



ESCUELA POLITÉCNICA NACIONAL



FACULTAD DE INGENIERÍA MECÁNICA

DISEÑO Y SIMULACIÓN DE UN MANIPULADOR ROBÓTICO ARTICULADO PARA SOLDADURA POR ARCO PARA CARPINTERIA METÁLICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO MECÁNICO**

ANDRÉS EDUARDO CHUQUILLA TOAQUIZA
andres.chuquilla@epn.edu.ec

DIRECTOR:
ING. ÁLVARO GONZALO AGUINAGA BARRAGÁN PhD.
alvaro.aguinaga@epn.edu.ec

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **ANDRÉS EDUARDO CHUQUILLA TOAQUIZA**, bajo mi supervisión.

Ing. Álvaro Gonzalo Aguinaga Barragán PhD.

DIRECTOR DE PROYECTO

DECLARACIÓN

Yo, **Andrés Eduardo Chuquilla Toaquiza**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



Andrés Eduardo Chuquilla Toaquiza

DEDICATORIA

A mis padres y hermanos por todo su apoyo y cariño durante estos años de preparación profesional.

AGRADECIMIENTO

A mis padres, por el apoyo, el cariño y su guía incondicional que me han brindado siempre, gracias a ustedes he logrado culminar este objetivo.

A todos quienes me brindaron su apoyo durante este tiempo de estudios.

A todas las personas que conforman la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional, por guiarme en este camino de formación como profesional.

ÍNDICE

CERTIFICACIÓN.....	ii
DECLARACIÓN.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTO.....	iv
ÍNDICE.....	v
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS.....	xii
RESUMEN.....	xiv
ABSTRACT.....	xv
INTRODUCCIÓN.....	1
Objetivo general.....	1
Objetivos específicos.....	2
1. MARCO TEÓRICO	3
1.1. Robótica.....	3
1.1.1. Origen y desarrollo de la robótica [3].....	3
1.1.2. Definición de robot manipulador industrial.....	4
1.1.3. Clasificación de los robots [4].....	4
1.1.3.1. Clasificación según el área de aplicación.....	4
1.1.3.2. Clasificación según la configuración.....	5
1.1.4. Morfología del robot [4]	5
1.1.4.1. Manipulador Articulado	7
1.1.5. Herramientas matemáticas para la ubicación espacial [4]	7
1.1.5.1. Representación de la posición.....	7
1.1.5.2. Representación de la orientación	8
1.1.5.3. Matrices de transformación homogénea	9
1.1.6. Cinemática del robot [4]	10
1.1.6.1. Cinemática directa.....	11
1.1.6.2. Cinemática inversa.....	12

1.1.6.3.	Matriz Jacobiana	13
1.1.6.4.	Jacobiana inversa	14
1.1.7.	Dinámica del robot [4].....	15
1.1.7.1.	Modelo dinámico de la estructura mecánica de un robot rígido	16
1.1.7.2.	Propiedades del modelo dinámico [6].....	18
1.1.8.	Control cinemático [4].....	19
1.1.8.1.	Funciones de control cinemático	19
1.1.8.2.	Espacio de las articulaciones [3]	20
1.1.8.3.	Espacio cartesiano [3].....	21
1.1.8.4.	Tipos de trayectorias.....	22
1.1.8.5.	Generación de trayectorias.....	23
1.1.8.6.	Muestreo de trayectorias.....	24
1.1.8.7.	Interpolación de trayectorias.....	25
1.1.9.	Control dinámico [4].....	26
1.1.9.1.	Control mono articular.....	26
1.2.	Mecatrónica [9]	28
1.2.1.	Definición	28
1.2.2.	Sistema mecatrónico	28
1.2.3.	Modelamiento de sistemas mecatrónicos.....	28
1.2.4.	Sistemas de medición	29
1.2.5.	Sistemas de control.....	29
1.2.5.1.	Sistemas en lazo abierto y cerrado [10]	29
1.2.5.2.	Sistemas de control analógico y digital	30
2.	METODOLOGÍA.....	31
2.1.	Definición del producto	31
2.1.1.	Identificar y establecer necesidades	31
2.1.1.1.	Requerimientos iniciales	31
2.1.1.2.	Casa de la calidad.....	31
2.1.1.3.	Especificaciones técnicas	33

2.2.	Diseño conceptual	33
2.2.1.	Análisis funcional.....	33
2.2.1.1.	Desarrollo del análisis funcional	34
2.2.1.2.	Análisis de los diagramas funcionales.....	34
2.2.2.	Definición de los módulos	35
2.2.3.	Soluciones para cada módulo.....	35
2.2.3.1.	Módulo 1: Generación de trayectorias.....	35
2.2.3.2.	Módulo 2: Movimiento del manipulador	41
2.2.3.3.	Modulo 3: Soldadura	48
2.2.4.	Solución conceptual	52
2.3.	Diseño de materialización.....	53
2.3.1.	Diseño del mecanismo	53
2.3.1.1.	Volumen de trabajo	54
2.3.2.	Cálculos estructurales	55
2.3.2.1.	Fuerzas y momentos.....	55
2.3.2.2.	Selección del perfil	59
2.3.3.	Análisis estructural por elementos finitos.....	60
2.3.3.1.	Procedimiento:	61
2.3.3.2.	Solución y resultados	63
2.3.3.3.	Análisis de resultados	66
2.3.4.	Modificación del diseño	66
2.3.4.1.	Cálculos estructurales estáticos	67
2.3.4.2.	Selección de perfil:	69
2.3.4.3.	Análisis estructural estático por elementos finitos.....	70
2.3.5.	Solución del diseño de materialización	75
2.4.	Modelo robótico	76
2.4.1.	Importar modelo CAD a Simulink.....	76
2.4.1.1.	Exportar CAD	77
2.4.1.2.	Modelo importado a Simulink.....	77

2.4.2.	Importar diagrama de bloques a Matlab	78
2.4.2.1.	Sistemas de coordenadas de referencia	78
2.4.2.2.	Modelo robótico con Toolbox Robotics System Toolbox	78
2.5.	Análisis cinemático y dinámico del manipulador.....	80
2.5.1.	Análisis cinemático	80
2.5.1.1.	Solución al problema cinemático directo	80
2.5.1.2.	Solución al problema cinemático inverso	81
2.5.1.3.	Matriz Jacobiana	81
2.5.2.	Análisis dinámico.....	82
2.5.2.1.	Solución al problema dinámico directo.....	82
2.5.2.2.	Solución al problema dinámico inverso	82
2.6.	Control.....	82
2.6.1.	Control cinemático.....	82
2.6.1.1.	Tareas del control cinemático	82
2.6.1.2.	Generación de trayectorias.....	83
2.6.1.3.	Programa generador de trayectorias	84
2.7.	Sistema mecatrónico del manipulador	87
2.7.1.	Sistema mecatrónico	87
2.7.2.	Seleccionar componentes	88
2.7.2.1.	Requerimientos	88
2.7.2.2.	Comparación de rendimiento de las opciones	89
2.7.2.3.	Selección del servomotor:.....	89
2.7.2.4.	Selección del encoder.....	91
2.7.2.5.	Selección del controlador.....	91
2.7.3.	Control dinámico.....	92
2.7.3.1.	Control mono articular PID del manipulador [29]	92
2.8.	Simulación.....	94
2.8.1.	Generar interfaz grafica.....	94
2.8.2.	Simular funciones del manipulador	95

3. RESULTADOS Y DISCUSIÓN	95
3.1. Sistema mecatrónico	95
3.2. Especificaciones técnicas del manipulador.....	96
3.3. Funcionamiento del manipulador al moverse en una trayectoria	98
3.3.1. Resultados obtenidos usando Robotics System Toolbox	98
3.3.1.1. Volumen de trabajo del manipulador	98
3.3.1.2. Programa generador de trayectorias	99
3.3.2. Simulación del manipulador usando Simulink	107
3.3.2.1. Simulación de seguimiento de trayectorias	107
3.3.3. Comparación del movimiento realizado.....	113
3.4. Costos	114
4. CONCLUSIONES Y RECOMENDACIONES	115
4.1. CONCLUSIONES	115
4.2. RECOMENDACIONES.....	116
5. BIBLIOGRAFÍA	117
ANEXOS.....	120

ÍNDICE DE FIGURAS

Figura 1.1. Precios de los robots comparados a los costos de mano de obra en la década de 1990.....	3
Figura 1.2. Configuraciones más frecuentes en robots industriales.	5
Figura 1.3. Ejemplo partes de un robot y su estructura mecánica.....	6
Figura 1.4. Los seis pares inferiores	6
Figura 1.5. a) Cadena cinemática abierta, b) Cadena cinemática cerrada.....	7
Figura 1.6. Ángulos de Euler XYZ.....	9
Figura 1.7. Relación entre la cinemática directa e inversa.....	11
Figura 1.8. Asignación de marcos de referencia a cada eslabón del robot.	12
Figura 1.9. Diagrama de relación entra las Jacobianas geométricas directa e inversa	14
Figura 1.10. Modelo de un eslabón de masa concentrada.	16
Figura 1.11. Funcionamiento del control cinemático.	20
Figura 1.12. Generación de trayectorias en el espacio cartesiano (arriba) y articulado (abajo).....	22

Figura 1.13. Perfiles de posición, velocidad y aceleración del interpolador trapezoidal.	26
Figura 1.14. Diagrama de bloques del control PID.....	27
Figura 1.15. Elementos básicos de un sistema mecatrónico.....	28
Figura 1.16. Diagrama de bloques de un sistema de control en lazo abierto.....	30
Figura 1.17. Diagrama de bloques de un sistema de control en lazo cerrado.	30
Figura 1.18. Elementos de un sistema de control digital en lazo cerrado.	31
Figura 2.1. Diagrama funcional: nivel 0.	34
Figura 2.2. Diagrama funcional: nivel1.	34
Figura 2.3. Ubicación manual del efector en los puntos de trayectoria.	36
Figura 2.4. Puntos y orientaciones de la trayectoria deseada.	36
Figura 2.5. Logo de Matlab	37
Figura 2.6. Logo de Python.....	38
Figura 2.7. Logo de C++.	38
Figura 2.8. Estructura mecánica del manipulador robótico.	42
Figura 2.9. Servomotor AC.....	42
Figura 2.10. Servomotor DC.	43
Figura 2.11. Motor a pasos.	44
Figura 2.12. Reductor.....	44
Figura 2.13.Efector final del manipulador.	45
Figura 2.14. Controlador de motor eléctrico.	45
Figura 2.15. Porta electrodo para soldadura SMAW.....	49
Figura 2.16. Soplete para soldadura de arco.	49
Figura 2.17. Solución conceptual del manipulador robótico.....	53
Figura 2.18. Partes y dimensiones del manipulador.	54
Figura 2.19. Volumen de trabajo el manipulador robótico.....	55
Figura 2.20. División del brazo manipulador en tres cuerpos distintos.....	56
Figura 2.21. Diagrama de cuerpo libre del mecanismo.....	57
Figura 2.22. Diagrama de cuerpo libre del cuerpo 1.	57
Figura 2.23. Diagramas de fuerza cortante y momento flector del cuerpo 1.....	58
Figura 2.24. Diagrama de cuerpo libre del cuerpo 2.	58
Figura 2.25. Diagramas de fuerza cortante y momento flector del cuerpo 2.....	59
Figura 2.26. Catálogo de tubos redondos de aluminio de la marca Cedal.....	60
Figura 2.27. Malla generada en ANSYS.....	62
Figura 2.28. Cargas y soportes del modelo de estudio.	62
Figura 2.29. Distribución de las tensiones de Von Mises.....	63
Figura 2.30. Distribución de los esfuerzos de Von Mises en el cuerpo crítico.	64

Figura 2.31. Distribución de factores de seguridad.	64
Figura 2.32. Distribución de factores de seguridad en el cuerpo crítico.	65
Figura 2.33. Deformación direccional en el cuerpo 1.	65
Figura 2.34. Deformación direccional del cuerpo 2.	66
Figura 2.35. Diagrama de cuerpo libre del cuerpo 1.	67
Figura 2.36. Diagramas de fuerza cortante y momento flector del cuerpo 1.	68
Figura 2.37. Diagrama de cuerpo libre del cuerpo 2.	68
Figura 2.38. Diagramas de fuerza cortante y momento flector del cuerpo 2.	69
Figura 2.39. Catálogo de tubos redondos de aluminio de la marca Cedal.	69
Figura 2.40. Malla generada en Ansys.	70
Figura 2.41. Cargas y soportes del modelo de estudio.	71
Figura 2.42. Distribución de los esfuerzos de Von Mises.	72
Figura 2.43. Distribución de esfuerzos de Von Mises en el cuerpo crítico.	72
Figura 2.44. Distribución de factores de seguridad.	73
Figura 2.45. Distribución de factores de seguridad del cuerpo crítico.	73
Figura 2.46. Deformación direccional del cuerpo 1.	74
Figura 2.47. Deformación direccional del cuerpo 2.	74
Figura 2.48. Modelo final del manipulador robótico.	76
Figura 2.49. Modelo CAD final del manipulador y la antorcha de soldadura.	77
Figura 2.50. Diagrama de bloques del manipulador importado a Simulink.	77
Figura 2.51. Vista en Matlab del manipulador importado.	78
Figura 2.52. Marcos de referencia para los eslabones del manipulador.	78
Figura 2.53. Relación entre el modelo importado y el modelo robótico.	79
Figura 2.54. Bloques pertenecientes a cada elemento del modelo robótico.	79
Figura 2.55. Sistemas de coordenadas locales de cada cuerpo del robot.	80
Figura 2.56. Programa generador de trayectorias para el control cinemático.	85
Figura 2.57. Elementos constituyentes del sistema mecatrónico del manipulador.	88
Figura 2.58. Comparación de rendimiento de los actuadores Dynamixel.	89
Figura 2.59. Servomotor XM540-W270-R.	90
Figura 2.60. Controlador OpenCR 1.0 y sus partes.	91
Figura 2.61. Placa de expansión OpenCm 485.	91
Figura 2.62. Diagrama bloques del controlador PID del servomotor Dynamixel.	93
Figura 2.63. Diagrama de bloques del manipulador realizado en Simulink.	94
Figura 2.64. Conexiones de las entradas y salidas de los elementos del manipulador.	94
Figura 2.65. Entradas y sensores ubicados en cada articulación del robot.	95
Figura 3.1. Elementos del Sistema mecatrónico.	95

Figura 3.2. Manipulador robótico articulado de seis grados de libertad.	97
Figura 3.3. Volumen de trabajo del manipulador robótico usando Robotics System ...	98
Figura 3.4. Definición de las geometrías rectangulares de movimiento.	99
Figura 3.5. Tiempos establecidos para el movimiento del manipulador.	100
Figura 3.6. Robot manipulador y trayectorias cartesianas.	100
Figura 3.7. Manipulador ubicado en el primer punto de la trayectoria.	101
Figura 3.8. Movimiento del manipulador en la primera trayectoria cuadrada.	101
Figura 3.9. Movimiento del manipulador en la segunda trayectoria cuadrada.	102
Figura 3.10. Movimiento del manipulador entre trayectorias especificadas.	102
Figura 3.11. Configuración del robot en su posición inicial.	103
Figura 3.12. Movimiento del manipulador en el espacio articulado.	104
Figura 3.13. Movimiento del manipulador en el espacio cartesiano.	105
Figura 3.14. Torques requeridos para el movimiento en la trayectoria definida.	106
Figura 3.15. Recorrido realizado por el extremo del manipulador.	107
Figura 3.16. Vista del brazo manipulador generado con Simscape con las trayectorias cuadradas.	108
Figura 3.17. Funcionamiento del manipulador robótico.	108
Figura 3.18. Funcionamiento del manipulador robótico.	109
Figura 3.19. Movimiento del manipulador en el espacio articulado producido en Simulink.	110
Figura 3.20. Movimiento del manipulador en el espacio cartesiano producido en Simulink.	111
Figura 3.21. Torques requeridos para el movimiento producido en Simulink.	112
Figura 3.22. Recorrido realizado por el manipulador.	113
Figura 3.23. Recorrido del manipulador generado en Matlab (azul) y obtenido en Simulink (rojo).	114

ÍNDICE DE TABLAS

Tabla 1.1. Clasificación de las aplicaciones de los robots industriales manipuladores..	4
Tabla 1.2. Comparación entre los espacios de las articulaciones y el cartesiano.....	22
Tabla 2.1. Especificaciones técnicas del manipulador robótico	33
Tabla 2.2. Alternativas de solución del módulo 1.	39
Tabla 2.3. Evaluación de peso específico de cada criterio.	40
Tabla 2.4. Evaluación del peso específico: facilidad de simulación.....	40
Tabla 2.5. Evaluación del peso específico: precisión.....	40
Tabla 2.6. Evaluación del peso específico: complejidad.	40

Tabla 2.7. Evaluación del peso específico del criterio: facilidad de simulación.....	41
Tabla 2.8. Conclusión de la evaluación de alternativas.	41
Tabla 2.9. Alternativas de solución del módulo 2.	46
Tabla 2.10. Evaluación de peso específico de cada criterio.	47
Tabla 2.11. Evaluación del peso específico del criterio: precio.....	47
Tabla 2.12. Evaluación del peso específico del criterio: precisión.....	47
Tabla 2.13. Evaluación del peso específico del criterio: tamaño.	48
Tabla 2.14. Evaluación del peso específico del criterio: energía.	48
Tabla 2.15. Conclusión de la evaluación de alternativas.	48
Tabla 2.16. