith, and R. Schneider, “Magnetoresistive sensors for nondestructive evaluation (invited paper),” in *Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring* (N. Meyendorf, G. Y. Baaklini, and B. Michel, eds.), SPIE, 2005.
- [20] J. Lee, J. Hwang, J. Jun, and S. Choi, “Nondestructive testing and crack evaluation of ferromagnetic material by using the linearly integrated hall sensor array,” *Journal of Mechanical Science and Technology*, vol. 22, pp. 2310–2317, 2008.
- [21] F. López-Huerta, G. Rojas-Nava, B. S. Soto-Cruz, and B. L. Herrera-May, “Diseño e implementación de un sistema de caracterización para microsensores de efecto hall,” *Revista Mexicana de Física*, vol. 59, no. 1, pp. 54–61, 2013.

- [22] R. Popovic, Z. Randjelovic, and D. Manic, “Integrated hall-effect magnetic sensors,” *Sensors and Actuators A: Physical*, vol. 91, no. 1-2, pp. 46–50, 2001.
- [23] S. R. Gyawali, “Design and construction of helmholtz coil for biomagnetic studies on soybean,” Master’s thesis, University of Missouri-Columbia, 2008.
- [24] A. Ogulmus, A. Cakan, and M. Tinkir, “Modeling and position control of scara type 3d printer,” *Int. J. Sci. Technol. Res*, vol. 5, no. 12, pp. 140–143, 2016.
- [25] S. Tsuda, D. Oleynikov, J. Gould, D. Azagury, B. Sandler, M. Hutter, S. Ross, E. Haas, F. Brody, and R. Satava, “Sages tovac safety and effectiveness analysis: da vinci® surgical system (intuitive surgical, sunnyvale, ca),” *Surgical endoscopy*, vol. 29, no. 10, pp. 2873–2884, 2015.
- [26] A. Fontanet, J. Marcos, L. Ribó, V. Massana, and J. Campmany, “Design and construction of 3d helmholtz coil system to calibrate 3d hall probes,” in *Journal of Physics: Conference Series*, vol. 1350, 2019.
- [27] G. Chen, L. Calabretta, M.-M. Xu, Y.-T. Song, Y.-H. Chen, K.-Z. Ding, J.-J. Li, O. Karamyshev, G. Karamysheva, and G. Shirkov, “Design and research of magnetic field mapping system for SC200,” *IEEE Transactions on Nuclear Science*, vol. 67, no. 1, pp. 369–373, 2020.
- [28] M. Blagojevic and D. Mancic, “The system for magnetic field mapping based on the integrated 3d hall sensor,” pp. 1–8, 2016.
- [29] V. Mattioni, E. Ida’, and M. Carricato, “Design of a planar cable-driven parallel robot for non-contact tasks,” *Applied Sciences*, vol. 11, p. 9491, Oct. 2021.
- [30] M. Carpio, R. Saltaren, J. Viola, C. García, J. Guerra, J. Cely, and C. Calderón, “A simulation study of a planar cable-driven parallel robot to transport supplies for patients with contagious diseases in health care centers,” *Robotics*, vol. 10, p. 111, Sept. 2021.
- [31] H. Kino, T. Imamura, N. Sakagami, and and, “Error analysis by kinetics for parallel-wire driven system using approximated inverse kinematics,” *Journal of Robotics and Mechatronics*, vol. 30, pp. 763–771, Oct. 2018.
- [32] C. Sun, H. Gao, Z. Liu, S. Xiang, H. Yu, N. Li, and Z. Deng, “Design and optimization of three-degree-of-freedom planar adaptive cable-driven parallel robots using the cable wrapping phenomenon,” *Mechanism and Machine Theory*, vol. 166, p. 104475, Dec. 2021.

