



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE INVESTIGACIÓN

**DISEÑO, CONSTRUCCIÓN Y CONTROL DE UN PROTOTIPO DE UN ROBOT SCARA
CON CINCO GRADOS DE LIBERTAD**

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN MECATRÓNICA

PRESENTADO POR:

21641090 ENZO FABRIZIO ALTAMIRANO TURCIOS

21751168 IVÁN HUMBERTO HERRERA PALACIOS

ASESOR: ING. ALICIA MARÍA REYES DUKE

CAMPUS SAN PEDRO SULA; JULIO, 2021

DEDICATORIA

A Dios, nuestras familias y seres queridos por todo su apoyo incondicional.

AGRADECIMIENTO

Primeramente, a Dios, por haberme dado vida, por la increíble familia que me otorgó y permitirme haber llegado hasta este momento tan importante en mi formación profesional

A mis padres, los pilares de mi vida, quienes admiro por enseñarme: excelentes valores, la importancia de la familia, luchar por mis metas, nunca darme por vencido y a confiar en mí. Por demostrarme su cariño, apoyo incondicional sin importar nuestras diferencias de opiniones, su inagotable fe en mí y porque en esta vida nunca me ha faltado nada.

A mi hermano por ser mi mejor amigo y estar siempre para mí en momentos difíciles y alegres, conforme hemos ido creciendo y madurando juntos siempre he tenido su apoyo incondicional, consejos y compañía.

Mis abuelos por enseñarme lo que es la verdadera determinación, coraje, esfuerzo, valentía y un corazón de oro. Por demostrarme un amor tan cálido como ningún otro y unas sonrisas tan grandes que siempre llevaré en mi corazón.

A mis seres queridos por amarme, cuidarme, motivarme, apoyarme siempre y permitirme ser parte de sus vidas.

A Bessy Ramos por ser mi mejor amiga, brindarme apoyo emocional en los peores momentos y motivarme a ser mejor.

A mis catedráticos de educación media y superior, por pavimentar mi camino y darme las herramientas para poder llegar hasta aquí.

A mi compañero de investigación Iván Herrera, por permitirme trabajar con él, por su excelente trabajo, responsabilidad, apoyo, su increíble dedicación y exigencia, que terminó afectándome y motivándome a ser cada vez mejor.

-Enzo Fabrizio Altamirano Turcios

Agradezco a Dios por darme la habilidad, determinación y paciencia durante mi carrera universitaria, por la sabiduría e inteligencia en cada examen y proyecto, por su gracia y favor y por tenerme en el hueco de su mano en todo momento.

Agradezco a mis padres y abuelos, a mi madre por sus cuidados y atenciones que facilitaron todas las noches de estudio, y por sus consejos a lo largo de mi vida; a mi padre por su instrucción, consejo y conocimiento compartido que me han asistido hasta el día de hoy; a mis abuelos por su cariño, disposición y cuidados que fueron imprescindibles para mí. Gracias a la ayuda y amor incondicional de cada uno de ustedes me fue posible culminar exitosamente esta etapa. De igual forma, a mis hermanos por su apoyo y asistencia en todo momento.

Agradezco a mis compañeros y seres queridos por acompañarme durante mi carrera universitaria, a Emy Orellana por apoyarme, ayudarme y estar siempre presente para mí; a Enzo Altamirano por ser un excelente compañero de tesis, realmente no imagino una mejor elección para llevar a cabo este gran trabajo.

Finalmente, agradezco a cada uno de los catedráticos que aman lo que hacen y me han enseñado con dedicación. Sin esa pasión por la enseñanza, por lograr nuestro aprendizaje y enriquecer nuestro conocimiento, mi experiencia universitaria no hubiera sido la misma, los aliento a seguir así.

-Iván Humberto Herrera Palacios

Agradecemos a nuestra asesora metodológica, la ingeniera Alicia María Reyes Duke, por su excelente dedicación, motivación, consejos y gran apoyo a lo largo del desarrollo de esta investigación.

ΕΠΙΓΡΑΦΕ

Education is the most powerful weapon which you can use to change the world.

-Nelson Mandela

RESUMEN EJECUTIVO

Los robots tradicionales son desarrollados para operaciones y aplicaciones específicas, siendo limitados por su propia construcción y/o configuración. El SCARA es uno de los robots más importantes y un estándar en la robótica industrial a nivel mundial, su construcción típica se compone de cuatro articulaciones, tres de revoluta y una prismática. Esta hace que tenga facilidad de movimiento horizontal, pero dificultad en movimiento vertical, su área de trabajo es de un círculo casi completo alrededor de su base con pequeños cambios de altura que están limitados por el nivel de su instalación, impidiéndole crecer en su rango de aplicaciones desde el momento que requieran cambios de altura significativos. Una de las formas ideadas, para superar las limitaciones de altura encontradas en un SCARA, es la implementación de un quinto grado de libertad en su base, optando por un diseño que ampliará su área de trabajo de forma vertical, permitiéndole subir y bajar significativamente sobre un SCARA de montaje tradicional, esperando que pueda cumplir con ciertas aplicaciones planteadas en la investigación y que se pueda encontrar un uso en la industria para este. Esta investigación muestra el diseño robótico en diferentes softwares CAD, construcción a través de manufactura aditiva y sustractiva, y control por medio de modelos matemáticos y programación, de un prototipo de robot SCARA con cinco grados de libertad. Los resultados experimentales demostraron que, un quinto grado de libertad implementado en la base de un SCARA es una forma viable de incrementar su área de trabajo vertical, haciendo que se pueda desempeñar en nuevas aplicaciones, que tenga una instalación versátil, económica y facilidad de adaptación a cambios de lugar u aplicación.

Palabras claves – cinco grados de libertad, diseño robótico, prototipo, SCARA, robótica, robots

ABSTRACT

Traditional robots are developed for specific operations and applications, being limited by their own construction and/or configuration. The SCARA is one of the most important robots and a standard in industrial robotics worldwide, its typical construction consists of four joints, three revolute and one prismatic. This makes it easy to move horizontally, but difficult to move vertically, its working area is almost a complete circle around its base with small height changes that are limited by the level of its installation, preventing it from growing in its range of applications as soon as significant height changes are required. One of the devised ways, to overcome the height limitations found in a SCARA, is the implementation of a fifth degree of freedom in its base, opting for a design that will expand its working area vertically, allowing it to go up and down significantly over a traditional mounting SCARA, hoping that it can meet certain applications raised in the research and that a use in the industry can be found for it. This research shows the robotic design in different CAD software, construction through additive and subtractive manufacturing, and control by means of mathematical models and programming, of a SCARA robot prototype with five degrees of freedom. The experimental results demonstrated that, a fifth degree of freedom implemented in the base of a SCARA is a viable way to increase its vertical working area, making it able to perform in new applications, to have a versatile, economical installation and ease of adaptation to changes of location or application.

Keywords – five degrees of freedom, prototype, SCARA, robotic design, robotics, robots

ÍNDICE DE CONTENIDO

I	Introducción	4
II	Planteamiento del problema	6
2.1	Precedentes del problema	6
2.2	Definición del problema.....	8
2.3	Justificación.....	10
2.4	Preguntas de investigación.....	11
2.5	Objetivos.....	11
2.5.1	Objetivo general	11
2.5.2	Objetivos específicos.....	12
III	Marco teórico	13
3.1	Análisis de la situación actual	13
3.1.1	Mundial	13
3.1.2	Continental.....	14
3.1.3	País.....	15
3.2	Robótica industrial	16
3.2.1	Morfología de un robot	17
3.2.2	Tipos de robots industriales en la actualidad.....	21
3.2.3	Industrias y sus aplicaciones	27
3.3	Robot SCARA.....	32
3.3.1	Historia y evolución del SCARA.....	32
3.3.2	Configuración mecánica y área de trabajo	36
3.3.3	Ejemplos de robots SCARA en la industria	37

3.3.4	Aplicaciones comunes	39
3.4	Cinemática en la robótica.....	40
3.4.1	Cinematica directa.....	41
3.4.2	Cinematica inversa	42
3.5	Manufactura aditiva.....	43
3.5.1	Tipos de manufactura aditiva.....	45
3.5.2	Implementación de manufactura aditiva en la robótica.....	48
3.6	Sistema de control.....	50
3.6.1	Microcontroladores	50
3.6.2	Protocolos de comunicación.....	51
3.6.3	Interfaz para el usuario.....	52
IV	Metodología.....	53
4.1	Enfoque	53
4.2	Alcance.....	53
4.3	Variables de investigación.....	53
4.4	Hipótesis	55
4.5	Técnicas e instrumentos aplicados	56
4.6	Metodología de estudio.....	57
4.6.1	Etapa I: Nivel de los sistemas	57
4.6.2	Etapa II: Nivel de los subsistemas.....	59
4.6.3	Etapa III: Realización de las partes	61
4.6.4	Etapa IV: Integración de las partes.....	66
4.6.5	Etapa V: Integración de los subsistemas.....	68

4.6.6	Etapa VI: Integración de los sistemas.....	69
4.7	Cronograma de Actividades.....	70
V	Resultados y análisis.....	71
5.1	Resultados indirectos.....	71
5.1.1	Etapa 3: Realización de las partes.....	71
5.1.2	Etapa 4: Integración de las partes.....	93
5.1.3	Etapa 5: Integración de los subsistemas.....	97
5.1.4	Etapa 6: Integración de los sistemas.....	108
5.2	Resultados directos.....	111
5.2.1	Incremento de área de trabajo.....	111
5.2.2	Desempeño en aplicaciones donde previamente no hubiera podido.....	113
5.3	Costos del prototipo final.....	124
VI	Conclusiones.....	126
VII	Recomendaciones.....	127
VIII	Aplicabilidad.....	128

ÍNDICE DE ILUSTRACIONES

Ilustración 1-Área de trabajo tradicional de un robot SCARA	9
Ilustración 2-Propuesta de SCARA con 5 GDL.....	10
Ilustración 3-Instalaciones anuales de robots industriales 2009-2019 ^a nivel mundial.....	14
Ilustración 4-Instalaciones anuales por continente de robots industriales 2010-2019.....	15
Ilustración 5-Instalaciones anuales de robots industriales en el año 2019.....	16
Ilustración 6-Tipos de articulaciones	19
Ilustración 7-Banda síncrona	21
Ilustración 8-Configuración mecánica y área de trabajo de un robot cartesiano	22
Ilustración 9-Configuración mecánica y área de trabajo de un robot cilíndrico.....	23
Ilustración 10-Configuración mecánica y área de trabajo de un robot esférico.....	24
Ilustración 11-Configuración mecánica y área de trabajo de un robot articulado.....	25
Ilustración 12-Configuración mecánica y ejemplo de un robot paralelo	26
Ilustración 13-Instalaciones de robots por industrias a nivel mundial 2017-2019	28
Ilustración 14-Primer prototipo del robot SCARA	33
Ilustración 15-Segundo prototipo del robot SCARA	34
Ilustración 16-Primer robot SCARA vendido internacionalmente.....	35
Ilustración 17-Primer robot SCARA con accionamiento directo: <i>AdeptOne</i>	35
Ilustración 18-Configuración mecánica tradicional de un robot SCARA	36
Ilustración 19-Robot SCARA modelo SR-3iA.....	37
Ilustración 20-Robot SCARA modelo RS3.....	37
Ilustración 21-Robot SCARA serie IX	38
Ilustración 22-Robot SCARA modelo i4L.....	38

Ilustración 23-Cinematica en robots.....	40
Ilustración 24-Áreas de aplicación donde se está usando MA.....	44
Ilustración 25-Industrias que hacen uso de la MA.....	44
Ilustración 26-SLA	45
Ilustración 27-FDM	46
Ilustración 28-SLS.....	46
Ilustración 29-LOM	47
Ilustración 30-DLP.....	47
Ilustración 31-Mano protésica Raptor Hand elaborada utilizando impresión 3d	48
Ilustración 32-Robot móvil con dos grados de libertad	49
Ilustración 33-Brazo robótico con 6 grados de libertad	49
Ilustración 34-Variables de investigación	54
Ilustración 35-Metodología en "V"	57
Ilustración 36-Sistemas y subsistemas del prototipo.....	58
Ilustración 37-Etapa II: Metodología en "V"	59
Ilustración 38-Etapa III: Metodología en "V"	61
Ilustración 39-Etapa IV: Metodología en "V"	66
Ilustración 40-Etapa V: Metodología en "V"	68
Ilustración 41-Etapa VI: Metodología en "V"	69
Ilustración 42-Cronograma de actividades	70
Ilustración 43-Prueba de estrés Von Mises placa inferior	72
Ilustración 44-Prueba de desplazamiento placa inferior	72
Ilustración 45-Prueba de estrés Von Mises eslabón 1	73

Ilustración 46-Prueba de desplazamiento eslabón 1	74
Ilustración 47-Prueba de estrés Von Mises eslabón 2	75
Ilustración 48-Prueba de desplazamiento eslabón 2	76
Ilustración 49-Prueba de estrés Von Mises eslabón 3	77
Ilustración 50-Prueba de desplazamiento eslabón 3	78
Ilustración 51-Prueba de estrés Von Mises acople	78
Ilustración 52-Prueba de desplazamiento acople	79
Ilustración 53-Curva de torque vs velocidad NEMA 17 HS16-2004S1	81
Ilustración 54-Diagrama de sistemas de coordenadas de SCARA con 5 GDL	82
Ilustración 55-Diagrama geométrico de cinemática inversa	85
Ilustración 56-Diagrama de cinemática directa MATLAB	87
Ilustración 57-Resultados de cinemática directa Matlab posición de inicio	88
Ilustración 58-Resultados de cinemática inversa Matlab posición de inicio	89
Ilustración 59-Diagrama de cinemática inversa MATLAB	90
Ilustración 60-Diagrama de cinemática directa MATLAB ejemplo 2	91
Ilustración 61-Resultados de cinemática directa Matlab ejemplo 2	92
Ilustración 62-Resultados de cinemática inversa Matlab ejemplo 2	92
Ilustración 63-Diagrama de cinemática inversa MATLAB ejemplo 2	93
Ilustración 64-Renderizado de estructura completa en SolidWorks	94
Ilustración 65-Diagrama de flujo programación principal	96
Ilustración 66-Sistema mecánico vista lateral	98
Ilustración 67-Sistema mecánico vista superior	99
Ilustración 68-Esquemático completo de sistema electrónico	100

Ilustración 69-Diseño de la placa electrónica.....	101
Ilustración 70-Placa electrónica manufacturada	102
Ilustración 71-Placa electrónica soldada.....	103
Ilustración 72-Interfaz gráfica.....	104
Ilustración 73-GUI posición de inicio.....	105
Ilustración 74-Recibimiento exitoso de posición de inicio.....	105
Ilustración 75-GUI cinemática directa	106
Ilustración 76-Recibimiento exitoso de coordenadas de cinemática directa	107
Ilustración 77-GUI cinemática inversa.....	107
Ilustración 78-Recibimiento exitoso de articulaciones de cinemática inversa.....	108
Ilustración 79-Robot SCARA con 5 GDL	109
Ilustración 80-Prueba punto a punto posición uno.....	110
Ilustración 81-Prueba punto a punto posición dos	111
Ilustración 82-Área de trabajo de robot SCARA con 5 GDL	112
Ilustración 83-Área de trabajo de robot Adept Cobra 450 de Omron	113
Ilustración 84-Empacado en cajas con altura posición uno	114
Ilustración 85-Empacado en cajas con altura posición dos.....	115
Ilustración 86-Simulación en Ace empacado en cajas con altura	116
Ilustración 87- <i>Pick and Place</i> en diferentes alturas posición uno	117
Ilustración 88- <i>Pick and Place</i> en diferentes alturas posición intermedia	118
Ilustración 89- <i>Pick and Place</i> en diferentes alturas posición dos.....	119
Ilustración 90-Simulación en Ace <i>Pick and Place</i> en diferentes alturas	120
Ilustración 91-Espacio de trabajo bajo posición uno	121

Ilustración 92-Espacio de trabajo bajo posición dos.....	122
Ilustración 93-Espacio de trabajo alto posición uno	123
Ilustración 94-Espacio de trabajo alto posición dos.....	124

ÍNDICE DE TABLAS

Tabla 1-Sistemas de transmisión más frecuentes.....	20
Tabla 2-Tabla de parámetros D-H.....	42
Tabla 3-Comparación de materiales PLA y ABS	61
Tabla 4-Parámetros D-H para SCARA con 5 GDL	83
Tabla 5-Desglose de costos del prototipo final	125

ÍNDICE DE FORMULAS

Fórmula 1-Transformaciones D-H	42
Fórmula 2-Matriz de transformación homogénea D-H	42
Fórmula 3-Ecuación de peso	62
Fórmula 4-Diametro de paso de la polea.....	62
Fórmula 5-Relacion de transmisión	63
Fórmula 6-Relacion de velocidad	63
Fórmula 7-Torque de salida	63
Fórmula 8-Potencia del sistema	63
Fórmula 9-Largo de banda.....	63
Fórmula 10-Torque producido por collar de empuje	64
Fórmula 11-Torque necesario para levantar la carga.....	64
Fórmula 12-Torque necesario para bajar la carga.....	64

ÍNDICE DE ANEXOS

Anexo 1-Prueba de transmisión banda polea.....	137
Anexo 2-Prueba de transmisión tornillo sin fin.....	137
Anexo 3-Esquemático de subsistema de potencia	138
Anexo 4-Esquemático de subsistema de accionamiento	138
Anexo 5-Esquemático circuito de control del subsistema de accionamiento	139
Anexo 6-Prueba de corriente de motores del sistema de accionamiento.....	139
Anexo 7-Prueba de voltaje de motores del sistema de accionamiento.....	140
Anexo 8-Programacion de RPM de motores del sistema de accionamiento	140
Anexo 9-Conexión de motores para el sistema de accionamiento	142
Anexo 10-Esquemático de subsistema Sensorial.....	142
Anexo 11-Programacion final de carrera y motores en posición de inicio.....	143
Anexo 12-Prueba de sensor de final de carrera siendo activado y deteniendo un motor.....	144
Anexo 13-Integración de cinemática directa en programación principal parte 1.....	144
Anexo 14-Integración de cinemática directa en programación principal parte 2.....	145
Anexo 15-Integración de cinemática inversa en programación principal t1t2	145
Anexo 16-Integración de cinemática inversa en programación principal L1 y L5 subida.....	146
Anexo 17-Integración de cinemática inversa en programación principal L1 y L5 bajada.....	147
Anexo 18-Prueba de <i>GUI</i> accionando un motor y una led por medio del Arduino.....	147
Anexo 19 – Fragmentos de proceso de construcción.....	148
Anexo 20 – Programación de <i>GUI</i> en VisualStudio.....	149
Anexo 21 – Programación final Arduino	154

LISTA DE SIGLAS Y GLOSARIO

ABS: *Acrylonitrile butadiene styrene:* Acrilonitrilo butadieno estireno

CAA: *Committee for Automatic Assembly:* Comité para Ensamblaje Automático

Deep Learning: Aprendizaje Profundo

DLP: *Digital Light Processing:* Procesamiento de luz digital

DOF: *Degree of freedom:* Grado de Libertad

FDM: *Fused deposition modelling:* Modelado de deposición fundida

GUI: *Graphical User Interface:* Interfaz gráfica para el usuario

IFR: *International Federation of Robotics:* Federación Internacional de Robótica

ISO: *International Organization for Standardization:* Asociación Internacional de Estándares

JSPE: *Japan Society of Precision Engineering:* Sociedad Japonesa de Ingeniería de Precisión

LOM: *Laminated object manufacturing:* Fabricación de objetos laminados

PCB: *Printed Circuit Board:* Placa de circuito impresa

PETG: *Polyethylene terephthalate Glycol:* Tereftalato de polietileno Glicol

Pick and Place: Elegir y Colocar.

PID: *Proportional Integral and Derivative:* Proporcional, Integral y Derivativo

PLA: *Polylactic Acid:* Ácido poliláctico

SCARA: *Selective Compliance Assembly Robot Arm:* Brazo Robótico de Montaje de Cumplimiento

Selectivo

SLA: *Stereolithography*: Estereolitografía

SLS: *Selective Laser Sintering*: Sinterización por láser selectiva

I INTRODUCCIÓN

En la década de los 80 el robot *SCARA* se volvió uno de los robots más importantes y un estándar para la robótica industrial mundial. Se realizó un estudio en la universidad de Yamanashi, donde se estimó que en el 2019 el robot *SCARA* conformaría el 30% o más de todos los robots industriales trabajando en el mundo. El autor de dicho estudio cree fuertemente que es de suma importancia discutir las necesidades de investigar y desarrollar acerca del robot *SCARA* (Yamafuji et al, 2019).

La construcción típica de un robot *SCARA* se compone de cuatro articulaciones, tres de revoluta y una prismática, implícitamente esto nos dice que cuenta con 4 grados de libertad. Esta construcción hace que el mismo tenga facilidad de movimiento horizontalmente, pero se le dificulte verticalmente. El área de trabajo de un robot *SCARA* tradicional viene siendo el de un círculo casi completo alrededor de su base con pequeños cambios de altura, que están limitados por el nivel de la instalación a la cual se encuentre el brazo, impidiéndole crecer en su rango de aplicaciones desde el momento en que estas requieran cambios de altura significativos y que solo pueda ser utilizado en un ámbito laboral específico.

Es indiscutible el impacto que ha tenido la robótica en la industria moderna, sin embargo, el ámbito industrial y sus demandas pasan en constante desarrollo, sus procesos cada vez se vuelven más complicados y las aplicaciones para los robots más complejas. Los robots tradicionales son desarrollados para operaciones y aplicaciones específicas, siendo limitados por su propia construcción y/o configuración; un fundamental principio que les impide satisfacer las necesidades de producción flexible de la industria moderna. (Chen et al., 2020).

Con esta investigación, se pretende desarrollar un robot *SCARA* de 5 grados de libertad, añadiendo este último en su base, siendo una articulación helicoidal, esto le permitirá tener un amplio rango de movilidad en direcciones verticales, quitando limitaciones previamente impuestas por su construcción tradicional e incrementando su área de trabajo. Se validará si con este cambio presenta ventajas específicas sobre uno de montaje tradicional.

A continuación, se presentan los capítulos que conforman la presente tesis y una breve

descripción de estos:

En el Capítulo II, se presentarán los antecedentes del problema, las preguntas y objetivos de investigación.

El Capítulo III presentará el análisis de la situación actual y definirá las teorías de sustento.

El Capítulo IV: Abordará la metodología que se usará en esta investigación, definirá el enfoque, el tipo de alcance, las variables de estudio e hipótesis.

El Capítulo V, analizará los resultados obtenidos de las simulaciones y las pruebas realizadas al prototipo de acuerdo con los objetivos planteados.

En el capítulo VI se presentarán las conclusiones a partir de los resultados obtenidos acorde a los objetivos planteados y

Por último, en el capítulo VII se darán recomendaciones en base a las conclusiones obtenidas.

II PLANTEAMIENTO DEL PROBLEMA

En este capítulo se mostrarán los precedentes del problema investigación, como es la importancia que tiene la robótica en la automatización de las industrias modernas, el gran papel que juega el robot SCARA en dichas industrias, y lo importante que es innovar para seguir desarrollando nuevos avances tecnológicos. Se definirá el problema a investigar como la justificación de esta y se van a detallar preguntas de investigación con sus objetivos.

2.1 PRECEDENTES DEL PROBLEMA

La robótica es una disciplina considerablemente vieja, y los robots han dado forma a los avances industriales y a las diferentes revoluciones industriales por muchas décadas (Grau et al., 2017). Los robots son indispensables en el ámbito industrial, desde que en la revolución industrial se implementaron en las diferentes fabricas para liberar a los operadores humanos de tareas peligrosas y seguidamente de incorporarlos a otros tipos de procesos de producción para incrementar la productividad, lo cual demandaba más flexibilidad e inteligencia de ellos (Garcia et al., 2007).

Es por esto por lo que muchas industrias en los últimos años han visto un gran crecimiento en la automatización, que es el uso de varios sistemas de control para operar equipos con menos o mínima intervención humana. Esta tiene muchos campos, siendo uno de ellos la automatización industrial, la cual trata con la automatización de manufactura, control de calidad, y procesos de manejo de materiales (Rojas & Barbieri, 2019).

La robótica va de la mano con la automatización, ya que ambas buscan la mínima intervención humana por medio de tecnología para facilitar un proceso e incrementar productividad. Uno de los robots con más relevancia para la robótica industrial, es el robot SCARA. Fue propuesto por Hiroshi Makino en 1978, debido a su estructura, articulaciones y movilidad, revolucionó el ensamblaje de objetos pequeños. No solo se usa para ensamblaje, sino también para aplicaciones de *Pick and place* y paletización (Angeles et al., 2000), en varios campos como ser la mecánica, electrónica, alimentación, medicina, agricultura, e ingeniería de construcción.

Debido a la creación del *SCARA* Japón se volvió en la década de los 80 el líder mundial en la manufactura de robots y en la producción de electrónica para el consumidor (Gasparetto & Scalera, 2019). El robot *SCARA* se volvió uno de los robots más importantes y un estándar para la robótica industrial mundial. Se realizó un estudio en la universidad de Yamanashi, donde se estimó que en el 2019 el robot *SCARA* conformaría el 30% o más de todos los robots industriales trabajando en el mundo. El autor de dicho estudio cree fuertemente que es de suma importancia discutir las necesidades de investigar y desarrollar acerca del robot *SCARA* (Yamafuji et al., 2019), por lo que a lo largo de los años el robot *SCARA* ha sido un tema de investigación y aplicación muy popular.

En el 2014, el Prof. Akram Hossain y Mr. Md Jubair Hossain programaron un robot *SCARA* para la manufactura de células para ensamblar y producir dispositivos médicos. Ellos eligieron el *SCARA* por su rapidez, precio y durabilidad (Hossain et al., 2014).

En el 2014, Junyi Liu, Fang Jian y Junling Liu desarrollaron un robot *SCARA* con modelaje y simulación. En esta investigación, describieron las especificaciones técnicas y estructurales de laboratorio para un robot *SCARA* de 4 grados de libertad utilizando el algoritmo de Denavit – Hartenberg para la cinemática inversa (Liu et al., 2014).

En el 2014, se realizó una investigación acerca de un diseño económico del robot *SCARA* para aplicaciones industriales. Esto con el objetivo de ofrecer una construcción fácil y precio accesible, pero manteniendo siempre el desempeño comparado a uno industrial (Shariatee et al., 2014).

En el 2015, se desarrolló un robot de 5 grados de libertad para soldadura basado en el *SCARA*. Dicho robot es multifuncional debido a la introducción de una extremidad funcional que consiste en dos grados de libertad, los cuales son mutuamente perpendiculares. De esta manera mejorando su aplicación para el campo de la soldadura donde un *SCARA* tradicional no podría mantener el mismo nivel (Cao et al., 2015).

En el 2015, se realizó una investigación acerca de un robot *SCARA* de 3 ejes con un sujetador universal. Esto con el objetivo de mejorar sus capacidades en el campo de ensamblaje, haciéndolo capaz de sujetar objetos irregulares con cualquier tipo de forma o tamaño. (A et al., 2015).

En el 2017, se realizó una investigación para el modelaje y control por medio de PID para un robot SCARA. Esto con el objetivo de ofrecer una mejor estrategia para el control del robot, ya que los robots SCARA se caracterizan por ser difíciles de modelar y controlar (Şen et al., 2017).

En 2017, se diseñó y fabricó un robot SCARA con 5 grados de libertad, el cual es un brazo robótico montado en un carro autónomo. Las razones de su desarrollo no fueron especificadas (Bhadru et al., 2017).

En el 2018, se realizó un estudio de investigación acerca del robot SCARA para automatización industrial de la marca Delta. Este con el objetivo de comparar dicho robot con uno Cartesiano para llegar a una mejor comprensión de operaciones industriales y las ventajas y desventajas de ambos robots en ellas (Suri et al., 2018).

En el 2020, se realizó un estudio de investigación para implementar un desarrollo del robot SCARA por medio del método modular. Este con el objetivo de utilizar sistemas robóticos reconfigurables para ampliar la capacidad de utilizar el robot en más ambientes, ya que el robot tradicional tiene una configuración fija la cual lo hace útil solamente en ambientes laborales específicos (Chen et al., 2020).

2.2 DEFINICIÓN DEL PROBLEMA

La construcción típica de un robot SCARA se compone de cuatro articulaciones, tres de revoluta y una prismática, implícitamente esto nos dice que cuenta con 4 grados de libertad. Las articulaciones de revoluta normalmente se encuentran en la base del robot, la unión de sus dos eslabones y en el final del segundo eslabón, mientras que la articulación prismática se encuentra en el final del segundo eslabón, observar ilustración 1. Esta construcción hace que el mismo pueda moverse de manera fácil en direcciones horizontales, pero que presente dificultad de movimiento en direcciones verticales (Yamafuji et al., 2019).

El área de trabajo de un robot SCARA tradicional, como se puede observar en la ilustración 1, viene siendo el de un círculo casi completo alrededor de su base con pequeños cambios de altura, que están limitados por el nivel de la instalación a la cual se encuentre el brazo, impidiéndole crecer en su rango de aplicaciones desde el momento en que estas requieran

cambios de altura significativos y que solo pueda ser utilizado en un ámbito laboral específico (Chen et al., 2020).

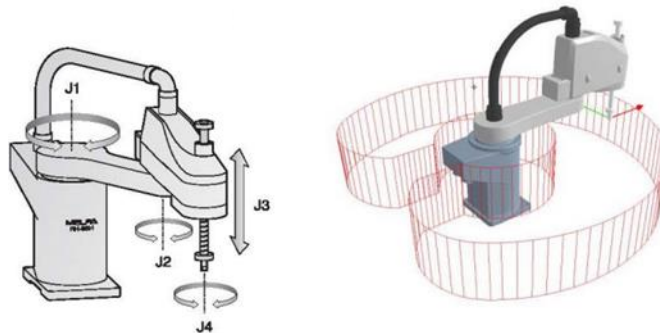


Ilustración 1-Área de trabajo tradicional de un robot SCARA

Fuente: (IFR,2019)

Con esta investigación, se pretende desarrollar un robot SCARA con 5 grados de libertad, añadiendo este último en su base, siendo una articulación helicoidal, validando si esto le permitirá tener las siguientes ventajas sobre las limitaciones impuestas por su construcción tradicional:

- Incremento de su área de trabajo.
- Desempeño en nuevas aplicaciones donde previamente no hubiera podido
- Instalación versátil y económica
- Facilidad de adaptación a cambios de lugar o aplicación dentro de la misma fabrica.

Para las finalidades de esta investigación se probarán las siguientes aplicaciones:

Empacado de objetos en cajas con una altura que un SCARA tradicional no puede alcanzar.

Empacado de objetos debajo del nivel de una banda o mesa y *Pick and Place* en bandas o mesas de diferentes alturas.

Sin embargo, no se descartan más posibles aplicaciones que alguna fabrica le pueda dar.

Además, para poder llevar a cabo el análisis de los resultados, se tendrá en cuenta solamente el alcance que tendrán los movimientos en el eje Z y su nueva área de trabajo, no la velocidad, precisión, ni capacidad de carga contra uno industrial de montaje tradicional, ya que esto es solo

un prototipo. A continuación, se puede observar en la ilustración 2 la propuesta del robot SCARA con 5 GDL.

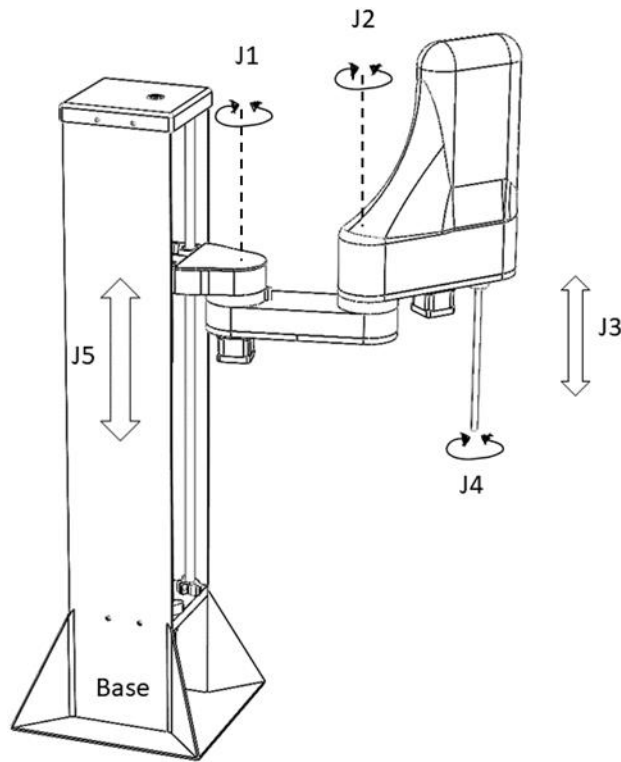


Ilustración 2-Propuesta de SCARA con 5 GDL

Fuente: (Elaboración propia)

2.3 JUSTIFICACIÓN

Por muchas décadas la robótica ha sido indispensable para la industria, cada vez los procesos se llevan a cabo de manera más eficiente gracias a esta, reduciendo errores humanos y cuidando a los mismos de situaciones o actividades de mucho peligro, minimizando accidentes y tasas de mortalidad en fábricas. Es indiscutible el impacto que ha tenido la robótica en la industria moderna, sin embargo, sus demandas pasan en constante desarrollo, sus procesos cada vez se vuelven más complicados y las aplicaciones para los robots más complejas.

La industria moderna demanda producción flexible y los robots tradicionales no cumplen con esas demandas, estos son desarrollados para operaciones y aplicaciones específicas, siendo limitados por su propia construcción y/o configuración, además, la mayoría son diseñados en

tamaños estándar para demandas específicas, lo que significa que, para cada variación de producción, los diseñadores deben desarrollar una nueva máquina, lo cual es tardado y costoso (Chen et al., 2020; Masike et al., 2014). Para superar las limitaciones del *SCARA* tradicional, ampliar su ámbito de aplicación y hacerlo adecuado para las necesidades de producción flexible en la industria moderna, se tiene que desarrollar un nuevo tipo de *SCARA* (Chen et al., 2020). Es por esto que se decidió agregar, a un *SCARA* tradicional, un grado de libertad, optando por un diseño que ampliará su área de trabajo de forma vertical, permitiéndole así tener más alcance y ser más flexible, esperando que pueda cumplir con las aplicaciones anteriormente mencionadas y que se pueda encontrar un uso en la industria.

2.4 PREGUNTAS DE INVESTIGACIÓN

1. ¿Cuál debe ser la relación de fuerzas entre motores y mecanismos utilizados para el movimiento correcto de las articulaciones?
2. ¿Cuáles son las dimensiones físicas adecuadas para la estructura?
3. ¿Qué componentes son necesarios para el sistema electrónico?
4. ¿Qué parámetros hay que tomar en consideración para el movimiento y control adecuado del brazo robótico?
5. ¿Es factible implementar impresión 3d para la construcción de un prototipo de robot *SCARA* de bajo costo?
6. ¿Cuáles serán los rangos de alcance de la nueva área de trabajo?

2.5 OBJETIVOS

En esta sección se definirán los objetivos generales tanto como los objetivos específicos del proyecto propuesto que ayudarán a medir los límites de la investigación.

2.5.1 Objetivo general

Desarrollar un prototipo de robot *SCARA* con un grado de libertad adicional (en el eje z) para validar si este presenta ventajas sobre las limitaciones de uno de montaje tradicional.

2.5.2 Objetivos específicos

- Calcular las relaciones de fuerzas entre motores y mecanismos de banda polea para las articulaciones del brazo robótico.
- Establecer un diseño de la estructura y circuito electrónico en programas CAD.
- Determinar la resolución de la cinemática para la programación correcta del movimiento del robot.
- Definir una interfaz de control para el usuario a través de un *IDE*.
- Analizar la factibilidad de implementar impresión 3d como método de reducción de costos en la estructura.
- Componer el prototipo a base de los diferentes sistemas planteados.

III MARCO TEÓRICO

Durante la ejecución de este capítulo se pretende abordar tanto conceptos básicos como avanzados los cuales ayudarán a desarrollar de manera correcta la investigación. Se desarrollarán los temas de robótica industrial desde sus inicios y su importancia en la actualidad, la morfología de los robots industriales y sus diferentes tipos, la historia y evolución del robot SCARA, la cinemática en la robótica, el algoritmo de Denavit-Hartenberg como solución de cinemática para los robots, manufactura aditiva en la robótica y algunas herramientas de control y comunicación.

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

En esta sección se presentarán los diferentes mercados internacionales de los robots industriales en los siguientes niveles: mundial, continental, de país. Esto con el objetivo de que el lector comprenda la demanda creciente que han tenido estos en los últimos años, al igual que puedan tener entendimiento de los países que más los desarrollan.

3.1.1 Mundial

Entre el 2005 y 2008, el promedio anual de robots vendidos fue de [115,000] unidades. Luego en el 2009, la crisis económica mundial ocasionó que este número bajara a [60,000] unidades ya que muchas inversiones habían sido pospuestas (*IFR, 2020*).

Desde 2010, la demanda de robots industriales ha aumentado considerablemente debido a la tendencia actual hacia la automatización y continuas innovaciones técnicas (*IFR, 2020*). Esto ha hecho que en la última década, los robots industriales se convirtieran en las herramientas básicas e indispensables para la automatización de la producción industrial (*Olszewski, 2020*).

En el 2018, las ventas de los robots industriales llegaron a un valor de USD16.5 billones, lo cual corresponde a [422,000] unidades vendidas y un aumento anual del 6% en comparación con el año anterior. Ese fue el sexto año consecutivo en valor máximo alcanzado en ventas. En el 2019, las ventas globales bajaron a USD 13.8 billones lo cual correspondía a [373,000] unidades representando una baja del 12% con el año anterior, pero sigue siendo el tercer volumen de ventas más alto jamás registrado. Esta disminución en ventas fue debido al conflicto internacional

entre Estados Unidos y China y los tiempos difíciles que enfrentaron las dos industrias principales de la robótica industrial, las cuales son la industria automotriz y electrónica (IFR, 2020).

A continuación, se muestran los datos que representan las instalaciones anuales de robots industriales entre 2009 y 2019.

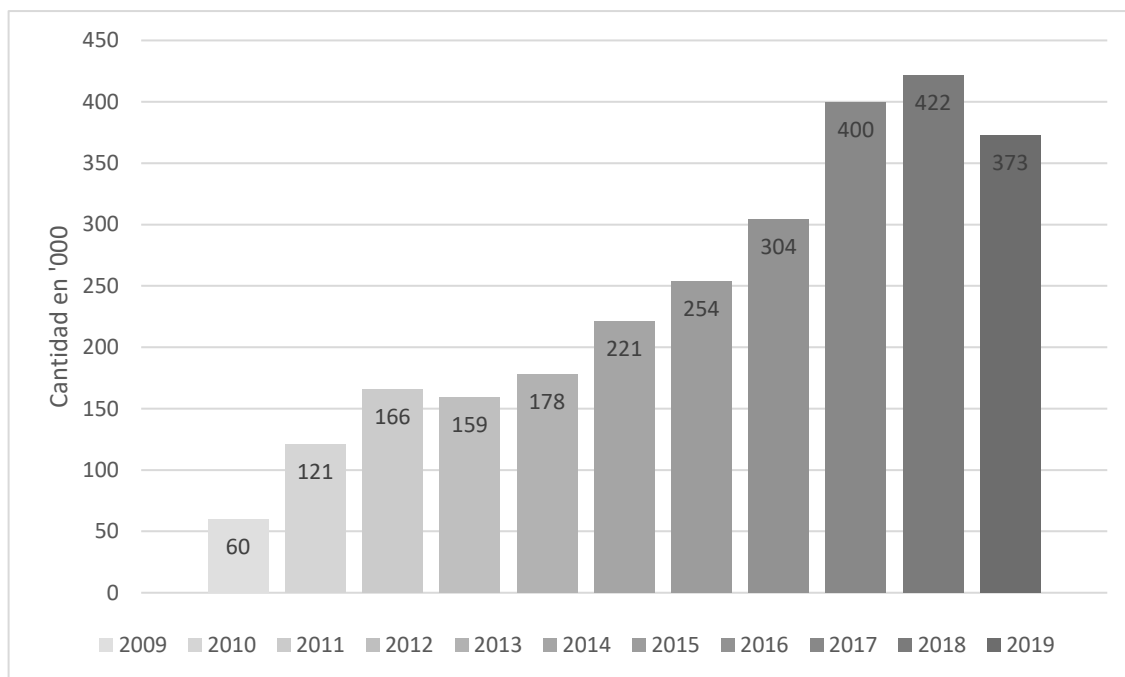


Ilustración 3-Instalaciones anuales de robots industriales 2009-2019^a nivel mundial

Fuente: Elaboración propia con información obtenida de (IFR, 2020)

3.1.2 Continental

A nivel continental, los tres mercados más grandes de robots industriales son: Asia, Europa y América, en ese respectivo orden. Del 2014 – 2018, se vio un periodo de crecimiento consecutivo de instalaciones, en donde Asia experimentó un crecimiento anual de 13%, Europa de 10% y América de 8%. En el 2018 Asia llegó a un pico de 283,000 robots industriales, mientras que Europa y América llegaron a [75,500] y [55,200] respectivamente. En el 2019, las instalaciones anuales bajaron, lo cual marcó el final del crecimiento consecutivo. Asia bajó en un 13% llegando a [245,000] instalaciones, Europa bajó en un 5% llegando a [71,900] instalaciones y América bajó en un 13% llegando a [47,800] instalaciones. Cabe destacar que, en el 2019, aún con las bajas

experimentadas, dos de cada tres robots (66%) implementados en el 2019 fueron instalados en Asia (IFR, 2020).

A continuación, se muestran los datos que representan las instalaciones anuales de robots industriales de los tres mercados principales a nivel continental entre 2010 y 2019.

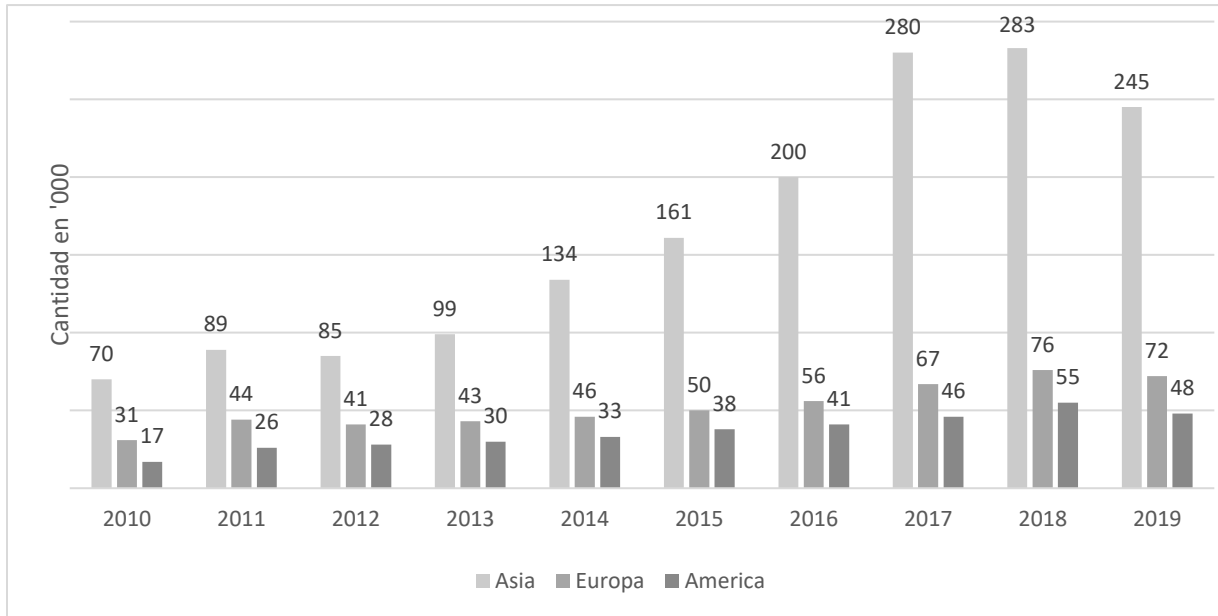


Ilustración 4-Instalaciones anuales por continente de robots industriales 2010-2019

Fuente: Elaboración propia con información obtenida de (IFR, 2020)

3.1.3 País

Existen cinco mercados principales para robots industriales a nivel de país: China, Japón, Estados Unidos, República de Corea y Alemania. Estos países cuentan con el 74% de las instalaciones globales de robots industriales (Olszewski, 2020).

China ha sido el mercado más grande del mundo desde 2013, representando el 38% de todas las instalaciones en 2017 y 2018. En el 2019, se instalaron [140,492] unidades, esto es un 9% menos que en el 2018 pero es más que el número combinado de las instalaciones en Europa y América. Japón es el segundo país en la jerarquía, en el 2019 se vio una baja del 10% representando [49,908] unidades, antes de eso había visto un crecimiento anual del 11% desde el 2014. El tercer país en la jerarquía es Estados Unidos, quien vio un crecimiento anual de ocho años

consecutivos y un máximo de [40,373] unidades instaladas en el 2018, presentó una baja del 17% en 2019 llegando a [33,3339] unidades. El cuarto país en la lista es la Republica de Corea, que ha visto un decremento consecutivo desde 2016 cuando llegó a su máximo de [41,373] unidades, en el 2019 llegó a [27,873] unidades, representando una reducción del 26% en comparación a su pico. Alemania representa el quinto mercado más grande, en el 2019 vio un decremento de 23% llegando a [20,473] unidades, desde el 2014-2019 no ha tenido ningún crecimiento anual en instalaciones (IFR, 2020).

A continuación, se muestran los datos que representan los 15 países con más instalaciones de robots industriales en el año 2019.

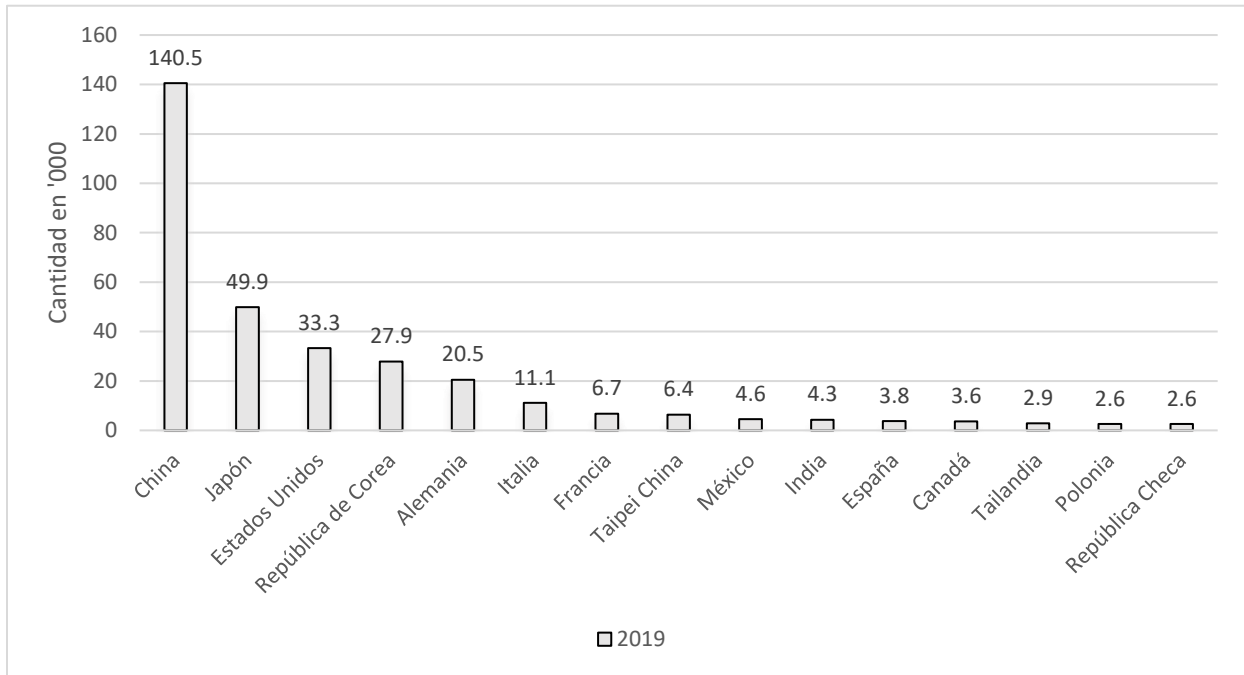


Ilustración 5-Instalaciones anuales de robots industriales en el año 2019

Fuente: Elaboración propia con información obtenida de (IFR, 2020)

3.2 ROBÓTICA INDUSTRIAL

La robótica industrial se remonta al año 1959 en el cual el ingeniero George Devol y Joseph Engelberger desarrollaron el primer robot industrial inspirados por la ciencia ficción. En 1961 se instaló el primer robot industrial en la planta *General Motors Ternstedt* en *Trenton, New Jersey*. (Grau, et al., 2017). Zamalloa et al., (2017) proponen una categorización crónica de los robots

industriales en cuatro generaciones: La primera generación data de 1950 a 1967, estos robots eran básicamente máquinas con programación simple de punto a punto, además, carecían de información sobre su ambiente, la segunda generación (1968-1977) ya contaba con sensores lo que hacía que tuvieran más conocimiento de sus alrededores y de sus sistemas, además de aprender por demostración, en la tercera generación (1978-1999) se agregaron controles dedicados (computadoras), lenguajes de programación para control de robots, volviéndolos reprogramables y contando con visión artificial parcial, por último, la cuarta generación (2000-actualidad) son robots que se caracterizan por tener funciones inteligentes de alto nivel, como la capacidad de realizar cálculos avanzados, razonamiento lógico, *deep learning*, sensores más avanzados que le ayudan a analizar información más compleja, además, que es en esta generación donde son introducidos los robots colaborativos.

En su mayoría, los robots industriales son manipuladores, la *IFR* define a los robots industriales en base a la definición que les da la *ISO*: manipulador multipropósito reprogramable, controlado automáticamente, programable en tres o más ejes, que puede ser fijo o móvil para su uso en aplicaciones de automatización industrial (*ISO*, 2012), esto nos indica tres cosas, la primera es que estos deben de ser reprogramables, la segunda es que sean multifuncionales y manipuladores, por último, que por lo menos tengan tres grados de libertad.

En la actualidad, según de la *IFR* hasta el año 2019, existen [2,722,077] robots operativos, estos son divididos en tres áreas, Asia, Europa y Las Américas, los cuales poseen respectivamente [1,688,000], [580,000] y [389,000] unidades operativas. La *IFR* afirma que la crisis económica global asociada a la pandemia COVID-19 va a dar forma a las ventas, a corto plazo se debe de esperar una contracción importante, a medio plazo esta crisis será un impulsor de la digitalización que creará oportunidades para la industria de la robótica, al largo plazo las perspectivas siguen siendo muy buenas.

3.2.1 Morfología de un robot

Un robot está formado por varias partes, en este subtema se definirán algunos conceptos claves para la investigación tomando en cuenta que se hablará específicamente de robots industriales que por lo general simulan brazos humanos.

Barrientos et al., (2007) dividen las partes de un robot en: "Estructura mecánica, transmisiones, sistema de accionamiento, sistema sensorial, sistema de potencia y control, y elementos terminales" (p. 31), mientras que Sánchez (2021) las divide en "estructura mecánica, transmisiones, sistemas de accionamiento, sistema sensorial, sistema de control y elementos terminales" (p.25), son criterios bastante parecidos pudiendo llegar a un consenso de las seis partes que componen a un robot, tomando la primera como la más acertada.

3.2.1.1 *Grados de libertad*

Saha (2010) define grado de libertad (GDL o *DOF*) de un sistema mecánico como "el número de coordenadas independientes o coordenadas mínimas necesarias para describir perfectamente su posición o configuración" (p. 80), sin embargo, Barrientos et al. (2007) lo explica de forma diferente como "cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior" (p.32), las dos definiciones se complementan y ayudan entre sí, generalmente en los brazos robóticos es fácil de averiguar cuantos grados de libertad poseen, ya que coinciden con el número de articulaciones que tienen, claro esto excluyendo a robots con cadena cinemática cerrada, como el Delta que es un tipo de robot paralelo donde se requiere más atención para calcular los GDL. Un brazo robótico que cuente con seis GDL tiene total libertad de movimientos, tres GDL para definir la posición y los otros tres para definir la orientación (Sánchez, 2021).

Una cadena cinemática, se podría definir como una serie de eslabones unidos por articulaciones. Una abierta significa que se puede llegar desde cualquier eslabón a cualquier otro mediante solamente un camino, en caso de al menos dos caminos, se trata de una cerrada (Barrientos et al., 2007).

3.2.1.2 *Articulaciones y eslabones*

Como se dijo anteriormente, un manipulador industrial está compuesto de varias partes, dentro de la estructura mecánica se encuentran los eslabones, que se pueden definir como cuerpos individuales estructuralmente rígidos, sobreentendiendo que la distancia entre sus dos

extremos no cambia mientras este se mueve. Los eslabones se conectan entre si mediante juntas, pares o articulaciones cinemáticas, una articulación conecta dos eslabones y proporciona restricciones para el movimiento relativo entre ellos, estas restricciones son en forma de concepto que especifican como estos se mueven respecto al otro (Saha, 2010).

Teniendo claro que una articulación es la unión de dos eslabones, a continuación, se presentan los diferentes tipos de articulaciones que se pueden encontrar en un robot industrial:

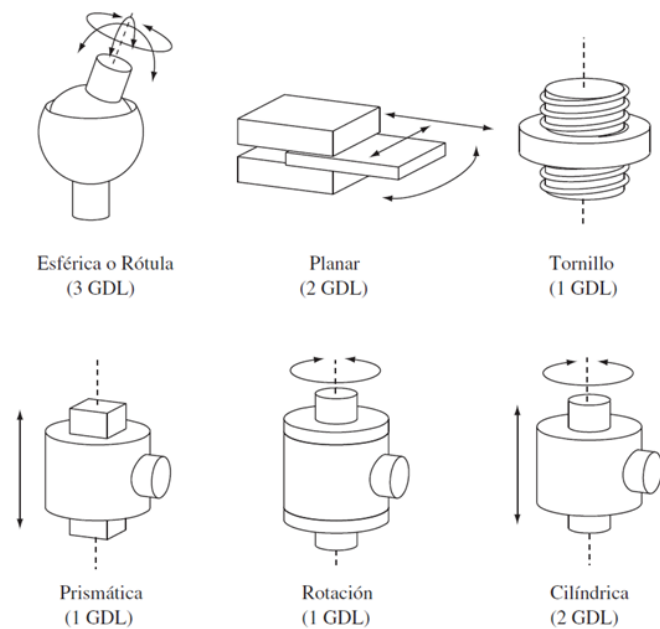


Ilustración 6-Tipos de articulaciones

Fuente: (Barrientos et al., 2007)

En la ilustración 6 se puede observar los diferentes movimientos que puede hacer cada articulación con sus respectivos grados de libertad. La articulación esférica (S) o rótula permite que un eslabón gire libremente en todas las orientaciones mostradas por las flechas en la imagen. La articulación planar (L) permite movimientos de traslación en 2 ejes y uno de rotación. La articulación de tornillo (H) o también llamada helicoidal, permite que los eslabones giren alrededor de un eje y se trasladen al mismo tiempo. La articulación prismática (P) permite que dos eslabones se deslicen entre si a lo largo de su eje. La articulación de rotación (R), revoluta o rotacional, permite que dos eslabones giren con respecto al otro, solo que gire, a diferencia de la

articulación cilíndrica (C) la cual permite el giro y traslado a lo largo del eje. Generalmente un robot industrial es compuesto de solo dos tipos de articulación, la prismática y la de rotación (Sánchez, 2021).

3.2.1.3 Transmisiones

En un robot se necesitan elementos que transmitan el movimiento de los actuadores a los eslabones, un actuador podría ser un motor eléctrico junto con uno de estos elementos anteriormente mencionados (Saha, 2010).

Barrientos et al. (2007) clasifican las transmisiones más frecuentes en base al tipo de movimiento posible en la entrada y salida, tomando en cuenta que las transmisiones pueden ser utilizadas para convertir movimiento circular en lineal o viceversa. A continuación, se presenta una tabla clasificada de las mismas, además, de ventajas e inconvenientes con estas:

Entrada-Salida	Denominación	Ventajas	Inconvenientes
Circular-Circular	Engranaje Correa dentada Cadena Paralelogramo Cable	Pares altos Distancia grande Distancia grande ----- -----	Holguras ---- Ruido Giro limitado Deformabilidad
Circular-Lineal	Tornillo sinfín Cremallera	Poca Holgura Holgura media	Rozamiento Rozamiento
Lineal-Circular	Paral. Articulado Cremallera	----- Holgura media	Control difícil Rozamiento

Tabla 1-Sistemas de transmisión más frecuentes

Fuente: Elaboración propia con información obtenida de (Barrientos, et al., 2007)

En esta investigación se hará uso de tornillo sinfín y correa dentada o banda síncrona, estos se utilizan bastante en la robótica, especialmente la correa dentada, que se puede observar en la ilustración 7; se puede jugar con el tamaño y números de dientes de las poleas para generar más torque y se puede usar en distancias grandes, su único problema es que tiene una vida útil breve, ya que para producir un buen agarre dependen de la tensión.

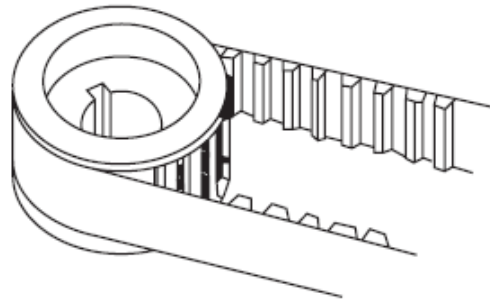


Ilustración 7-Banda síncrona

Fuente: (Saha, 2010)

3.2.2 Tipos de robots industriales en la actualidad

Saha (2010) menciona que, en términos generales, los robots se clasifican en tres grandes familias que son: industriales, no industriales y para usos especiales. Anteriormente solo se definió lo que eran los robots industriales, ya que es el tema principal de la investigación y se hará énfasis en ellos.

Clasificar los robots industriales puede ser un tema difícil, ya que existen muchas maneras de hacerlo atendiendo a diferentes criterios o características, algunas de estas serán dependientes de su propia esencia, aplicaciones o tareas que podrían realizar. Barrientos et al., (2007) mencionan que se puede clasificar los robots industriales según su generación, al área de aplicación, al tipo de actuadores que posee, al número de ejes, a la configuración y al tipo de control. Saha (2010) considera que se pueden clasificar los robots industriales según su aplicación, sistema de coordenadas, sistema de potencia y método de programación.

Si fuéramos a enfocarnos en todas las diferentes maneras antes presentadas, probablemente se extendería demasiado y se volvería muy complicado. El propósito de esta sección no es indagar profundamente en todas las clasificaciones existentes, sino enfocarse en la que mayor impacto tenga en la industria moderna. Es por esta razón que Sánchez (2021) ofrece solo una clasificación, siendo orientada a las configuraciones más frecuentes para realizar funciones industriales.

Sánchez (2021) mencionaba que dichas configuraciones se obtienen dependiendo de la combinación de articulaciones. Esto hace énfasis a la clasificación orientada “a la configuración” mencionada por Barrientos et al., en donde definía que la configuración de un robot queda determinada por el tipo de movimientos permitidos entre 2 eslabones consecutivos de la cadena cinemática, y también a la clasificación de la *IFR* en donde clasifica solamente basado en la estructura mecánica en donde reiteran que los ejes deberían de ser la característica principal.

Tomando en consideración estos criterios, para los propósitos de esta investigación los robots industriales se clasificarán en: Cartesiano, Cilindrico, Esferico, Angular o Articular, Paralelo y *SCARA*, ya que sus configuraciones son las más frecuentes encontradas en la industria.

3.2.2.1 Robot cartesiano

Entre las primeras generaciones de robots industriales utilizados en diversas industrias se encuentran los cartesianos (Luncanu & Stan, 2019).

Según su configuración, el robot cartesiano está conformado por tres articulaciones de tipo prismáticas, y sus ejes están correlacionados a un sistema de coordenadas cartesiano (*IFR*, 2020). Esto significa que solamente puede moverse de manera lineal en las coordenadas X, Y y Z, teniendo un área de trabajo en forma de caja o prisma rectangular (Saha, 2010). En la ilustración 8, se puede observar su configuración mecánica y su área de trabajo.

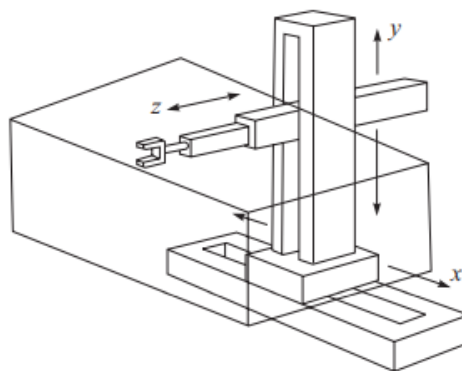


Ilustración 8-Configuración mecánica y área de trabajo de un robot cartesiano

Fuente: (Saha, 2010)

Algunas de las grandes ventajas que ofrece este robot debido a su sistema de posición son su alta flexibilidad, capacidad de posición rápida, y tiempo reducido de la programación de su trayectoria (Luncanu & Stan, 2019).

Los robots industriales con este tipo de cadena cinemática lineal tienen un rango extenso de uso en las siguientes aplicaciones: sistemas de manufactura flexible, donde son utilizados como puertos de trabajo para los sistemas de suministro de máquinas herramientas, sistemas de montaje automático, sistemas de soldadura automático y transporte y almacenamiento automático de piezas, herramientas y dispositivos. (Luncanu & Stan, 2019). Otras aplicaciones para este robot son: manejo de materiales, manejo para el moldeado de plásticos y reabastecimiento de máquinas (IFR, 2020).

3.2.2.2 Robot cilíndrico

Según su configuración, el robot cilíndrico está conformado por una articulación de revoluta y dos prismáticas. Esta configuración es similar a la del robot cartesiano, pero con la diferencia que la articulación prismática de la base se reemplaza con una de revoluta con su eje girado 90 grados con respecto al eje z, esto permite que sus ejes formen un sistema de coordenadas cilíndrico. Su brazo se mueve por medio de teta, Y y Z, es decir, tiene una rotación de base, una elevación y un alcance, haciendo que su área de trabajo sea en forma de cilindro (Saha, 2010). En la ilustración 9, se puede observar su configuración mecánica y su área de trabajo.

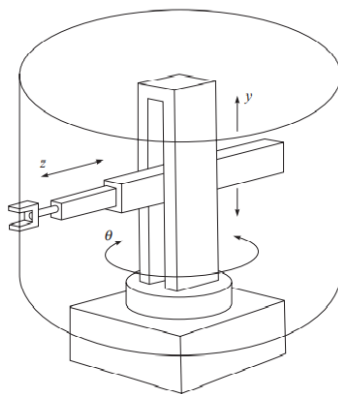


Ilustración 9-Configuración mecánica y área de trabajo de un robot cilíndrico

Fuente: (Saha, 2010)

Normalmente es utilizado para actividades de ensamblaje consideradas en dispositivos de máquina como soldadura y tratamiento por equipos de fundición a presión (Chaturvedi et al., 2020). Esta estructura de brazo mecánico es relativamente pequeña con alta dimensión de precisión, comúnmente utilizado en el manejo de procesos (Jiang, 2018).

3.2.2.3 Robot esférico

Según su configuración, el robot esférico, también denominado polar, está conformado por dos articulaciones de revoluta y una prismática. Esta configuración es similar a la de un robot cilíndrico, en donde la articulación de la base es de revoluta, pero con la diferencia que la segunda articulación prismática encargada de trasladarse verticalmente se cambia por una de revoluta con su eje girado 90 grados con respecto al eje z. Esto permite que sus ejes formen un sistema de coordenadas esférico. La posición del brazo se puede describir convenientemente por medio de las coordenadas esféricas teta, phi y Z, esta configuración hace que su área de trabajo sea la de un volumen esférico. En la ilustración 10, se puede observar su configuración mecánica y su área de trabajo (Saha, 2010).

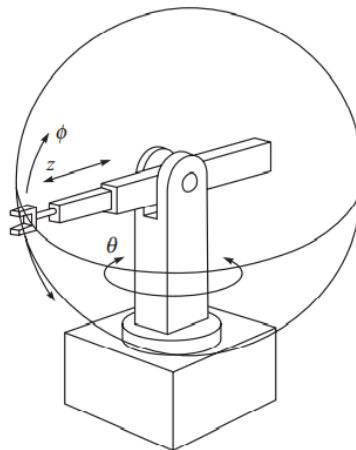


Ilustración 10-Configuración mecánica y área de trabajo de un robot esférico

Fuente: (Saha, 2010)

Este tipo de robot no se encuentra tan presente en el mercado actual, pero sin él y su tecnología esférica, no se hubiese creado el robot *Unimate*, el cual fue el primer robot industrial

creado en 1961. Aunque solo tenían 3 ejes, los sistemas esféricos robóticos como el del *Unimate*, fueron capaces de acelerar lo suficiente la producción como para que los fabricantes se dieran cuenta de su gran potencial, especialmente en el campo automotriz. Estos robots se pueden utilizar en algunas de las aplicaciones robóticas más básicas como pintura, soldadura y ensamblaje (Gasparetto & Scalera, 2019).

3.2.2.4 Robot articulado

Según su configuración, el robot articulado, también denominado angular, está compuesto por al menos tres articulaciones de revoluta (*IFR*, 2020). Esta configuración viene siendo similar a la del robot esférico, con la diferencia que la tercera articulación, la prismática, se reemplaza por una de revoluta con su eje girado 90 grados con respecto al eje z. Esta configuración hace que su área de trabajo sea esférica, sin embargo, su superficie interna es difícil de determinar. En la ilustración 11 se puede observar su configuración mecánica y área de trabajo (Saha, 2010).

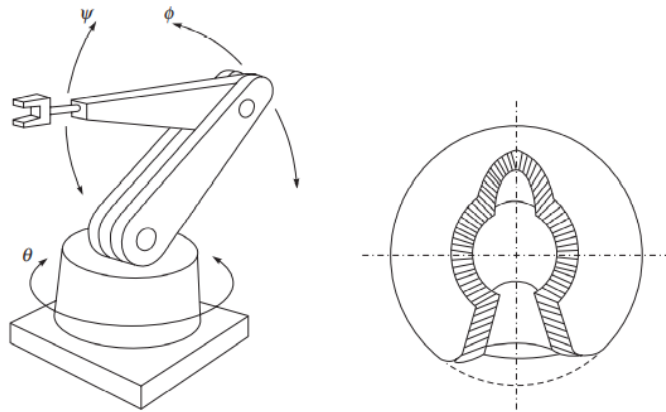


Ilustración 11-Configuración mecánica y área de trabajo de un robot articulado

Fuente: (Saha, 2010)

La configuración de este robot es una de las más comunes, siendo utilizada en el 25% de todos los brazos robóticos en la industria, debido a que se adapta de buena manera a las coordenadas antropomórficas del brazo humano (Rajpar et al., 2021). Barrientos et al., (2007) mencionan que la mayor parte de robots manipuladores actuales, responden a la estructura articulada, representando el 45%.

Su característica principal es la de imitar la estructura básica del brazo humano. Tiene un espacio relativo más grande y un área de trabajo absoluta más grande que otro robot bajo la condición del mismo tamaño. Tiene una gran libertad, apto para casi cualquier trabajo con diferente pista o ángulo (Li et al., 2017).

Es adecuado para la operación de automatización mecánica en muchos campos industriales, como ensamblaje automático, pintura en aerosol, manipulación y soldadura (Li et al., 2017). Otras aplicaciones comunes para este robot son: manipulación para fundición de metales, ensamblaje, pulido, sellado, alimentación de máquinas y rectificado (IFR, 2020).

3.2.2.5 Robot SCARA

Se omitirá el robot SCARA en esta sección, ya que es el objeto principal de esta investigación y se verá más a fondo en la siguiente sección.

3.2.2.6 Robot paralelo

Como lo menciona su nombre, este tipo de robot tiene una estructura tipo paralela, a diferencia de todos los robots previamente mencionados con una estructura tipo serial, también posee una cadena cinemática cerrada, a diferencia de los seriales que poseen una abierta (Saha, 2010). Su composición cinemática obliga a que el control y estudio de su modelaje se realice de manera independiente y diferente a los robots con cadena cinemática abierta. En la ilustración 12 se puede observar la estructura mecánica del robot paralelo junto con un ejemplo de un robot industrial paralelo (Barrientos et al., 2007).



Ilustración 12-Configuración mecánica y ejemplo de un robot paralelo

Fuente: (Barrientos et al., 2007)

Típicamente el número de eslabones es igual al número de grados de libertad, con el propósito de que cada eslabón sea controlado por un actuador diferente. Debido a esto cada actuador puede dividir la carga por igual logrando que estos robots tiendan a tener una alta capacidad de carga (Metodij, 2014). En la actualidad, los robots paralelos son usados popularmente en la industria debido a su flexibilidad, alta rigidez, y capacidad de carga elevada (Hoang et al., 2015).

Metodij (2014) menciona que han sido utilizados en aplicaciones como simuladores de avión, máquinas mineras, máquinas para caminar, manejo de alimentos y piezas electrónicas. Algunas otras aplicaciones para este tipo de robot son: *Pick and place*, ensamblaje y manejo de materiales (IFR, 2020).

3.2.3 Industrias y sus aplicaciones

La necesidad de robots en las diferentes industrias es indiscutible. Ellos permitieron la flexibilidad mejorada y configuración compleja necesaria para variedad de productos. Con ellos, las diferentes empresas del mundo pudieron satisfacer la demanda del consumidor y mantenerlo satisfecho. Estos beneficios son reportados por todas las industrias que han adoptado la automatización por medio de robots (Bader & Rahimifard, 2018).

La mayoría de las aplicaciones para los robots se comparten entre las diferentes industrias, con solo unas pocas siendo específicas para cada una de ellas. Las aplicaciones más comunes según Bekey & Yuh (2008) son:

- Manipulación de materiales
- Soldadura
- Pintura
- Ensamblaje
- *Pick and place*

En la ilustración 13, se pueden observar las instalaciones anuales de robots industriales por industria a nivel mundial desde el año 2017 hasta el año 2019.

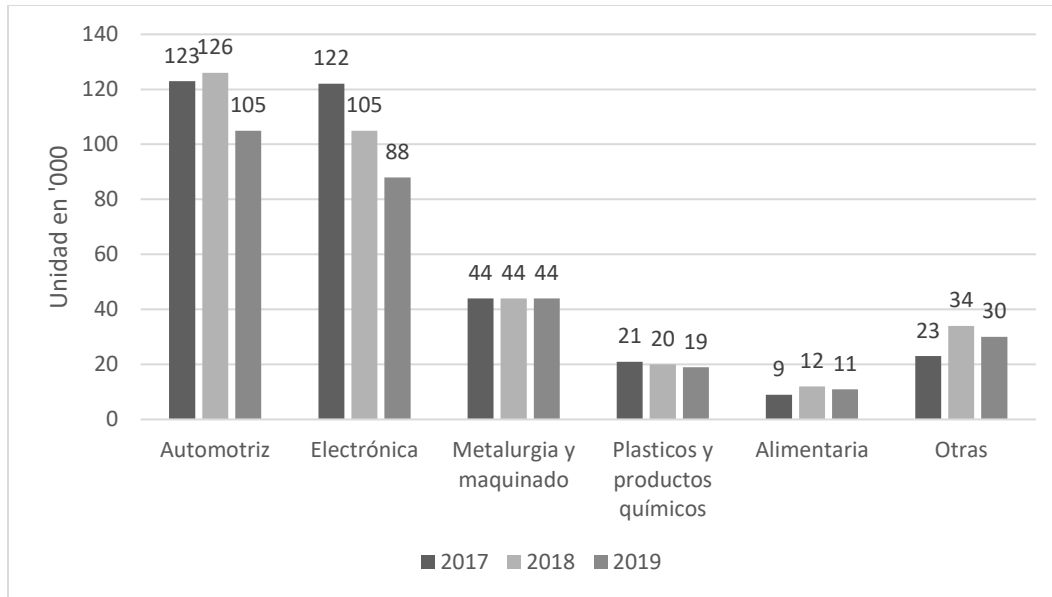


Ilustración 13-Instalaciones de robots por industrias a nivel mundial 2017-2019

Fuente: Elaboración propia con información de (IFR, 2020)

A continuación, se detallarán las industrias en donde los robots tienen mayor impacto y que tipos de aplicaciones desempeñan en ellas.

3.2.3.1 Industria automotriz

Es la más importante para la robótica industrial. La cantidad de robots en esta industria incrementó en un 12% entre los años 2011-2016 (Bader & Rahimifard, 2018). Casi el 28% de todas las instalaciones de robots industriales tienen lugar aquí. Después de dos años muy fuertes en ventas, se llegó a un pico de [125,500] unidades en el 2018. En el 2019, se experimentó una baja del 16% llegando a [105,300] unidades. (IFR, 2020).

Según Bader & Rahimifard (2018) los robots industriales en esta área suelen ser usados en tareas repetitivas como ser:

- Cortado
- Estampado
- Soldadura
- Montaje

También mencionaban que se utilizan en procesos de final de línea como ser: pintura en aerosol y revestimiento.

Karabegović & Husak (2013) mencionan que en general, se utilizan para:

- Recoger y transportar piezas pesadas
- Moldear carcasas
- Pintar las diferentes piezas
- Montaje de diferentes piezas
- Control de procesos.

3.2.3.2 *Industria electrónica*

Según Bader & Rahimifard (2018), la venta de robots a esta industria tuvo un crecimiento anual de 19% entre los años 2011-2016 como resultado de la creciente demanda por dispositivos electrónicos innovativos. Según la *IFR* (2020), las instalaciones de robots industriales en esta area ha incrementado en un 24% anual desde el año 2013 llegando a un pico de [121,000] unidades en el año 2017. Sin embargo, desde el 2018, ha tenido una caída debido a que la demanda global por dispositivos electrónicos ha bajado.

Según Mathia (2010) la automatización robótica se aplica a casi todas las etapas del ciclo de producción de la electrónica. Entre las aplicaciones principales en las *PCB* se encuentran:

- Manejo de materiales y componentes
- Líneas de ensamblaje
- Grabado
- Inspecciones y pruebas
- Manipular y rellenar
- Recubrir con precisión
- Aplicar selladores y adhesivos

Fanuc (2020) una de las grandes empresas en la producción de robots industriales menciona las aplicaciones anteriores y agrega una nueva: apretado de tornillos.

3.2.3.3 *Industria de metalurgia y maquinado*

La industria de metalurgia se refiere a la fabricación de metales. Implica el proceso de refinamiento del metal con el fin de obtener un producto final. En esta industria, los robots se aplican como elemento esencial de nuevas líneas de fabricación automatizadas, con características de flexibilidad. Los robots aumentan la eficiencia de los equipos existentes, en este sentido, la necesidad de nuevas inversiones para maquinaria se reducen (Karabegovic et al., 2012).

Según Karabegovic et al., (2012) las aplicaciones de robots en la industria de metalurgia y maquinado, ya sea en el sistema completo de automatización o en líneas de fabricación automatizadas se resumen en:

- Transporte de material
- Servicios de maquinas
- Operaciones de procesamiento
- Trabajos de instalación
- Control de productos

Según Kuka (2020), otra de las grandes empresas que manufacturan robots industriales, algunas aplicaciones para los robots industriales en esta área son:

- fundición de metales (aluminio, hierro, acero)
- Soldadura de fricción, precisión y magnética
- Taladrado y maquinado
- Forjado a temperaturas arriba de 1000 grados Celsius

3.2.3.4 *Industria de plásticos y productos químicos*

Con alrededor de 20,000 nuevos robots instalados en 2018, la industria del plástico y los productos químicos los utilizan para tareas como:

- Manipulación
- Dispensado

- Ensamblaje
- Procesamiento de materiales
- Inyección de moldes

Además de las aplicaciones previamente mencionadas, según Sadiku et al., (2018), los robots en la industria de productos químicos se utilizan para:

- Tareas de descubrimiento de fármacos en un entorno farmacéutico.
- Inspección de productos de manera repetitiva
- Filtrado
- Procesamiento de especímenes

Se estima que el mercado crezca más de un 10% para el 2021 (IFR, 2020).

3.2.3.5 *Industria alimentaria y de bebidas*

Bader & Rahimifard (2018) mencionan que la implementación de robots en esta área sigue baja en comparación con otras industrias. Esto es debido en su mayoría a ciertos factores contribuyentes:

- Estrictos requisitos de higiene: Para ser utilizados aquí, deben ser construidos de acero inoxidable y pasar por una serie de requisitos ya que normalmente tienen contacto directo con el alimento.
- Características complejas de los productos alimenticios: Cualquier manejo brusco puede afectar propiedades relacionadas a su aspecto, sabor, olor, y textura. La mayoría de los productos en esta área, tienen formas y tamaños diferentes.
- Escasez de operadores calificados: Se debe a la falta de información de los beneficios que los robots pueden tener en esta industria

Según Bader & Rahimifard (2020) las aplicaciones de robots en la industria alimentaria se pueden clasificar en tres categorías principales:

- **Manejo de materiales:** Se refiere al transporte de materiales, en partes o en conjunto, de un lugar a otro. Esta categoría incluye manejo de alimentos sin

protección y empaçado. Son tareas repetitivas y simples, por lo cual se pueden utilizar robots sencillos.

- **Ensamblaje:** se refiere a la unión de dos o más partes o la totalidad de los productos alimenticios, dando como resultado un todo o un subconjunto de un producto. Un ejemplo es producir un plato en capas o un sándwich o formar operaciones tales como dar forma a pan pretzel o glaseado personalizado sobre horneado.
- **Procesos de acabado:** Esto incluye, transferencia de productos terminados a través de empaçado primario, secundario y terciario, seguido de paletizado para almacenamiento, y distribución. Los productos en esta etapa suelen ser más consistentes en tamaño y forma. Un ejemplo incluye el envasado de aceitunas en un tarro y luego paletizar estos tarros.

En las industrias de bebidas, los robots se utilizan principalmente para paletizar botellas y cajas. Los tipos de robots industriales tradicionales pueden embotellar unas 24,000 botellas por hora, pueden transportar alrededor de 30,000 botellas por hora (alrededor de 1500 cajas) y es posible compensar algunas posibles desventajas de tiempo en otras fases de producción. La manipulación de la botella debe realizarse durante un ciclo de tiempo corto, sin una gran aceleración y la programación del robot debe permitir movimientos rápidos y exactos (Karabegović et al., 2008).

3.3 ROBOT SCARA

Debido a que el robot SCARA es el enfoque principal de esta investigación, se hará más énfasis en su información. Se detallará su historia y evolución, su configuración mecánica, área de trabajo, aplicaciones comunes y diferentes tipos que se pueden encontrar en el mercado actual.

3.3.1 Historia y evolución del SCARA

En 1968, se estableció el Comité para Ensamblaje Automático (CAA) de la Sociedad Japonesa de Ingeniería de Precisión (JSPE) y comenzó la investigación y desarrollo acerca de la automatización de ensamblaje. En la década de los setenta, ocurrió un cambio de la producción en masa de pocos productos a la producción limitada de una gran variedad de productos en el

mercado japonés. Este cambio, aceleró la automatización de procesos de ensamblaje de productos y se empezaron a desarrollar tecnologías de producción. Japón se vio en la necesidad de una máquina para ensamblaje automático, pero la tecnología robótica en ese momento no lograba alcanzar la velocidad y precisión necesarias para los trabajos de ensamblaje (Yamafuji, 2019).

En octubre de 1977, el robot de ensamblaje SIGMA fue presentado por d'Auria de la empresa italiana Olivetti Inc., en Tokio, y sorprendió a todos los ingenieros japoneses con su rapidez para lograr el objetivo deseado para los robots de ensamblaje. Sin embargo, el Dr. Hiroshi Makino comenzó a diseñar un robot para el ensamblaje sobre la base de una idea única dos meses después de la presentación de SIGMA. Analizando la estructura cartesiana del robot SIGMA y su rigidez en las direcciones XY, llegó a la conclusión que sería mejor una menor rigidez en las direcciones XY y mayor rigidez en la dirección Z. A finales de diciembre de 1977, el Dr. Makino terminó un dibujo del concepto básico del robot SCARA. El dibujo indicaba un robot de tres ejes, con brazos de doble articulación de enlace horizontal y un eje vertical: Z. Tenía una estructura simple con características para moverse fácilmente en la dirección horizontal, pero con dificultad para moverse en dirección vertical. En marzo de 1978, se lanzó un consorcio de investigación y desarrollo para robots SCARA por la Universidad de Yamanashi (profesor H. Makino y socio profesor K. Yamafuji) y 13 compañías japonesas (Yamafuji, 2019).

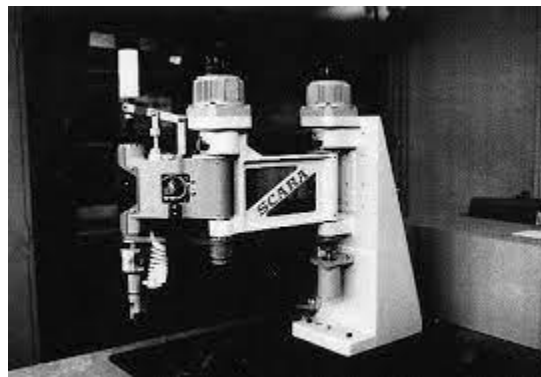


Ilustración 14-Primer prototipo del robot SCARA

Fuente: (Fujitsu,1978)

En Julio de 1978, se manufacturó el primer prototipo del robot *SCARA* por Fujitsu y seguidamente el segundo prototipo por Nitto-Seiko, el cual presentaba mejoras sobre el primero (Yamafuji, 2019).

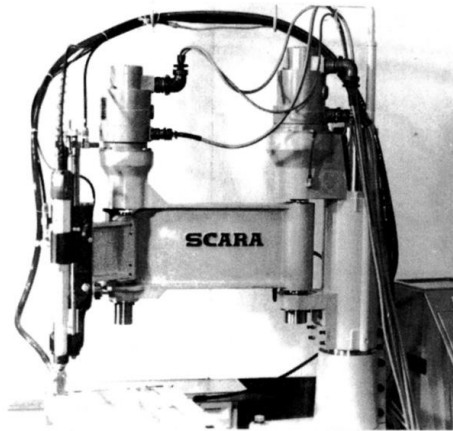


Ilustración 15-Segundo prototipo del robot *SCARA*

Fuente: (Nitto-Seiko,1978)

Ya que el robot presentaba cumplimiento selectivo menor en dirección horizontal comparado a su dirección vertical, era completamente apto para ser empleado en tareas de ensamblaje de objetos pequeños (Gasparetto & Scalera, 2019).

En 1981 terminó el consorcio acordado, en 1982 la compañía reconocida IBM anunció que obtendría de Sankyo Seiki un robot llamado "Sankyo Skilam" y que lo pondría en el mercado como el IBM 7535. Este fue el primer reconocimiento internacional del robot *SCARA*, el cual provocó un nuevo salto hacia adelante (Yamafuji, 2019).



Ilustración 16-Primer robot SCARA vendido internacionalmente.

Fuente: (Yamafuji, 2019)

En 1984, se desarrolló el primer robot SCARA con accionamiento directo, el *AdeptOne*. Este tipo de robot ofrecía precisión más alta y operación más rápida, ya que los motores se conectaban directo a las articulaciones, eliminando la necesidad de mecanismos intermedios (Gasparetto & Scalera, 2019).



Ilustración 17-Primer robot SCARA con accionamiento directo: *AdeptOne*

Fuente: (Gasparetto & Scalera, 2019).

Luego de esto, los robots SCARA se emplearon en las líneas de ensamblaje de varios productos, en especial en el mercado electrónico en Japón. Por esta razón Japón se volvió en los 80 no solo el líder mundial en la manufactura robótica, sino también el líder en la producción de bienes de consumo electrónicos (Gasparetto & Scalera, 2019). En el 2006, el robot SCARA fue introducido al Salón de la fama en la Universidad de Carnegie Mellon (Yamafuji, 2019).

3.3.2 Configuración mecánica y área de trabajo

Un robot *SCARA* según la *IFR*, (2020) es un robot que tiene dos articulaciones de revoluta paralelas para proporcionar cumplimiento en un plano. Según Saha (2010) un robot *SCARA*: “es un tipo cilíndrico cuyo alcance se logra mediante el uso de una articulación de revoluta en lugar de una articulación prismática” (p.28).

Según su configuración, el robot *SCARA* tiene cuatro grados de libertad y está compuesto con una arquitectura común: tres articulaciones de revoluta y una prismática, la última normalmente ubicada en su base o en la parte libre de su cadena cinemática (Angeles et al., 2000).

Esta configuración hace que se pueda mover fácilmente en direcciones horizontales, con poca rigidez en las direcciones *XY*, pero que presente más dificultad para moverse en direcciones verticales con alta rigidez en la dirección *Z* (Yamafuji, 2019).

Por esto, podemos concluir que el área de trabajo de un robot *SCARA* tradicional, sea el de un círculo casi completo alrededor de su base con pequeños cambios de altura, que están limitados por el nivel de la instalación a la cual se encuentre el brazo. En la ilustración 18 se puede observar su configuración mecánica tradicional, y en la ilustración 1 se pueden observar un ejemplo industrial junto con su área de trabajo.



Ilustración 18-Configuración mecánica tradicional de un robot *SCARA*

Fuente:(Cano, 2018)

3.3.3 Ejemplos de robots SCARA en la industria

Los fabricantes principales del robot SCARA a nivel industrial son: Epson, Omron, ABB, Fanuc, Denso, IAI, Staubli, Kawasaki, Yaskawa y Yamaha. Siendo un robot tan popular, existe una gran variedad de modelos con características diferentes como ser: rangos de alcance, dimensiones de la estructura, capacidad de carga y tipo de montaje.

A continuación, se presentarán los diferentes tipos de instalación en los SCARA industriales:



Ilustración 19-Robot SCARA modelo SR-3iA

Fuente: (Fanuc, 2021)

Este es un modelo de la marca FANUC de montaje tradicional.



Ilustración 20-Robot SCARA modelo RS3

Fuente: (EPSON 2021)

Este es un modelo de la marca EPSON de montaje de techo.



Ilustración 21-Robot SCARA serie IX

Fuente: (IAI 2021)

Este es un modelo de la marca IAI de eje invertido y montaje tradicional.



Ilustración 22-Robot SCARA modelo i4L

Fuente: (Omron 2020)

Este es un modelo de la marca Omron de montaje en pared.

3.3.4 Aplicaciones comunes

Desde 1978, el robot SCARA fue desarrollado principalmente para tareas de ensamblaje y se ha utilizado ampliamente en las diferentes industrias, especialmente en la automotriz y electrónica (Chen et al., 2020). Esto es debido a su configuración estructural o arquitectura, que los hace presentar una serie de características y ventajas diferentes.

Según Chen et al., (2020) presentan:

- Buena flexibilidad en las direcciones X y Y
- Fuerte rigidez en la dirección Z

Según Suri et al. (2018) presentan:

- Índices de precisión predefinidos
- Pedestales compactos y esto los hace más eficientes en un espacio limitado

Según Hossain et al., (2014) presentan:

- Velocidad
- Robustez
- Precio accesible
- Durabilidad

Estas características y ventajas los hacen ideales para:

- Tareas de ensamblaje de placas de circuitos impresos y otros componentes electrónicos
- Manipulación de objetos ligeramente pesados
- Tareas de repetición como: *Pick and place*, paletizado, carga y descarga de máquinas
- Aplicaciones que requieran aceleración constante a través del movimiento como el dispensado
- Transporte de piezas
- Soldadura de rapidez y precisión
- Pruebas e inspecciones de piezas electrónicas como chips
- Envasado

- Alimentación flexible

3.4 CINEMÁTICA EN LA ROBÓTICA

“La cinemática es la rama de la Física que se ocupa de estudiar el movimiento de los cuerpos independientemente de las causas que lo producen”(Trenzado, 2015), aplicado a un robot, estudia el movimiento de este, tomando siempre en cuenta un marco de referencia. Barrientos et al., (2007) en el capítulo 4 de su libro, explican que la cinemática en la robótica “se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares”. En esencia lo que interesa de un brazo robótico industrial es mover y poner su efector final en los puntos de interés, detrás de esto está la cinemática del robot, con la que se resuelve los problemas de posicionamiento, refiriéndose a la posición de un punto en el efector final y su orientación con los ángulos y posiciones de las articulaciones, todo esto con respecto a su base. Existen dos tipos de problemas en la cinemática de un robot, la cinemática directa que es donde las posiciones de las articulaciones ya están determinadas y se debe encontrar la configuración del efector final, y la cinemática inversa que es donde la posición del efector final está determinada y se deben encontrar los ángulos de las articulaciones (Saha, 2010).

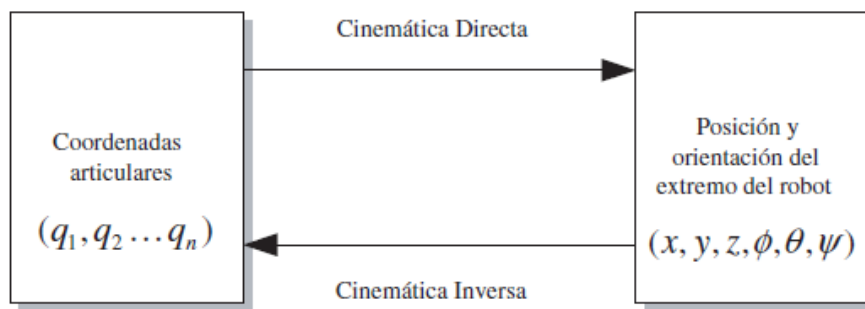


Ilustración 23-Cinemática en robots

Fuente: (Barrientos, 2007)

3.4.1 Cinematica directa

La cinemática directa permite conocer la posición y orientación que tiene el extremo del robot cuando sus variables, las que fijan la posición y orientación de sus articulaciones, toman valores determinados. Para obtener la cinemática directa de un robot, se pueden utilizar dos métodos: Geométrico y mediante matrices de transformación homogéneas, siendo este último el que se utilizará en esta investigación, debido a que la aplicación del método geométrico queda limitado a robots con pocos grados de libertad y el de matrices homogéneas permite de una manera sistemática obtener el modelo cinemático directo para robots de n grados de libertad (Barrientos, 2007).

La resolución del problema cinemático directo mediante las transformaciones de matrices homogéneas se reduce a encontrar una matriz de transformación homogénea T , con esta es posible representar las rotaciones y traslaciones relativas entre los eslabones del robot. Por lo general, la matriz de transformación homogénea, que representa la posición y orientación entre dos eslabones consecutivos se suele denominar matriz $i-1A_i$. La matriz T será función de las coordenadas articulares, esta relaciona la posición y orientación del extremo del robot con respecto a un sistema de referencia fijo situado en la base del mismo (Barrientos, 2007).

3.4.1.1 Algoritmo de Denavit-Hartenberg

Uno de los métodos más utilizados a nivel mundial para la resolución de transformaciones de matrices homogéneas, es el convenio de Denavit-Hartenberg. El modelo Denavit-Hartenberg sigue siendo la convención más utilizada para representar la estructura cinemática del robot. Esta aparece por primera vez en 1955, y consiste en varios pasos con los cuales de forma sistemática se obtiene la resolución del sistema. Una vez establecidos los sistemas de coordenadas asociados a cada eslabón según estos pasos, es posible pasar de uno al siguiente por medio de cuatro transformaciones básicas, las cuales son: Rotación alrededor del eje Z (θ), traslación a lo largo del eje Z (d), traslación a lo largo del eje X (a) y rotación alrededor del eje X (α) (Faria et al., 2019).

Según los grados de libertad del robot, será el número de sistemas, para obtener cada transformación se debe de llenar la siguiente tabla, donde las columnas representan las transformaciones y las filas el número de los sistemas:

i	θ	d	a	α
i+1				

Tabla 2-Tabla de parámetros D-H

Fuente: Elaboración propia con información obtenida de (Saha et al., 2010)

Posteriormente, de cada línea se obtiene una matriz de transformación multiplicando cada columna como se indica en la fórmula 1.

$${}^{i-1}\mathbf{A}_i = \mathbf{Rotz}(\theta) \mathbf{T}(0,0, d_i) \mathbf{T}(a,0,0) \mathbf{Rotx}(\alpha)$$

Fórmula 1-Transformaciones D-H

Fuente: (Barrientos, 2007)

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fórmula 2-Matriz de transformación homogénea D-H

Fuente: (Barrientos, 2007)

El algoritmo D-H es muy útil ya que proporciona una representación mínima garantizada, un método intuitivo para determinar sus parámetros y lo más importante, funciona en álgebra lineal sencilla cuyas matrices son computacionalmente rápido de resolver(Faria et al., 2019).

3.4.2 Cinemática inversa

La cinemática inversa consiste en determinar las variables de las articulaciones con respecto a una orientación y posición específica del efector final. El problema de la cinemática inversa es mucho más complejo que el de la cinemática directa por las siguientes razones:

- No siempre es posible encontrar una solución explícita, debido a que las ecuaciones son no lineales
- Pueden existir múltiples soluciones o soluciones infinitas, esto puede ocurrir en robots cinemáticamente redundantes
- Debido a la arquitectura del robot, es posible que no haya soluciones admisibles

Por lo general, para hacerle frente al problema de la cinemática inversa se utilizan dos métodos: Algebraico y geométrico. En esta investigación se utilizará el geométrico, debido a que es más exacto ya que se conoce la dirección exacta de los ángulos y solo se tomará en cuenta los grados de libertad necesarios para posicionar el extremo del robot (Saha et al., 2010).

Este método geométrico consiste en encontrar un número suficiente de relaciones geométricas en las que intervengan las coordenadas articulares y del extremo del robot. Se hace un diagrama de sus eslabones con una representación a sus tamaños reales, se busca obtener relaciones trigonométricas a través de diferentes identidades para encontrar los valores de las articulaciones relevantes que se toman en cuenta para posicionar el extremo del robot (Barrientos, 2007).

3.5 MANUFACTURA ADITIVA

La impresión 3d, también conocida como manufactura aditiva (MA), consiste en la creación de un objeto añadiendo capas de material, una tras otra, hasta que dicho objeto este completo. La MA está creciendo cada vez más siendo usada en varias áreas, Wohlers (2018), un reporte anual sobre el estado de la MA, nos especifica en la ilustración 24 en que áreas de aplicación se está utilizando la MA. Se puede observar que las aplicaciones que producen piezas reales y funcionales, ya sea directa o indirectamente, representan gran parte del porcentaje, Wohlers espera que estos porcentajes crezcan sustancialmente en los siguientes años cuando más y más industrias vayan adoptando la MA como parte de su arsenal de manufactura (Diegel et al., 2019).

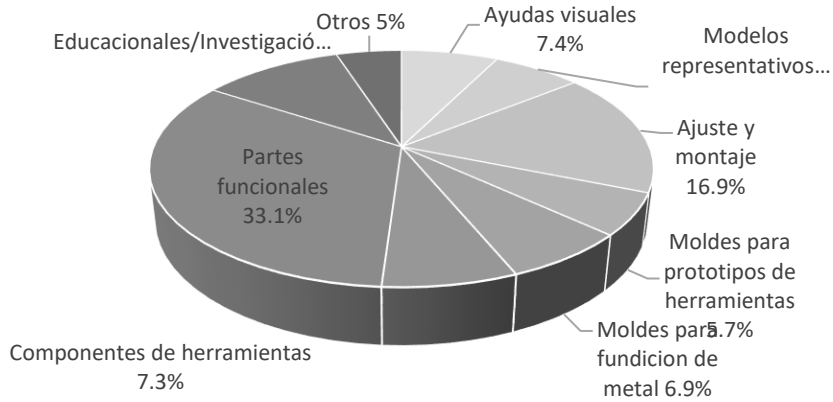


Ilustración 24-Áreas de aplicación donde se está usando MA

Fuente: Elaboración propia con información obtenida de (Wohlert, 2018)

En este mismo reporte nos brindan la ilustración 25 sobre el porcentaje de la MA usada en diferentes industrias, debido a que esta ha mejorado en calidad en los últimos años, varias compañías han empezado a usarla como una vía de producción (Diegel et al., 2019).

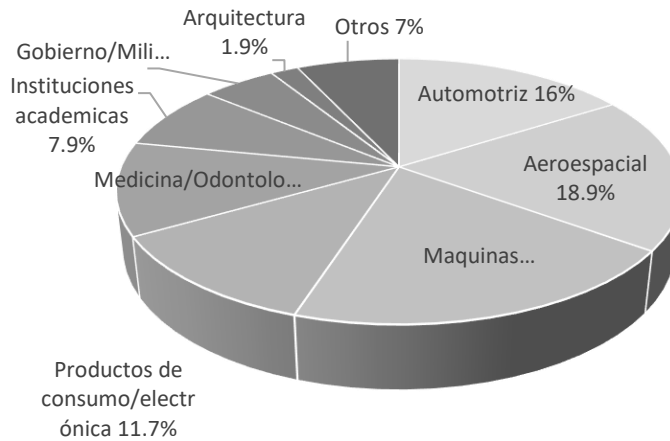


Ilustración 25-Industrias que hacen uso de la MA

Fuente: Elaboración propia con información obtenida de (Wohlert, 2018)

Es importante saber que la MA nunca va a reemplazar por completo a la manufactura tradicional, es una tecnología complementaria que si se usa de la manera correcta puede añadir un gran valor a las compañías (Diegel et al., 2019).

El uso de la MA tiene varias ventajas, Diegel et al., (2019) nos señalan algunas de estas: la complejidad de las piezas que se pueden producir, ensambles instantáneos (impresión de mecanismos complejos y funcionales en una sola pieza), consolidación de piezas (unión de varios componentes a una sola pieza), customización, libertad de diseño, poco peso, manufactura a pedido (piezas pueden ser producidas como y cuando se requieran). No todas las impresoras 3d utilizan la misma tecnología, existen diferentes formas de imprimir en 3d.

3.5.1 Tipos de manufactura aditiva

Jasveer & Jianbin (2018) listan y explican cinco diferentes tipos de tecnología de impresión 3d que más se usan actualmente:

- *Stereolithography (SLA)*: usa un láser ultravioleta y un recipiente con resina para imprimir piezas, una plataforma dentro del recipiente baja desde la superficie (arriba hacia abajo) de la resina para hacer el cambio de capa mientras que el láser marca el diseño en la misma, este proceso es reconocido como la impresión 3d original.

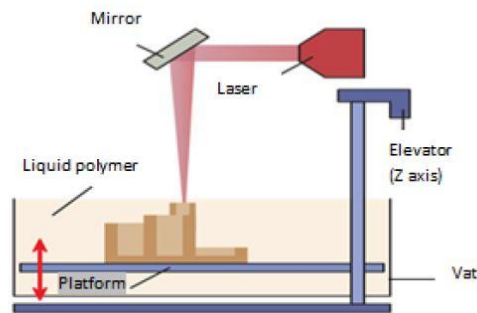


Ilustración 26-SLA

Fuente: (Jasveer & Jianbin, 2018)

- *Fused deposition modelling (FDM)*: calienta y extruye filamento de termoplástico de acuerdo con el modelo 3d suplido, se forma un objeto de abajo hacia arriba capa por capa, es más barato que el *SLA*, sin embargo, es menos preciso y toma más tiempo.

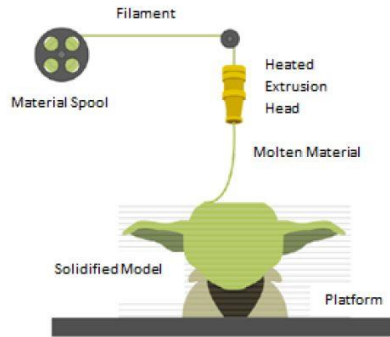


Ilustración 27-FDM

Fuente: (Jasveer & Jianbin, 2018)

- *Selective Laser Sintering (SLS)*: polvo de plástico, cerámica o vidrio es unido a través de un rayo láser que traza la forma capa por capa de acuerdo con el modelo suministrado, el polvo está apretado sobre una plataforma que baja a medida que el láser va marcando, este método es capaz de imprimir figuras geométricas complejas que otros métodos no pueden.

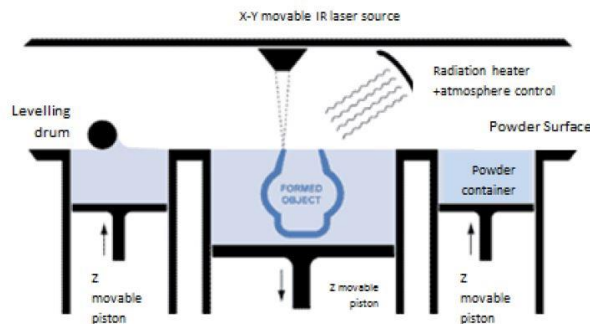


Ilustración 28-SLS

Fuente: (Jasveer & Jianbin, 2018)

- *Laminated Object Manufacturing (LOM)*: se usa láminas de plástico o papel con adhesivo que se ponen a lo largo de la plataforma de construcción, se pasa un rodillo caliente sobre la superficie haciendo que el adhesivo se derrita, un láser corta el material de acuerdo con la pieza dada, la plataforma baja en cada capa realizada (arriba hacia abajo). Es muy útil

para piezas grandes, sin embargo, no puede hacer piezas geoméricamente complejas ni prototipos funcionales.

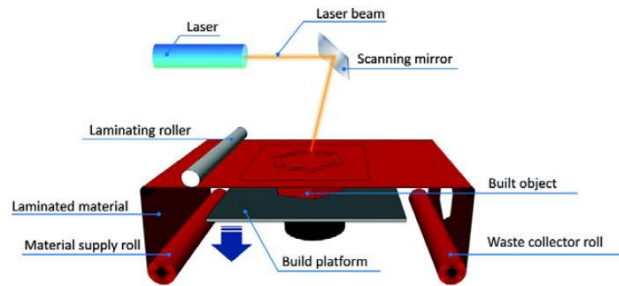


Ilustración 29-LOM

Fuente: (Jasveer & Jianbin, 2018)

- *Digital Light Processing (DLP)*: es muy parecido al *SLA*, la principal diferencia es que en vez de usar un láser usa luz convencional, y resina fotosensible, la luz marca el diseño suministrado mientras la plataforma sube (abajo hacia arriba) en cada cambio de capa. Este método produce piezas con alta precisión y resolución, sin embargo, no se pueden hacer prototipos funcionales, a pesar de ser barato comparado al *SLA* es caro en comparación a los demás métodos.

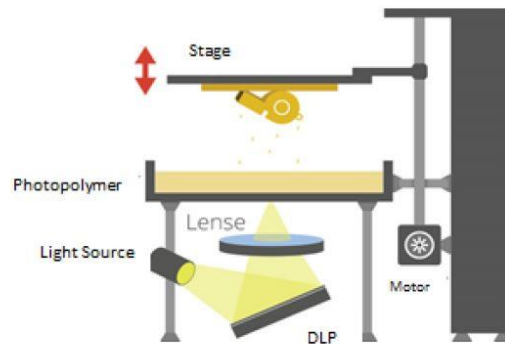


Ilustración 30-DLP

Fuente: (Jasveer & Jianbin, 2018)

La impresión 3d ha revolucionado la manera en que la manufactura es realizada, mejora la fabricación de los diseños, reduce el tiempo de entrega y el costo para nuevos productos. En esta investigación se hará uso del método *FDM* para la impresión del prototipo, debido a su bajo costo,

su capacidad de producir prototipos funcionales, su precisión y accesibilidad. La impresión por *FDM* está limitada a termoplásticos, dentro de las opciones consideradas para esta investigación, *ABS*, *PETG* y *PLA*, se escogió *PLA* debido a su bajo costo, facilidad de impresión y accesibilidad.

3.5.2 Implementación de manufactura aditiva en la robótica

La impresión 3d es una tecnología muy utilizada para prototipos, ya que permite un diseño flexible y funcional con bajo costo, da una gran libertad para la creación de objetos con propiedades estructurales y mecánicas deseadas (Stefanov et al., 2019a).

A continuación, se presentará una breve recopilación de investigaciones donde se utilizó la impresión 3d como método principal de fabricación de prototipo, esto validará, el uso de esta en la investigación.

En 2015, R. Borjas & W. Flores debido al precario sistema de salud público de Honduras y en busca de ayudar a la población hondureña con discapacidades físicas, se auxiliaron de la impresión 3d para el desarrollo de prótesis de mano, exponiendo que con este método se reducirían los precios y el tiempo de creación de estas.



Ilustración 31-Mano protésica Raptor Hand elaborada utilizando impresión 3d

Fuente: (Flores & Borjas, 2015)

En 2019, Stefanov et al., hicieron uso de la impresión 3d como método de prototipado para la construcción de un robot móvil con dos grados de libertad y con un diseño minimalista comparable con los robots bípedos.

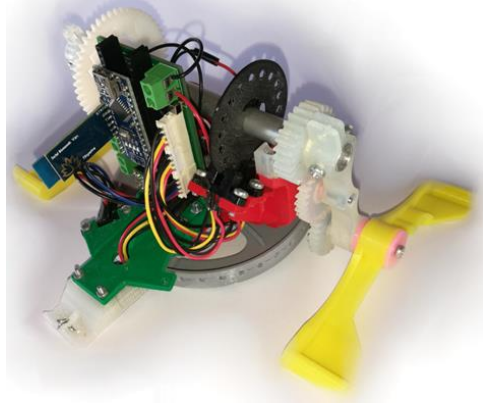


Ilustración 32-Robot móvil con dos grados de libertad

Fuente: (Stefanov et al., 2019b)

En 2019, Adamu, en la Universidad de Malaya en Malaysia, teniendo en mente la necesidad de los estudiantes de aprender y desarrollar habilidades para un correcto desempeño en la creciente industria 4.0, construyó un brazo robótico de 6 grados de libertad de bajo costo impreso en 3d, con propósitos de aprendizaje, manifestando que con esto los estudiantes aprenderán sobre impresión 3d, cinemática de un robot y cierto nivel de control.



Ilustración 33-Brazo robótico con 6 grados de libertad

Fuente: (Adamu, 2019)

3.6 SISTEMA DE CONTROL

Previamente se vio la importancia de la cinemática, y como esta es indispensable para obtener los parámetros necesarios de un movimiento correcto del robot analizado, pero para poder llevar a cabo dichos movimientos, un robot necesita también de un sistema de control. Este recibe información, la procesa y luego transmite comandos al sistema de actuación para que se lleven a cabo los movimientos que se requieren (Pandrea et al., 2017).

Para poder llevar a cabo este proceso de control, se requieren elementos fundamentales que realizan funciones específicas en un orden determinado. Todos los robots industriales en el mercado poseen estos elementos, *software* y *hardware* con los que controlan sus robots, sin embargo, son de uso exclusivo para estos, impidiendo así hacer uso de ellos para fines de esta investigación.

A continuación, se expondrán los diferentes elementos con los que se realizará el sistema de control, con el fin de darles una breve explicación.

3.6.1 Microcontroladores

(Dogan, 2008) nos dice:

“Un microcontrolador es un ordenador de un único chip. La palabra micro indica que el dispositivo es pequeño, y controlador indica que el dispositivo se puede usar en aplicaciones de control”.

Según Peña (2008) un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento, unidades de memoria, puertos de entrada y salida y periféricos. El propósito fundamental de los microcontroladores es de leer y ejecutar los comandos que el usuario escribe, en su memoria reside el programa que contiene los comandos para realizar una tarea, pero este se puede reprogramar haciéndolo útil en diversas aplicaciones. Todo microcontrolador requiere un programa y un lenguaje para poder ser capaz de recibir sus instrucciones, sin esto el microcontrolador carece de utilidad.

Existen una gran variedad de familias de microcontroladores, cada una con diferentes gamas: baja, media, alta y mejorada. Las familias más populares son los Atmega de Atmel y los PIC de Microchip.

En esta investigación se usará un microcontrolador de la familia Atmel, por su facilidad de uso, fuente abierta y cantidad de recursos disponibles.

3.6.2 Protocolos de comunicación

Según Subero (2021) en los sistemas de microcontroladores, hay dos tipos de comunicación: paralela y serial.

- **Paralela:** Un sistema de comunicación paralelo utiliza múltiples líneas de señal para transmitir datos. La comunicación paralela es rápida y era el método preferido de transferencia de datos en los sistemas informáticos más antiguos.
- **Serial:** En la comunicación serial, los datos se envían como un flujo de bits sobre muchas menos líneas de comunicación que en un sistema paralelo. Aunque la comunicación en serie es más lenta que la comunicación paralela, si la tasa de transferencia de datos es igual, el hardware moderno es lo suficientemente rápido como para eliminar la necesidad de sistemas paralelos.

Según Subero (2021), las comunicaciones en serie pueden tener lugar de dos maneras: de forma asíncrona o síncrona.

- **Asíncrona:** Las comunicaciones asíncronas envían datos como un flujo de bits. Este flujo de bits suele constar de tres partes. En la cabecera del flujo hay un bit de inicio que indica cuándo han comenzado a transmitirse los datos. También hay un bit de parada al final que indica cuando los datos han detenido la transmisión. Entre los bits de inicio y de parada, hay un paquete de datos que contiene los datos que queremos transmitir.
- **Síncrona:** Las comunicaciones seriales síncronas eliminan la sobrecarga asociada con el envío de bits de inicio y parada y, por lo tanto, requieren menos carga de software. Esto se logra utilizando un reloj para sincronizar la transferencia de datos, pero requiere de una línea adicional para cargar la señal del reloj.

En la actualidad, existen una gran cantidad de métodos de comunicación en serie, también llamados protocolos en serie. Los protocolos seriales más utilizados son: USART, I2C y SPI.

3.6.2.1 *Protocolo USART*

Aunque es capaz de ser usado de forma síncrona, esas características rara vez se utilizan, por lo cual la manera asíncrona, UART, es la que se encuentra en la mayoría de los sistemas de microcontroladores. La UART asíncrona tiene dos líneas de comunicación que están conectadas al pin de transmisión (TX) y al pin de recepción (RX). Estos pines son compartidos entre el transmisor que envía los datos y el receptor que es el receptor de estos datos transferidos. Para que la comunicación de datos comunicación de datos, también debe haber una conexión a tierra común entre los dispositivos (Subero, 2021).

3.6.3 Interfaz para el usuario

Las interfaces de usuario han existido desde que existen las computadoras, incluso mucho antes de que se estableciera el campo de la interacción persona-computadora (Jørgensen & Myers, 2008).

Según Takoordyal (2020) una interfaz de usuario (*UI*) se refiere a los medios por los cuales un usuario interactúa con un sistema informático.

En el pasado, se utilizaba bastante la interfaz de líneas de comando, ya que fue la primera interfaz en ser desarrollada y era bastante simple. Hoy en día, la mayoría de las interfaces proporcionan una interfaz gráfica de usuario (*GUI*), lo que permite al usuario una forma atractiva de visualizar información e interactuar con la aplicación. Las interfaces *GUI* proporcionan objetos comunes, incluidos cuadros de texto y botones, que ayudan al usuario a adaptarse rápidamente a nuevas aplicaciones. Además, fotografías, imágenes, íconos, vídeo y sonido a menudo se pueden incluir para mantener el interés del usuario. Menús y otros objetos de navegación también se incluyen comúnmente para ayudar al usuario a moverse por la aplicación con éxito (Prettyman, 2020).

IV METODOLOGÍA

La metodología es la explicación del camino recorrido para responder al problema central o para probar la hipótesis levantada. La metodología debe describir exactamente la forma en que se llevó a cabo el trabajo, de manera que éste pueda ser replicable con la información descrita en esta sección. En este capítulo se definirá el enfoque y alcance que tendrá la metodología de estudio, las variables de investigación, las hipótesis formuladas, las técnicas e instrumentos aplicados para el desarrollo del prototipo y se definirán y explicarán las diferentes etapas de la metodología seleccionada.

4.1 ENFOQUE

En esta investigación se utilizará un enfoque cuantitativo, debido a que el problema es muy específico y tiene datos medibles y observables, la literatura juega un rol fundamental al entendimiento, se compararan los resultados con predicciones y se realizaran análisis de cinemática, estructura y dispositivos, los cuales darán resultados numéricos primordiales para el diseño del robot.

4.2 ALCANCE

Esta investigación tiene un alcance descriptivo, ya que se considera el fenómeno estudiado, se describen características de los diferentes componentes que conforman el sistema, se miden conceptos, se definen variables y se formulan hipótesis.

4.3 VARIABLES DE INVESTIGACIÓN

Ya con el enfoque y el alcance de la investigación definidos, se identifican las variables dependientes e independientes. Estas son todos los elementos o factores que se evaluarán durante el proyecto y que adquieren distintos valores.

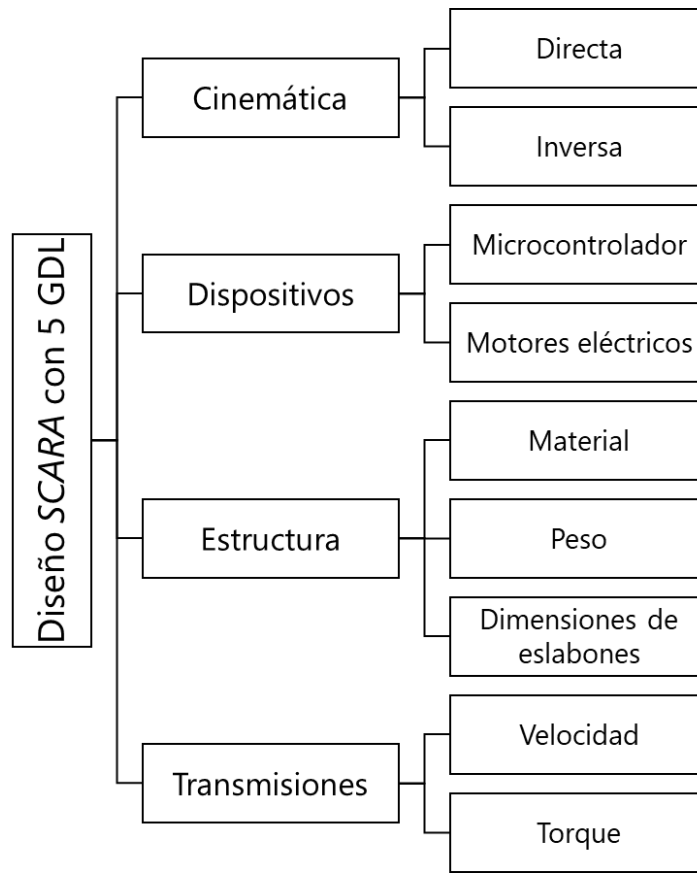


Ilustración 34-VARIABLES de investigación

Fuente: Elaboración propia

Nuestra variable dependiente es el diseño de SCARA con 5 grados de libertad, el cual se verá afectado y sujeto a cambios debido a sus variables independientes las cuales son:

Cinemática

En la cinemática se encuentran dos variables independientes, las cuales son cinemática directa e inversa.

- Cinemática Directa: Afecta directamente al movimiento de las articulaciones, a como este va a actuar con diferentes comandos.
- Cinemática Inversa: Afecta al movimiento de las articulaciones después de ingresar coordenadas específicas.

Dispositivos

- Microcontrolador: Determina el *IDE* y lenguaje de programación a utilizar, capacidad de entradas y salidas, almacenamiento y procesamiento.
- Motores eléctricos: Determina la velocidad de movimiento y fuerza base de las articulaciones e influye en la selección de componentes eléctricos.

Estructura

- Material: Determina las propiedades físicas que tendrá la estructura como ser: rigidez y punto de fusión.
- Peso: Afecta la fuerza requerida por las articulaciones para que puedan moverse y determinan el estrés al cual estarán sometidos los diferentes eslabones.
- Dimensiones de eslabones: Determinan la capacidad o rango del área de trabajo.

Transmisiones

- Velocidad: Determina el factor por el cual va a incrementar o disminuir la velocidad de las articulaciones.
- Torque: Determina el factor por el cual va a incrementar o disminuir el torque de las articulaciones.

4.4 HIPÓTESIS

A continuación, se presentarán las hipótesis alternativas de la investigación, las cuales guiaran el estudio, proporcionaran explicaciones y apoyaran la prueba de teorías.

1. El grado de libertad adicional le otorgará todas las ventajas enlistadas sobre las limitaciones de un SCARA de montaje tradicional.
2. El grado de libertad adicional le otorgará algunas de las ventajas enlistadas sobre las limitaciones en un SCARA de montaje tradicional.
3. El grado de libertad adicional no le otorgó ninguna ventaja sobre las limitaciones de un SCARA de montaje tradicional.

4.5 TÉCNICAS E INSTRUMENTOS APLICADOS

En el desarrollo de esta investigación se ha logrado obtener la información que la sustenta en diferentes fuentes como:

- Libros
- Artículos de revistas científicas
- Tesis de diferentes universidades
- Manuales

Para el desarrollo del prototipo se hará uso de los siguientes *software* ingenieriles:

- *SolidWorks* 2021
- *Matlab* R2020B
- *Ace* 4.0
- *Arduino* IDE
- *Proteus* 8.9
- *FlatCAM* 8.9
- *Ultimaker Cura* 4.2.1
- *Visual Studio*

El software de modelado 3D *SolidWorks* se utilizará para el diseño del prototipo al igual que para realizar pruebas de esfuerzos y deformaciones de los eslabones. *Matlab* se utilizará para ejecutar las pruebas de la cinemática directa a través del algoritmo D-H e inversa a través de métodos geométricos. *Arduino IDE* se utilizará para programar lógica en el microcontrolador. *Proteus* se utilizará para realizar el esquema de todas las conexiones electrónicas para luego, realizar el diseño de la placa de circuito impreso. *FlatCAM* y *Ultimaker Cura* serán utilizados para la manufactura del prototipo, procesarán el diseño del circuito impreso y el troceado del diseño 3d respectivamente. *Visual Studio* se utilizará para crear la interfaz gráfica para el usuario con la cual se controlará el robot. *Ace* 4.0 se utilizará para realizar simulaciones como método de comparación con un SCARA de montaje tradicional.

4.6 METODOLOGÍA DE ESTUDIO

Para el desarrollo general de esta investigación, se decidió utilizar la metodología de "modelo en V" propuesta por Vasic y Lazarevic, la cual fue adoptada de ingeniería de software y adaptada a los requerimientos mecatrónicos. El objetivo principal de esta es superar los clásicos procedimientos secuenciales de diseño de productos y el desarrollo de productos aislados en un dominio, con una reducción sustancial de costes y tiempo. Principalmente esta se basa en seis etapas las cuales se especifican en la ilustración 35.

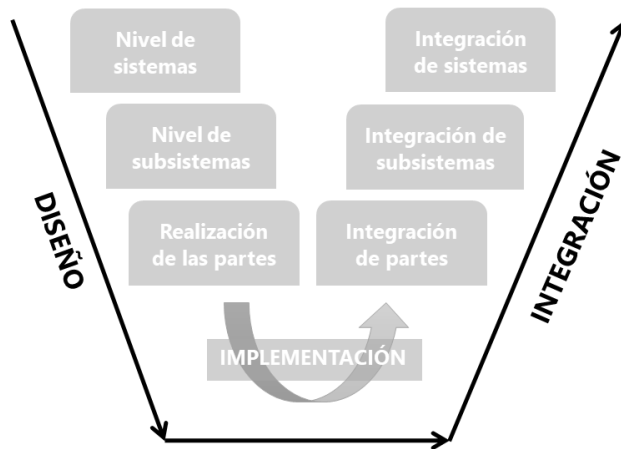


Ilustración 35-Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

4.6.1 Etapa I: Nivel de los sistemas

Para el desarrollo del prototipo de un robot SCARA con cinco grados de libertad, se definen los siguientes sistemas: sistema mecánico, sistema electrónico y sistema de control, cada uno con sus respectivos subsistemas.

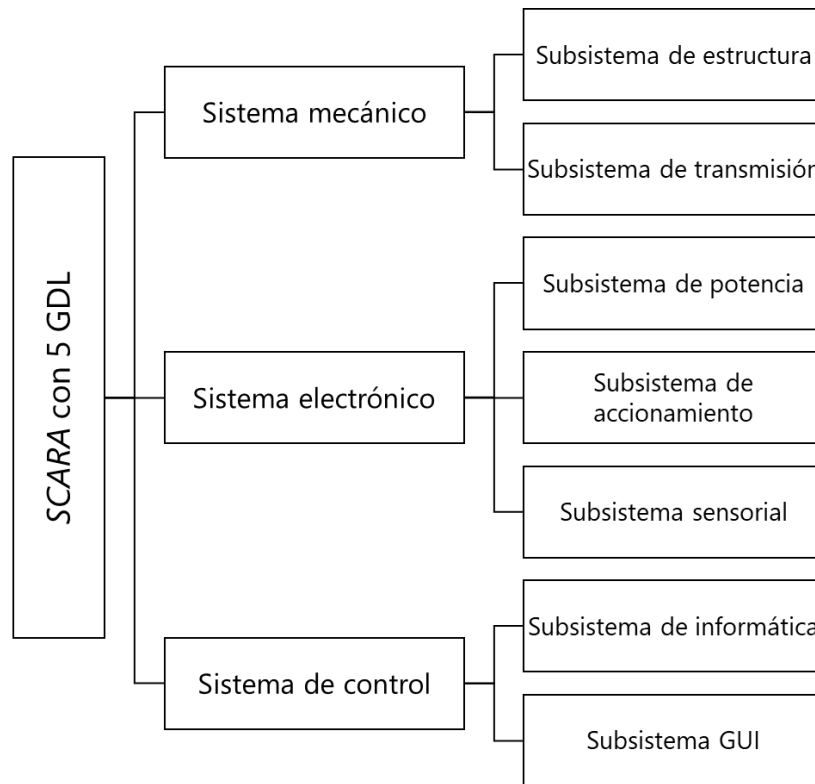


Ilustración 36-Sistemas y subsistemas del prototipo

Fuente: Elaboración propia

4.6.1.1 *Sistema mecánico*

Este sistema está encargado del diseño estructural, el cual asegurará una estructura rígida, estética y funcional; y las relaciones de fuerzas entre articulaciones, las cuales aseguran una velocidad y fuerza adecuada para el movimiento fluido del brazo robótico. Está constituido por los subsistemas de estructura y de transmisiones.

4.6.1.2 *Sistema electrónico*

Este sistema está encargado de la parte electrónica del prototipo, es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Cada uno de sus subsistemas representa a uno de estos circuitos: de potencia, de accionamiento y sensorial, que en conjunto integran el sistema electrónico.

4.6.1.3 Sistema de control:

Este sistema está encargado de administrar, ordenar, dirigir y regular el comportamiento del prototipo, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados, además, de permitirle al usuario poder operar el brazo robótico. Está compuesto por los subsistemas de informática y de *GUI*.

4.6.2 Etapa II: Nivel de los subsistemas

Para el desarrollo del prototipo de un robot SCARA con cinco grados de libertad, se definen los siguientes subsistemas de estructura, de transmisión, de potencia, de accionamiento, sensorial, de informática y *GUI*.

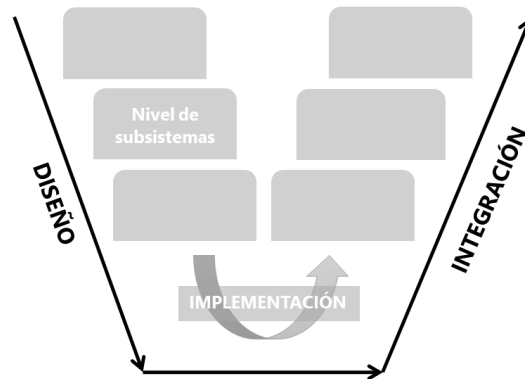


Ilustración 37-Etapa II: Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

4.6.2.1 Subsistema de estructura

Tiene como objetivo acoplar todos los eslabones a través de sus diferentes articulaciones de forma correcta y permitir que tengan un movimiento fluido. El subsistema está compuesto por diferentes partes, comenzando por el material de los eslabones que componen la estructura, las articulaciones que permiten el movimiento de los eslabones y, por último, diferentes componentes como rodamientos, varillas de acero, tornillos, tuercas y arandelas que son los que le dan la capacidad de sostenerse.

4.6.2.2 Subsistema de transmisión

Tiene como objetivo transmitir el movimiento de los actuadores a los eslabones, en este prototipo se utilizarán dos tipos de transmisión que ayudarán a modificar el torque y velocidad: Correa dentada que es una transmisión de entrada y salida circular, y tornillo sinfín que es una transmisión de entrada circular y salida lineal. Además, dentro de este se encuentran componentes como tensadores de banda, poleas y acoples flexibles que son los que ayudan a tener un funcionamiento óptimo.

4.6.2.3 *Subsistema de potencia*

Tiene como objetivo suministrar y transferir energía eléctrica al brazo robótico, en este prototipo se utilizará una fuente de alimentación que convertirá corriente alterna (CA) en corriente continua (CC) de 24V con 10A. Además, dentro de este se encuentran componentes como reguladores de voltaje los cuales son encargados de modificar el voltaje a un valor adecuado para ciertas áreas del circuito, capacitores y resistencias que mantendrán dichos voltajes estables.

4.6.2.4 *Subsistema de accionamiento*

Tiene como objetivo generar el movimiento de los elementos del robot según las órdenes dadas por el sistema de control, en este prototipo se utilizarán motores eléctricos CC NEMA 17 con sus respectivos drivers y cables de extensión, un microcontrolador ATMEGA328pu encargado del procesamiento y módulos de comunicación serial, que establecerán comunicación con la interfaz de usuario.

4.6.2.5 *Subsistema sensorial*

Tiene como objetivo delimitar el área de trabajo procurando salvaguardar la integridad de la estructura, esto a través de sensores, que en el caso de este prototipo serán sensores de fin de carrera.

4.6.2.6 *Subsistema de informática*

Tiene como objetivo almacenar y procesar información, es un conjunto de partes de *hardware* y *software*, principalmente este subsistema se enfocará en el software ya que el hardware fue cubierto en el sistema electrónico. Este incluye la programación, de la cual una gran parte es la resolución de la cinemática del robot, a través del algoritmo de D-H y métodos geométricos.

4.6.2.7 *Subsistema GUI*

Es un conjunto de componentes visuales interactivos, que tienen como objetivo mostrar objetos que transmiten información y representan acciones que puede realizar el usuario. Este prototipo tendrá una interfaz gráfica diseñada y creada exclusivamente para este.

4.6.3 Etapa III: Realización de las partes

Ya definidos los subsistemas y sus objetivos, se procede a detallar la realización de las partes para cada uno de estos.

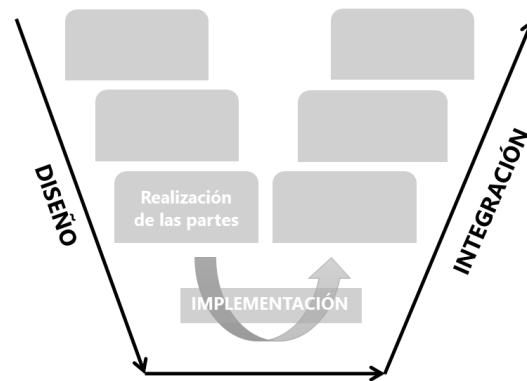


Ilustración 38-Etapa III: Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

4.6.3.1 *Subsistema estructural*

Para la realización de los eslabones se consideraron diferentes materiales de impresión 3d como PLA, y ABS:

Materiales	Densidad[kg/m³]	Límite elástico[N/m²]	Tensión de rotura[N/m²]	Punto de fusión [°C]
PLA	1290	40x10 ⁶	64.5x10 ⁶	200
ABS	1020	44.8x10 ⁶	38.8x10 ⁶	260

Tabla 3-Comparación de materiales PLA y ABS

Fuente: Elaboración propia

Entre estos se escogió el PLA por su accesibilidad, facilidad de manufactura y el límite elástico y tensión de rotura con respecto al ABS. Para la base, debido a que esta tiene que ser bastante

rígida y ser capaz de sostener el resto del robot, se consideraron materiales como láminas de hierro, aluminio y acero inoxidable, se escogió el hierro debido a su precio en comparación con los demás, accesibilidad, facilidad de soldadura y rigidez, además, que se considera que cumplen la misma función para los propósitos del prototipo. Se definió en el diseño que el SCARA alcanzará 445mm horizontalmente y 700mm verticalmente, por lo que el eslabón L2 de articulación a articulación mide 108[mm] y los eslabones L3 y L4 miden 169[mm] de articulación a articulación, alcanzando así 400[mm]. Cada eslabón cuenta con sus orificios respectivos para acoplar los motores, otros eslabones, tensadores de banda, sensores de final de carrera y los diferentes tornillos y tuercas, de igual forma, la altura de los eslabones es el adecuado para ensamblar los componentes dentro de ellos. Se realizarán pruebas de estrés y desplazamiento en *SolidWorks* a cada eslabón individual para comprobar su rigidez y diseño por medio de la siguiente fórmula:

$$W = m * g$$

Fórmula 3-Ecuación de peso

Donde:

W = peso del objeto

M = masa del objeto

G = fuerza de gravedad

4.6.3.2 *Subsistema de transmisión*

Para la realización de las partes de este subsistema, se harán varios cálculos y pruebas de los dientes necesarios en las poleas, para producir la potencia adecuada en las articulaciones para mover los eslabones, además, se obtendrá el largo de las bandas en *SolidWorks* en base a la distancia entre cada polea. Se utilizarán las siguientes formulas obtenidas de (Budynas et al., 2015):

$$pd = \frac{PN}{\pi}$$

Fórmula 4-Diametro de paso de la polea

Donde:

P = paso de la polea

N = número de dientes en la polea

$$M_r = \frac{pdSalida}{pdEntrada}$$

Fórmula 5-Relacion de transmisión

$$Rv = \frac{N2}{N1}$$

Fórmula 6-Relacion de velocidad

Donde:

N_1 = número de dientes polea grande

N_2 = número de dientes polea pequeña

Cabe destacar que este es el caso cuando la polea grande está ubicada en la salida

$$T_s = M_r * T_e$$

Fórmula 7-Torque de salida

Donde:

T_s = Torque de salida

$$H = T_e \omega_e = T_s \omega_s$$

Fórmula 8-Potencia del sistema

Donde:

H = potencia del sistema

ω_s = velocidad angular de salida

$$L = 2C + \frac{\pi(D_1 + D_2)}{2} + \frac{(D_1 - D_2)^2}{4C}$$

Fórmula 9-Largo de banda

Donde:

D_1 = diámetro de polea grande

D_2 = diámetro de polea pequeña

C = distancia entre centros de las poleas

Para la transmisión de tornillo sin fin, se utilizaron las siguientes fórmulas para conocer la potencia requerida del motor para el correcto funcionamiento, obtenidas de (Budynas et al., 2015):

$$T_c = \frac{F f_c d_c}{2}$$

Fórmula 10-Torque producido por collar de empuje

Donde:

F = carga, en este caso el brazo

f_c = coeficiente de fricción entre el collar y la superficie fija

d_c = diámetro medio de collar de empuje

$$T_R = \frac{F d_m}{2} \left(\frac{\pi f d_m + l \cos \alpha}{\pi d_m \cos \alpha - f l} \right) + T_c$$

Fórmula 11-Torque necesario para levantar la carga

Donde:

d_m = diámetro medio del tornillo sin fin

f = coeficiente de fricción entre el collar y el tornillo sin fin

l = avance lineal por cada vuelta

$\alpha = 14.5^\circ$

$$T_L = \frac{F d_m}{2} \left(\frac{\pi f d_m - l \cos \alpha}{\pi d_m \cos \alpha + f l} \right) + T_c$$

Fórmula 12-Torque necesario para bajar la carga

Se escogió un tornillo sin fin de 700mm de longitud para alcanzar el movimiento vertical requerido en la base y un tornillo sin fin de 400mm de longitud para alcanzar el movimiento vertical requerido en su efector final.

4.6.3.3 *Subsistema de potencia*

Para la realización de las partes de este subsistema, se escogerá una fuente de alimentación para el circuito de accionamiento, en base a los cálculos de potencia. El circuito de control será alimentado por el adaptador de comunicación serial que proveerá 5[V].

4.6.3.4 *Subsistema de accionamiento*

Para la realización de las partes de este subsistema, primero se tomaron en consideración varios microcontroladores como el PIC18F45K22, Atmega328pu y Atmega2560. Se escogió el Atmega327pu por su fácil integración a una PCB con respecto al Atmega2560, y sus librerías para motores paso a paso y cantidad de recursos con respecto a un PIC18F45K22. Para los motores, se tomaron en consideración paso a paso y servos, pero se escogieron los motores paso a paso debido a su facilidad de uso ya que no es un circuito de control cerrado. Dentro de estos se escogió el modelo NEMA 17 debido a su pequeño tamaño, compatibilidad con la mayoría de *los drivers* comunes, y bajo precio y peso en comparación con los NEMA 23. Los *drivers* que se escogieron son los DRV8825 por encima de los A4988 debido a la corriente máxima de 2A en este y la cantidad de recursos disponibles para utilización.

4.6.3.5 *Subsistema sensorial*

Para la realización de las partes de este subsistema, se escogieron sensores de final de carrera genéricos debido a su funcionamiento idéntico a la mayoría de los sensores en el mercado.

4.6.3.6 *Subsistema informático*

Para la realización de las partes del subsistema, se implementará el algoritmo de Denavit – Hartenberg, el cual dará como resultado los parámetros exactos para la cinemática directa del robot y ecuaciones geométricas para resolver la cinemática inversa del robot, se realizarán pruebas en Matlab para corroborar que la resolución del cinemática fuera correcta para su

utilización en la programación. Se realizará un diagrama de flujo que representa la lógica general que esta seguirá al ser procesada por el microcontrolador.

4.6.3.7 *Subsistema GUI*

Este subsistema está conformado solamente por la interfaz gráfica, la cual se diseñará con el objetivo que sea fácil de usar, intuitiva, estética.

4.6.4 Etapa IV: Integración de las partes

En esta etapa se describirá el proceso de integración de las diferentes partes que conforman los subsistemas, al igual que pruebas del funcionamiento adecuado de cada uno de estos.

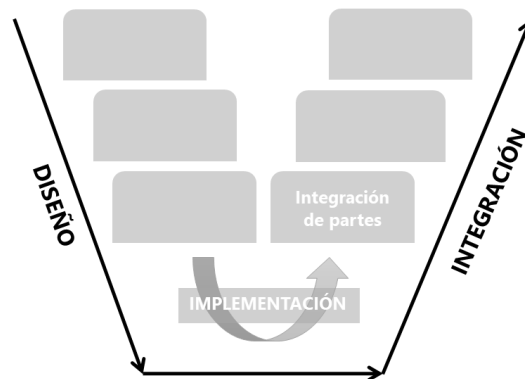


Ilustración 39-Etapa IV: Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

4.6.4.1 *Subsistema estructural*

Para la integración de las partes de este subsistema, se realizará un ensamblaje de aproximadamente 70 piezas en SolidWorks para tener el diseño completo en digital, posteriormente, se manufacturarán las piezas con impresión 3d y luego se realizará el ensamblaje de cada uno de los eslabones en su respectivo lugar, utilizando tornillos, tuercas y arandelas necesarias para el encaje y sujeción adecuado de cada uno de estos.

4.6.4.2 *Subsistema de transmisión*

Para la integración de las partes de este subsistema, se utilizarán dos transmisiones de banda dentada, una del eslabón L2 al L3 y otra del L3 al L4, dos transmisiones de tornillo sinfín, una en la base y otra en el efector final. Se iniciará manufacturando con impresión 3d las poleas de las

articulaciones para después ensamblar las poleas conductoras a los diferentes eje de los motores y las conducidas a las diferentes articulaciones, luego, las bandas que irán entre las poleas con sus respectivos tensadores, posteriormente, se acoplarán los tornillos sinfín a los motores y las tuercas de latón a los eslabones correspondientes. Después del ensamblaje, se procedió a realizar pruebas físicas con un motor y su polea en la entrada del sistema, la polea conductora en la salida y programación independiente del prototipo para corroborar su correcto funcionamiento.

4.6.4.3 *Subsistema de potencia*

Para la integración de las partes de este subsistema, se realizará un esquemático del circuito de alimentación para los circuitos de accionamiento y de control, con sus partes necesarias como ser capacitores, resistencias, adaptador serial y una fuente de alimentación para el circuito de accionamiento.

4.6.4.4 *Subsistema de accionamiento*

Para la integración de las partes de este subsistema, se realizará un esquemático del circuito de accionamiento en donde se integrarán los diferentes motores NEMA 17, los drivers y un circuito de control donde se encontrará el microcontrolador con su respectivo oscilador. Posteriormente, se realizaron pruebas de corriente y voltaje a los motores al ser conectados mediante los *drivers* para la calibración del mismo. Por último, se realizaron pruebas con programación independiente para corroborar movimiento y RPM adecuada para los motores.

4.6.4.5 *Subsistema sensorial*

Se realizará un esquemático de conexión para todos los sensores de final de carrera. Dichos sensores se acoplarán a la estructura, para delimitar las diferentes articulaciones. Posteriormente se realizarán pruebas con programación independiente para corroborar que, en efecto, dicho sensor delimita los motores.

4.6.4.6 *Subsistema informático*

Para la integración de las partes de este subsistema, se debe previamente resolver la cinemática directa a través del algoritmo de Denavit – Hartenberg y la cinemática inversa a través de los métodos geométricos, para posteriormente integrar dichas respuestas a la programación principal que procesara el microcontrolador.

4.6.4.7 Subsistema GUI

Este subsistema está conformado solamente por la interfaz gráfica, la cual después de su diseño y pruebas preliminares, se comprobará que funciona siendo conectada por medio del módulo de comunicación serial al Arduino para ser capaz de accionar un servo motor a través de ella.

4.6.5 Etapa V: Integración de los subsistemas

En esta etapa se describirá el proceso de integración de los diferentes subsistemas que conforman los sistemas principales, al igual que pruebas del funcionamiento adecuado de cada uno de estos.

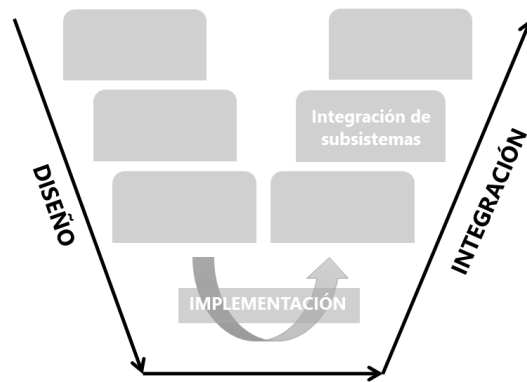


Ilustración 40-Etapa V: Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

4.6.5.1 Sistema mecánico

Una vez realizadas las pruebas de los subsistemas de estructura y transmisión, corroborando su funcionamiento adecuado, se procedió a integrar las transmisiones en la estructura. Posteriormente, se realizaron pruebas físicas de movimiento y rigidez a la estructura junto con sus transmisiones para corroborar el correcto desempeño de estas.

4.6.5.2 Sistema electrónico

Una vez realizadas las pruebas de los subsistemas de potencia, accionamiento y sensorial, corroborando su funcionamiento adecuado, se procedió a realizar el esquemático completo uniendo los diferentes circuitos, para después manufacturar la PCB y soldar todos los componentes a esta.

4.6.5.3 Sistema de control

Una vez realizadas las pruebas de los subsistemas de informática y *GUI*, corroborando su funcionamiento adecuado, se procedió a integrarlos mediante comunicación serial entre el Arduino y la PC del usuario para permitirle operar por medio de la *GUI*, todo esto con el objetivo de comprobar el correcto funcionamiento.

4.6.6 Etapa VI: Integración de los sistemas

En esta etapa se describirá el proceso de integración de los diferentes sistemas principales que conforman *SCARA*, al igual que pruebas del funcionamiento adecuado de cada uno de estos.

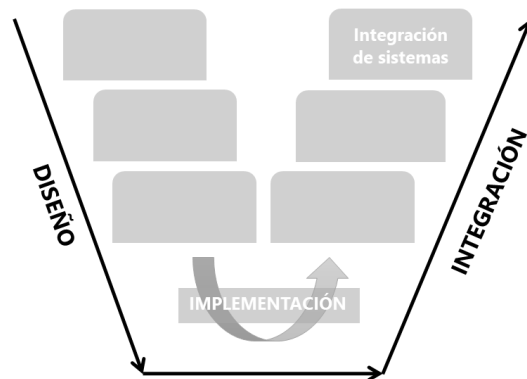


Ilustración 41-Etapa VI: Metodología en "V"

Fuente: Elaboración propia con información obtenida de (Vasic & Lazarevic, 2008)

Una vez realizadas todas las pruebas de los diferentes sistemas que componen el *SCARA*, se procederá a integrarlos. Se iniciará integrando el sistema electrónico al sistema mecánico, instalando en la base la PCB junto con la fuente de alimentación, los motores a los diferentes eslabones junto con los sensores de fin de carrera. Luego se integrará el sistema de control al sistema electrónico, subiendo la programación completa al microcontrolador y conectando mediante el adaptador serial la PC del operador a la PCB del *SCARA*. Posteriormente, se realizarán pruebas de punto a punto para corroborar el correcto funcionamiento del *SCARA* completo.

4.7 CRONOGRAMA DE ACTIVIDADES

Se realizó un cronograma de actividades detallando cada tarea a realizar a lo largo del desarrollo de la investigación en base a un tiempo de trabajo de 10 semanas.

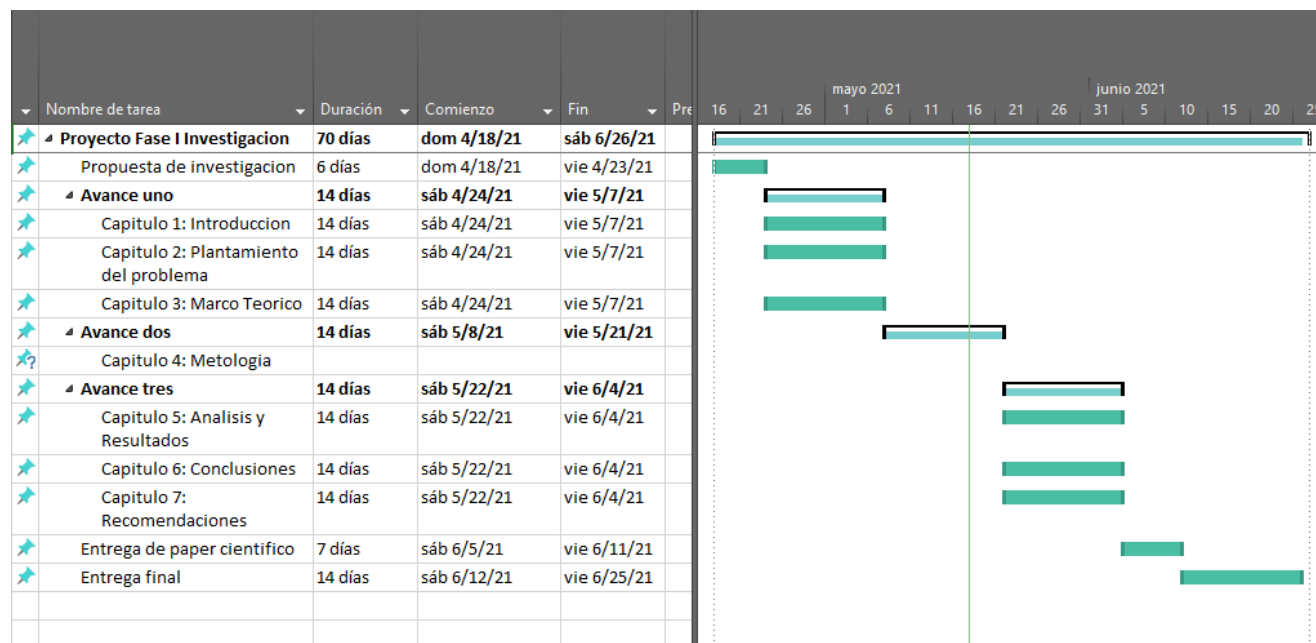


Ilustración 42-Cronograma de actividades

Fuente: Elaboración propia

V RESULTADOS Y ANÁLISIS

En este capítulo se presentará los resultados cuantitativos obtenidos durante el desarrollo de la investigación, junto con el análisis e interpretación de los mismos. Dichos resultados se dividirán en dos partes: Directos e indirectos.

5.1 RESULTADOS INDIRECTOS

Los indirectos abarcarán las diferentes etapas de la metodología que requieran de justificación y validación por medio de pruebas, simulaciones y/o cálculos. Debido a que las primeras dos etapas no requieren ningún tipo de análisis ni resultados, se obviarán de esta sección.

5.1.1 Etapa 3: Realización de las partes

En esta sección se mostrarán los resultados de la etapa 3 de la metodología.

5.1.1.1 *Subsistema estructural*

Durante la realización de las partes del subsistema estructural, se realizaron en *SolidWorks* pruebas estáticas de estrés y desplazamiento para comprobar que los eslabones no sufrirán deformaciones por exceso de estrés y que no se desplazarán debido al peso que soportarán. Para el cálculo de los pesos se utilizó la fórmula 3 previamente mencionada en la metodología. Las pruebas se realizaron en la placa inferior, eslabón L2, eslabón L3 y eslabón L4 ya que son los que soportarán los diferentes pesos del robot. Con estas pruebas, se demostró que el largo escogido para los eslabones no afectó estructuralmente al robot.

La placa inferior tendrá que soportar un peso de 40[N] los cuales 30[N] son del brazo robótico y el resto es el peso de la carga, además de eso soportará el peso de cuatro tubos de acero que aplicaran una fuerza de 5.8[N] cada uno.

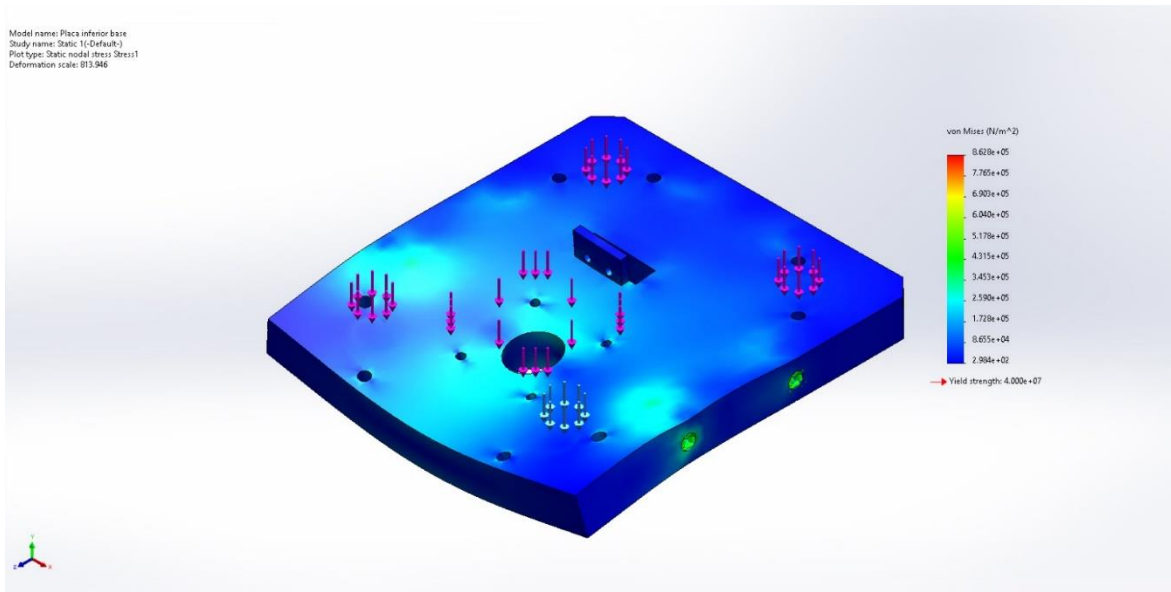


Ilustración 43-Prueba de estrés Von Mises placa inferior

Fuente: Elaboración propia

Como se puede observar en la Ilustración 43, la pieza muestra un esfuerzo máximo de $8.628e+05[N/m^2]$ y un mínimo de $2.984e+02[N/m^2]$, y no se observa ningún punto de deformación o ruptura, lo cual representa un buen resultado.

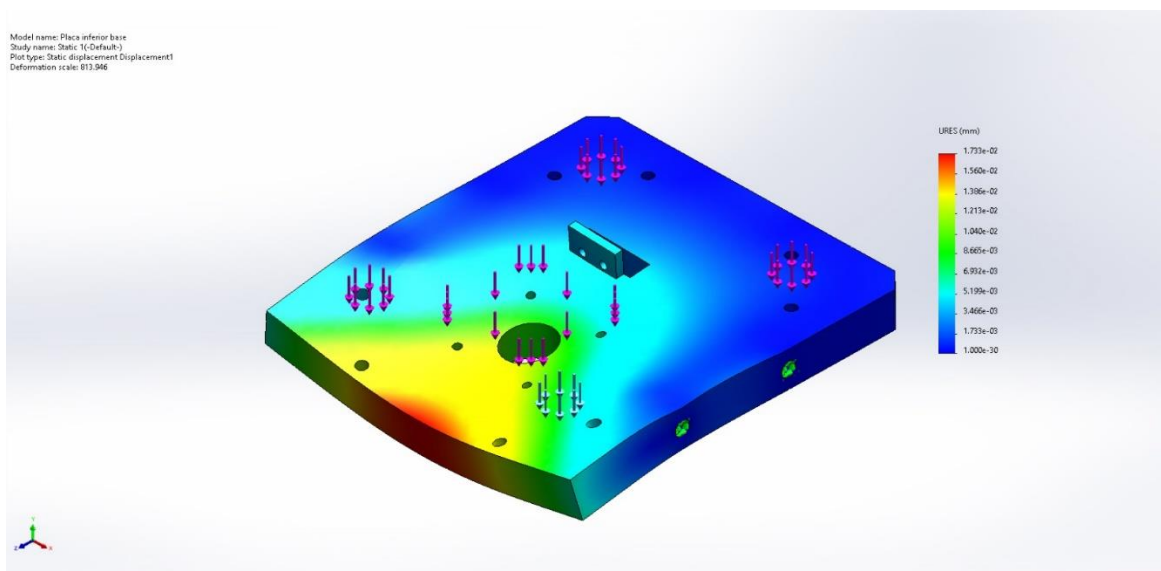


Ilustración 44-Prueba de desplazamiento placa inferior

Fuente: Elaboración propia

Como se puede observar en la Ilustración 44, la pieza muestra un desplazamiento máximo de $1.733e-02$ [mm] y un mínimo de $0.000e+00$ [mm], estos valores representan un desplazamiento despreciable por lo cual es un buen resultado.

El eslabón 1 tendrá que soportar un peso de 27.5 [N] en su articulación de revoluta, 40 [N] en su articulación de tornillo y 2.95 [N] en dirección contraria de un motor acoplado en su parte baja.

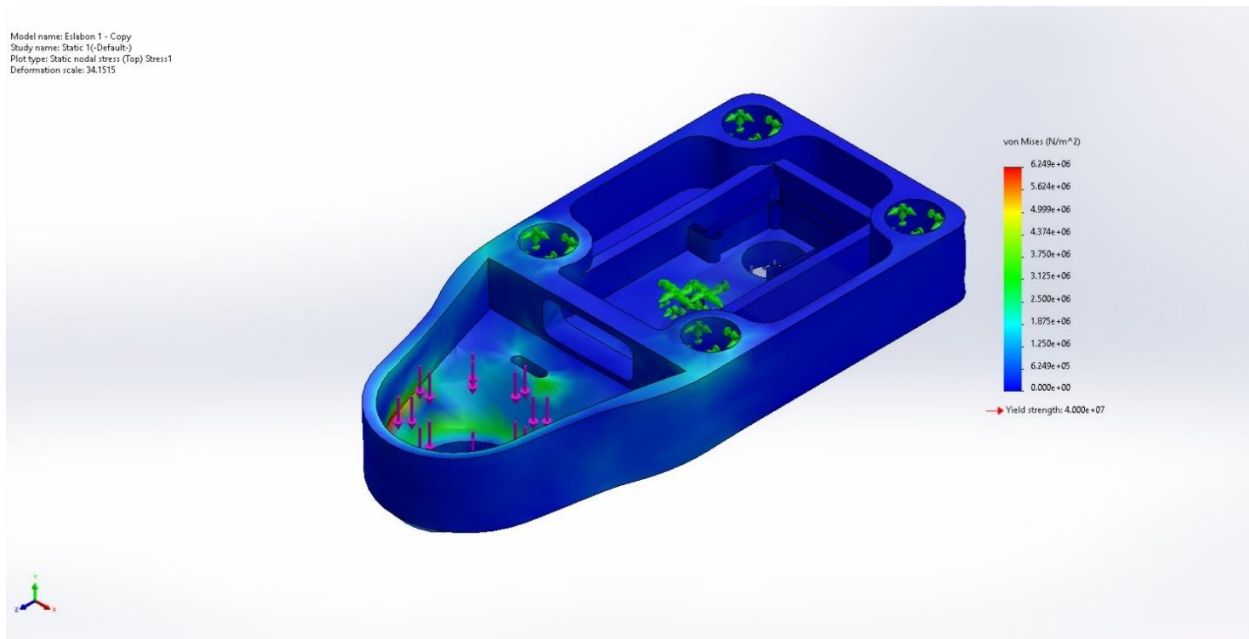


Ilustración 45-Prueba de estrés Von Mises eslabón 1

Fuente: Elaboración propia

Como se puede observar en la Ilustración 45, la pieza muestra un esfuerzo máximo de $6.249e+06$ [N/m²] y un mínimo de $0.000e+00$ [N/m²], y no se observa ningún punto de deformación o ruptura, lo cual representa un buen resultado.

Model name: Eslabon 1 - Copy
Study name: Static 1(-Default-)
Plot type: Static displacement Displacement1
Deformation scale: 34.1515

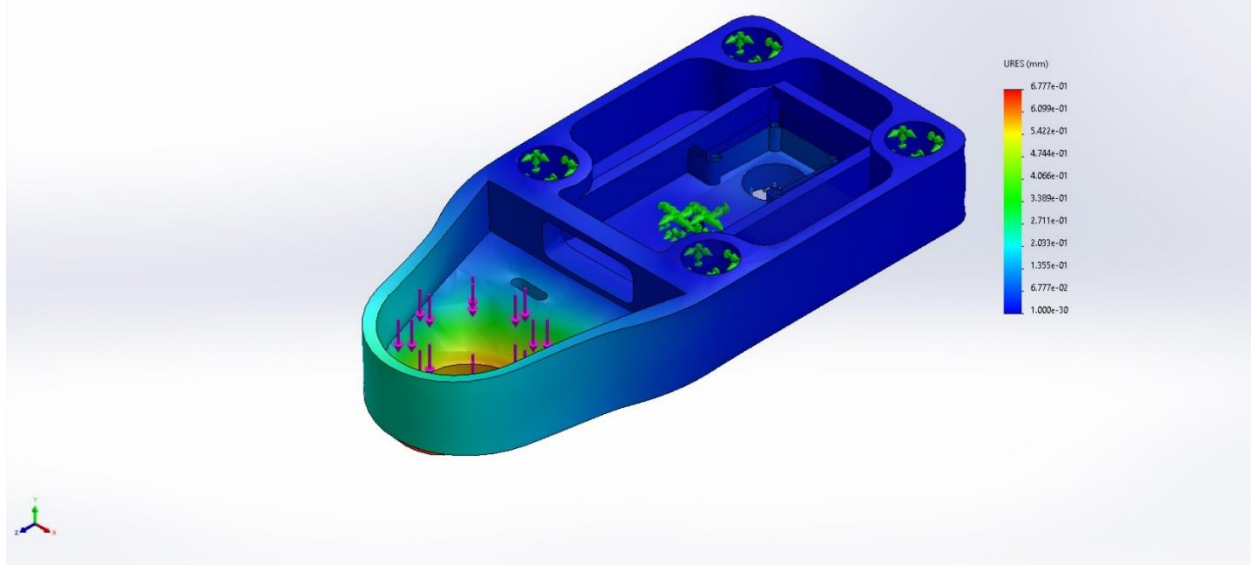


Ilustración 46-Prueba de desplazamiento eslabón 1

Fuente: Elaboración propia

Como se puede observar en la Ilustración 46, la pieza muestra un desplazamiento máximo de $6.777e-01$ [mm] y un mínimo de $0.000e+00$ [mm], estos valores representan un desplazamiento despreciable por lo cual es un buen resultado.

El eslabón 2 tendrá que soportar un peso de 21.41 [N] en su articulación de revoluta y 2.95 [N] en dirección contraria de un motor acoplado en su parte baja.

Model name: Assem1
Study name: Static 1(-Default-)
Plot type: Static nodal stress Stress1
Deformation scale: 141.146

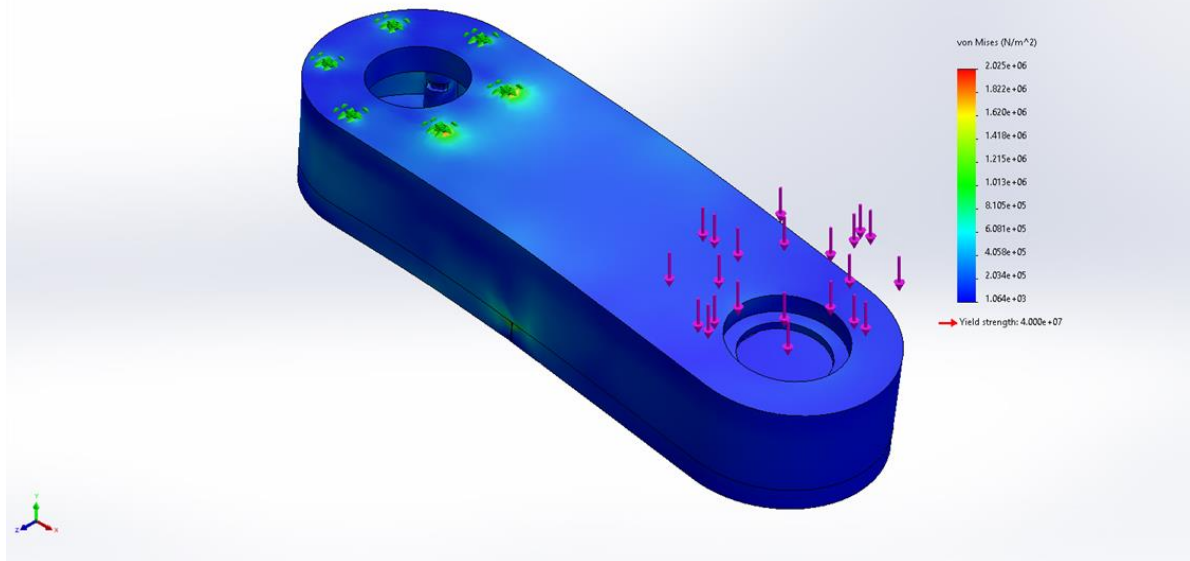


Ilustración 47-Prueba de estrés Von Mises eslabón 2

Fuente: Elaboración propia

Como se puede observar en la Ilustración 47, la pieza muestra un esfuerzo máximo de 2.025×10^6 [N/m²] y un mínimo de 1.064×10^3 [N/m²], y no se observa ningún punto de deformación o ruptura, lo cual representa un buen resultado.

Model name: Assem1
Study name: Static 1(-Default-)
Plot type: Static displacement Displacement1
Deformation scale: 141.146

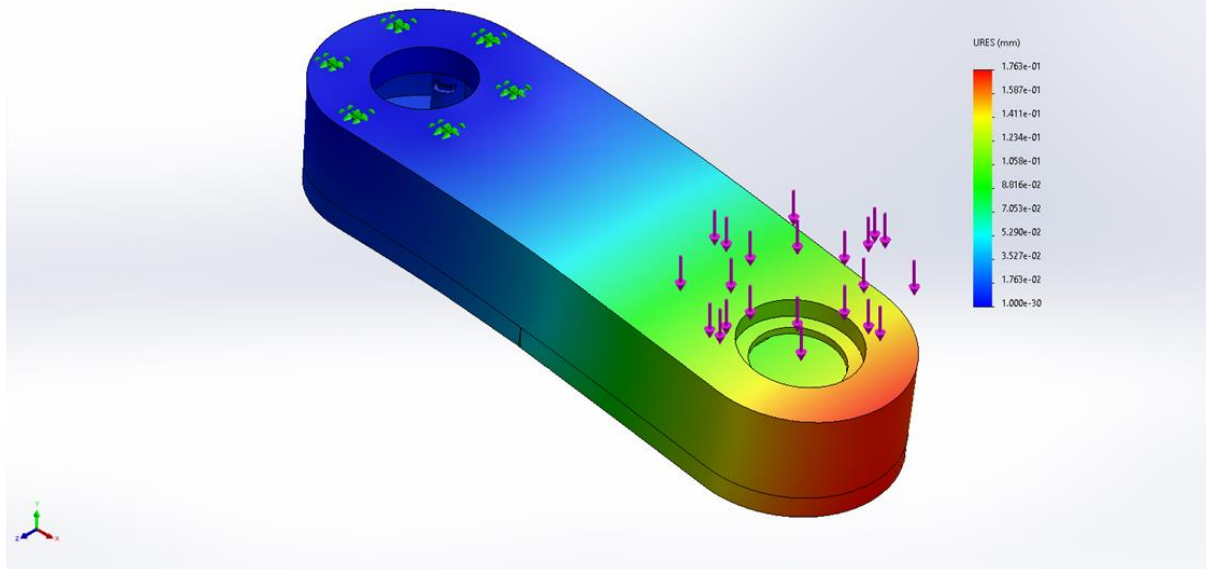


Ilustración 48-Prueba de desplazamiento eslabón 2

Fuente: Elaboración propia

Como se puede observar en la Ilustración 48, la pieza muestra un desplazamiento máximo de $1.763e-01$ [mm] y un mínimo de $0.000e+00$ [mm], estos valores representan un desplazamiento despreciable por lo cual es un buen resultado.

El eslabón 3 tendrá que soportar un peso de 21.41 [N] en su articulación de revoluta y 2.95 [N] en dirección contraria de un motor acoplado en su parte baja.

Model name: Eslabon 3
Study name: Static Z(-Default-)
Plot type: Static modal stress Stress1
Deformation scale: 28.3094

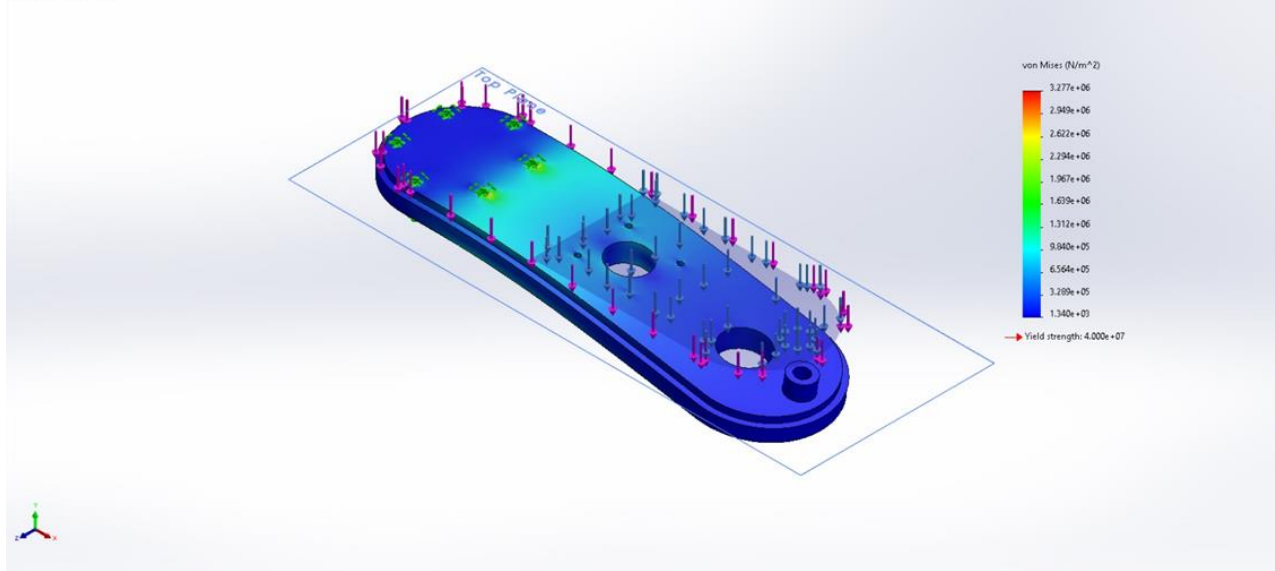


Ilustración 49-Prueba de estrés Von Mises eslabón 3

Fuente: Elaboración propia

Como se puede observar en la Ilustración 49, la pieza muestra un esfuerzo máximo de $3.277e+06[N/m^2]$ y un mínimo de $1.340e+03[N/m^2]$, y no se observa ningún punto de deformación o ruptura, lo cual representa un buen resultado.

Model name: Eslabon 3
Study name: Static Z(-Default-)
Plot type: Static displacement Displacement1
Deformation scale: 28.3094

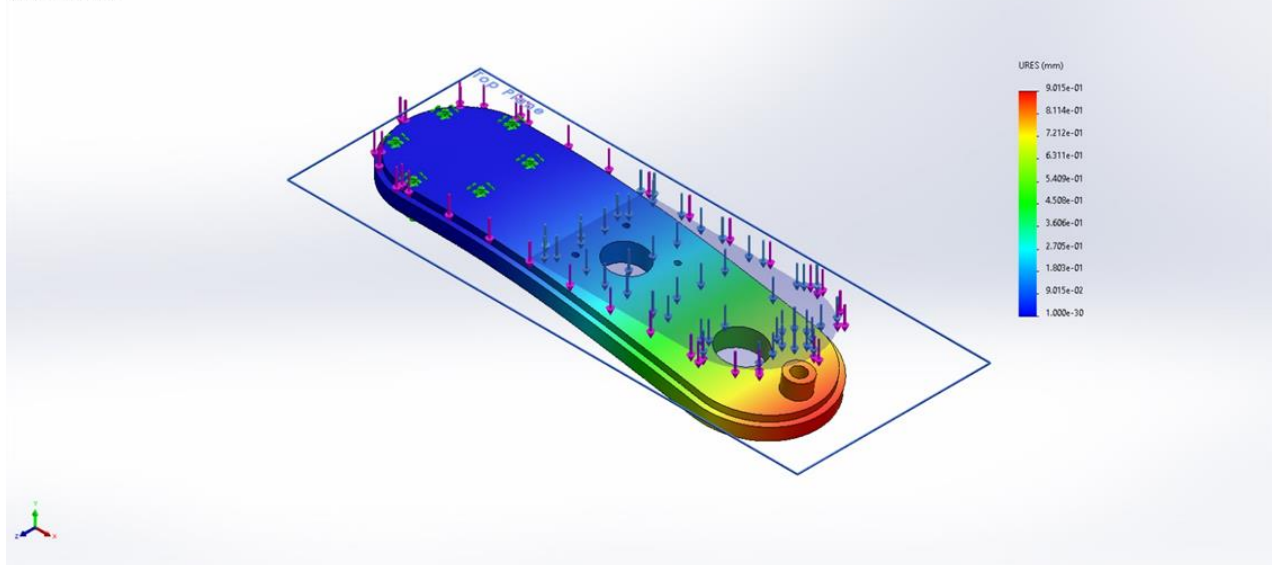


Ilustración 50-Prueba de desplazamiento eslabón 3

Fuente: Elaboración propia

Como se puede observar en la Ilustración 50, la pieza muestra un desplazamiento máximo de $9.015e-01$ [mm] y un mínimo de $0.000e+00$ [mm], estos valores representan un desplazamiento despreciable por lo cual es un buen resultado.

El acople tendrá que soportar un peso de 27.5 [N] que corresponde al peso del brazo robótico.

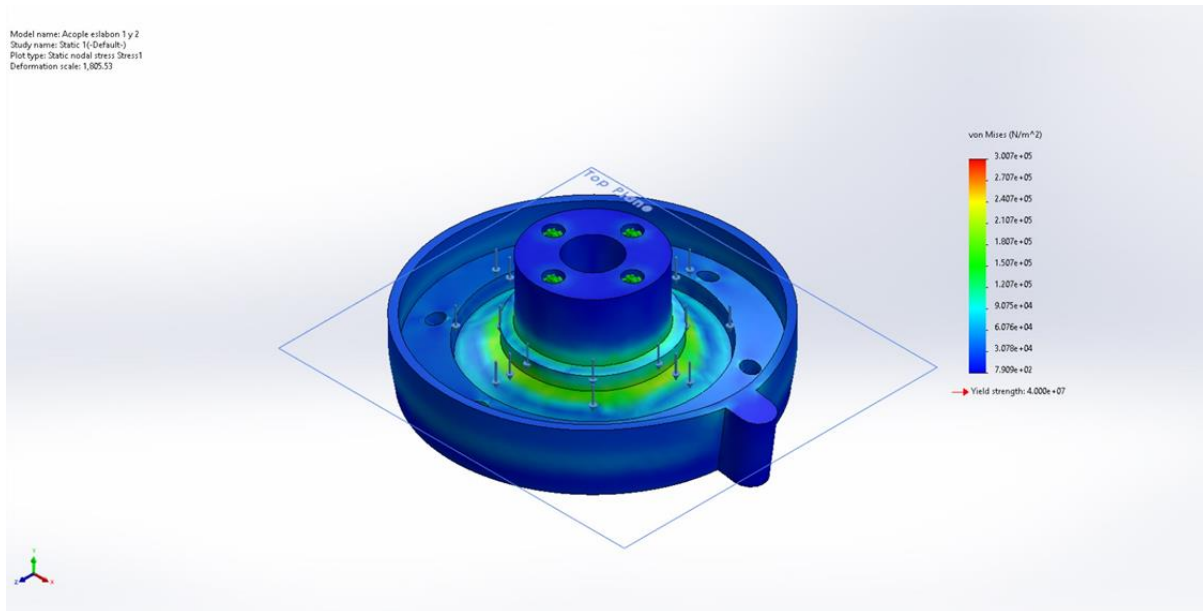


Ilustración 51-Prueba de estrés Von Mises acople

Fuente: Elaboración propia

Como se puede observar en la Ilustración 51, la pieza muestra un esfuerzo máximo de $3.007e+05$ [N/m²] y un mínimo de $7.909e+02$ [N/m²], y no se observa ningún punto de deformación o ruptura, lo cual representa un buen resultado.

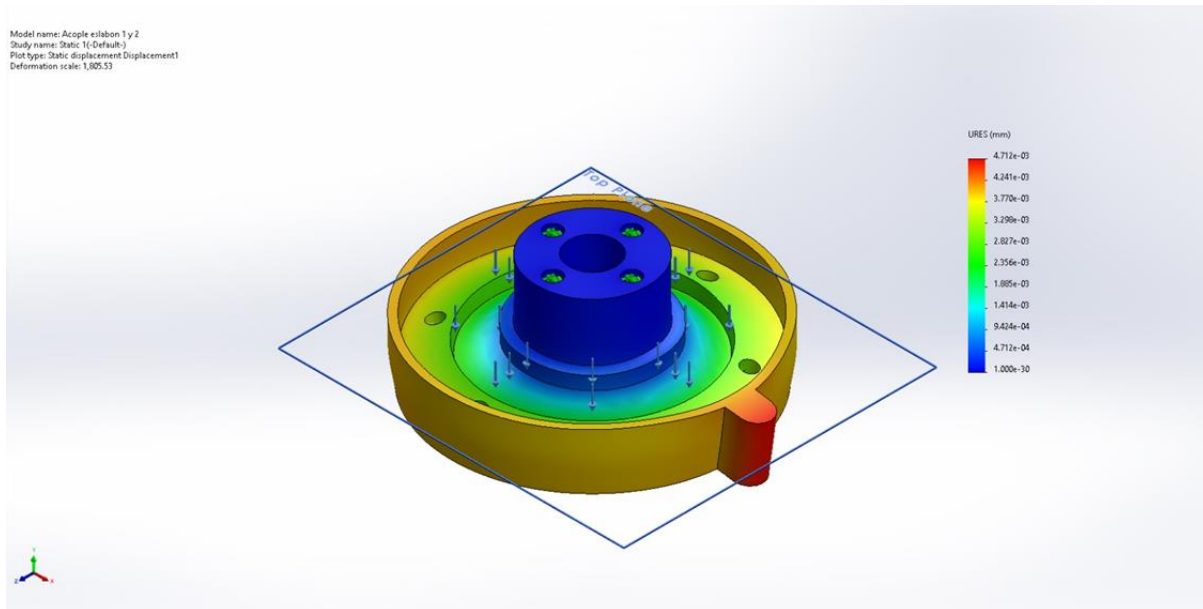


Ilustración 52-Prueba de desplazamiento acople

Fuente: Elaboración propia

Como se puede observar en la Ilustración 52, la pieza muestra un desplazamiento máximo de $4.712e-03$ [mm] y un mínimo de $0.000e+00$ [mm], estos valores representan un desplazamiento despreciable por lo cual es un buen resultado.

5.1.1.2 Subsistema de transmisión

Para poder diseñar el subsistema de transmisión, se tuvo que hacer uso de las fórmulas previamente mencionadas en la metodología.

Para la transmisión de tornillo sinfín se realizaron los siguientes cálculos:

$$T_R = \frac{(40N) * (0.007m)}{2} \left(\frac{\pi * (0.1) * (0.007m) + (0.008m) \cos(14.5^\circ)}{\pi * (0.007m) \cos(14.5^\circ) - (0.1) * (0.008m)} \right) + \frac{(40N) * (0.15) * (0.022m)}{2}$$

$$T_R = 0.1339N \cdot m = 13.39N \cdot cm$$

$$T_L = \frac{(40N) * (0.007m)}{2} \left(\frac{\pi * (0.1) * (0.007m) - (0.008m) \cos(14.5^\circ)}{\pi * (0.007m) \cos(14.5^\circ) + (0.1) * (0.008m)} \right) + \frac{(40N) * (0.15) * (0.022m)}{2}$$

$$T_L = 0.0308N \cdot m = 3.08N \cdot cm$$

Para la transmisión de banda y polea dentada se realizaron los siguientes cálculos:

$$Pde = \frac{2mm * 20mm}{\pi}$$

$$Pde = 12.73mm$$

$$Pds = \frac{2mm * 92mm}{\pi}$$

$$Pds = 58.56mm$$

$$M_r = \frac{58.56mm}{12.73mm} = 4.6$$

$$R_v = \frac{N_2}{N_1} = M_r^{-1}$$

$$R_v = \frac{20}{92} = 0.2173$$

$$R_v = \frac{v_s}{v_e}$$

$$V_e = \frac{100}{0.2173} = 460rpm$$

$$T_e = \text{grafico nema para } 460rpm \text{ (47Ncm)}$$

$$T_s = M_r \cdot T_e$$

$$T_s = 4.6 \cdot 47Ncm$$

A continuación se presenta la ilustración 53 donde se puede observar la curva de torque vs velocidad de los motores NEMA 17-HS16-2004S1 que son los que se usaron en el desarrollo de esta investigación.

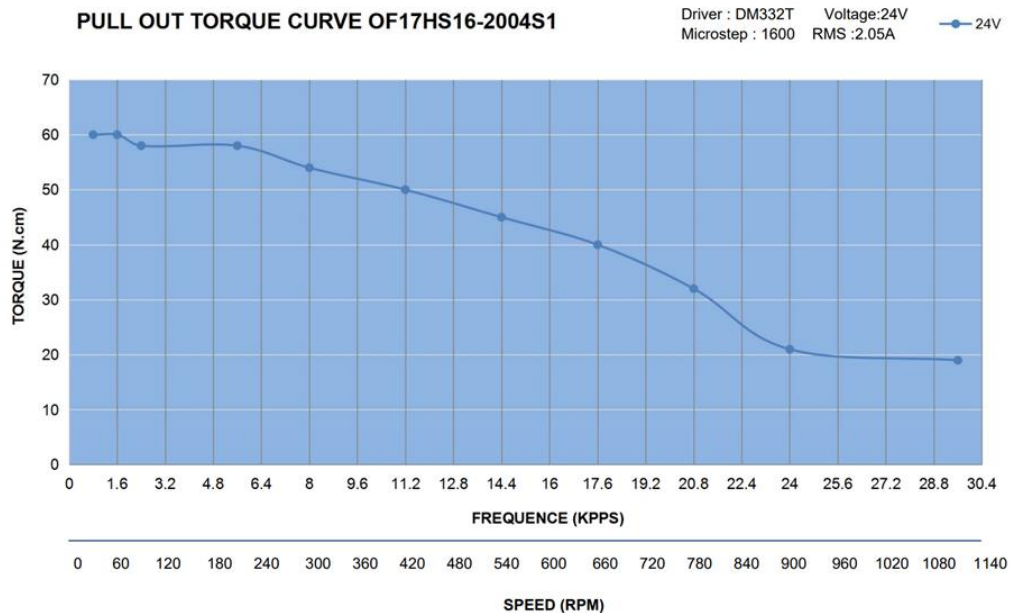


Ilustración 53-Curva de torque vs velocidad NEMA 17 HS16-2004S1

Fuente: Fabricante

5.1.1.3 Subsistema de potencia

Para el subsistema de potencia se optó por una fuente de poder de 24V 10A, 240W. Esto debido a que se tienen cinco motores con corriente nominal de 2A y las pruebas hechas por el fabricante de los motores en las que muestra el torque a diferentes velocidades, fueron hechas con 24V. Utilizando la fórmula de potencia $P = VI$ y multiplicando por el número de motores, se obtiene un resultado final de 240W. En general, se necesita menos que la corriente nominal, esto dependerá de las velocidades y las condiciones de carga por motor, lo cual deja la elección de la fuente de poder en un buen rango para suplir las necesidades del circuito.

5.1.1.4 Subsistema de accionamiento

No hay resultados indirectos en este subsistema dentro de esta etapa.

5.1.1.5 Subsistema sensorial

No hay resultados indirectos en este subsistema dentro de esta etapa.

5.1.1.6 Subsistema informático

En este subsistema se tuvo que hacer la resolución de la cinemática directa e inversa para el robot y el diagrama de flujo que representará la lógica general de la programación.

Para la cinemática directa, se utilizó el algoritmo de Denavit – Hartenberg como método de resolución. Siguiendo los pasos del algoritmo, lo primero que se tuvo que hacer fue un diagrama del robot en el que se enumeraron los eslabones y articulaciones, posteriormente se localizaron arbitrariamente los ejes Z de cada articulación, los ejes X a lo largo de la normal común entre los ejes Z, y los ejes Y siguiendo la regla de la mano derecha. El sistema 0 correspondiendo a la articulación 1 y el sistema 5 correspondiendo al efector final, una vez terminados estos pasos se tuvo como resultado el diagrama de sistemas que se puede observar en la Ilustración 54.

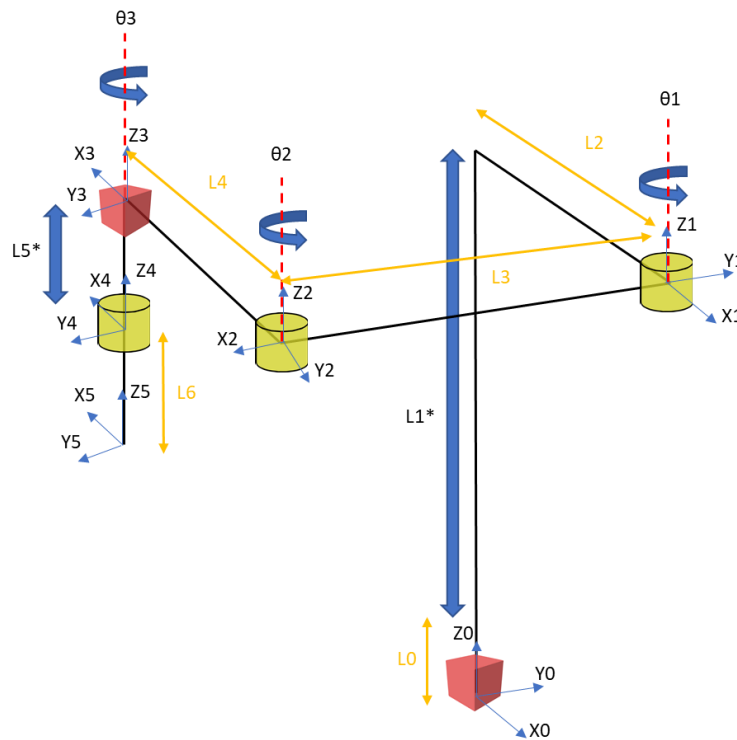


Ilustración 54-Diagrama de sistemas de coordenadas de SCARA con 5 GDL

Fuente: Elaboración propia

Seguidamente, se tuvo que pasar del sistema $i-1$ al sistema actual i , a través de los cuatro parámetros de D-H: Rotación en Z, traslación en Z, traslación en X y rotación en X. Esta resolución da como resultado, los parámetros D-H que se pueden observar en la tabla 4, los cuales se ingresan en la fórmula 1 previamente mencionada.

i	θ	d	a	α
1	0	L0+L1	L2	0
2	-90+ θ_1	0	L3	0
3	-90+ θ_2	0	L4	0
4	0	-L5	0	0
5	θ_3	-L6	0	0

Tabla 4-Parámetros D-H para SCARA con 5 GDL

Fuente: Elaboración propia

$$H_{(i-1) \rightarrow i} = Rotz(\theta_i) \cdot Tz(0,0,d_i) \cdot Tx(a_i,0,0) \cdot Rotx(\alpha_i)$$

A partir de los parámetros de la tabla se encuentran las matrices de transformación homogénea, las cuales describen el paso del sistema i-1 a un sistema i. Estos parámetros deben ser sustituidos en la matriz base de Denavit – Hartenberg (H), la cual se puede observar en la fórmula 2. Se utilizó el *software* MATLAB para la resolución de estas matrices.

$$H_{(i-1) \rightarrow i} = \begin{pmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Al sustituir los parámetros del sistema de coordenadas 0 al 1 en la matriz base, nos da como resultado la matriz de transformación homogénea H01, la cual se puede observar a continuación:

$$H_{01} = \begin{pmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_0 + L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Al sustituir los parámetros del sistema de coordenadas 1 al 2 en la matriz base, nos da como resultado la matriz de transformación homogénea H12, la cual se puede observar a continuación:

$$H_{12} = \begin{pmatrix} \cos(\theta_1 - 90) & -\sin(\theta_1 - 90) & 0 & L_3 * \cos(\theta_1 - 90) \\ \sin(\theta_1 - 90) & \cos(\theta_1 - 90) & 0 & L_3 * \sin(\theta_1 - 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Al sustituir los parámetros del sistema de coordenadas 2 al 3 en la matriz base, nos da como resultado la matriz de transformación homogénea H23, la cual se puede observar a continuación:

$$H_{23} = \begin{pmatrix} \cos(\theta_2 - 90) & -\sin(\theta_2 - 90) & 0 & L_4 * \cos(\theta_2 - 90) \\ \sin(\theta_2 - 90) & \cos(\theta_2 - 90) & 0 & L_4 * \sin(\theta_2 - 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Al sustituir los parámetros del sistema de coordenadas 3 al 4 en la matriz base, nos da como resultado la matriz de transformación homogénea H34, la cual se puede observar a continuación:

$$H_{34} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -L_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Al sustituir los parámetros del sistema de coordenadas 4 al 5 en la matriz base, nos da como resultado la matriz de transformación homogénea H45, la cual se puede observar a continuación:

$$H_{45} = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & -L_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La multiplicación de estas matrices da como resultado el movimiento necesario para pasar del sistema de coordenadas 0 al 5, el cual está representado por la matriz H05, se puede observar a continuación:

$$H_{05} = \begin{pmatrix} \cos(\theta_1 + \theta_2 + \theta_3 - 180) & -\sin(\theta_1 + \theta_2 + \theta_3 - 180) & 0 & L_2 + L_4 * \cos(\theta_1 + \theta_2 - 180) + L_3 * \cos(\theta_1 - 90) \\ \sin(\theta_1 + \theta_2 + \theta_3 - 180) & \cos(\theta_1 + \theta_2 + \theta_3 - 180) & 0 & L_4 * \sin(\theta_1 + \theta_2 - 180) + L_3 * \sin(\theta_1 - 90) \\ 0 & 0 & 1 & L_0 + L_1 - L_5 - L_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Para la resolución de la cinemática inversa se utilizó el método geométrico, analizando únicamente los dos eslabones relacionados a las articulaciones de revoluta, debido a que el robot cuenta con dos GDL en el eje z, lo que hace que el número de soluciones para este resulte infinito.

Tomando en consideración lo anterior, se realizó un análisis similar al de un robot de 2 GDL, el diagrama se puede observar a continuación:

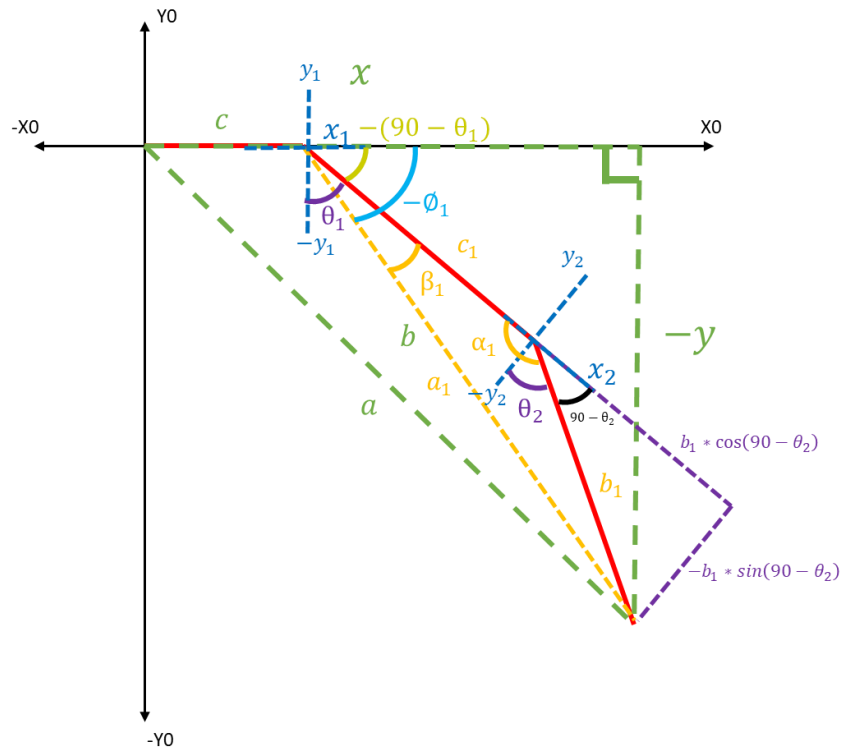


Ilustración 55-Diagrama geométrico de cinemática inversa

Fuente: Elaboración propia

A través del método geométrico se deben encontrar relaciones de ángulos que incluyan a theta 1 y theta 2, luego, se procede a la resolución de los mismos haciendo relaciones de tangente, para posteriormente convertirlo en arcotangente2 (atan2) que es una función creada para la correcta determinación del cuadrante de una coordenada (x, y), ya que arcotangente (atan) solo funciona en el cuadrante 1 y 4. Se evita utilizar cosenos y senos inversos debido a que es difícil computarlos, lo que hace que el procesamiento sea lento, esto afecta al desempeño real del manipulador.

Estos son los valores reales de las variables dentro del diagrama:

$$C = L_2 | C_1 = L_3 | b_1 = L_4$$

Se buscó una relación que incluya a theta 1:

$$-\phi_1 = -(90 - \theta_1) + \beta_1$$

$$\theta_1 = 90 - \beta_1 - \phi_1$$

Se procedió a resolver cada una de las variables, por medio de relaciones trigonométricas de un triángulo rectángulo, para obtener theta 1:

$$\tan \phi_1 = \frac{-y}{x - c} \Rightarrow \phi_1 = \text{atan2}(-y, x - L_2)$$

$$\tan \beta_1 = \frac{-b_1 * \sin(90 - \theta_2)}{c_1 + b_1 * \cos(90 - \theta_2)} \Rightarrow \beta_1 = \text{atan2}(-L_4 * \sin(90 - \theta_2), L_3 + L_4 * \cos(90 - \theta_2))$$

$$\Rightarrow \theta_1 = 90 - \text{atan2}(-L_4 * \sin(90 - \theta_2), L_3 + L_4 * \cos(90 - \theta_2)) - \text{atan2}(-y, x - L_2)$$

Se buscó una relación que incluya a theta 2:

$$180 = \alpha_1 + (90 - \theta_2)$$

Se manipuló la relación para poder igualarla a una de las identidades de la ley de cosenos:

$$\Rightarrow 180 - \alpha_1 = 90 - \theta_2 \Rightarrow \cos(180 - \alpha_1) = \cos(90 - \theta_2) \Rightarrow -\cos \alpha_1 = \sin \theta_2$$

Se procedió a utilizar la identidad de la ley de cosenos para obtener el seno de theta2, lo que resulte se llamara D:

$$\cos \alpha_1 = \frac{b_1^2 + c_1^2 - a_1^2}{2 * b_1 * c_1} = -\sin \theta_2$$

$$\sin \theta_2 = \frac{-b_1^2 - c_1^2 + a_1^2}{2 * b_1 * c_1}$$

$$a_1^2 = (x - L_2)^2 + y^2$$

$$\Rightarrow \sin \theta_2 = \frac{-L_4^2 - L_3^2 + (x - L_2)^2 + y^2}{2 * L_4 * L_3} = D$$

Usando la identidad trigonométrica fundamental se despejó para poder reemplazar el seno theta 2 en la misma, para posteriormente obtener la relación tangente:

$$\sin^2 \theta_2 + \cos^2 \theta_2 = 1$$

$$\cos \theta_2 = \pm \sqrt{1 - \sin^2 \theta_2} \Rightarrow \cos \theta_2 = \pm \sqrt{1 - D^2}$$

$$\frac{\sin \theta_2}{\cos \theta_2} = \frac{D}{\pm \sqrt{1 - D^2}} = \tan \theta_2$$

Por último, se encontró theta 2 utilizando atan2:

$$\Rightarrow \theta_2 = \text{atan2} \left(D, \pm \sqrt{1 - D^2} \right)$$

A continuación, se presentan ejemplos del funcionamiento de la cinemática del robot, realizados en MATLAB. Al encender el robot siempre buscará su posición de inicio, dictada por la ubicación de los sensores de fin de carrera, en la que sus articulaciones están en 0, sin embargo, para el objetivo de las simulaciones solo se tomarán en cuenta θ_1 y θ_2 , debido a que se comparará con los movimientos de la cinemática inversa. En la ilustración 56 se puede observar la posición de inicio del robot.

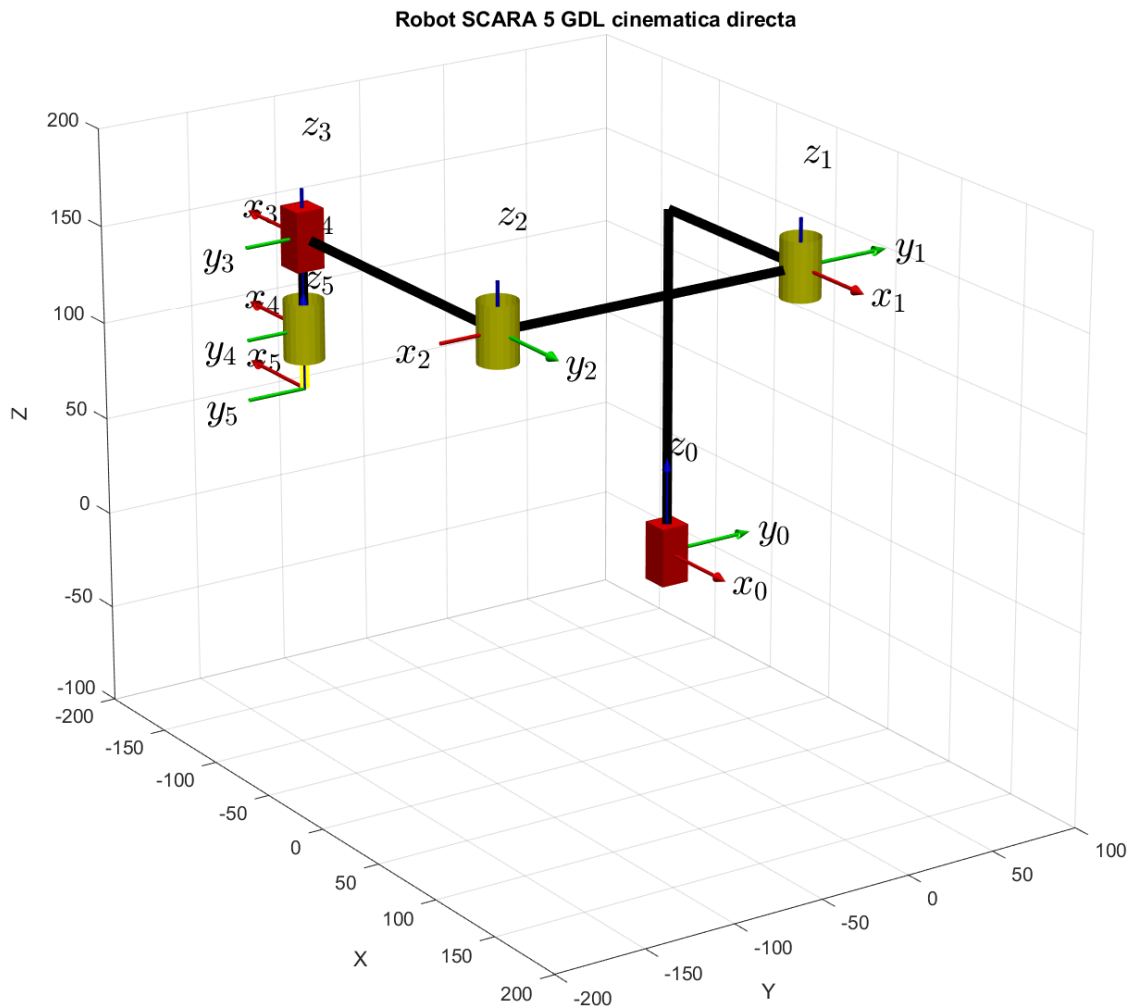


Ilustración 56-Diagrama de cinemática directa MATLAB

Fuente: Elaboración propia

Con la resolución de la cinemática directa Matlab comprueba que, en efecto, θ_1 y θ_2 se encuentran en 0. A través de la matriz H05, se obtienen las coordenadas X, Y, Z del efector final, siendo -61, -169, 100 respectivamente. En la ilustración 57 se pueden observar estos resultados.

```
theta1 =  
  
0  
  
theta2 =  
  
0  
  
H05 =  
  
-1.0000    0.0000    0 -61.0000  
-0.0000   -1.0000    0 -169.0000  
0          0      1.0000  100.0000  
0          0          0    1.0000
```

Ilustración 57-Resultados de cinemática directa Matlab posición de inicio

Fuente: Elaboración propia

Con las coordenadas X, Y obtenidas de la simulación de la cinemática directa, se procedió a ingresar estas al modelo de la cinemática inversa, este buscará llegar a ese punto, resolviendo para θ_1 y θ_2 (theta1i y theta2i). La coordenada Z es independiente, por lo que no afecta a las simulaciones. Como se puede observar a continuación en la Ilustración 58, se introdujeron coordenadas dando como resultado los mismos ángulos de la cinemática directa.

```
>> Dibujo_Inversa

theta1i =

    0

theta2i =

    0

H05 =

   -1.0000    0.0000         0   -61.0000
   -0.0000   -1.0000         0  -169.0000
         0         0    1.0000  100.0000
         0         0         0    1.0000
```

Ilustración 58-Resultados de cinemática inversa Matlab posición de inicio

Fuente: Elaboración propia

Como se puede observar en la Ilustración 59, el modelo cinemático inverso adoptó las mismas posiciones del modelo cinemático directo, corroborando su correcto funcionamiento.

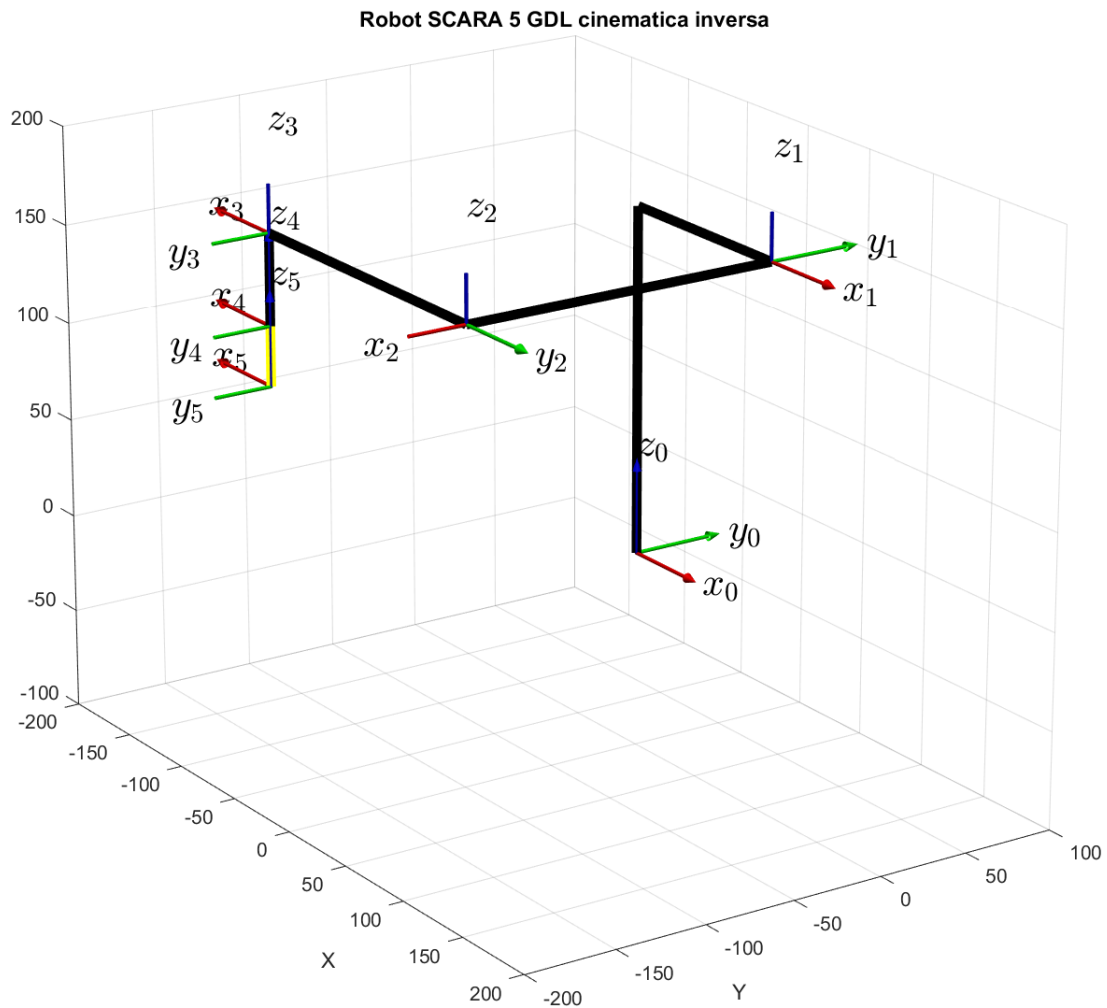


Ilustración 59-Diagrama de cinemática inversa MATLAB

Fuente: Elaboración propia

Como segundo ejemplo, en la cinemática directa introdujeron los valores de 125 grados a θ_1 y 35 grados a θ_2 . En la ilustración 60 se puede observar el diagrama con la nueva posición.

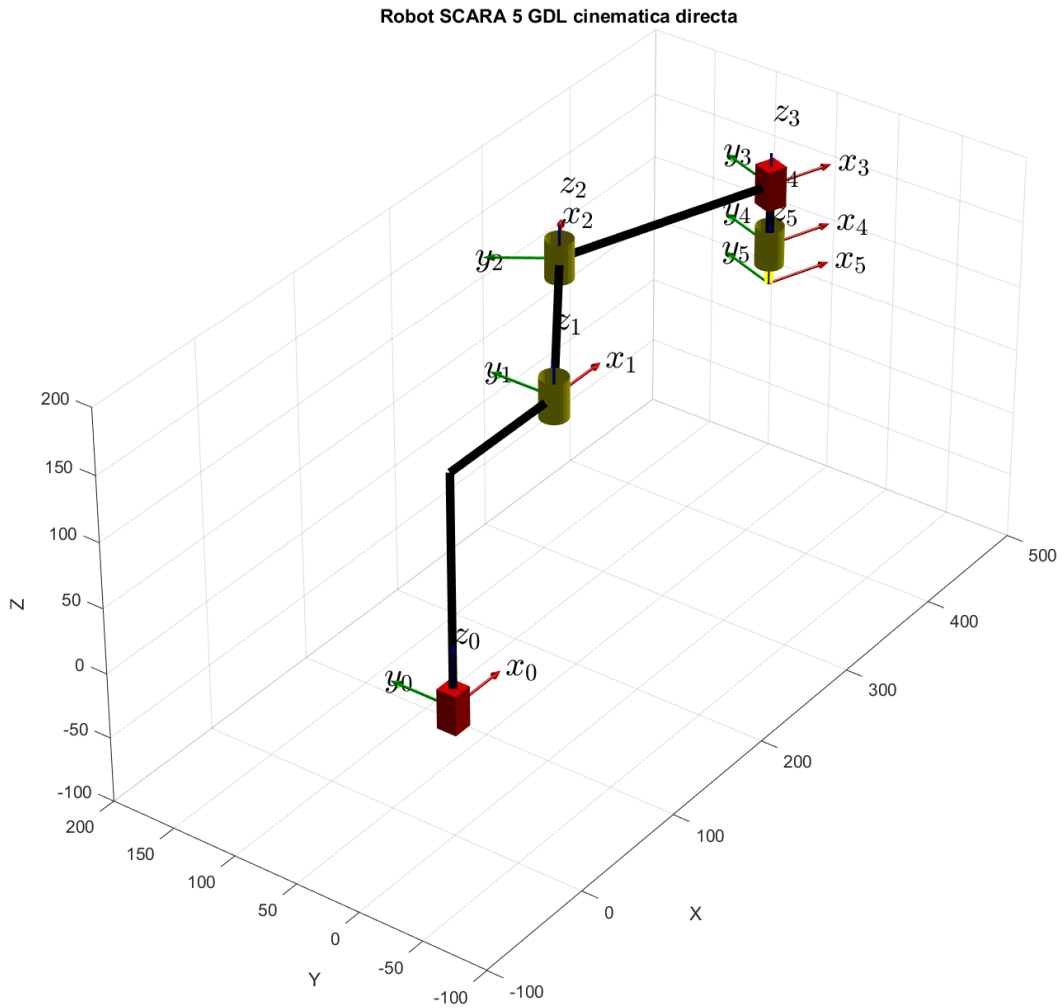


Ilustración 60-Diagrama de cinemática directa MATLAB ejemplo 2

A través de la matriz H_{05} , se obtienen las coordenadas X, Y, Z del efector final, siendo 405.2447, 39.1330, 100.0 respectivamente. En la ilustración 61 se pueden observar estos resultados.

```

theta1_grados =

    125.0000

theta2_grados =

    35

H05 =

    0.9397    0.3420         0  405.2447
   -0.3420    0.9397         0   39.1330
         0         0    1.0000  100.0000
         0         0         0    1.0000

```

Ilustración 61-Resultados de cinemática directa Matlab ejemplo 2

Fuente: Elaboración propia

Con las coordenadas X, Y obtenidas de la simulación de la cinemática directa, se procedió a ingresar estas al modelo de cinemática inversa, resolviendo para θ_1 y θ_2 (theta1i y theta2i). La coordenada Z es independiente, por lo que no afecta a las simulaciones. Como se puede observar a continuación en la ilustración 62, se introdujeron coordenadas dando como resultado los mismos ángulos de la cinemática directa.

```

>> Dibujo_Inversa

theta1i_grados =

    125.0000

theta2i_grados =

    35.0000

H05 =

    0.9397    0.3420         0  405.2447
   -0.3420    0.9397         0   39.1330
         0         0    1.0000  100.0000
         0         0         0    1.0000

```

Ilustración 62-Resultados de cinemática inversa Matlab ejemplo 2

Fuente: Elaboración propia

Como se puede observar en la ilustración 63, el modelo cinemático inverso adoptó las mismas posiciones del modelo cinemático directo, corroborando su correcto funcionamiento.

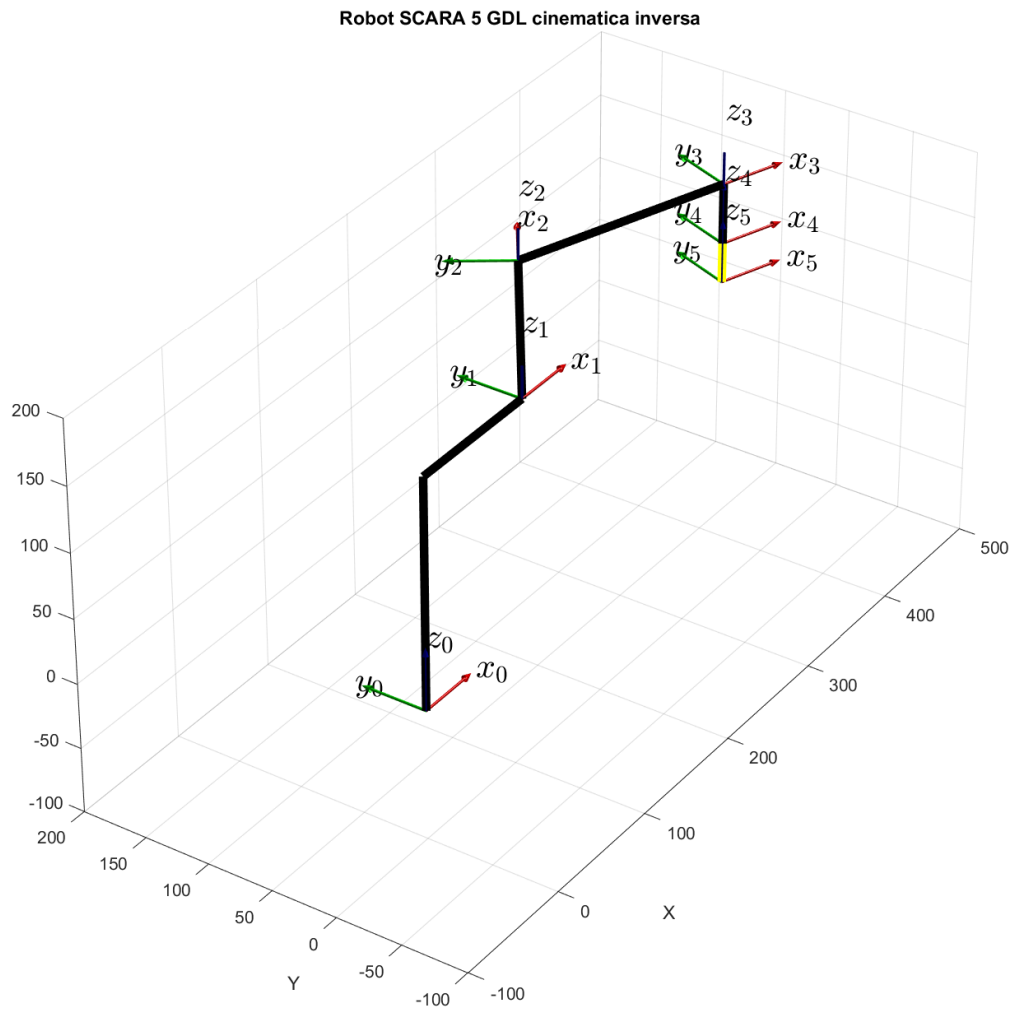


Ilustración 63-Diagrama de cinemática inversa MATLAB ejemplo 2

Fuente: Elaboración propia

5.1.1.7 Subsistema GUI

No hay resultados indirectos en este subsistema dentro de esta etapa.

5.1.2 Etapa 4: Integración de las partes

En esta sección se mostrarán los resultados de la etapa 4 de la metodología.

5.1.2.1 Subsistema estructural

Se realizó el ensamblaje de aproximadamente 70 piezas que componen el SCARA de 5 GDL estructuralmente en SolidWorks, el ensamblaje cuenta con una base de 870mm de altura hecha de metal, 3 eslabones con sus respectivas tapas, acoples para sus articulaciones, poleas y plataforma base todo esto hecho con PLA de color rojo y azul, además, de los diferentes componentes, como rodamientos, tornillos sinfín, motores paso a paso y varillas lisas. A continuación, se muestra un renderizado del ensamblaje:

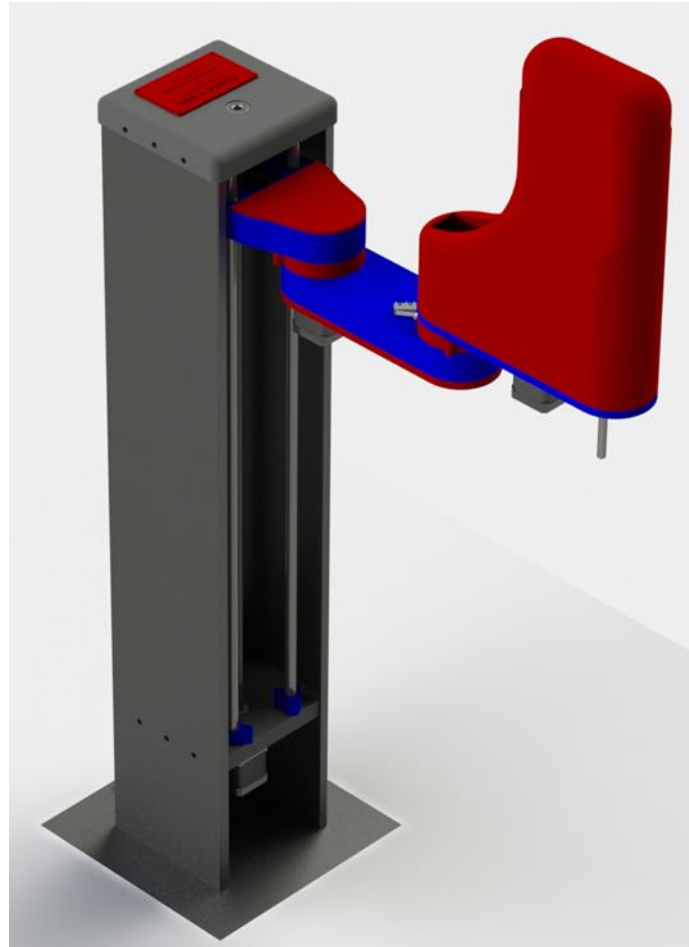


Ilustración 64-Renderizado de estructura completa en SolidWorks

Fuente: Elaboración propia

Debido a que no se puede ensamblar físicamente la estructura sin antes encajar el resto de los componentes que irán dentro de ella, se harán pruebas de movimiento hasta que el resto de los subsistemas y sistemas estén acoplados.

5.1.2.2 *Subsistema de transmisión*

Se ensamblaron los sistemas de transmisión y se realizaron pruebas con un motor en la entrada de los sistemas, para corroborar que funcionaban correctamente. Estas pruebas se pueden encontrar en los anexos 1 y 2.

5.1.2.3 *Subsistema de potencia*

Se realizó el esquemático de potencia para los circuitos de accionamiento y control. Este se puede encontrar en el anexo 3.

5.1.2.4 *Subsistema de accionamiento*

Se realizaron los esquemáticos de accionamiento y control al igual que la programación independiente para probar las RPM de los motores, al igual que pruebas de voltaje y corriente en los drivers para su calibración. Estos esquemáticos y pruebas se pueden encontrar en los anexos 4-9.

5.1.2.5 *Subsistema sensorial*

Se realizó el esquemático del sistema sensorial, y se realizaron las pruebas con programación independiente para corroborar que en efecto el sensor delimitaba correctamente los motores. Dicho esquemático y prueba se pueden encontrar en los anexos 10-12.

5.1.2.6 *Subsistema informático*

Para la integración de la resolución de la cinemática a la programación principal se hizo uso de una librería de álgebra lineal, a través de la cual se pudieron ingresar las diferentes matrices de transformación homogéneas, según el orden del algoritmo de Denavit – Hartenberg, se realizó una función para la cinemática directa y otra para la cinemática inversa. Esta integración se puede encontrar en los anexos 13-17. Para la programación principal, se hizo un diagrama de flujo que representa la lógica completa que tendrá que seguir el microcontrolador, esta se puede observar a continuación en la Ilustración 65.

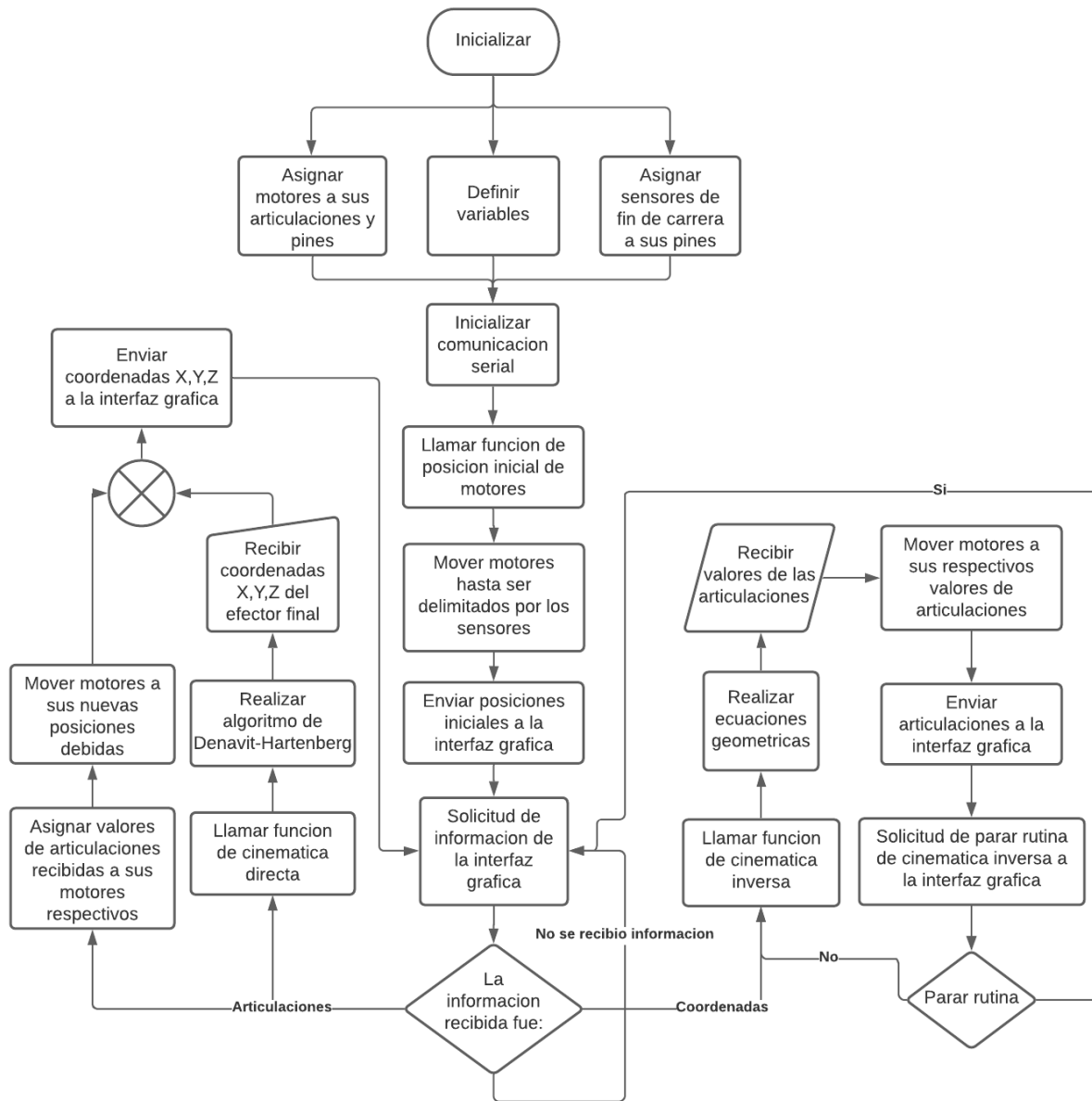


Ilustración 65-Diagrama de flujo programación principal

Fuente: Elaboración propia

5.1.2.7 *Subsistema GUI*

Se diseñó una GUI básica en la cual se realizó una prueba donde se ingresó un ángulo a un servo motor y se mandó por medio del módulo de comunicación serial al Arduino, para que este fuera capaz de mover el servo motor. Esta *GUI* se puede observar en el anexo 18.

5.1.3 Etapa 5: Integración de los subsistemas

En esta sección se mostrarán los resultados de la etapa 5 de la metodología.

5.1.3.1 *Sistema mecánico*

Para el sistema mecánico, se tuvieron que integrar los subsistemas de estructura y transmisiones. La estructura es capaz de sostenerse sola, se encontraron algunos problemas con el sistema mecánico, en el acople entre el eslabón L2 y L3 ocurre una falla que hace que el robot este inclinado 10mm en su efector final, con respecto a las medidas del ensamblaje en SolidWorks, a pesar de varios intentos para corregir esto dentro del tiempo estipulado, no se tuvo éxito, por lo cual no se imprimió la tapadera del eslabón 3 para evitar mayor inclinación del efector final, sin embargo, se realizaron pruebas de movimiento manuales llegando a la conclusión que esto no afecta el movimiento de las articulaciones del robot. A continuación, se puede observar el sistema mecánico ensamblado con los cables desconectados:

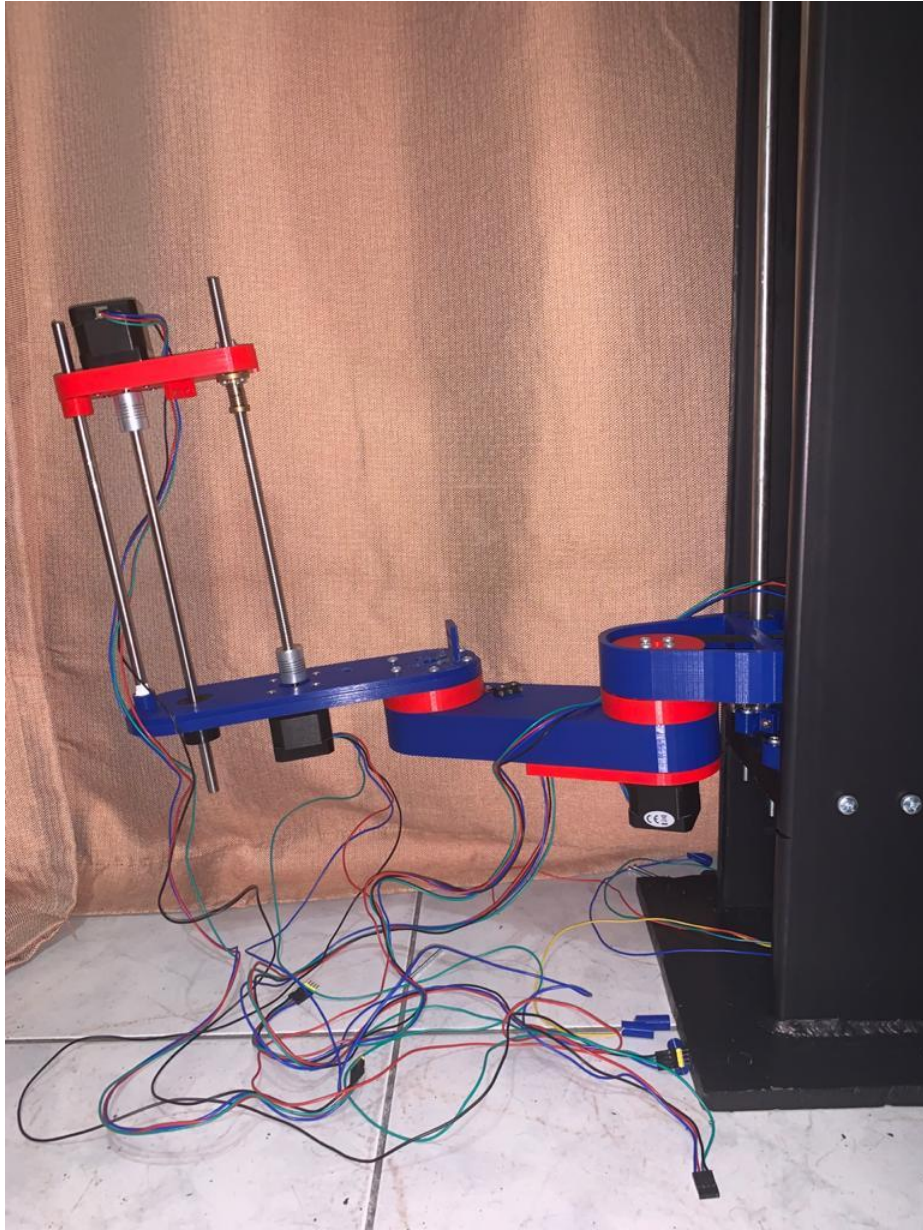


Ilustración 66-Sistema mecánico vista lateral

Fuente: Elaboración propia

A continuación, se puede observar el sistema mecánico ensamblado con los cables desconectados en su vista superior.



Ilustración 67-Sistema mecánico vista superior

Fuente: Elaboración propia

5.1.3.2 *Sistema electrónico*

Para la integración del sistema electrónico, se tiene, el esquemático y diseño del circuito impreso, que hace parte del sistema electrónico, este está compuesto por varios circuitos, el de accionamiento que es el encargado de la conexión de los motores paso a paso con sus respectivos

drivers y el del efector final; el circuito de control donde se encuentran las conexiones al microcontrolador; el circuito sensorial que se encarga de la conexión de los sensores de final de carrera y por último el circuito de potencia que es el encargado de la alimentación del resto de los circuitos.

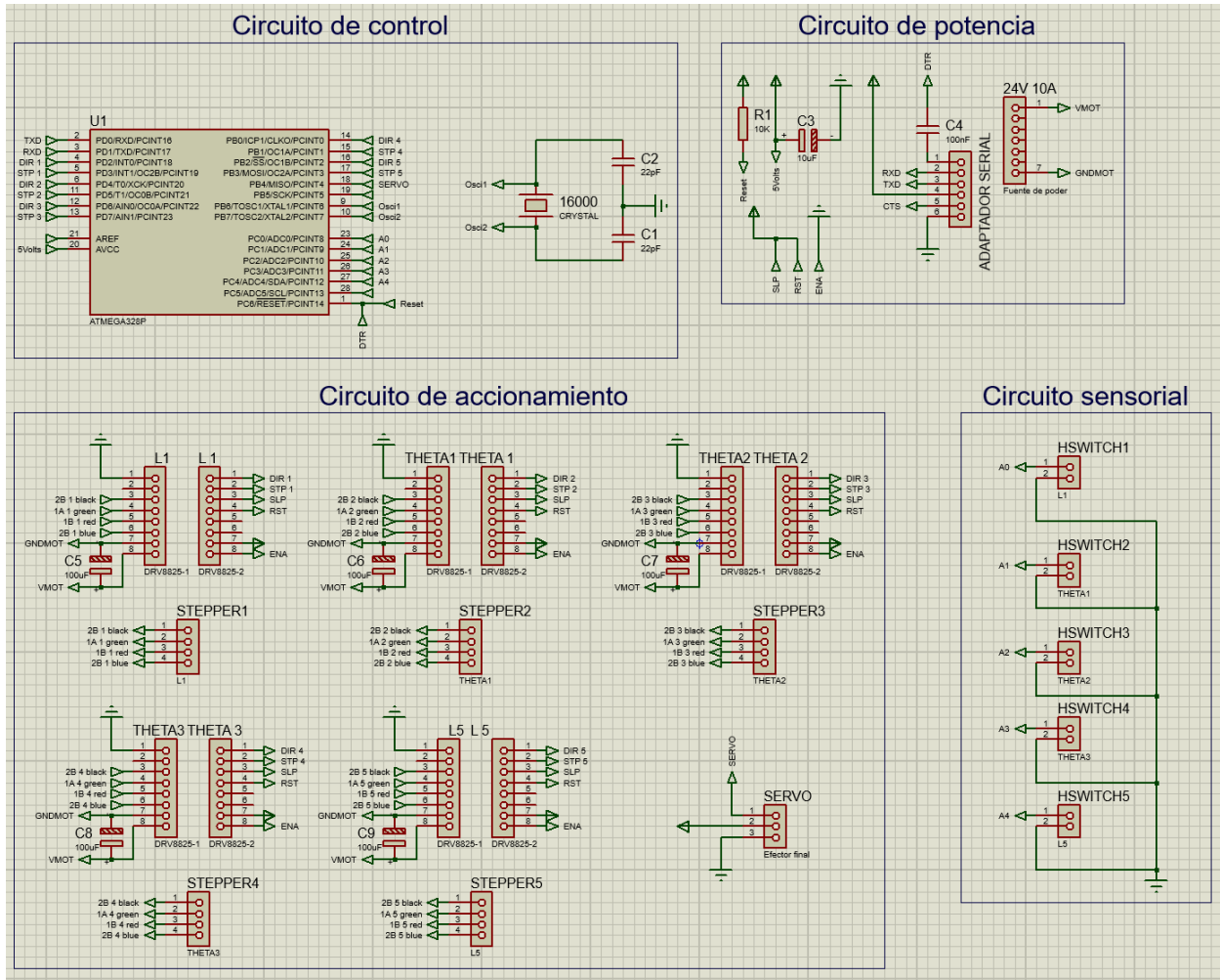


Ilustración 68-Esquemático completo de sistema electrónico

Fuente: Elaboración propia

A continuación, se puede observar la manufactura del circuito impreso desde el diseño de sus pistas:

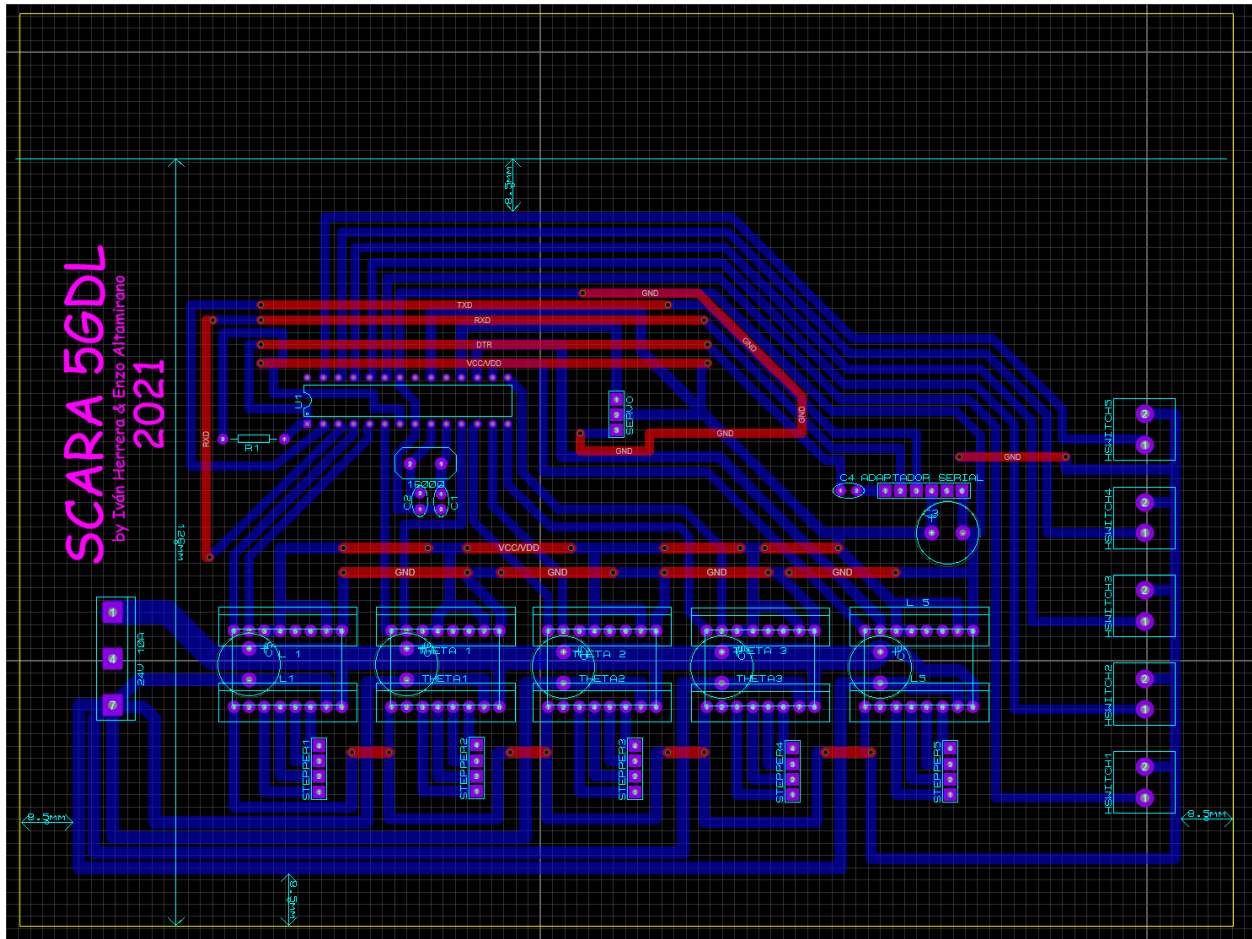


Ilustración 69-Diseño de la placa electrónica

Fuente: Elaboración propia

A continuación, se puede observar la manufactura del circuito impreso desde el finalizado de sus pistas a su maquinado:

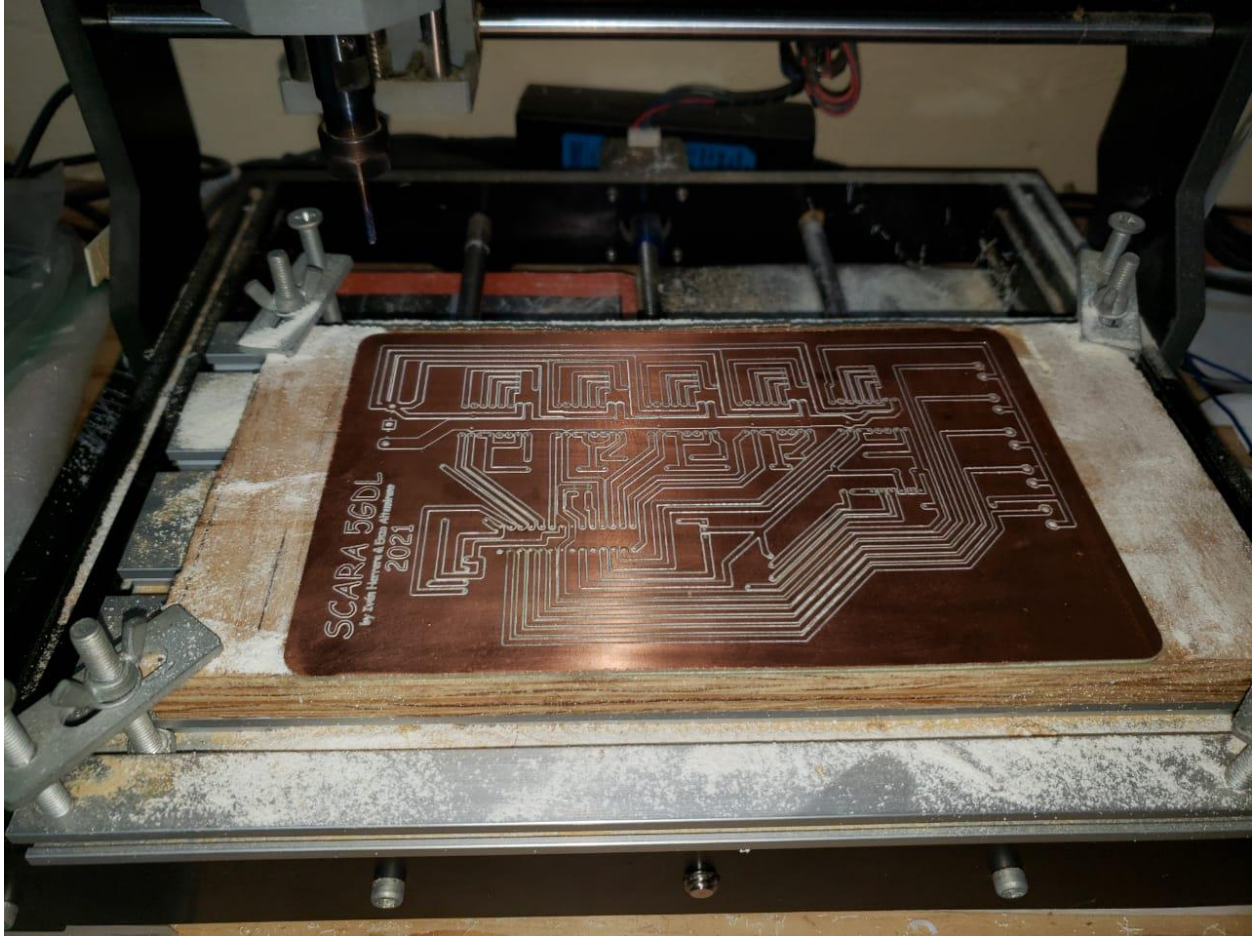


Ilustración 70-Placa electrónica manufacturada

Fuente: Elaboración propia

A continuación, se puede observar la manufactura del circuito impreso en su última etapa, la cual es el soldado de sus componentes:

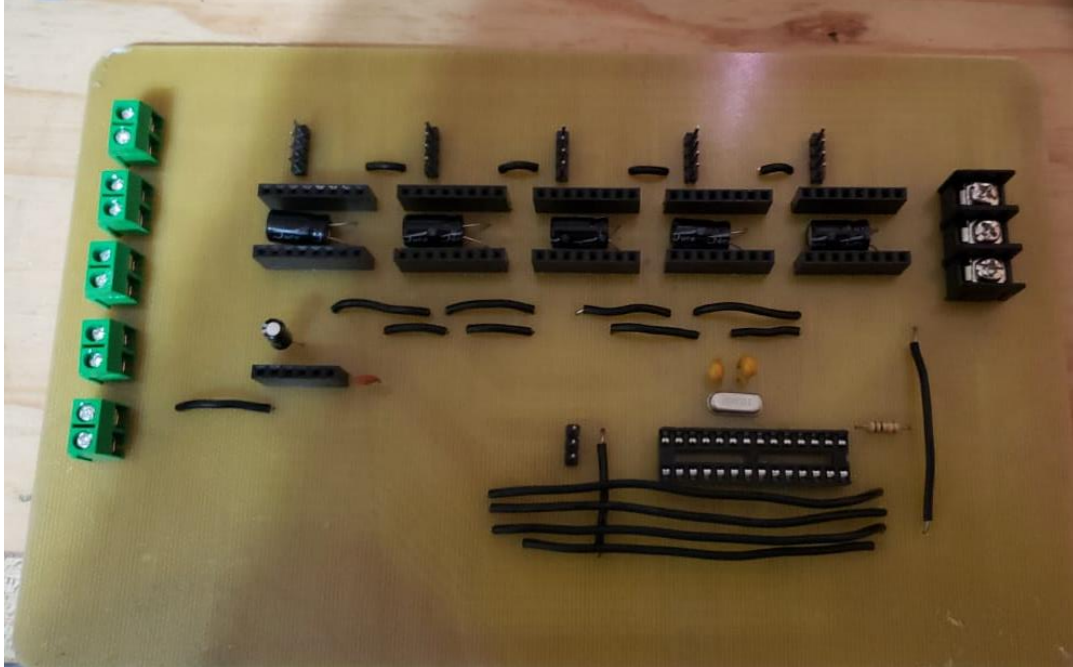


Ilustración 71-Placa electrónica soldada

Fuente: Elaboración propia

5.1.3.3 *Sistema de control*

Para la integración del sistema de control, se integraron los subsistemas de informática y *GUI*. A continuación, se puede observar la interfaz gráfica, cuenta con control para cinemática directa mediante incremento y decremento de las articulaciones, Theta 1, Theta 2, Theta 3, L1, L5 y su efector final, también cuenta con visualización de coordenadas XYZ, botones para guardar posiciones deseadas y otro para borrarlas, estos correspondientes a la cinemática inversa, junto con el botón de "correr secuencia de posiciones" y "parar programa" que sirven para la ejecución de la misma. Cuenta con deslizadores para la velocidad y aceleración en porcentajes y por último con un botón para reiniciar el microcontrolador.



Ilustración 72-Interfaz gráfica

Fuente: Elaboración propia

En el subsistema informático anteriormente se mostró la programación principal, con eso en cuenta, se obtuvieron resultados exitosos al ser capaces de realizar la comunicación entre la *GUI* y el microcontrolador, logrando mandar instrucciones de cinemática directa e inversa desde la *GUI* hacia el microcontrolador y obteniendo la respuesta debida. A continuación, se mostrarán tres ejemplos de su funcionamiento:

Lo primero que se realiza es la posición de inicio en el microcontrolador, por lo cual este al terminar, envía hacia la *GUI* las posiciones en 0 de todas las articulaciones. Estas articulaciones son procesadas con la cinemática directa obteniendo las coordenadas X, Y, Z del efector final, pudiéndolo visualizar en la *GUI*. A continuación, se puede observar en la Ilustración 73, como la *GUI* despliega todas las articulaciones en 0 y las posiciones X, Y, Z correspondientes.



Ilustración 73-GUI posición de inicio

Fuente: Elaboración propia

En la ilustración 74, se puede observar que en efecto la GUI recibió las posiciones de inicio, ya que cada vez que esto sucede, se recibe una "i" de inicio que es uno de los casos programados dentro de ella.

```

Output
Show output from: Debug
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
'GUI_FINAL_ROBOT_SCARA.exe' (CLR v4.0.30319: GUI_FINAL_ROBOT_SCARA.exe):
i-61.00,-169.00,160.00,0.00,0.00,0.00,0.00,0.00,180.00
  
```

Ilustración 74-Recibimiento exitoso de posición de inicio

Fuente: Elaboración propia

Como segundo ejemplo, se hizo uso del control de la cinemática directa por parte de la *GUI*. Con ella se envió hacia el microcontrolador, un movimiento de +50 a la articulación L1, posteriormente estas se procesaron por medio de la cinemática directa obteniendo las coordenadas X, Y, Z del efector final. Como se puede observar en la ilustración 75, se ingresó +50 a L1 y la coordenada en Z que se recibió cambió a 210, siendo lo correspondiente a ese cambio en L1.



Ilustración 75-GUI cinemática directa

Fuente: Elaboración propia

En la ilustración 76, se puede observar que en efecto la *GUI* recibió las coordenadas X, Y, Z ya que cada vez que esto sucede, se recibe una "c" de coordenadas que es uno de los casos programados dentro de ella. Cabe destacar, que el cambio fue solamente en L1, por lo cual el valor de X, Y no cambian.

```
Output
Show output from: Debug
The thread 0x73b0 has exited with code 0 (0x0).
The thread 0x4e94 has exited with code 0 (0x0).
The thread 0x3cb8 has exited with code 0 (0x0).
c-61.00,-169.00,210.00
```

Ilustración 76-Recibimiento exitoso de coordenadas de cinemática directa

Fuente: Elaboración propia

Como último ejemplo, se hará uso del control de cinemática inversa por parte de la *GUI*. Con ella se envió hacia el microcontrolador, una coordenada a alcanzar de 200 en Z se recibió en el microcontrolador, procesándose en la cinemática inversa, obteniendo las articulaciones.

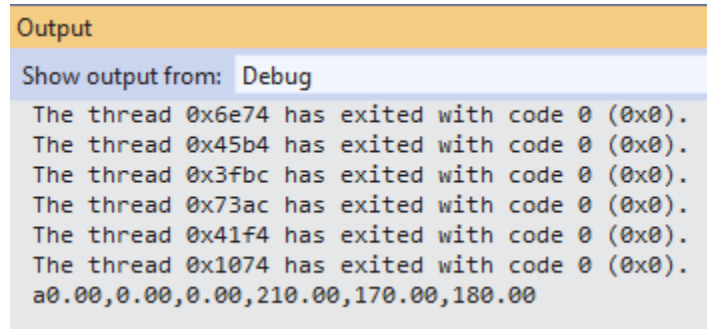
Como se puede observar en la ilustración 77, se recibieron valores de L1=210 y L5=170 en las articulaciones para la GUI, siendo lo correspondiente para una posición de Z=200.



Ilustración 77-GUI cinemática inversa

Fuente: Elaboración propia

En la ilustración 78, se puede observar que en efecto la *GUI* recibió las articulaciones ya que cada vez que esto sucede, se recibe una "a" de articulaciones que es uno de los casos programados dentro de ella. Cabe destacar, que el cambio fue solamente en Z, por lo tanto, el valor de las demás articulaciones no cambia.



```
Output
Show output from: Debug
The thread 0x6e74 has exited with code 0 (0x0).
The thread 0x45b4 has exited with code 0 (0x0).
The thread 0x3fbc has exited with code 0 (0x0).
The thread 0x73ac has exited with code 0 (0x0).
The thread 0x41f4 has exited with code 0 (0x0).
The thread 0x1074 has exited with code 0 (0x0).
a0.00,0.00,0.00,210.00,170.00,180.00
```

Ilustración 78-Recibimiento exitoso de articulaciones de cinemática inversa

Fuente: Elaboración propia

5.1.4 Etapa 6: Integración de los sistemas

En esta última etapa, se integraron los tres sistemas dando como resultado el SCARA con 5 GDL como se puede observar en la Ilustración 79. En el anexo 19 se muestran fragmentos del proceso de construcción.

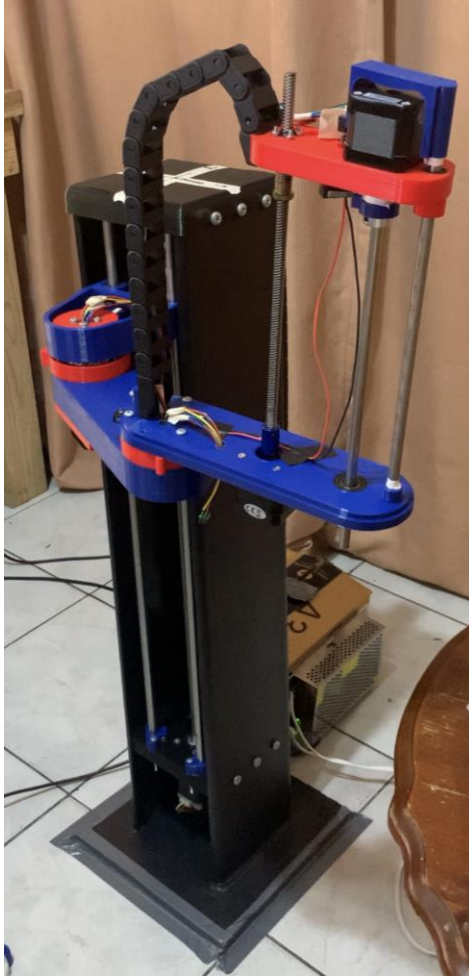


Ilustración 79-Robot SCARA con 5 GDL

Fuente: Elaboración propia

Se realizaron pruebas de punto a punto para corroborar su correcto funcionamiento como se puede observar en las ilustraciones X y X. A continuación, en la ilustración 80 se puede observar como el robot alcanzó el primer punto identificado por una tuerca.



Ilustración 80-Prueba punto a punto posición uno

Fuente: Elaboración propia

A continuación, en la ilustración 81 se puede observar como el robot alcanzó el segundo punto identificado por una tuerca.

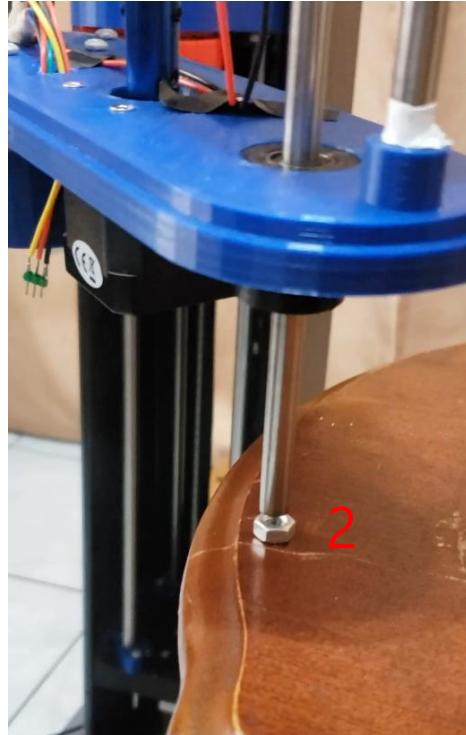


Ilustración 81-Prueba punto a punto posición dos

Fuente: Elaboración propia

5.2 RESULTADOS DIRECTOS

Los resultados directos validarán o negarán las ventajas del prototipo sobre un *SCARA* de montaje tradicional, previamente propuestas en el planteamiento del problema.

5.2.1 Incremento de área de trabajo

Como primer resultado directo validando las ventajas propuestas en la definición del problema, se tiene el incremento del área de trabajo con respecto al de un *SCARA* de montaje tradicional, para los propósitos de esta investigación se comparará el prototipo realizado con el Adept Cobra 450 de la marca Omron, sin embargo, solamente se tendrá en cuenta el alcance de los movimientos en el eje *z*, y su nueva área de trabajo, no la velocidad, precisión ni capacidad de carga ya que esto es solo un prototipo.

Como se puede observar en las Ilustración 82, el prototipo alcanza 700[mm] verticalmente lo cual le permitirá funcionar en aplicaciones que requieran de cambios de alturas significativos, el

alcance horizontal no es relevante ya que existen diferentes modelos de SCARAs de montaje tradicional que tienen alcances horizontales de hasta 1100[mm].

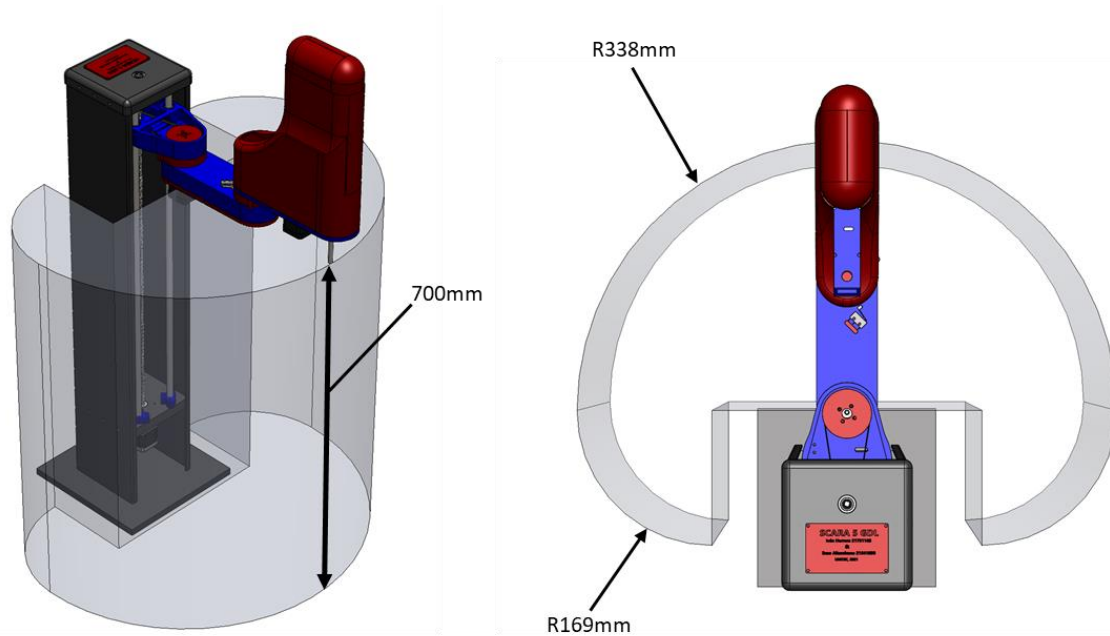


Ilustración 82-Área de trabajo de robot SCARA con 5 GDL

Fuente: Elaboración propia

Mientras que el Adept Cobra 450 solo es capaz de alcanzar 200[mm] verticalmente. A continuación, se puede observar su área de trabajo en la ilustración 83.

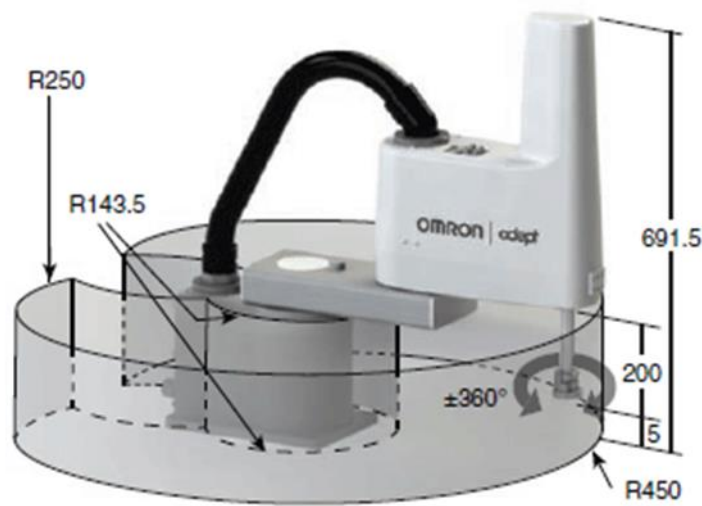


Ilustración 83-Área de trabajo de robot Adept Cobra 450 de Omron

Fuente: (Omron, 2018)

5.2.2 Desempeño en aplicaciones donde previamente no hubiera podido

Para las finalidades de esta investigación se probarán las siguientes aplicaciones:

- Empacado de objetos en cajas con una altura que un SCARA tradicional no puede alcanzar.
- Empacado de objetos debajo del nivel de una banda o mesa y Pick and Place en bandas o mesas de diferentes alturas.

Sin embargo, no se descartan más posibles aplicaciones que alguna fabrica le pueda dar. Cabe destacar que los ejemplos mostrados fueron realizados con la cinemática inversa del robot.

5.2.2.1 *Empacado en cajas con altura*

En este primer caso, se tiene el empacado de objetos en cajas altas. Para el ejemplo se ideó un empacado de botellas improvisado la caja utilizada es de 520[mm] de altura, el prototipo es capaz de dejarla al fondo de la caja, a pesar de ser este el ejemplo mostrado, se pueden empacar otro tipo de objetos donde la aplicación sea más adecuada. Además, las cajas pueden variar de altura según sea el caso. En la ilustración 84 se puede observar la botella en el punto uno que simula una banda.



Ilustración 84-Empacado en cajas con altura posición uno

Fuente: Elaboración propia

A continuación, en la ilustración 85, se puede observar como el prototipo es capaz de dejar la botella en el punto dos, que es el fondo de la caja, simulando el empaqueo de la misma, demostrando así, que el robot es capaz de empaacar objetos dentro de cajas con gran altura.



Ilustración 85-Empacado en cajas con altura posición dos

Fuente: Elaboración propia

A continuación, en la ilustración 86 realizada en el simulador de Ace 4.0 se puede observar como el Cobra 450 no es capaz de empacar objetos en cajas de altura debido a que colisiona con la misma.

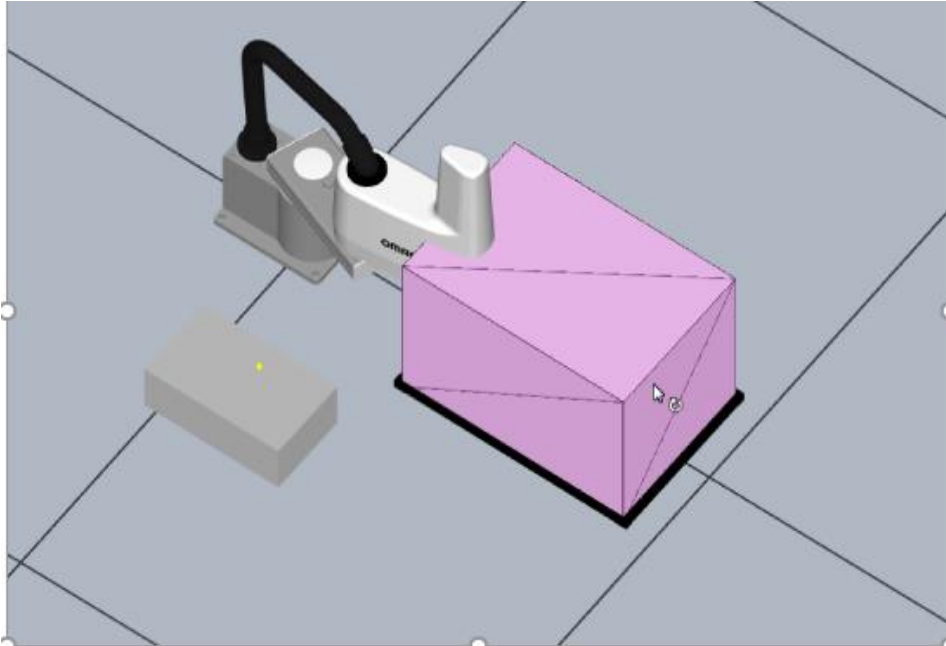


Ilustración 86-Simulación en Ace empacado en cajas con altura

Fuente: Elaboración propia

5.2.2.2 *Pick and Place en diferentes alturas*

En este segundo caso, se tiene *Pick and Place* en bandas, mesas, o lugares en diferentes alturas. Para el ejemplo fueron utilizadas cajas, que imitan los posibles casos de *Pick and Place* mencionados, la caja de color negro pequeña tiene una altura de 125[mm] y la caja grande de color café tiene una altura de 390[mm]. Con motivos demostrativos fue realizado un efector final para que el robot fuera capaz de recoger objetos, sin embargo, esto no está documentado en la investigación ya que no forma parte de la misma, además, que se pueden acoplar diferentes tipos de efector final.

En la ilustración 87, se puede observar el efector final en el punto uno, que simula la banda de menor altura.



Ilustración 87-*Pick and Place* en diferentes alturas posición uno

Fuente: Elaboración propia

En la ilustración 88, se puede observar el efector final en el punto dos, que simula el punto intermedio entre las dos bandas para evitar algún tipo de choque.

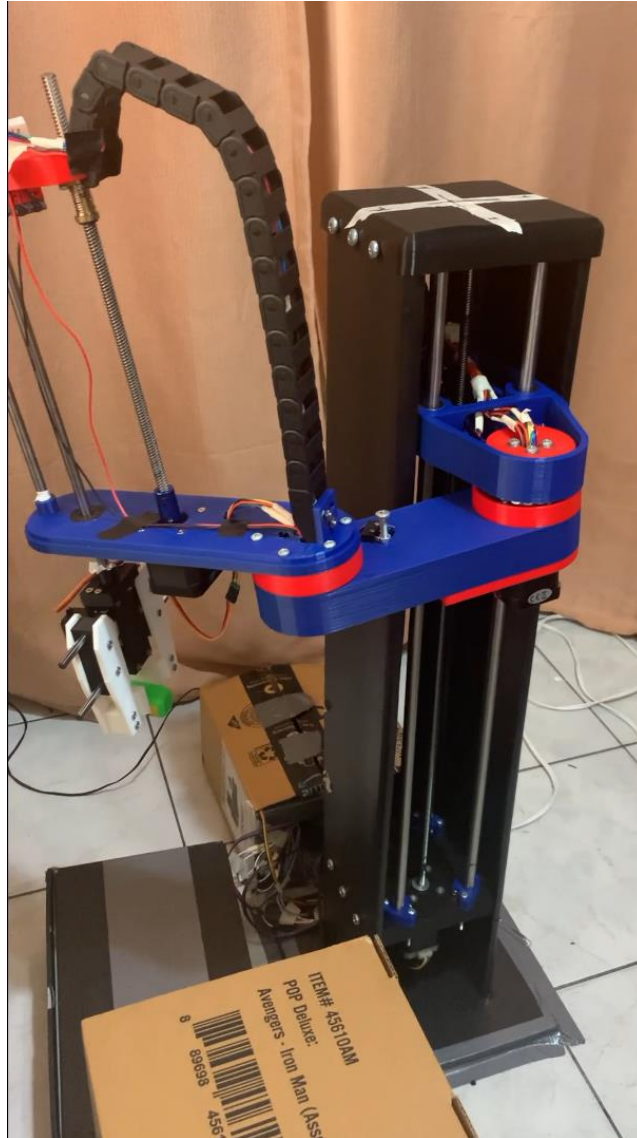


Ilustración 88-*Pick and Place* en diferentes alturas posición intermedia

Fuente: Elaboración propia

En la ilustración 89, se puede observar el efector final en el punto tres, que simula la banda con mayor altura, demostrando así, que el robot es capaz de hacer *Pick and Place* en puntos con alturas significativamente diferentes.

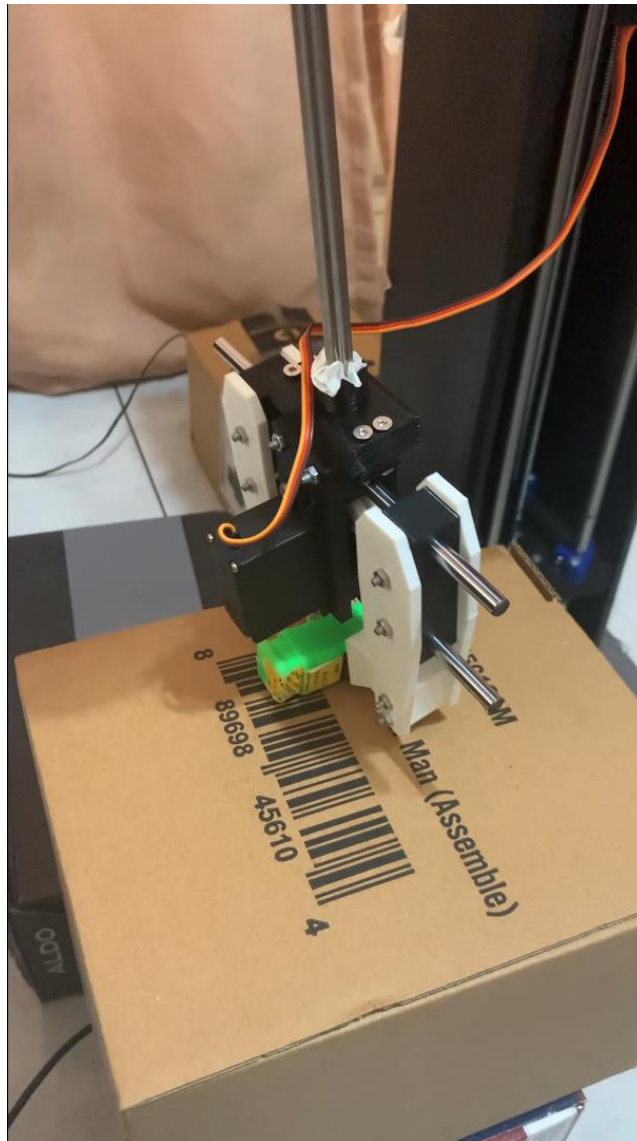


Ilustración 89-*Pick and Place* en diferentes alturas posición dos

Fuente: Elaboración propia

En la ilustración 90, realizada en el simulador de Ace 4.0 se puede observar como el Cobra 450 no es capaz de realizar *Pick and Place* en puntos con alturas significativamente diferentes debido a que colisiona con la misma. Las cajas utilizadas en la simulación fueron de las mismas dimensiones que las reales.

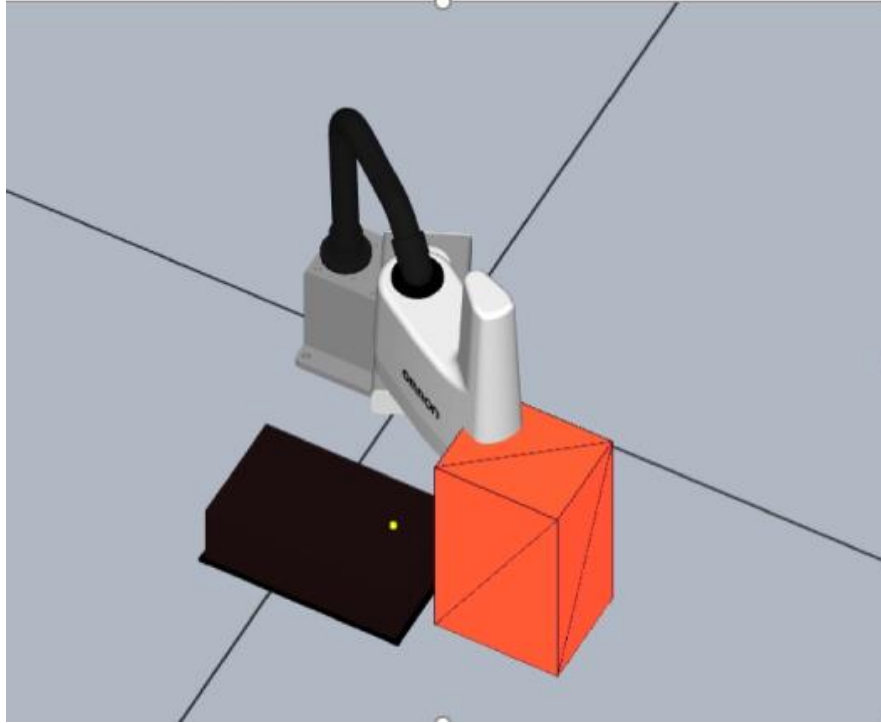


Ilustración 90-Simulación en Ace *Pick and Place* en diferentes alturas

Fuente: Elaboración propia

5.2.2.3 *Instalación versátil y económica, Facilidad de adaptación a cambios de lugar o aplicación*

Como tercer y último resultado directo, se tiene la validación de una instalación versátil y económica, junto con la facilidad de adaptación a cambios de lugar u aplicación. Para poder validar la facilidad de adaptación a cambios de lugar o aplicación, se desarrollaron dos ejemplos: En el primero, es una secuencia de Pick and Place en puntos con una altura de 130[mm] y el segundo es una secuencia en una altura de 510[mm], validando así, que el prototipo es capaz de cambiar de lugar o aplicación dentro de una misma fabrica sin tener que recurrir a inversión adicional en infraestructura, como bases de metal o mesas, para poder posicionarlo verticalmente en el lugar deseado o ajustes al espacio de trabajo. Al mismo tiempo, se demuestra que en efecto su instalación seria versátil y económica al no tener que incurrir en gastos adicionales de infraestructura para la misma, ya que los SCARAs de montaje tradicional operan a la misma altura de su instalación, en cambio, el prototipo puede ajustar su altura y trabajar en una posición fija.

En la ilustración 91, se puede observar como el robot ubica el objeto en la posición uno dentro del espacio de trabajo a 130[mm] de altura.

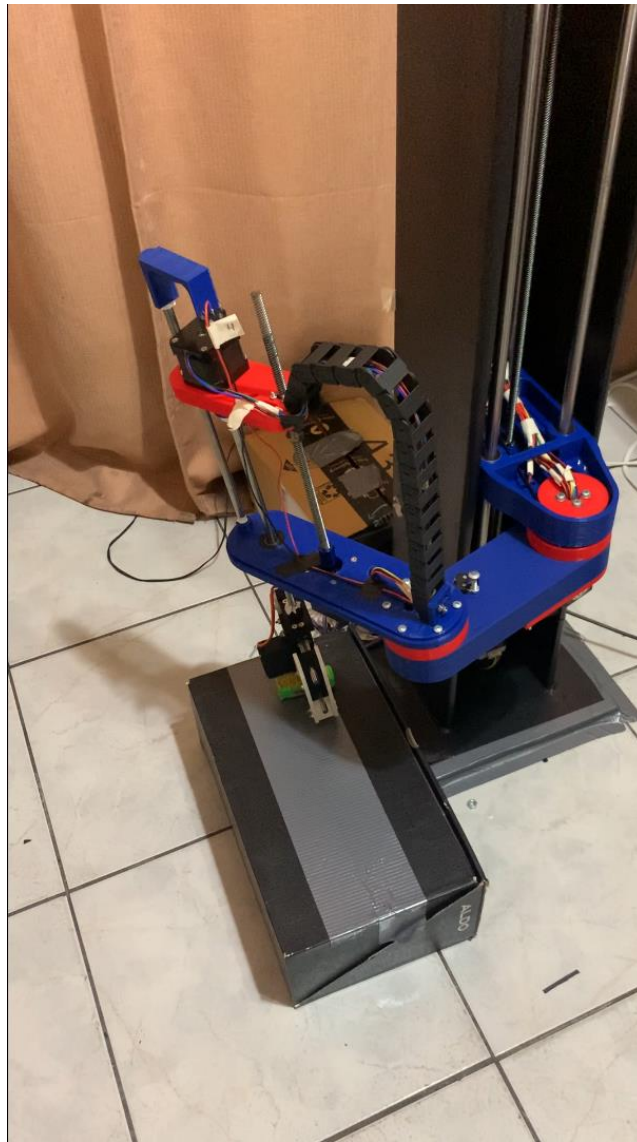


Ilustración 91-Espacio de trabajo bajo posición uno

Fuente: Elaboración propia

En la ilustración 92, se puede observar como el robot ubica el objeto en la posición dos dentro del espacio de trabajo a 130[mm] de altura.

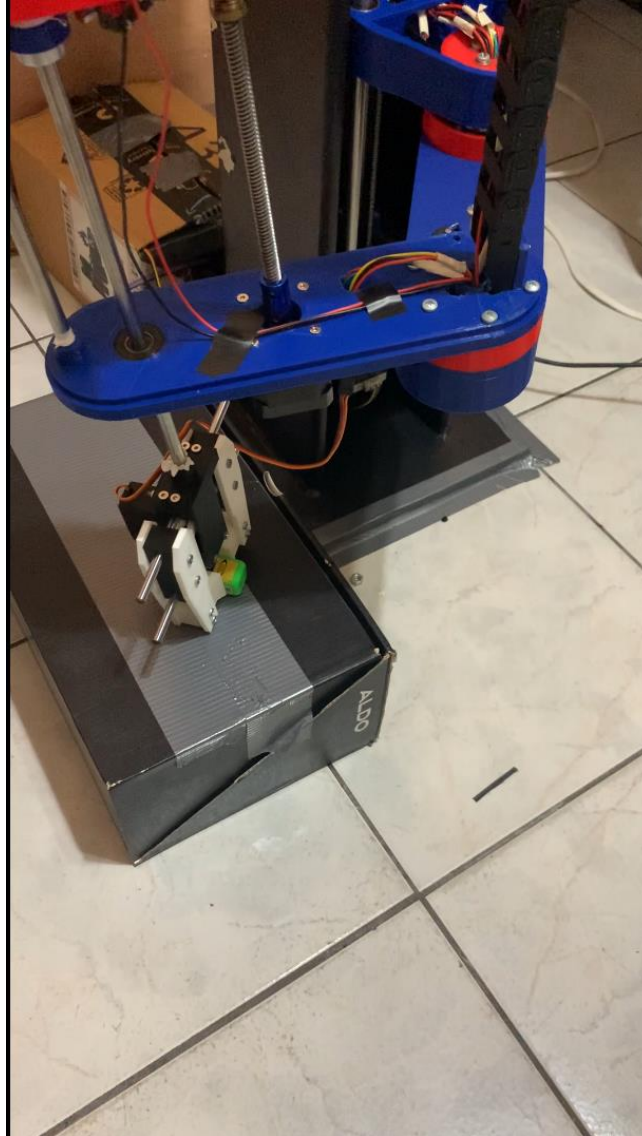


Ilustración 92-Espacio de trabajo bajo posición dos

Fuente: Elaboración propia

En la ilustración 93, se puede observar como el robot ubica el objeto en la posición uno dentro del espacio de trabajo a 510[mm] de altura.

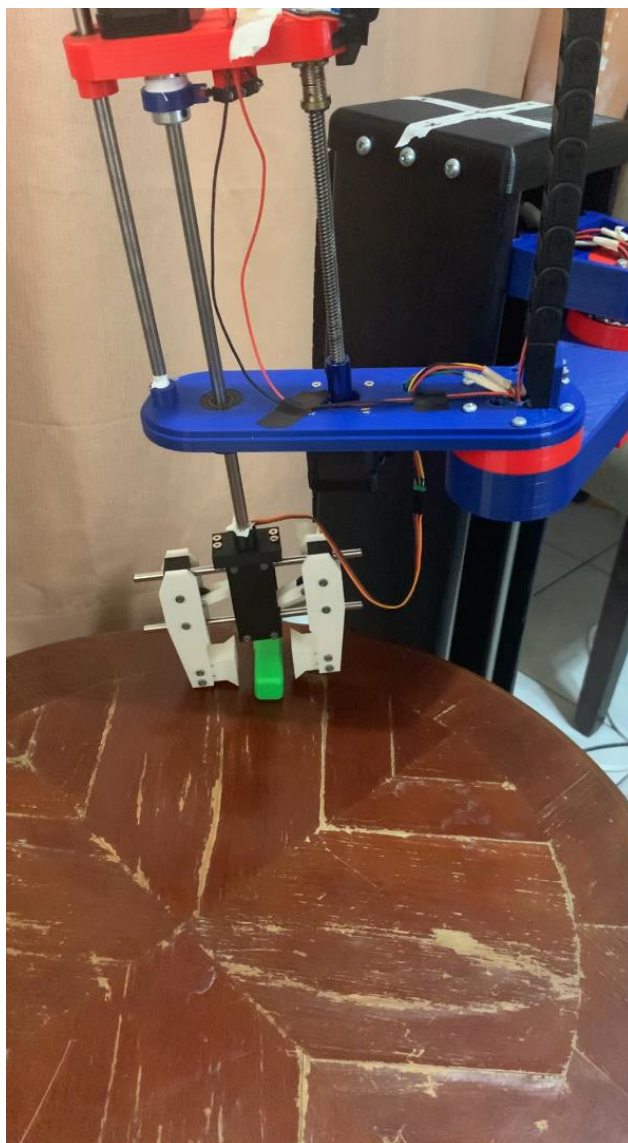


Ilustración 93-Espacio de trabajo alto posición uno

Fuente: Elaboración propia

En la ilustración 94, se puede observar como el robot ubica el objeto en la posición dos dentro del espacio de trabajo a 510[mm] de altura.

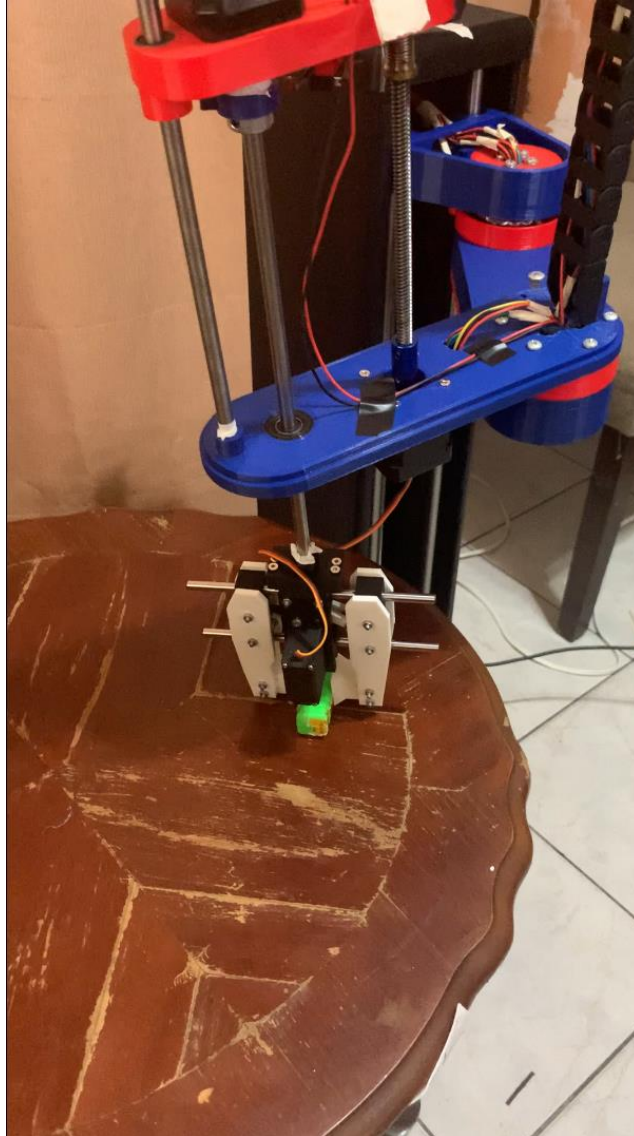


Ilustración 94-Espacio de trabajo alto posición dos

Fuente: Elaboración propia

5.3 COSTOS DEL PROTOTIPO FINAL

A continuación, se muestra el costo total del prototipo contando todos los componentes utilizados en la estructura, los precios son tomando en cuenta que se posee una maquina CNC e impresora 3D.

Elementos	Costo	Porcentaje
Costo total del prototipo	\$ 890.70	100.00%
Estructura	\$ 492.30	55.27%
Banda dentada	\$ 17.00	1.91%
Platina de hierro	\$ 32.70	3.67%
Tubo Estructural	\$ 118.40	13.29%
Soldadura y pintura	\$ 14.30	1.61%
Material de impresion 3d(PLA)	\$ 44.00	4.94%
Tornillos y materiales varios	\$ 59.90	6.73%
Rodamientos	\$ 61.00	6.85%
Varillas lisas	\$ 62.00	6.96%
Acoples	\$ 15.00	1.68%
Tornillos sin fin	\$ 35.00	3.93%
Cadena de cables	\$ 23.00	2.58%
Otros	\$ 10.00	1.12%
Electrónica	\$ 303.60	34.09%
Motores	\$ 65.00	7.30%
Sensores de fin de carrera	\$ 7.00	0.79%
Cables	\$ 16.00	1.80%
Baquelita virgen	\$ 10.60	1.19%
Microcontrolador	\$ 49.90	5.60%
Drivers	\$ 69.40	7.79%
Fuente de poder	\$ 21.00	2.36%
Componentes varios	\$ 64.70	7.26%
Costos de importación	\$ 94.80	10.64%

Tabla 5-Desglose de costos del prototipo final

Fuente: Elaboración propia

VI CONCLUSIONES

En este capítulo se exponen las conclusiones realizadas a partir de los resultados obtenidos.

1. Se acepta la hipótesis alternativa número uno, ya que el grado de libertad adicional le otorgó al SCARA todas las ventajas enlistadas sobre las limitaciones de uno de montaje tradicional.
2. Se logró resolver la cinemática directa del robot utilizando el algoritmo de Denavit-Hartenberg y la cinemática inversa utilizando una combinación del método geométrico para las articulaciones de rotación, y varias condiciones con valores predefinidos dentro de la programación para las infinitas soluciones de sus dos articulaciones de traslación vertical, obteniendo así, un control exitoso del SCARA a pesar de ser un diseño arbitrario.
3. Para el robot SCARA a nivel de prototipaje el uso de transmisiones de banda – polea representa una alternativa económica, sin embargo, es necesario el uso de motores de accionamiento directo para la implementación del robot SCARA con 5 GDL a nivel industrial.
4. El prototipo realizado puede llegar a una altura de 700[mm], sin embargo, es posible implementar un tornillo sin fin más largo u otro tipo de mecanismo de traslación vertical lo suficientemente preciso para aumentar su alcance vertical según sea requerido para la aplicación.

VII RECOMENDACIONES

En este capítulo se exponen las recomendaciones a partir de las conclusiones formuladas, se recomienda:

1. La implementación de robots SCARA con 5 GDL para las aplicaciones mostradas en esta investigación o para usos alternativos que se les pueda encontrar en la industria.
2. El uso del algoritmo de Denavit – Hartenberg para la resolución de cinemática directa en robots de diseño arbitrario y la combinación del método geométrico con varias condiciones y valores predefinidos en la programación para la resolución de la cinemática inversa de robots con infinitas soluciones en su eje Z.
3. Hacer uso de motores de accionamiento directo para las articulaciones del prototipo del robot SCARA con 5 GDL, superando así, las limitaciones e inconvenientes presentados con una transmisión de banda-polea.
4. Realizar un análisis de la efectividad de nuevos alcances verticales implementados, según la necesidad de la aplicación, para determinar su viabilidad.

VIII APLICABILIDAD

En este capítulo, se presentan varias recomendaciones para una posible continuación y mejora de la investigación.

1. Aumentar el diámetro en las articulaciones para distribuir mejor el peso que soportan los rodamientos de empuje.
2. Realizar un análisis dinámico del robot.
3. Utilizar motores de accionamiento directo para evitar el uso de sistemas de transmisión.
4. Implementar un sistema de visión artificial para detección de objetos con el fin de incrementar la automatización del robot.
5. Elegir algún metal como material de construcción para superar los problemas encontrados en el ensamblaje físico de la estructura.
6. Utilizar un microcontrolador más potente ya que con el actual en ocasiones se presentaban problemas por la cantidad de procesamiento.

BIBLIOGRAFÍA

1. A, B., AA, N. K., & TSB, S. (2015). 3 axis SCARA robot with universal gripper. *2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15)*, 1–5. <https://doi.org/10.1109/ITACT.2015.7492687>
2. Adamu, Y. A. (2019). *DEVELOPMENT OF A 6-DOF 3D PRINTED INDUSTRIAL ROBOT FOR TEACHING AND LEARNING*. <https://doi.org/10.13140/RG.2.2.33285.17124>
3. Angeles, J., Morozov, A., & Navarro, O. (2000). A novel manipulator architecture for the production of SCARA motions. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 3, 2370–2375. <https://doi.org/10.1109/ROBOT.2000.846381>
4. Bader, F., & Rahimifard, S. (2020). A methodology for the selection of industrial robots in food handling. *Innovative Food Science & Emerging Technologies*, 64, 102379. <https://doi.org/10.1016/j.ifset.2020.102379>
5. Bader, F., & Rahimifard, S. (2018). Challenges for Industrial Robot Applications in Food Manufacturing. *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, 1–8. <https://doi.org/10.1145/3284557.3284723>
6. Barrientos, A. (2007). *Fundamentos de robótica*.
7. Bekey, G., & Yuh, J. (2008). The Status of Robotics. *IEEE Robotics & Automation Magazine*, 15(1), 80–86. <https://doi.org/10.1109/M-RA.2007.907356>

8. Bhadru, B., Narendar, D., & Kiran, B. (2015). *Design and Fabrication of SCARA Robot with 5 Degree of Freedom*. 6(3), 5.
9. Budynas, R. G., Nisbett, J. K., & Shigley, J. E. (2015). *Shigley's mechanical engineering design* (Tenth edition). McGraw-Hill Education.
10. Cano, J. Á. M. (2018). *Estudio para la puesta en marcha de un robot SCARA AdeptThree-XL*. 62.
11. Cao, F., Liu, J., Zhou, C., Zhao, Y., Fu, Z., & Yan, W. (2015). A Novel 5-DOF welding robot based on SCARA. *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2016–2019. <https://doi.org/10.1109/ICIEA.2015.7334444>
12. Chaturvedi, R., Islam, A., & Sharma, K. (2020). Anticipated Investigation of a Cylindrical Robot ARM by Means of Compound Materials. *Clinical Medicine*, 7(4), 10.
13. Chen, D., Zeng, Z., Guan, Y., Zhu, H., & Zhang, T. (2020). SCARA Robots Developed with Modular Method. *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, 454–459. <https://doi.org/10.1109/ICMA49215.2020.9233716>
14. Diegel, O., Nordin, A., & Motte, D. (2019). *A Practical Guide to Design for Additive Manufacturing*. Springer Singapore. <https://doi.org/10.1007/978-981-13-8281-9>
15. Faria, C., Vilaca, J. L., Monteiro, S., Erlhagen, W., & Bicho, E. (2019). Automatic Denavit-Hartenberg Parameter Identification for Serial Manipulators. *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 610–617. <https://doi.org/10.1109/IECON.2019.8927455>

16. Flores, W., & Borjas, R. (2015). *Fabricación de una Prótesis Humana utilizando una impresora 3D en Honduras*. <https://doi.org/10.13140/RG.2.1.3121.7120>
17. Garcia, E., Jimenez, M. A., De Santos, P. G., & Armada, M. (2007). The evolution of robotics research. *IEEE Robotics & Automation Magazine*, 14(1), 90–103. <https://doi.org/10.1109/MRA.2007.339608>
18. Gasparetto, A., & Scalera, L. (2019). A Brief History of Industrial Robotics in the 20th Century. *Advances in Historical Studies*, 08(01), 24–35. <https://doi.org/10.4236/ahs.2019.81002>
19. Grau, A., Indri, M., Bello, L. L., & Sauter, T. (2017). Industrial robotics in factory automation: From the early stage to the Internet of Things. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 6159–6164. <https://doi.org/10.1109/IECON.2017.8217070>
20. Hoang, T., Vuong, T., & Pham, B. (2015). *Study and Development of Parallel Robots Based On 5-Bar Linkage*.
21. Hossain, A., Hossain, Md., & Jahan, M. (2014). Programming a SCARA Robot for a. *2014 ASEE Annual Conference & Exposition Proceedings*, 24.1011.1-24.1011.19. <https://doi.org/10.18260/1-2--22944>
22. Industrial Robots. (2017). In N. Pandrea, D. Popa, & N.-D. Stănescu, *Classical and Modern Approaches in the Theory of Mechanisms* (pp. 369–390). John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119221777.ch9>
23. International Federation of Robotics. (2020). *World Robotics Industrial Robots*

24. ISO. (2012). *ISO 8373:2012(en)*. Obtenido de <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
25. Jasveer, S., & Jianbin, X. (2018a). Comparison of Different Types of 3D Printing Technologies. *International Journal of Scientific and Research Publications (IJSRP)*, 8(4). <https://doi.org/10.29322/IJSRP.8.4.2018.p7602>
26. Jasveer, S., & Jianbin, X. (2018b). Comparison of Different Types of 3D Printing Technologies. *International Journal of Scientific and Research Publications (IJSRP)*, 8(4). <https://doi.org/10.29322/IJSRP.8.4.2018.p7602>
27. Jiang, X. (2018). *Design and Development of Robotic Arm for Cutting Trees*. 41.
28. Jørgensen, A. H., & Myers, B. A. (2008). User interface history. *Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI '08*, 2415. <https://doi.org/10.1145/1358628.1358696>
29. Karabegovic, E., Karabegovic, I., & Hadzalic, E. (2012). Industrial Robot Application Trend in World's Metal Industry. *Engineering Economics*, 23(4), 368–378. <https://doi.org/10.5755/j01.ee.23.4.2567>
30. Karabegović, I., Catovic, F., & Hodžić, D. (2008). *Industrial Robots Applications in the Process Industry*.
31. Karabegović, I., & Husak, E. (2013). Industrial Robot Application in Automobile Industry in Countries of Asia. *International Journal of Computer Science and Artificial Intelligence*, 95–101. <https://doi.org/10.5963/IJCSAI0303002>

32. Li, Z., Li, G., Sun, Y., Jiang, G., Kong, J., & Liu, H. (2017). Development of articulated robot trajectory planning. *International Journal of Computing Science and Mathematics*, 8, 52. <https://doi.org/10.1504/IJCSM.2017.083152>
33. Liu, J., Jian, F., & Liu, J. (2014). Four Degrees of Freedom SCARA Robot Modeling and Simulation. *2014 International Symposium on Computer, Consumer and Control*, 410–412. <https://doi.org/10.1109/IS3C.2014.113>
34. Luncanu, A., & Stan, G. (2019). Research on the influence of straightness deviations on positioning precision of Cartesian industrial robots. *IOP Conference Series: Materials Science and Engineering*, 591, 012081. <https://doi.org/10.1088/1757-899X/591/1/012081>
35. Masike, R., Garikayi, T., & Tigere, G. (2014). *Conceptual Design Framework for Developing a Cylindrical Robot Manipulator for Material Handling in Zimbabwean Small to Medium Enterprises: A Finite Element Approach*. 1(1), 6.
36. Mathia, K. (2010). *Robotics for Electronics Manufacturing: Principles and Applications in Cleanroom Automation*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511712173>
37. Metodij, S. K. I. (2014). *COMPARISON OF THE CHARACTERISTICS BETWEEN SERIAL AND PARALLEL ROBOTS*. 18.
38. Mittal, V. O. (Ed.). (1998). *Assistive technology and artificial intelligence: Applications in robotics, user interfaces, and natural language processing*. Springer.
39. Novas Peña, D. (2008). *Microcontroladores*. 31.

40. Olszewski, M. (2020). Modern Industrial Robotics. *Pomiary Automatyka Robotyka*, 24(1), 5–20. https://doi.org/10.14313/PAR_235/5
41. Prettyman, S. (2020). *Learn PHP 8: Using MySQL, JavaScript, CSS3, and HTML5*. Apress. <https://doi.org/10.1007/978-1-4842-6240-5>
42. Rajpar, A. H., Eladawi, A. E., Ali, I., & Bashir, M. (2021). Reconfigurable Articulated Robot Using Android Mobile Device. *Journal of Robotics*, 2021, 1–8. <https://doi.org/10.1155/2021/6695198>
43. Rojas, A. M., & Barbieri, G. (2019). A Low-Cost and Scaled Automation System for Education in Industrial Automation. *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 439–444. <https://doi.org/10.1109/ETFA.2019.8869535>
44. Sadiku, M. N. O., Musa, S. M., & Musa, O. M. (2018). Robots in the Chemical Industry. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3267294>
45. Saha, S. (2010). *Introducción a la robótica*. McGraw-Hill España. <https://elibro.net/ereader/elibrodemo/36580>
46. Sánchez, J. L. (2021). *Fundamentos de robótica*. Ecoe Ediciones. Obtenido de <https://elibro.net/es/lc/unitechn/titulos/175520>
47. Şen, M. A., Bakırcıoğlu, V., & Kalyoncu, M. (2017). *Modelling and PID Control of Scara Robot*.
48. Shariatee, M., Akbarzadeh, A., Mousavi, A., & Alimardani, S. (2014). Design of an economical SCARA robot for industrial applications. *2014 Second RSI/ISM International*

Conference on Robotics and Mechatronics (ICRoM), 534–539.
<https://doi.org/10.1109/ICRoM.2014.6990957>

49. Stefanov, A., Chavdarov, I., Nedanovski, D., & Boiadzhiev, G. (2019a). Dynamics and Control of a 3D Printed Walking Robot. *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–5.
<https://doi.org/10.23919/SOFTCOM.2019.8903684>

50. Stefanov, A., Chavdarov, I., Nedanovski, D., & Boiadzhiev, G. (2019b). Dynamics and Control of a 3D Printed Walking Robot. *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–5.
<https://doi.org/10.23919/SOFTCOM.2019.8903684>

51. Subero, A. (2021). *Programming Microcontrollers with Python: Experience the Power of Embedded Python*. Apress. <https://doi.org/10.1007/978-1-4842-7058-5>

52. Suri, S., Jain, A., Verma, N., & Prasertpoj, N. (2018). SCARA Industrial Automation Robot. *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, 173–177. <https://doi.org/10.1109/PEEIC.2018.8665440>

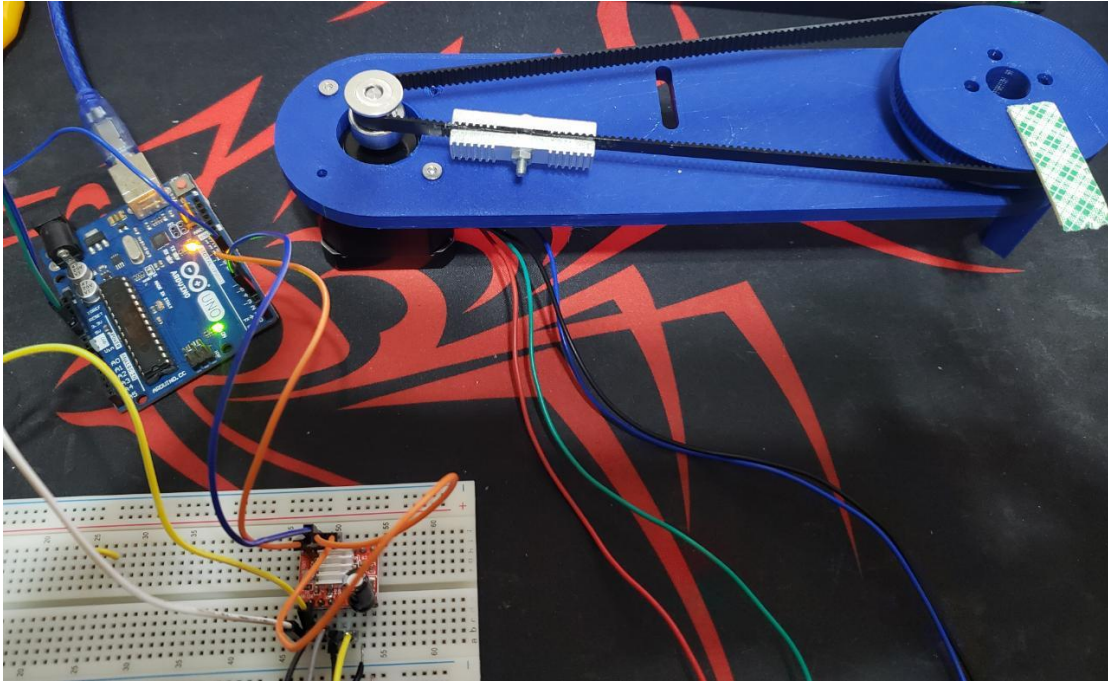
53. Takoordyal, K. (2020). User Interface. In K. Takoordyal, *Beginning Unity Android Game Development* (pp. 105–137). Apress. https://doi.org/10.1007/978-1-4842-6002-9_4

54. Trenzado Diepa, J. L. (2015). *Física*. Universidad de Las Palmas de Gran Canaria. Servicio de Publicaciones y Difusión Científica.
<http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3227504>

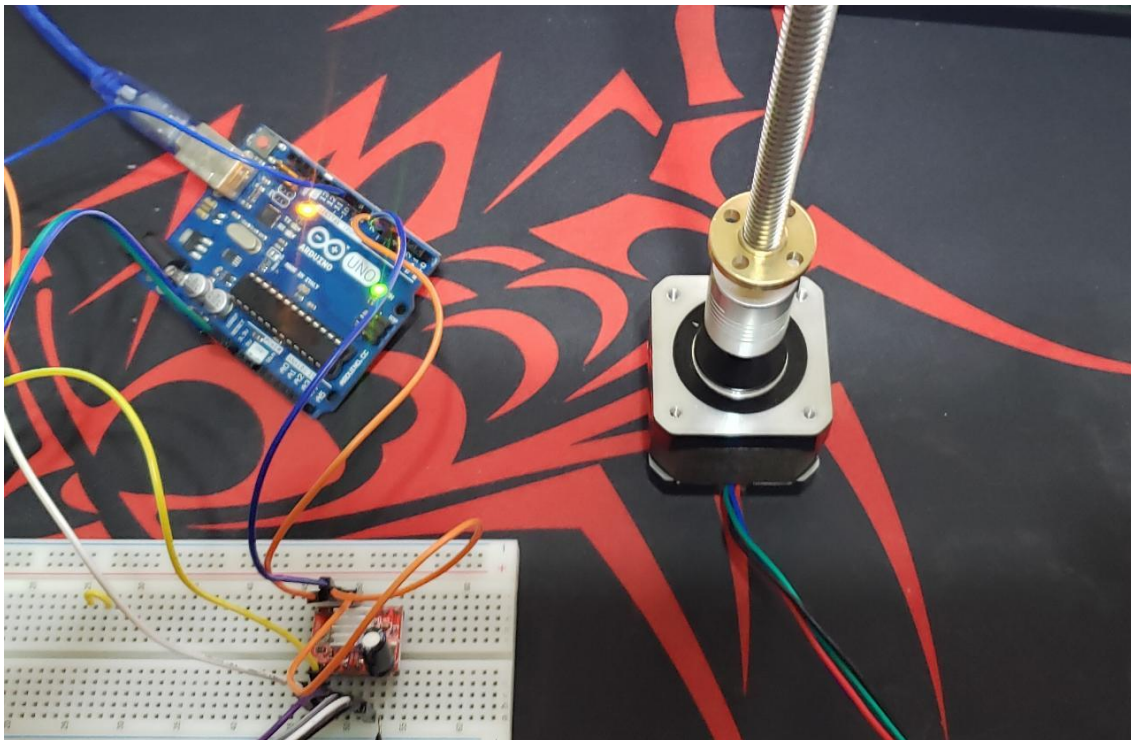
55. Vasic, V. S., & Lazarevic, M. P. (2008). Standard Industrial Guideline for Mechatronic Product Design . *Faculty of Mechanical Engineering*.
56. Wohlers, T. T. (2018). *WOHLERS REPORT: 3d printing and additive manufacturing state of the industry*. WOHLERS Associates.
57. Yamafuji, K. & The University of Electro-Communications 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan. (2019). Development of SCARA Robots. *Journal of Robotics and Mechatronics*, 31(1), 10–15. <https://doi.org/10.20965/jrm.2019.p0010>
58. Zamalloa, I., Kojcev, R., Hernández, A., Muguruza, I., Usategui, L., Bilbao, A., & Mayoral, V. (2017). Dissecting Robotics—Historical overview and future perspectives. *ArXiv:1704.08617 [Cs]*. <http://arxiv.org/abs/1704.08617>

ANEXOS

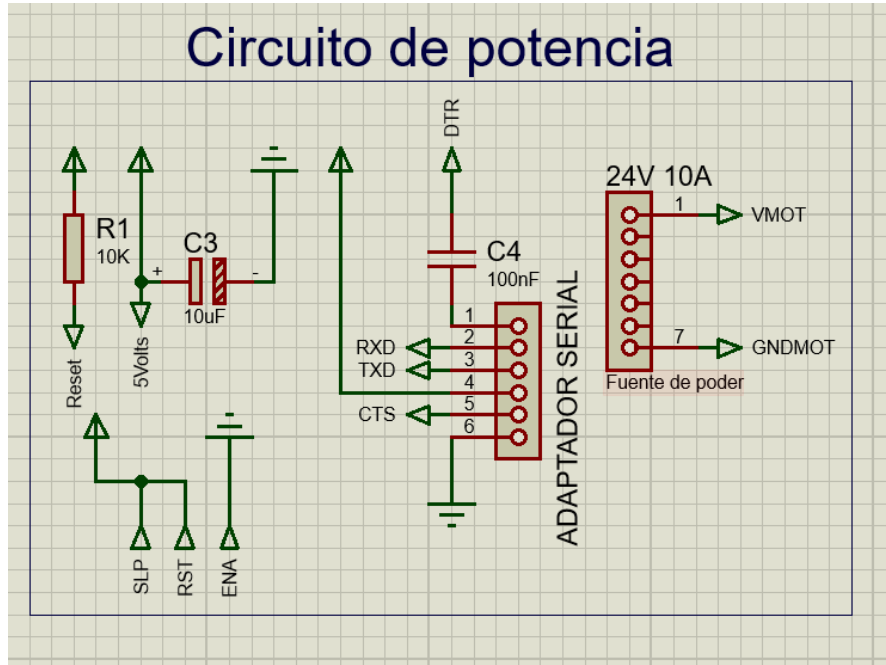
ANEXO 1-PRUEBA DE TRANSMISIÓN BANDA POLEA



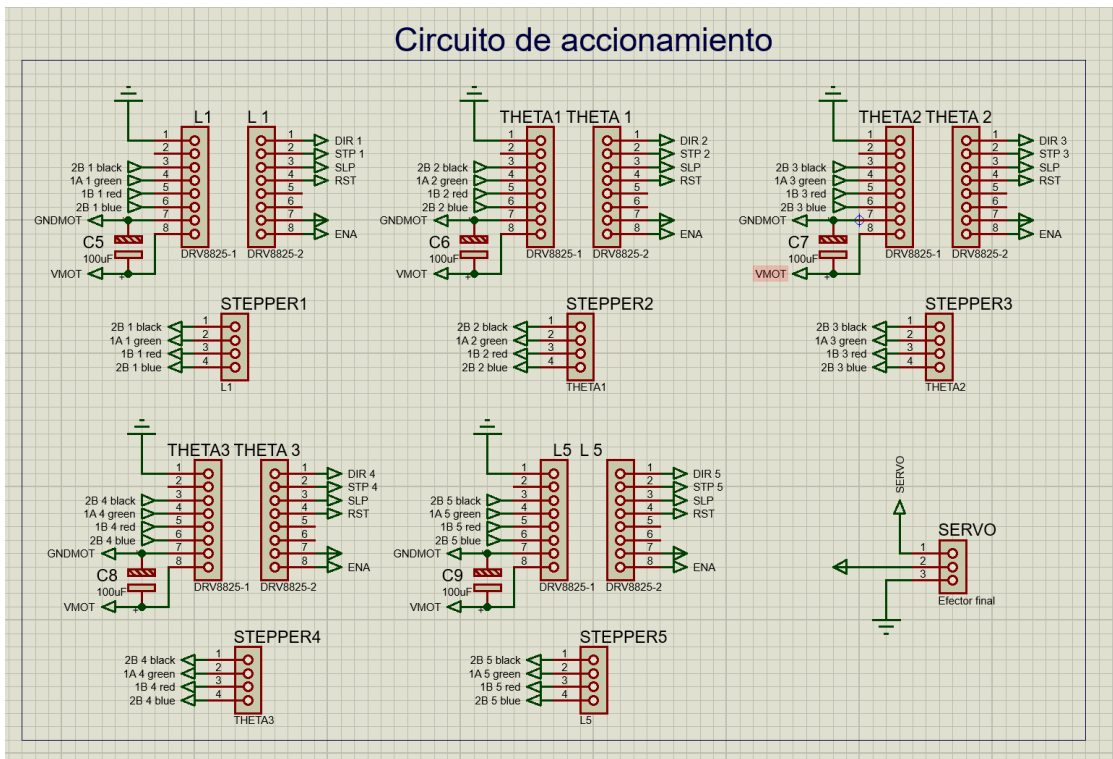
ANEXO 2-PRUEBA DE TRANSMISIÓN TORNILLO SIN FIN



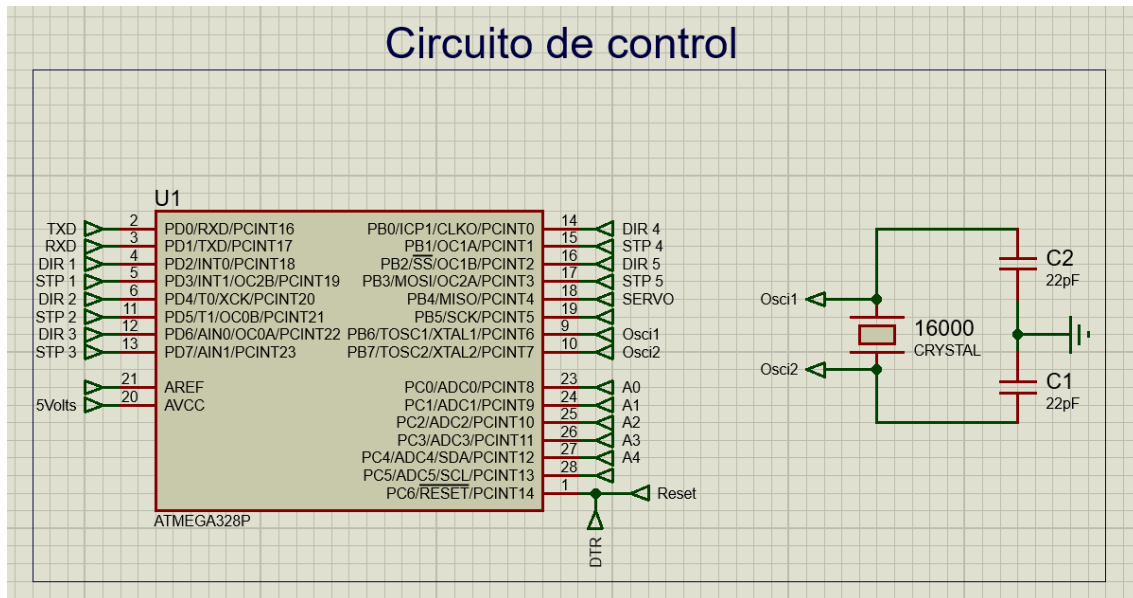
ANEXO 3-ESQUEMÁTICO DE SUBSISTEMA DE POTENCIA



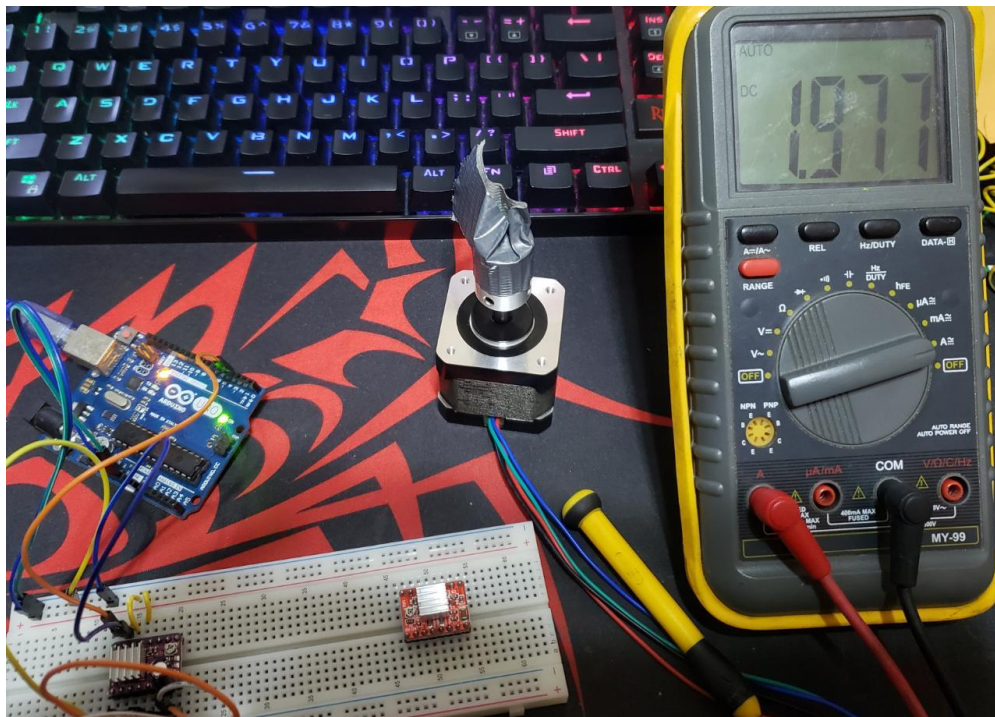
ANEXO 4-ESQUEMÁTICO DE SUBSISTEMA DE ACCIONAMIENTO



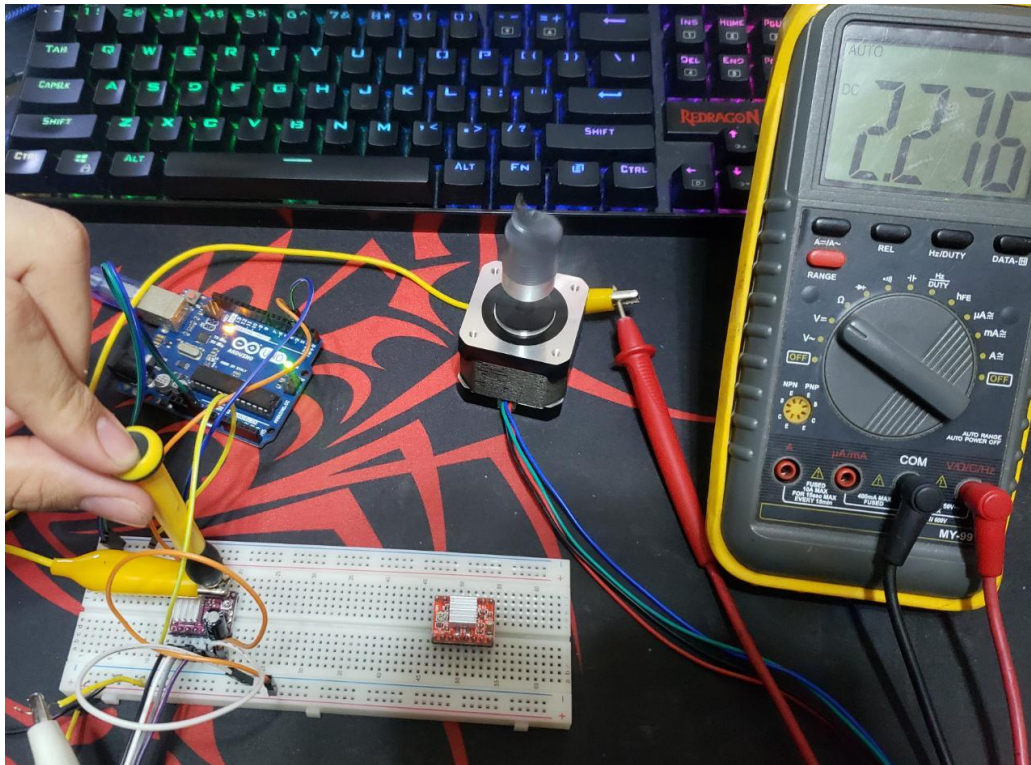
ANEXO 5-ESQUEMÁTICO CIRCUITO DE CONTROL DEL SUBSISTEMA DE ACCIONAMIENTO



ANEXO 6-PRUEBA DE CORRIENTE DE MOTORES DEL SISTEMA DE ACCIONAMIENTO



ANEXO 7-PRUEBA DE VOLTAJE DE MOTORES DEL SISTEMA DE ACCIONAMIENTO



ANEXO 8-PROGRAMACION DE RPM DE MOTORES DEL SISTEMA DE ACCIONAMIENTO

```
int stepPin = 2; //Set to whatever pin will trigger steps.
int dirPin = 3; //Set to the pin that tells the stepper driver if it's forward or backward.
int enablePin = 4; //
unsigned long rpm = 460; //Target max RPM of the stepper for this test.
unsigned long stepsPerRevolution = 200; //Enter the number of steps for your motor to make a full revolution.
bool forward = true; //Toggle to switch motor direction.
unsigned long accelTime = 2; //Second to accelerate the motor up to the max RPM (use 0 to just start the motor at full speed)
```

```

int stepPin = 2; //Set to whatever pin will trigger steps.
int dirPin = 3; //Set to the pin that tells the stepper driver if it's forward or backward.
int enablePin = 4; //
unsigned long rpm = 460; //Target max RPM of the stepper for this test.
unsigned long stepsPerRevolution = 200; //Enter the number of steps for your motor to make a full revolution.
bool forward = true; //Toggle to switch motor direction.
unsigned long accelTime = 2; //Second to accelerate the motor up to the max RPM (use 0 to just start the motor at full speed)

unsigned long firstStamp; //When the warmup began.
unsigned long lastStamp; //Stamp from the previous toggle.
unsigned long curStamp; //Stamp from the current iteration.
unsigned long rampTime = accelTime * 1000000; //How long to accelerate in microseconds.
bool warmup = true; //Track if we've made it through the initial acceleration stage or not.
bool pinOn = false;
unsigned long toggleTime;

void setup() {
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(stepPin, pinOn); //Set the step pin low.
  digitalWrite(dirPin, forward); //Set the direction.
  digitalWrite(enablePin, LOW); //Pull the enable pin low to turn on the stepper driver.
  firstStamp = lastStamp = micros(); //Note the start time.

  //Determine the number of microseconds per toggle of the step pin at the target RPM.
  toggleTime = 60000000; //Number of micros in one minute
  toggleTime /= rpm; //divid by number of times per minute to revolve
  toggleTime /= stepsPerRevolution; //divide by number of steps to complete a full revolution. Now we have the number of micros per step.
  toggleTime /= 2; //Cut in half so half the time it's on, half off.
}

void loop() {
  curStamp = micros(); //Get the timestamp for this loop.

  unsigned long toggleTarget = toggleTime;

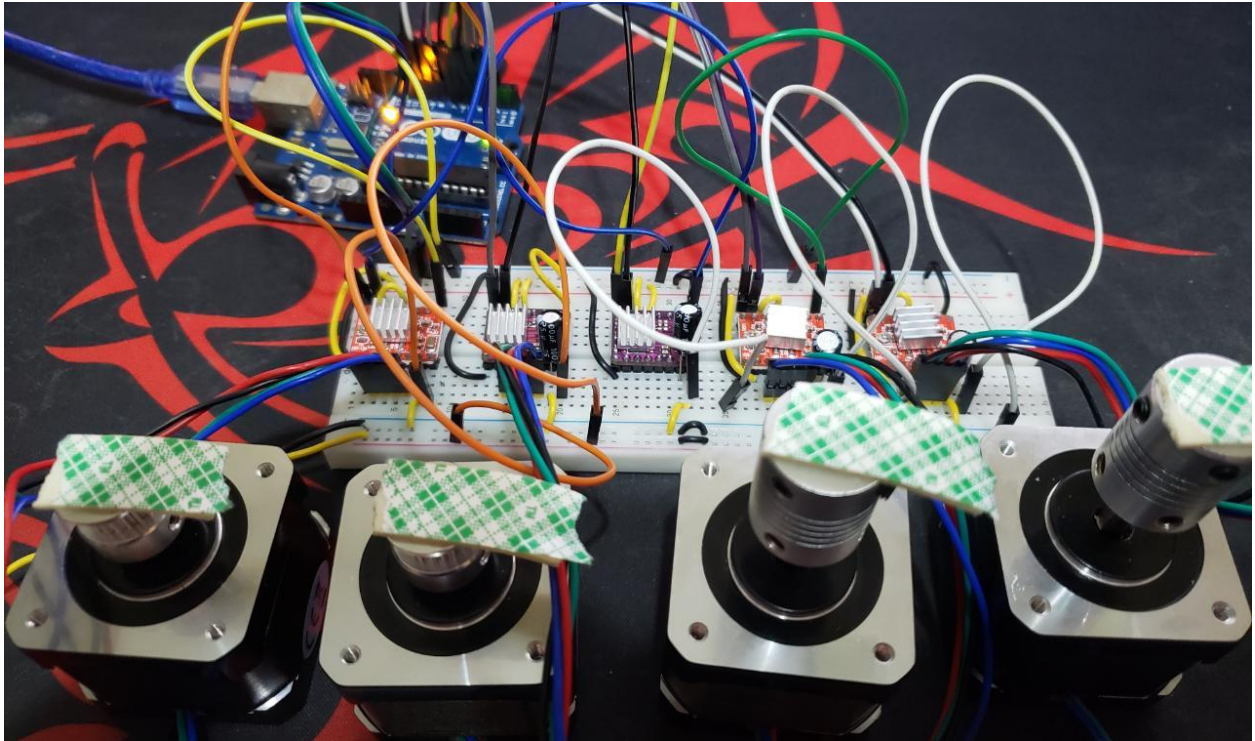
  if (warmup){ //Still accelerating at the outset, we'll increase the toggle time proportionally.
    if (curStamp - firstStamp > rampTime) warmup = false; //We've amde it to the end of the speed ramp.
    else {
      //Increase the toggle time proportional to how far along the acceleration ramp we are, longer at the start, no increase at the end.
      unsigned long tempToggle = (rampTime - (curStamp - firstStamp)) * toggleTime / rampTime; //less and less added time to the toggle.
      toggleTarget += tempToggle; //Add this extra toggle delay to the motor start.
    }
  }

  unsigned long elapsed = curStamp - lastStamp;
  if (elapsed >= toggleTarget) {
    //Toggle the step pin.
    pinOn = !pinOn;
    digitalWrite(stepPin, pinOn);
    digitalWrite(LED_BUILTIN, pinOn);

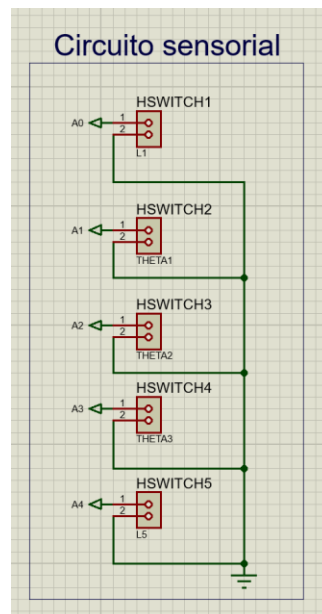
    //Update the tracking variables.
    lastStamp += toggleTarget; //Advance the stamp by the amount of time that was meant to have occurred between pins.
  }
}

```

ANEXO 9-CONEXIÓN DE MOTORES PARA EL SISTEMA DE ACCIONAMIENTO



ANEXO 10-ESQUEMÁTICO DE SUBSISTEMA SENSORIAL



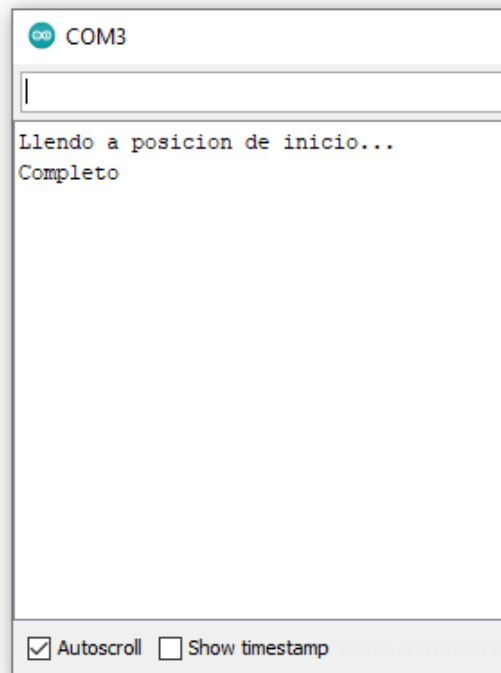
ANEXO 11-PROGRAMACION FINAL DE CARRERA Y MOTORES EN POSICIÓN DE INICIO

```
void setup() {
  Serial.begin(9600);
  // put your setup code here, to run once:
  pinMode(HomeSwitch1, INPUT_PULLUP);
  pinMode(HomeSwitch2, INPUT_PULLUP);
  pinMode(HomeSwitch3, INPUT_PULLUP);
  pinMode(HomeSwitch4, INPUT_PULLUP);
  pinMode(HomeSwitch5, INPUT_PULLUP);
  delay(5);

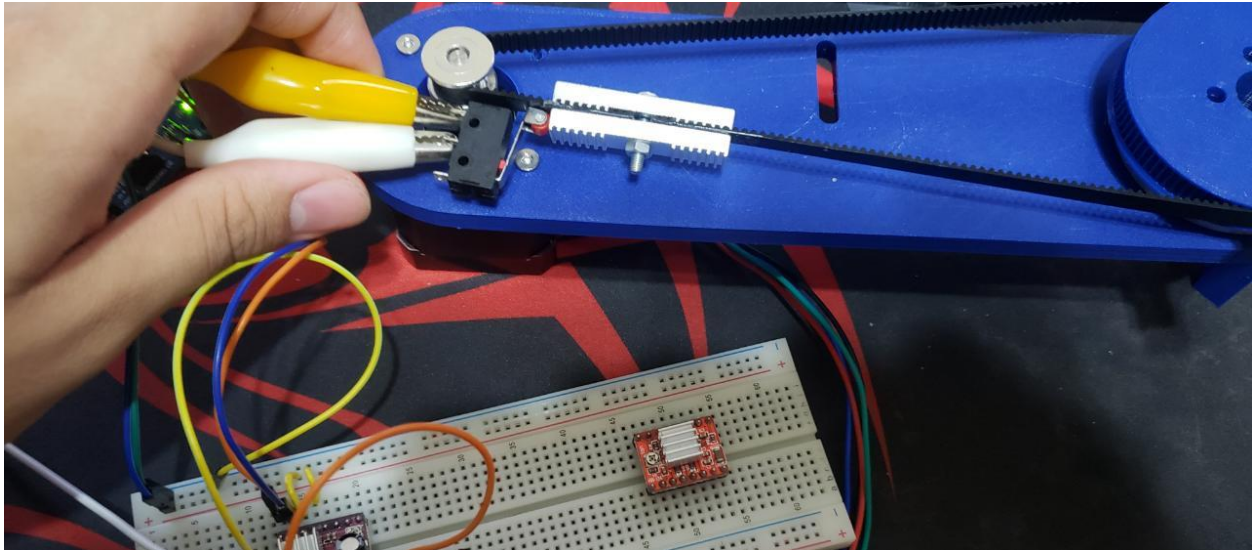
  //Stepper1
  stepper1.setMaxSpeed(500); //150 rpm
  stepper1.setAcceleration(250); //la mitad de la velocidad
  Serial.println("Llendo a posicion de inicio...");

  while( digitalRead(HomeSwitch1)){
    stepper1.moveTo(Homing);
    stepper1.run();
    Homing --;
    delay(10);
  }
  stepper1.setCurrentPosition(0);
  stepper1.setMaxSpeed(500); //150 rpm
  stepper1.setAcceleration(250); //la mitad de la velocidad
  Homing=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch

  while(!digitalRead(HomeSwitch1)){
    stepper1.moveTo(Homing);
    stepper1.run();
    Homing ++;
    delay(10);
  }
  stepper1.setCurrentPosition(0);
  Serial.println("Completo");
  stepper1.setMaxSpeed(1534); //460 rpm
  stepper1.setAcceleration(767); //la mitad de la velocidad
```



ANEXO 12-PRUEBA DE SENSOR DE FINAL DE CARRERA SIENDO ACTIVADO Y DETENIENDO UN MOTOR



ANEXO 13-INTEGRACIÓN DE CINEMÁTICA DIRECTA EN PROGRAMACIÓN PRINCIPAL PARTE 1

```
float DH(float theta1,float theta2, float theta3, float L1, float L5,int i){//cinematica directa

int L2 = 108;
int L0 = 176;//originalmente 176
int L3 = 169;
int L4 = 169;
int L6 = 30;
float x,y,z;
//Matrices que conforman DH01
BLA::Matrix<4,4> rotz01 = {cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> transXYZ01 = {1,0,0,L2,0,1,0,0,0,0,1,(L0+L1),0,0,0,1};
BLA::Matrix<4,4> rotx01 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
//Matriz DH01
BLA::Matrix<4,4> DH01 = rotz01*transXYZ01*rotx01;

//Matrices que conforman DH12
BLA::Matrix<4,4> rotz12 = {cos(-HALF_PI+theta1),-sin(-HALF_PI+theta1),0,0,sin(-HALF_PI+theta1),cos(-HALF_PI+theta1),0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> transXYZ12 = {1,0,0,L3,0,1,0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> rotx12 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
//Matriz DH12
BLA::Matrix<4,4> DH12 = rotz12*transXYZ12*rotx12;

//Matrices que conforman DH23
BLA::Matrix<4,4> rotz23 = {cos(-HALF_PI+theta2),-sin(-HALF_PI+theta2),0,0,sin(-HALF_PI+theta2),cos(-HALF_PI+theta2),0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> transXYZ23 = {1,0,0,L4,0,1,0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> rotx23 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
//Matriz DH23
BLA::Matrix<4,4> DH23 = rotz23*transXYZ23*rotx23;
```

ANEXO 14-INTEGRACIÓN DE CINEMÁTICA DIRECTA EN PROGRAMACIÓN PRINCIPAL PARTE 2

```
//Matrices que conforman DH34
BLA::Matrix<4,4> rotz34 = {cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> transXYZ34 = {1,0,0,0,0,1,0,y,0,0,1,(-6-L5),0,0,0,1};
BLA::Matrix<4,4> rotx34 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
//Matriz DH34
BLA::Matrix<4,4> DH34 = rotz34*transXYZ34*rotx34;

//Matrices que conforman DH45
BLA::Matrix<4,4> rotz45 = {cos(theta3),-sin(theta3),0,0,sin(theta3),cos(theta3),0,0,0,0,1,0,0,0,0,1};
BLA::Matrix<4,4> transXYZ45 = {1,0,0,0,0,1,0,0,0,0,1,-L6,0,0,0,1};
BLA::Matrix<4,4> rotx45 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
//Matriz DH45
BLA::Matrix<4,4> DH45 = rotz45*transXYZ45*rotx45;

//Multiplicaciones para llegar a DH05
BLA::Matrix<4,4> DH02 = DH01*DH12;
BLA::Matrix<4,4> DH03 = DH02*DH23;
BLA::Matrix<4,4> DH04 = DH03*DH34;
BLA::Matrix<4,4> DH05 = DH04*DH45;

x=DH05(0,3);
y=DH05(1,3);
z=DH05(2,3);

if (i==1) {
    return x;
}else if (i==2){
    return y;
}else if (i==3){
    return z;
}
}
```

ANEXO 15-INTEGRACIÓN DE CINEMÁTICA INVERSA EN PROGRAMACIÓN PRINCIPAL T1T2

```
float CI(float x,float y,float z,float codo,float i, float L1, float L5){//Cinematica inversa

//Pf Coordenada Z de inversa
float Pf=z;
float L2 = 108;
float L3 = 169;
float L4 = 169;
float D;
float theta1, theta2,Llinv,L5inv;
Llinv = L1;
L5inv = L5;
float Pi = 140+Llinv-L5inv;
float CambioPosicion=Pf-Pi;
//THETA2
D= (y*y+(x-L2)*(x-L2)-L4*L4-L3*L3)/(2*L4*L3);
if (codo==0){
    theta2= atan2(D,+sqrt(1-D*D));
}else if(codo==1){
    theta2= atan2(D,-sqrt(1-D*D));
}
//THETA1
theta1= HALF_PI-atan2(-L4*sin(HALF_PI-theta2), L3+L4*cos(HALF_PI-theta2))-atan2(-y,x-L2);
```

ANEXO 16-INTEGRACIÓN DE CINEMÁTICA INVERSA EN PROGRAMACIÓN PRINCIPAL L1 Y L5 SUBIDA

```
// COORDENADA L1 y L5
if(CambioPosicion>=0){//PARA SUBIR

    if(Pf>545.5){// 545.5 es la posicion maxima para alcanzar;if posicion fuera de rango= todo queda igual

        Llinv=Llinv;
        L5inv=L5inv;

    }else if(L5inv>=CambioPosicion){//Si el efector final puede alcanzar el punto z por si mismo

        Llinv+=30;//primer movimiento 30 arriba en L1 independiente de todo lo demas al MOTOR
        // stepper1.moveTo(Llinv*25);
        // stepper1.runToPosition();
        L5inv=L5inv-CambioPosicion;//movimiento supuesto al MOTOR
        // stepper5.moveTo(L5inv*25);
        // stepper5.runToPosition();
        Llinv=Llinv-30;//tercer movimiento 30 abajo compensando lo que L1 subio de mas al MOTOR
        // stepper1.moveTo(Llinv*25);
        // stepper1.runToPosition();

    }else if(L5inv<CambioPosicion){// Si el efector final no puede alcanzar el punto z por si mismo

        Llinv+=30;//primer movimiento 30 arriba en L1 independiente de todo lo demas al MOTOR
        Llinv=Llinv+(CambioPosicion-L5inv);//movimiento supuesto al MOTOR
        // stepper1.moveTo(Llinv*25);
        // stepper1.runToPosition();
        L5inv=L5inv;//movimiento supuesto al MOTOR
        // stepper5.moveTo(L5inv*25);
        // stepper5.runToPosition();
        L5inv=L5inv+30;//tercer movimiento 30 abajo compensando lo que L1 subio de mas al MOTOR
        //stepper5.moveTo(L5inv*25);
        //stepper5.runToPosition();
    }
}
```

ANEXO 17-INTEGRACIÓN DE CINEMÁTICA INVERSA EN PROGRAMACIÓN PRINCIPAL L1 Y L5 BAJADA

```
}else{// PARA BAJAR
CambioPosicion = CambioPosicion*-1;
//Para bajar seria bueno primero mover las articulaciones y luego lo que bajaria
if(Pf<0){

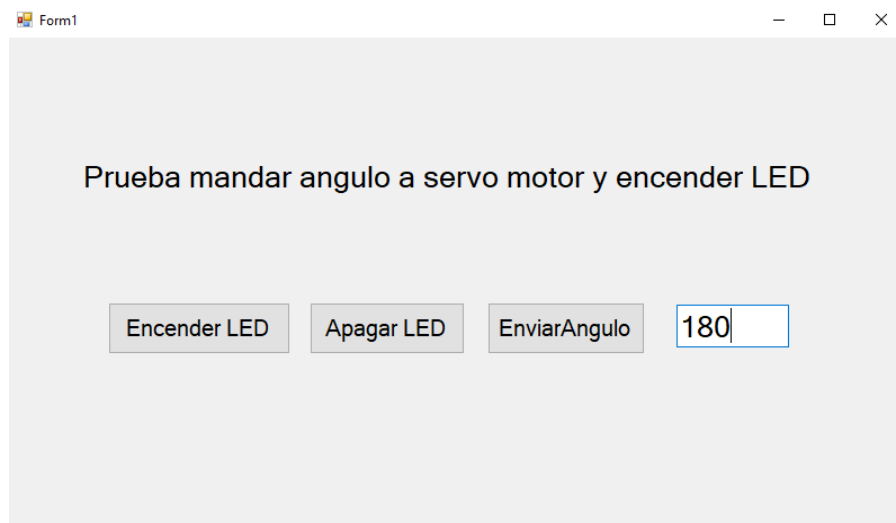
Llinv=Llinv;
L5inv=L5inv;

}else if(CambioPosicion<=(140-L5inv)){//Si el efector final puede alcanzar el punto z por si mismo
L5inv+=CambioPosicion;
// stepper5.moveTo(L5inv*25);
// stepper5.runToPosition();

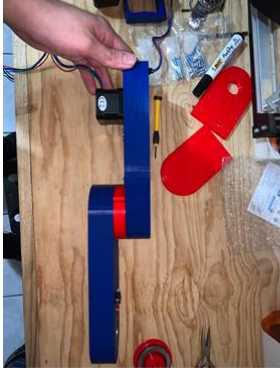
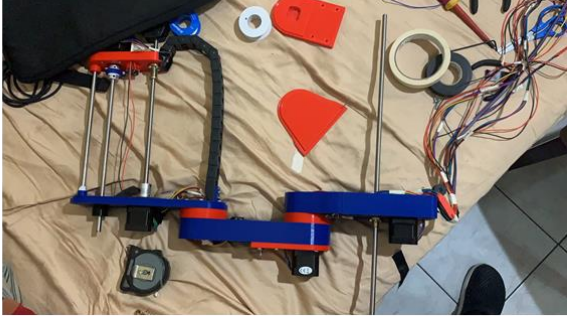
}else if(CambioPosicion>(140-L5inv)){//Si el efector final no puede alcanzar el punto z por si mismo
int L5temp=L5inv+(140-L5inv);
//Se refiere a lo que ya estaba bajado, mas lo que faltaba por bajar (140-L5), bajando L5 en su totalidad
Llinv=Llinv-(CambioPosicion-(L5temp-L5inv));
//Se necesiba bajar en L1 lo que faltó en el cambio de posicion, al restarle lo que se bajo con L5
// stepper1.moveTo(Llinv*25);
// stepper1.runToPosition();
L5inv=L5temp;
// stepper5.moveTo(L5inv*25);
// stepper5.runToPosition();

}
}
if (i==1) {
return theta1;
}else if (i==2){
return theta2;
}else if(i==3){
return Llinv;
}else if(i==4){
return L5inv;
}
}
```

ANEXO 18-PRUEBA DE **GUI** ACCIONANDO UN MOTOR Y UNA LED POR MEDIO DEL ARDUINO



ANEXO 19 – FRAGMENTOS DE PROCESO DE CONSTRUCCIÓN



ANEXO 20 – PROGRAMACIÓN DE GUI EN VISUALSTUDIO

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace GUI_FINAL_ROBOT_SCARA
12 {
13     3 references
14     public partial class Form1 : Form
15     {
16         private string ToArduino;
17         private double T1;
18         private double T2;
19         private double T3;
20         private double L1;
21         private double L5;
22         private double gripper;
23         private double T1dif;
24         private double T2dif;
25         private double T3dif;
26         private double L1dif;
27         private double L5dif;
28         private double gripperDif;
29         private double Vsteps;
30         private string VstepsStr;
31         private double Asteps;
32         private string AstepsStr;
33         public delegate void d1(string indata); //Bridge between code and GUI, public-can be accsed from anywhere within the code
34         private string Posicion;
35         private string Posiciones;
36         private int counter;
37         private string counterStr;
38         private int paran;
39         private string paranStr;
40         private string CicloPos;
41
42     1 reference
43     public Form1()
44     {
45         InitializeComponent();
46         serialPort1.Open();
47         textBox1.Text = "50";
48         textBox2.Text = "50";
49         textBox3.Text = "50";
50         textBox4.Text = "100";
51         textBox5.Text = "80";
52         textBox9.Text = "50";
53
54         trackBar1.Value = 100;
55         textBox10.Text = Convert.ToString(trackBar1.Value);
56         Vsteps = Convert.ToDouble(textBox10.Text) * 30.6;
57         VstepsStr = Convert.ToString(Vsteps);
58
59         trackBar2.Value = 100;
60         textBox11.Text = Convert.ToString(trackBar2.Value);
61         Asteps = Convert.ToDouble(textBox11.Text) * 30.6;
62         AstepsStr = Convert.ToString(Asteps);
63
64     }
65
66     1 reference
67     private void button2_Click(object sender, EventArgs e)
68     {
69         T1dif = Convert.ToDouble(textBox1.Text);
70         T1 = T1 + T1dif;
71         if (T1 > 180)
72         {
73             T1 = 180;
74         }
75         textBox12.Text = Convert.ToString(T1);
76         progressBar1.Value = Convert.ToInt32(T1);
77         ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
78         serialPort1.Write(ToArduino);
79     }

```

```

83 | I reference
84 | private void button1_Click(object sender, EventArgs e)
85 | {
86 |     T1dif = Convert.ToDouble(textBox1.Text);
87 |     T1 = T1 - T1dif;
88 |     if (T1 < 0)
89 |     {
90 |         T1 = 0;
91 |     }
92 |     textBox12.Text = Convert.ToString(T1);
93 |     progressBar1.Value = Convert.ToInt32(T1);
94 |     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
95 |     serialPort1.Write(ToArduino);
96 | }

I reference
97 | private void button4_Click(object sender, EventArgs e)
98 | {
99 |     T2dif = Convert.ToDouble(textBox2.Text);
100 |     T2 = T2 + T2dif;
101 |     if (T2 > 180)
102 |     {
103 |         T2 = 180;
104 |     }
105 |     textBox13.Text = Convert.ToString(T2);
106 |     progressBar2.Value = Convert.ToInt32(T2);
107 |     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
108 |     serialPort1.Write(ToArduino);
109 | }

I reference
111 | private void button3_Click(object sender, EventArgs e)
112 | {
113 |     T2dif = Convert.ToDouble(textBox2.Text);
114 |     T2 = T2 - T2dif;
115 |     if (T2 < 0)
116 |     {
117 |         T2 = 0;
118 |     }
119 |     textBox13.Text = Convert.ToString(T2);
120 |     progressBar2.Value = Convert.ToInt32(T2);

121 |     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
122 |     serialPort1.Write(ToArduino);
123 | }
124 | }

I reference
125 | private void button6_Click(object sender, EventArgs e)
126 | {
127 |     T3dif = Convert.ToDouble(textBox3.Text);
128 |     T3 = T3 + T3dif;
129 |     if (T3 > 360)
130 |     {
131 |         T3 = 360;
132 |     }
133 |     textBox14.Text = Convert.ToString(T3);
134 |     progressBar3.Value = Convert.ToInt32(T3);
135 |     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
136 |     serialPort1.Write(ToArduino);
137 | }
138 | }

I reference
139 | private void button5_Click(object sender, EventArgs e)
140 | {
141 |     T3dif = Convert.ToDouble(textBox3.Text);
142 |     T3 = T3 - T3dif;
143 |     if (T3 < 0)
144 |     {
145 |         T3 = 0;
146 |     }
147 |     textBox14.Text = Convert.ToString(T3);
148 |     progressBar3.Value = Convert.ToInt32(T3);
149 |     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
150 |     serialPort1.Write(ToArduino);
151 | }
152 | }

I reference
153 | private void button8_Click(object sender, EventArgs e)
154 | {
155 |     L1dif = Convert.ToDouble(textBox4.Text);
156 |     L1 = L1 + L1dif;
157 |     if (L1 > 545)
158 |     {
159 |         L1 = 545;
160 |     }

```

```

161     textBox15.Text = Convert.ToString(L1);
162     progressBar4.Value = Convert.ToInt32(L1);
163     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
164     serialPort1.Write(ToArduino);
165 }
166
167 1 reference
168 private void button7_Click(object sender, EventArgs e)
169 {
170     L1dif = Convert.ToDouble(textBox4.Text);
171     L1 = L1 - L1dif;
172     if (L1 < 0)
173     {
174         L1 = 0;
175     }
176     textBox15.Text = Convert.ToString(L1);
177     progressBar4.Value = Convert.ToInt32(L1);
178     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
179     serialPort1.Write(ToArduino);
180 }
181
182 1 reference
183 private void button10_Click(object sender, EventArgs e)
184 {
185     L5dif = Convert.ToDouble(textBox5.Text);
186     L5 = L5 + L5dif;
187     if (L5 > 170)
188     {
189         L5 = 170;
190     }
191     textBox16.Text = Convert.ToString(L5);
192     progressBar5.Value = Convert.ToInt32(L5);
193     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
194     serialPort1.Write(ToArduino);
195 }
196
197 1 reference
198 private void button9_Click(object sender, EventArgs e)
199 {
200     L5dif = Convert.ToDouble(textBox5.Text);
201     L5 = L5 - L5dif;
202     if (L5 < 0)
203     {
204         L5 = 0;
205     }
206     textBox16.Text = Convert.ToString(L5);
207     progressBar5.Value = Convert.ToInt32(L5);
208     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
209     serialPort1.Write(ToArduino);
210 }
211
212 1 reference
213 private void trackBar1_Scroll(object sender, EventArgs e)
214 {
215     textBox10.Text = Convert.ToString(trackBar1.Value);
216     Vsteps = Convert.ToDouble(textBox10.Text) * 30.6;
217     VstepsStr = Convert.ToString(Vsteps);
218 }
219
220 1 reference
221 private void trackBar2_Scroll(object sender, EventArgs e)
222 {
223     textBox11.Text = Convert.ToString(trackBar2.Value);
224     Asteps = Convert.ToDouble(textBox11.Text) * 30.6;
225     AstepsStr = Convert.ToString(Asteps);
226 }
227
228 1 reference
229 private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
230 {
231     string indata = serialPort1.ReadLine(); //Will read the string in the serial buffer from the GUI, and stores it in the string indata
232     d1 writeit = new d1(writeForm); //Function in the GUI that will process the information that is sent to it
233     Invoke(writeit, indata); //Will actually start the data bridge working
234 }
235
236 1 reference
237 public void writeForm(string indata) // C strings are arrays, first character in array can be placed in char
238 {
239     char firstchar;
240     string FromArduino;
241     firstchar = indata[0];
242     Console.WriteLine(indata);
243     switch (firstchar)
244     {
245         ...

```



```

238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320

```

```

case 'i':
    FromArduino = indata.Substring(1);
    string[] PosInicial = FromArduino.Split(',');
    textBox6.Text = PosInicial[0];
    textBox7.Text = PosInicial[1];
    textBox8.Text = PosInicial[2];

    textBox12.Text = PosInicial[3];
    T1 = Convert.ToDouble(textBox12.Text);
    progressBar1.Value = Convert.ToInt32(T1);
    textBox13.Text = PosInicial[4];
    T2 = Convert.ToDouble(textBox13.Text);
    progressBar2.Value = Convert.ToInt32(T2);
    textBox14.Text = PosInicial[5];
    T3 = Convert.ToDouble(textBox14.Text);
    progressBar3.Value = Convert.ToInt32(T3);
    textBox15.Text = PosInicial[6];
    L1 = Convert.ToDouble(textBox15.Text);
    progressBar4.Value = Convert.ToInt32(L1);
    textBox16.Text = PosInicial[7];
    L5 = Convert.ToDouble(textBox16.Text);
    progressBar5.Value = Convert.ToInt32(L5);
    textBox17.Text = PosInicial[8];
    gripper = Convert.ToDouble(textBox17.Text);
    break;

case 'a':

    FromArduino = indata.Substring(1);
    string[] articulaciones = FromArduino.Split(',');
    textBox12.Text = articulaciones[0];
    T1 = Convert.ToDouble(textBox12.Text);
    progressBar1.Value = Convert.ToInt32(T1);
    textBox13.Text = articulaciones[1];
    T2 = Convert.ToDouble(textBox13.Text);
    progressBar2.Value = Convert.ToInt32(T2);
    textBox14.Text = articulaciones[2];
    T3 = Convert.ToDouble(textBox14.Text);
    progressBar3.Value = Convert.ToInt32(T3);
    textBox15.Text = articulaciones[3];
    L1 = Convert.ToDouble(textBox15.Text);
    progressBar4.Value = Convert.ToInt32(L1);
    textBox16.Text = articulaciones[4];
    L5 = Convert.ToDouble(textBox16.Text);
    progressBar5.Value = Convert.ToInt32(L5);
    textBox17.Text = articulaciones[5];
    gripper = Convert.ToDouble(textBox17.Text);
    break;

case 'c':
    FromArduino = indata.Substring(1);
    string[] coordenadas = FromArduino.Split(',');
    textBox6.Text = coordenadas[0];
    textBox7.Text = coordenadas[1];
    textBox8.Text = coordenadas[2];
    break;
}

1 reference
private void boton11_Click(object sender, EventArgs e)
{
    Posicion = textBox6.Text + "," + textBox7.Text + "," + textBox8.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox17.Text;
    Posiciones = Posiciones + Posicion + "/";
    counter++;
    Posicion = "";
}

1 reference
private void boton15_Click(object sender, EventArgs e)
{
    counterStr = Convert.ToString(counter);
    parar = 0;
    pararStr = Convert.ToString(parar);
    CicloPos = pararStr + counterStr + Posiciones;
    serialPort1.Write("c"+ CicloPos);

    //Este cdodigo era solo para probar que se creaba bien el string y comprobar las posiciones guardadas.
    /*char Secondchar = CicloPos[0];
    string Pos = CicloPos.Substring(1);
    string[] descomprimidoPOS = Pos.Split('/');
    string[] Posicion1 = descomprimidoPOS[0].Split(',');

```

```

321 string[] Posicion2 = descomprimidoPOS[1].Split(',');
322 string[] Posicion3 = descomprimidoPOS[2].Split(',');
323 textBox5.Text = Convert.ToString(Secondchar);
324 textBox12.Text = Posicion1[0];
325 textBox1.Text = Posicion1[1];
326 textBox13.Text = Posicion1[2];
327 textBox2.Text = Posicion2[0];
328 textBox14.Text = Posicion2[1];
329 textBox3.Text = Posicion2[2];
330 textBox15.Text = Posicion3[0];
331 textBox4.Text = Posicion3[1];
332 textBox16.Text = Posicion3[2];*/
333
334
335 }
336
337 1 reference
338 private void button14_Click(object sender, EventArgs e)
339 {
340     Posiciones = "";
341     counter = 0;
342 }
343 1 reference
344 private void button16_Click(object sender, EventArgs e)
345 {
346     parar = 1;
347     pararStr = Convert.ToString(parar);
348     // counterStr = Convert.ToString(counter);
349     // CicloPos = pararStr + counterStr + Posiciones;
350     // serialPort1.Write("c" + CicloPos);
351     serialPort1.Write(pararStr);
352 }
353
354 1 reference
355 private void button13_Click(object sender, EventArgs e)
356 {
357     gripperDif = Convert.ToDouble(textBox9.Text);
358     gripper = gripper + gripperDif;
359     if (gripper > 180)

```

```

359     {
360         gripper = 180;
361     }
362     textBox17.Text = Convert.ToString(gripper);
363     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
364     serialPort1.Write(ToArduino);
365 }
366
367 1 reference
368 private void button12_Click(object sender, EventArgs e)
369 {
370     gripperDif = Convert.ToDouble(textBox9.Text);
371     gripper = gripper - gripperDif;
372     if (gripper < 90)
373     {
374         gripper = 90;
375     }
376     textBox17.Text = Convert.ToString(gripper);
377     ToArduino = "a" + textBox12.Text + ',' + textBox13.Text + ',' + textBox14.Text + ',' + textBox15.Text + ',' + textBox16.Text + ',' + VstepsStr + ',' + AstepsStr + ',' + textBox17.Text;
378     serialPort1.Write(ToArduino);
379 }
380 1 reference
381 private void progressBar3_Click(object sender, EventArgs e)
382 {
383 }
384
385 1 reference
386 private void textBox4_TextChanged(object sender, EventArgs e)
387 {
388 }
389
390 1 reference
391 private void button17_Click(object sender, EventArgs e)
392 {
393     string resetArduino = "r";
394     serialPort1.Write(resetArduino);
395 }
396 }

```

ANEXO 21 – PROGRAMACIÓN FINAL ARDUINO

```
1 //STEPPER 1 ES L1
2 //STEPPER 2 ES THETA1
3 //STEPPER 3 ES THETA2
4 //STEPPER 4 ES THETA3
5 //STEPPER 5 ES L5
6
7 #include <Servo.h>
8 #include <AccelStepper.h>
9 #include <BasicLinearAlgebra.h>
10 #include <Separador.h>
11
12 #define HomeSwitch1 A8//en el pin A0; L1
13 #define HomeSwitch2 A9//en el pin A1; THETA1
14 #define HomeSwitch3 A10//en el pin A2; THETA2
15 #define HomeSwitch4 A11//en el pin A3; THETA3
16 #define HomeSwitch5 A12//en el pin A4;L5
17
18 #define PI 3.1415926535897932384626433832795
19 #define HALF_PI 1.5707963267948966192313216916398
20 #define TWO_PI 6.283185307179586476925286766559
21
22 Separador s;
23 //SERVO
24 Servo servol;
25 //STEPPERS
26 AccelStepper stepper1(1,3,2);//Para indicar que es un driver; Steps; Direccion;
27 AccelStepper stepper2(1,5,4);//Para indicar que es un driver; Steps; Direccion;
28 AccelStepper stepper3(1,7,6);//Para indicar que es un driver; Steps; Direccion;
29 AccelStepper stepper4(1,9,8);//Para indicar que es un driver; Steps; Direccion;
30 AccelStepper stepper5(1,11,10);//Para indicar que es un driver; Steps; Direccion;
31
32 //Le agregue valores iniciales a x,y,z para al terminar homing se envíen estos valores a la GUI sin necesidad de usar ninguna cinematica.
33 float x=-61,y=-169,z=160,thetal,theta2,theta3,L1,L5,Gripper=180,thetal1,theta21,theta31,L1,L51,codo,data[10],theta2COD0, data1[10],Pp,L1inv,L5inv,Pi,CambioPosicion,thetal1Lost,theta21Lost;
34 String ForGUI,FromGUI,Posiciones,Posicion,xstring,ystring,zstring,theta2COD0string,theta31string,dataPackage,GripperString;
35 char firstchar,parar,CharRepInv;
36 int RepInv,MT1,MT2,MT3,ML1,ML5,thetal1LOSTint,theta21LOSTint, counterDif;
37 #define **data[0]=thetal
38 * data[1]=theta2
39 * data[2]=theta3
40
41 * data[3]=L1
42 * data[4]=L5
43 * data[5]=Velocidad en steps
44 * data[6]=Aceleracion en steps
45 * data[7]=Posicion Servo
46 * data[8]=Disponible
47 * data[9]=Disponible
48 * data[10]=Disponible*/
49
50 void setup() {
51   Serial.begin(115200);
52   pinMode(HomeSwitch1, INPUT_PULLUP);
53   pinMode(HomeSwitch2, INPUT_PULLUP);
54   pinMode(HomeSwitch3, INPUT_PULLUP);
55   pinMode(HomeSwitch4, INPUT_PULLUP);
56   pinMode(HomeSwitch5, INPUT_PULLUP);
57
58   servol.attach(12); //make output 12 the servo channel
59
60   delay(50);
61   stepper1.setMaxSpeed(2500); //150 rpm
62   stepper1.setAcceleration(2500); //la mitad de la velocidad
63   Homing();
64   servol.write(Gripper);
65   delay(3000);
66   ForGUI= ForGUI + '1' + x + ',' + y + ',' + z + ',' + thetal + ',' + theta2 + ',' + theta3 + ',' + L1 + ',' + L5 + ',' + Gripper;
67   Serial.println(ForGUI);
68   ForGUI= "";
69 }
70
71 void loop() {
72   if (Serial.available()){
73     FromGUI = Serial.readString(); // Read the incoming data from Processing
74     firstchar = FromGUI.charAt(0);
75     void(* resetFunc) (void) = 0;//declare reset function at address 0
76     switch(firstchar){
77
```

```

78     case 'r':
79         resetFunc(); //call reset
80         break;
81
82     case 'a':
83         dataPackage = FromGUI.substring(1);
84     for (int j = 0; j < 9 ; j++) {
85         int index = dataPackage.indexOf(","); // locate the first ","
86         data[j] = atof(dataPackage.substring(0, index).c_str()); //Extract the number from start to the ","
87         dataPackage = dataPackage.substring(index + 1); //Remove the number from the string
88     }
89
90     //desglose de variables para los motores independiente del DH
91     theta1 = data[0];
92     theta2 = data[1];
93     theta3 = data[2];
94     L1 = data[3];
95     L5 = data[4];
96     Gripper = data[7];
97
98     //Esto convierte los float a int, MoveTo requiere que metamos un int
99     ML1 = (int) L1;
100    MT1 = (int) theta1;
101    MT2 = (int) theta2;
102    MT3 = (int) theta3;
103    ML5 = (int) L5;
104
105    ML1 = ML1 * 50;
106    MT1 = MT1 * 5;
107    MT2 = MT2 * 5;
108    MT3 = MT3 * 1;
109    ML5 = ML5 * 50;
110
111    //Guardar datos de velocidad y aceleracion a los motores en Steps
112    stepper1.setMaxSpeed(data[5]);
113    stepper1.setAcceleration(data[6]);
114    stepper2.setMaxSpeed(data[5]*0.3);
115
116    stepper2.setAcceleration(data[6]*0.3);
117    stepper3.setMaxSpeed(data[5]*0.3);
118    stepper3.setAcceleration(data[6]*0.3);
119    stepper4.setMaxSpeed(data[5]*0.3);
120    stepper4.setAcceleration(data[6]*0.3);
121    stepper5.setMaxSpeed(data[5]*0.2475);
122    stepper5.setAcceleration(data[6]*0.2475);
123
124    //Mandar dato al servo motor
125    // servol.write(Gripper);
126
127    //Mandarle a cada stepper que se mueva a la posicion de su articulacion recién recibida desde la GUI
128    stepper1.moveTo(ML1);
129    stepper2.moveTo(MT1);
130    stepper3.moveTo(MT2);
131    stepper4.moveTo(MT3);
132    stepper5.moveTo(ML5);
133
134    while (stepper1.currentPosition() != ML1 || stepper2.currentPosition() != MT1 || stepper3.currentPosition() != MT2 || stepper4.currentPosition() != MT3 || stepper5.currentPosition() != ML5) {
135        stepper1.run();
136        stepper2.run();
137        stepper3.run();
138        stepper4.run();
139        stepper5.run();
140    }
141
142    //Llamar funcion de DH(denavit hartenberg) para que nos devuelva las coordenadas x,y,z.
143    x=DR(data[0]*(PI/180),data[1]*(PI/180),data[2]*(PI/180),data[3],data[4],1);
144    y=DR(data[0]*(PI/180),data[1]*(PI/180),data[2]*(PI/180),data[3],data[4],2);
145    z=DR(data[0]*(PI/180),data[1]*(PI/180),data[2]*(PI/180),data[3],data[4],3);
146
147    ForGUI = ForGUI + 'c' + x + ',' + y + ',' + z;
148    Serial.println(ForGUI);
149    ForGUI = "";
150    break;
151
152    case 'c':
153        parer=FromGUI.charAt(1);
154        CharRepInv = FromGUI.charAt(2);
155        RepInv = CharRepInv - '0';
156    }

```

```

154     posiciones = FromGUI.substring(3);
155
156
157 while(parar=='0'){
158     for(int i=0; i<RepInv; i++){
159         Posicion = s.separa(posiciones, '/', i);
160
161         for(int k=0; k<6; k++){
162             int indexi = Posicion.indexOf(",");
163             datai[k] = atof(Posicion.substring(0, indexi).c_str()); //Extract the number from start to the ","
164             Posicion = Posicion.substring(indexi + 1); //Remove the number from the string
165         }
166         /*datai[0]=x
167         datai[1]=y
168         datai[2]=z
169         datai[3]=theta2
170         datai[4]=theta3
171         datai[5]=gripper*/
172
173         if((datai[0]>0 && datai[1]>0) || (datai[0]<0 && datai[1]>0)){
174             codo=1;
175         }
176
177         if((datai[0]<0 && datai[1]<0) || (datai[0]>0 && datai[1]<0)){
178             codo=0;
179         }
180
181         //Inversa le ingresa x,y,z,L1,L5
182         delay(100);
183         theta1 = CI(datai[0], datai[1], datai[2], codo, 1) * (180/PI);
184         delay(100);
185         theta2i = CI(datai[0], datai[1], datai[2], codo, 2) * (180/PI);
186         theta3i = datai[4];
187         Gripper=datai[5];
188
189         stepper2.setMaxSpeed(700);
190         stepper2.setAcceleration(700);
191         stepper3.setMaxSpeed(700);
192         stepper3.setAcceleration(700);

```

```

193
194 //Se debe reasignar el valor al que se va mover el motor dependiendo de las articulaciones de la inversa
195 MT1 = (int) thetali;
196 delay(100);
197 MT2 = (int) theta2i;
198 delay(100);
199 MT3 = (int) theta3i;
200
201 //IFs de compensacion de steps perdidos THETA 1
202 if(thetal<thetali){
203     thetaliLost=thetali*0.12;
204     thetaliLOSTint = int(thetaliLost);
205     stepper2.moveTo((MT1*5)+thetaliLOSTint);
206 }
207
208 else if(thetal>thetali){
209     thetaliLost=thetali*0.12;
210     thetaliLOSTint = int(thetaliLost);
211     stepper2.moveTo((MT1*5)-thetaliLOSTint);
212 }
213
214 else{
215     stepper2.moveTo(MT1*5);
216 }
217
218 //IFs de compensacion de steps perdidos THETA 2
219 if(theta2<theta2i){
220     theta2iLost=theta2i*0.12;
221     theta2iLOSTint = int(theta2iLost);
222     stepper3.moveTo((MT2*5)+theta2iLOSTint);
223 }
224
225 else if(theta2>theta2i){
226     theta2iLost=theta2i*0.12;
227     theta2iLOSTint = int(theta2iLost);
228     stepper3.moveTo((MT2*5)-theta2iLOSTint);
229 }
230

```

```

231     else{
232         stepper3.moveTo(MT2*5);
233     }
234
235
236     // stepper2.moveTo(MT1*5);
237     // stepper3.moveTo(MT2*5);
238     stepper4.moveTo(MT3*1);
239
240 // COORDENADA L1 y L5
241 //Pf Coordenada Z de inversa
242 Pf=datai[2];
243 Llinv = L1;
244 L5inv = L5;
245 Pi = 160-0+Llinv-L5inv;
246 CambioPosicion=Pf-Pi;
247 if(CambioPosicion>=0){//PARA SUBIR
248
249     //CAMBIAR NUEVA POSICION MAXIMA
250     if(Pf>715){// 715=160+545 es la posicion maxima para alcanzar;if posicion fuera de rango= todo queda igual
251
252         Llinv=Llinv;
253         L5inv=L5inv;
254
255     }else if(L5inv>=CambioPosicion){//Si el efector final puede alcanzar el punto z por si mismo
256
257         L5inv=L5inv-CambioPosicion;//movimiento supuesto al MOTOR
258         stepper5.moveTo(L5inv*50);
259         stepper5.runToPosition();
260
261         stepper3.runToPosition();
262         stepper2.runToPosition();
263         stepper4.runToPosition();
264
265     }else if(L5inv<CambioPosicion){// Si el efector final no puede alcanzar el punto z por si mismo
266
267         Llinv=Llinv+(CambioPosicion-L5inv);//movimiento supuesto al MOTOR

```

```

268     L5inv=L5inv;//movimiento supuesto al MOTOR
269     stepper5.moveTo(L5inv*50);
270     stepper5.runToPosition();
271     stepper1.moveTo(Llinv*50);
272     stepper1.runToPosition();
273
274     stepper3.runToPosition();
275     stepper2.runToPosition();
276     stepper4.runToPosition();
277     }
278
279 }else{// PARA BAJAR
280     CambioPosicion = CambioPosicion*-1;
281     //Para bajar seria bueno primero mover las articulaciones y luego lo que bajaria
282     if(Pf<0){
283
284         Llinv=Llinv;
285         L5inv=L5inv;
286
287     }else if(CambioPosicion<=(170-L5inv)){//Si el efector final puede alcanzar el punto z por si mismo
288
289         stepper3.runToPosition();
290         stepper2.runToPosition();
291         stepper4.runToPosition();
292
293         L5inv+=CambioPosicion;
294         stepper5.moveTo(L5inv*50);
295         stepper5.runToPosition();
296
297     }else if(CambioPosicion>(170-L5inv)){//Si el efector final no puede alcanzar el punto z por si mismo
298
299         stepper3.runToPosition();
300         stepper2.runToPosition();
301         stepper4.runToPosition();
302
303         int L5temp=L5inv+(170-L5inv);
304         //Se refiere a lo que ya estaba bajado, mas lo que faltaba por bajar (140-L5), bajando L5 en su totalidad
305         Llinv=Llinv-(CambioPosicion-(L5temp-L5inv));
306         //Se necesiba bajar en Ll lo que falto en el cambio de posicion, al restarle lo que se bajo con L5
307         ...

```



```

307     stepper1.moveTo(Llinv*50);
308     stepper1.runToPosition();
309     L5inv=L5temp;
310     stepper5.moveTo(L5inv*50);
311     stepper5.runToPosition();
312
313 }
314 }
315     //Mandar dato al servo motor
316     servol.write(Gripper);
317
318     //Se deben sobrescribir L1 y L5 originales, para que estos sean los nuevos valores.
319     L1=Llinv;
320     L5=L5inv;
321     theta1=thetali;
322     theta2=theta2i;
323
324     ForGUI = ForGUI + 'a' + theta1 + ',' + theta2i + ',' + theta3i + ',' + Llinv + ',' + L5inv + ',' + Gripper;
325     Serial.println(ForGUI);
326     delay(1000);
327     ForGUI= "";
328 }
329
330 //Leer nuevamente dentro del while para saber si ya podemos salir de el
331 if (Serial.available()){
332     String PararStr = Serial.readString();
333     parar = PararStr.charAt(0);
334 }
335
336 }
337 break;
338 }
339 }
340 }
341
342
341 float DH(float theta1,float theta2, float theta3, float L1, float L5,int i){//cinematica directa
342 float DH(float theta1,float theta2, float theta3, float L1, float L5,int i){//cinematica directa
343
344     int L2 = 108;
345     int L0 = 160;//originalmente 176
346     int L3 = 169;
347     int L4 = 169;
348     int L6 = 0; //111
349     float x,y,z;
350     //Matrices que conforman DH01
351     BLA::Matrix<4,4> rotz01 = {cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1,0,0,0,1};
352     BLA::Matrix<4,4> transXYZ01 = {1,0,0,L2,0,1,0,0,0,0,1,(L0+L1),0,0,0,1};
353     BLA::Matrix<4,4> rotx01 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
354     //Matriz DH01
355     BLA::Matrix<4,4> DH01 = rotz01*transXYZ01*rotx01;
356
357     //Matrices que conforman DH12
358     BLA::Matrix<4,4> rotz12 = {cos(-HALF_PI+theta1),-sin(-HALF_PI+theta1),0,0,sin(-HALF_PI+theta1),cos(-HALF_PI+theta1),0,0,0,0,1,0,0,0,1};
359     BLA::Matrix<4,4> transXYZ12 = {1,0,0,L3,0,1,0,0,0,0,1,0,0,0,0,1};
360     BLA::Matrix<4,4> rotx12 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
361     //Matriz DH12
362     BLA::Matrix<4,4> DH12 = rotz12*transXYZ12*rotx12;
363
364     //Matrices que conforman DH23
365     BLA::Matrix<4,4> rotz23 = {cos(-HALF_PI+theta2),-sin(-HALF_PI+theta2),0,0,sin(-HALF_PI+theta2),cos(-HALF_PI+theta2),0,0,0,0,1,0,0,0,1};
366     BLA::Matrix<4,4> transXYZ23 = {1,0,0,L4,0,1,0,0,0,0,1,0,0,0,0,1};
367     BLA::Matrix<4,4> rotx23 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
368     //Matriz DH23
369     BLA::Matrix<4,4> DH23 = rotz23*transXYZ23*rotx23;
370
371     //Matrices que conforman DH34
372     BLA::Matrix<4,4> rotz34 = {cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1,0,0,0,1};
373     BLA::Matrix<4,4> transXYZ34 = {1,0,0,0,0,1,0,y,0,0,1,(-L5),0,0,0,1};
374     BLA::Matrix<4,4> rotx34 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
375     //Matriz DH34
376     BLA::Matrix<4,4> DH34 = rotz34*transXYZ34*rotx34;
377
378     //...

```

```

378 //Matrices que conforman DH45
379 BLA:Matrix<4,4> rotz45 = {cos(theta3),-sin(theta3),0,0,sin(theta3),cos(theta3),0,0,0,0,1,0,0,0,0,1};
380 BLA:Matrix<4,4> transXYZ45 = {1,0,0,0,0,1,0,0,0,0,1,-L6,0,0,0,1};
381 BLA:Matrix<4,4> rotx45 = {1,0,0,0,0,cos(0),-sin(0),0,0,sin(0),cos(0),0,0,0,0,1};
382 //Matriz DH45
383 BLA:Matrix<4,4> DH45 = rotz45*transXYZ45*rotx45;
384
385 //Multiplicaciones para llegar a DH05
386 BLA:Matrix<4,4> DH02 = DH01*DH12;
387 BLA:Matrix<4,4> DH03 = DH02*DH23;
388 BLA:Matrix<4,4> DH04 = DH03*DH34;
389 BLA:Matrix<4,4> DH05 = DH04*DH45;
390
391 x=DH05(0,3);
392 y=DH05(1,3);
393 z=DH05(2,3);
394
395 if (i==1) {
396     return x;
397 }else if (i==2){
398     return y;
399 }else if (i==3){
400     return z;
401 }
402 }
403
404 float CI(float x,float y,float z,float codo,float i){//Cinematica inversa
405
406
407     float L2 = 108;
408     float L3 = 169;
409     float L4 = 169;
410     float D;
411     float thetal, theta2;
412     //THETA2
413     D= (y*y+(x-L2)*(x-L2)-L4*L4-L3*L3)/(2*L4*L3);
414     if (codo==0){
415         theta2= atan2(D,+sqrt(1-D*D));

```

```

416     }else if(codo==1){
417         theta2= atan2(D,-sqrt(1-D*D));
418     }
419     //THETA1
420     thetal= HALF_PI-atan2(-L4*sin(HALF_PI-theta2), L3+L4*cos(HALF_PI-theta2))-atan2(-y,x-L2);
421
422
423     if (i==1) {
424         return thetal;
425     }else if (i==2){
426         return theta2;
427     }
428 }
429 void Homing(){ // posicion de inicio
430     int HomingTL=-1;//Homing para thetas y L1, es negativo ya que con las bobinas invertidas negativo es clockwise
431     int HomingL5=1;// Homing para L5, tiene que ser anticlockwise ya que su home esta arriba
432
433     //Stepper 4 THETA 3
434     stepper4.setMaxSpeed(667); //100 rpm
435     stepper4.setAcceleration(667); //
436     while( digitalRead(HomeSwitch5)){
437         stepper4.moveTo(HomingTL);
438         stepper4.run();
439         HomingTL --;
440         delay(10);
441     }
442     stepper4.setCurrentPosition(0);
443     stepper4.setMaxSpeed(667); //100 rpm
444     stepper4.setAcceleration(667); //
445     HomingTL=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch
446
447     while(!digitalRead(HomeSwitch5)){
448         stepper4.moveTo(HomingTL);
449         stepper4.run();
450         HomingTL ++;
451         delay(10);
452     }
453     stepper4.setCurrentPosition(0);
454     // stepper4.setMaxSpeed(1534); //460 rpm

```

```

455 // stepper4.setAcceleration(767); //la mitad de la velocidad
456
457 //Stepper 5 L5
458 stepper5.setMaxSpeed(1000); //150 rpm
459 stepper5.setAcceleration(1000); //la mitad de la velocidad
460 HomingL5=-1;
461 while( digitalRead(HomeSwitch4)){
462     stepper5.moveTo(HomingL5);
463     stepper5.run();
464     HomingL5 --;
465     delay(10);
466 }
467 stepper5.setCurrentPosition(0);
468 stepper5.setMaxSpeed(500); //150 rpm
469 stepper5.setAcceleration(250); //la mitad de la velocidad
470 HomingL5=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch
471
472 while(!digitalRead(HomeSwitch4)){
473     stepper5.moveTo(HomingL5);
474     stepper5.run();
475     HomingL5 ++;
476     delay(10);
477 }
478 stepper5.setCurrentPosition(0);
479 // stepper5.setMaxSpeed(1534); //460 rpm
480 // stepper5.setAcceleration(767); //la mitad de la velocidad
481
482 //stepper 2 THETA1
483 stepper2.setMaxSpeed(667); //150 rpm
484 stepper2.setAcceleration(667); //la mitad de la velocidad
485 HomingTL=-1;
486 while( digitalRead(HomeSwitch2)){
487     stepper2.moveTo(HomingTL);
488     stepper2.run();
489     HomingTL --;
490     delay(10);

```

```

491 }
492 stepper2.setCurrentPosition(0);
493 stepper2.setMaxSpeed(667); //150 rpm
494 stepper2.setAcceleration(667); //la mitad de la velocidad
495 HomingTL=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch
496
497 while(!digitalRead(HomeSwitch2)){
498     stepper2.moveTo(HomingTL);
499     stepper2.run();
500     HomingTL ++;
501     delay(10);
502 }
503 stepper2.setCurrentPosition(0);
504 // stepper2.setMaxSpeed(1534); //460 rpm
505 // stepper2.setAcceleration(767); //la mitad de la velocidad
506
507 //Stepper 3 THETA2
508 stepper3.setMaxSpeed(667); //150 rpm
509 stepper3.setAcceleration(667); //la mitad de la velocidad
510 HomingTL=-1;
511 while( digitalRead(HomeSwitch3)){
512     stepper3.moveTo(HomingTL);
513     stepper3.run();
514     HomingTL --;
515     delay(10);
516 }
517 stepper3.setCurrentPosition(0);
518 stepper3.setMaxSpeed(667); //150 rpm
519 stepper3.setAcceleration(667); //la mitad de la velocidad
520 HomingTL=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch
521
522 while(!digitalRead(HomeSwitch3)){
523     stepper3.moveTo(HomingTL);
524     stepper3.run();
525     HomingTL ++;
526     delay(10);
527 }
528
529 stepper3.setCurrentPosition(0);
530 //stepper3.setMaxSpeed(1534); //460 rpm
531 //stepper3.setAcceleration(767); //la mitad de la velocidad
532
533 //Stepper1 L1
534 HomingTL=-1;
535 while( digitalRead(HomeSwitch1)){
536     stepper1.moveTo(HomingTL);
537     stepper1.run();
538     HomingTL --;
539     delay(10);
540 }
541 stepper1.setCurrentPosition(0);
542 stepper1.setMaxSpeed(2500); //150 rpm
543 stepper1.setAcceleration(2500); //la mitad de la velocidad
544 HomingTL=1; // ya que el movimiento del robot es counter clockwise y hay que desactivar el switch
545
546 while(!digitalRead(HomeSwitch1)){
547     stepper1.moveTo(HomingTL);
548     stepper1.run();
549     HomingTL ++;
550     delay(10);
551 }
552 stepper1.setCurrentPosition(0);
553 //stepper1.setMaxSpeed(921); //460 rpm
554 //stepper1.setAcceleration(921); //la mitad de la velocidad
555 }

```