- [33] S. Qian, B. Zi, W.-W. Shang, and Q.-S. Xu, “A review on cable-driven parallel robots,” *Chinese Journal of Mechanical Engineering*, vol. 31, Aug. 2018.
- [34] A. Ming, “Study on multiple degree-of-freedom positioning mechanism using wires (part 1) - concept, design and control,” *Int. J. Jpn Soc. Eng*, vol. 28, pp. 131–138, 2004.
- [35] R. Verhoeven, *Analysis of the Workspace of Tendon-based Stewart Platforms*. 2004.
- [36] R. P. Areny, *Sensores y acondicionadores de senal*. Alfaomega Grupo Editor, Dec. 2002.
- [37] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. Wiley Series in Discrete Mathematics and Optimization, Nashville, TN: John Wiley & Sons, 4 ed., Jan. 2013.
- [38] T. Chen and M. C. Messner, “Training material models using gradient descent algorithms,” *International Journal of Plasticity*, vol. 165, 2023.
- [39] M. McCauley, “Accelstepper library for arduino.”



---

# Rutinas de programación

En este apartado se encontraran las rutinas usadas a lo largo del documento.

## A.1. Arduino

### A.1.1. Medición

```
1 #include <Wire.h>
2 #include <Adafruit_ADS1X15.h>
3
4 float vol0 = 0;
5 Adafruit_ADS1115 ads;
6
7 void setup() {
8   Serial.begin(9600);
9
10  ads.setGain(GAIN_TWOTHIRDS);
11
12  ads.begin();
13
14 }
15
16 void loop(){
17   short adc0 = ads.readADC_SingleEnded(0);
18   vol0 = (float) (adc0 * 0.1875*8/5 - 2500 )/250;
19   Serial.println(vol0);
20 }
```

Código A.1: Inicialización y lectura del módulo ADC, entorno: Arduino.

```

1 void add_sample (float sample)
2 {
3     float previous = history[index];
4     history[index] = sample;
5     index++;
6     if(index >= N)
7         index = 0;
8     total -= previous;
9     total += sample;
10 }

```

Código A.2: Función para agregar muestras al promedio, entorno: Arduino.

```

1 float latest_mean ()
2 {
3     return float (total)/N;
4 }

```

Código A.3: Función para realizar el promedio, entorno: Arduino.

```

1 void Zero ()
2 {
3     for(int i = 0; i <=5*N; i++)
4     {
5         short adc0 = ads.readADC_SingleEnded(0);
6         a = (float) (adc0 * 0.1875*8/5)/250;
7         add_sample(a);
8     }
9     a = latest_mean ();
10 }

```

Código A.4: Función de ajuste a 0, entorno: Arduino.

### A.1.2. Posicionamiento

```

1 #include <AccelStepper.h>
2 #include <MultiStepper.h>
3
4 #define x_paso 2
5 #define x_dire 5
6 #define x_habi 8
7 #define y_paso 3
8 #define y_dire 6
9 #define y_habi 8

```

```

10 #define z_paso 4
11 #define z_dire 7
12 #define z_habi 8
13
14 #define motorInterfaceType 1
15
16 long positions [] = {0,0,0};
17
18 AccelStepper X = AccelStepper(motorInterfaceType, x_paso, x_dire);
19 AccelStepper Y = AccelStepper(motorInterfaceType, y_paso, y_dire);
20 AccelStepper Z = AccelStepper(motorInterfaceType, z_paso, z_dire);
21
22 MultiStepper Mot;
23
24 void setup() {
25     X.setMaxSpeed(1000);
26     Y.setMaxSpeed(1000);
27     Z.setMaxSpeed(1000);
28
29     Mot.addStepper(X);
30     Mot.addStepper(Y);
31     Mot.addStepper(Z);
32 }
33
34 void loop() {
35
36     Mot.moveTo(positions);
37     Mot.runSpeedToPosition();
38
39 }

```

Código A.5: Pruebas inicialización y control de los motores, entorno: Arduino.

```

1 #include <AccelStepper.h>
2 #include <MultiStepper.h>
3
4 #define x_paso 2
5 #define x_dire 5
6 #define x_habi 8
7 #define y_paso 3
8 #define y_dire 6
9 #define y_habi 8
10 #define z_paso 4
11 #define z_dire 7

```

```

12 #define z_habi 8
13
14 #define motorInterfaceType 1
15
16 AccelStepper X = AccelStepper(motorInterfaceType, x_paso, x_dire);
17 AccelStepper Y = AccelStepper(motorInterfaceType, y_paso, y_dire);
18 AccelStepper Z = AccelStepper(motorInterfaceType, z_paso, z_dire);
19
20 MultiStepper Mot;

```

Código A.6: Inicialización del control de posición, entorno: Arduino.

```

1 int ppm = 129;
2
3 float pd[] = {0,129};
4
5 long positions[] = {0,0,0};
6 float positionsmm[] = {0,0,0};
7 float IniPositions[] = {65.7,118,65.7};

```

Código A.7: Variables posicionales, entorno: Arduino.

```

1 float h = 140;
2 float w = 75;
3 float r = 11;
4
5 float alph = M_PI*3/2;
6 float bet = M_PI*5/6;
7 float gamm = M_PI/6;
8
9 float Zanch[] = {r*cos(bet), r*sin(bet)};
10 float Yanch[] = {r*cos(alph), r*sin(alph)};
11 float Xanch[] = {r*cos(gamm), r*sin(gamm)};
12
13 float Zpos[] = {-w, h};
14 float Ypos[] = {0, 0};
15 float Xpos[] = {w, h};

```

Código A.8: Variables cinemáticas, entorno: Arduino.

```

1 void IKinematics()
2 {
3     float Xsz[] = {pd[0] + Zanch[0], pd[1] + Zanch[1]};
4     float Xsy[] = {pd[0] + Yanch[0], pd[1] + Yanch[1]};
5     float Xsx[] = {pd[0] + Xanch[0], pd[1] + Xanch[1]};

```

```

6
7 float Lz = sqrt(sq(Xsz[0] - Zpos[0]) + sq(Xsz[1] - Zpos[1]));
8 float Ly = sqrt(sq(Xsy[0] - Ypos[0]) + sq(Xsy[1] - Ypos[1]));
9 float Lx = sqrt(sq(Xsx[0] - Xpos[0]) + sq(Xsx[1] - Xpos[1]));
10
11 positionsmm[0] = Lz - IniPositions[0];
12 positionsmm[1] = Ly - IniPositions[1];
13 positionsmm[2] = Lx - IniPositions[2];
14
15 positions[0] = (positionsmm[0]) *ppm;
16 positions[1] = (positionsmm[1]) *ppm;
17 positions[2] = (positionsmm[2]) *ppm;
18 }

```

Código A.9: Función de cinemática inversa, entorno: Arduino.

```

1 void Execute(float x, float y)
2 {
3
4 pd[0] = x;
5 pd[1] = y;
6
7 IKinematics();
8
9 Mot.moveTo(positions);
10 Mot.runSpeedToPosition();
11 }

```

Código A.10: Función de ejecución de la cinemática inversa, entorno: Arduino.

### A.1.3. Algoritmo

```

1
2 float sd()
3 {
4 float sum = 0;
5 float med = latest_mean();
6 for(int i = 0; i <N; i++)
7 {
8 sum += sq(history[i] - med);
9 }
10 float sd = sqrt(sum/(N-1));
11 return sd;

```

```
12 }
```

Código A.11: Función para la desviación estándar de cada medida, entorno: Arduino.

```
1
2 float ft = 1;
3
4 byte Operation = 'S';
5
6 float xgrad = -1;
7 float ygrad = -1;
8 float grad = 1;
9
10 float err = 1;
```

Código A.12: Variables de operación, entorno: Arduino.

```
1 void processIncomingByte2 (const byte inByte)
2 {
3   switch (inByte)
4     {
5
6     case '\n':
7       break;
8
9     case '\r':
10      break;
11
12    default:
13      Operation = inByte;
14      break;
15
16    }
17 }
18
19 float sign(float num)
20 {
21   float s = 0;
22   if(num > 0) s = 1;
23   if(num == 0) s = 0;
24   if (num < 0) s = -1;
25   return s;
26 }
27
28 void Alg2()
```

```

29 {
30   float med = abs(latest_mean());
31   float x = pd[0];
32   float y = pd[1];
33   Execute(x,y + ygrad);
34   Measure();
35   err = med - abs(latest_mean());
36   ygrad = grad*sign(err/(y - pd[1]));
37
38   med = abs(latest_mean());
39   x = pd[0];
40   y = pd[1];
41   Execute(x + xgrad ,y);
42   Measure();
43   err = med - abs(latest_mean());
44   xgrad = grad*sign(err/(x - pd[0]));
45 }

```

Código A.13: Funciones de procesamiento serial, signo de la operación gradiente y algoritmo de posicionamiento, entorno: Arduino.

```

1  int MaxArr(float M[])
2  {
3    int res = 0;
4    for(int i = 1; i < 4; i++)
5    {
6      if(abs(M[i-1]) < abs(M[i]))
7        {res = i;}
8
9    }
10   return res;
11 }
12
13 void Map(float x1, float x2, float y1, float y2)
14 {
15   Serial.println("Inicia Mapeo");
16   for(int x = x1; x < x2; x++)
17   {
18     for(int y = y1; y < y2; y++)
19     {
20       Execute(x,y);
21       Serial.print("X: ");
22       Serial.print(x);
23       Serial.print(" Y: ");

```

```

24     Serial.print(y);
25
26     for(int i = 0; i<3; i++)
27     {Measure();}
28     Serial.print("  B[mT]: ");
29     Serial.print(latest_mean(),10);
30     Serial.print("  T: ");
31     Serial.println(millis());
32
33     while (Serial.available () > 0){
34     Operation = 'S';
35     break;
36     }
37     }
38     }
39     Serial.println(" Finaliza Mapeo—————");
40 }

```

Código A.14: Función para máximo de un vector y función para realizar un barrido automatizado del área de trabajo, entorno: Arduino.

## A.2. MatLab

### A.2.1. Simulación del campo

```

1 function [I] = Gauss(m, sigma)
2
3 [x,y] = meshgrid(-m:m,-m:m);
4 Exp_comp = -(x.^2+y.^2)/(2*sigma*sigma);
5 I = 1 - (Exp_comp / min(min(Exp_comp)));
6 end

```

Código A.15: Campo magnético simulado, entorno: MatLab.

### A.2.2. Ejecución del algoritmo

```

1 clc
2 clear all
3 close all
4

```



```

5 figure(1)
6 s = 101;
7 m = ceil((s - 1)/2);
8 I = Gauss(m,100);
9
10 L = cat(3, I,I,I);
11 imshow(imresize((L),10,'nearest'));

```

Código A.16: Inicialización del campo simulado, entorno: MatLab.

```

1 x = randi(s,1,1);
2 y = randi(s,1,1);
3
4 med = I(x,y);
5 L(x,y) = med;
6 L(x,y,:) = [1;0;0];
7 imshow(imresize((L),10,'nearest'));
8
9 xgrad = -1;
10 ygrad = -1;
11
12 a = 1;
13 err = 1;
14 i = 1;

```

Código A.17: Inicialización del sensor simulado, entorno: MatLab.

```

1 while abs(err) > 0 && i < 50
2     y1 = y + ygrad;
3     med1 = I(x,y1);
4     err = med - med1;
5     med = med1;
6     ygrad = a*sign(err/(y - y1));
7     y = y1;
8     L(x,y,:) = [0;0;1];
9     imshow(imresize((L),5,'nearest'));
10    drawnow
11

```

```

12     [x y]
13
14     x1 = x + xgrad;
15     med1 = I(x1,y);
16     err = med - med1;
17     med = med1;
18     xgrad = a*sign(err/(x - x1));
19     x = x1;
20     L(x,y,:) = [0;0;1];
21     imshow(imresize((L),5,'nearest'));
22     drawnow
23
24     [x y]
25
26     i = i+1
27 end
28
29 L(x,y,:) = [0;1;0];
30 imshow(imresize((L),5,'nearest'));
31 drawnow

```

Código A.18: Algoritmo de posicionamiento en operación por iteraciones, entorno: MatLab.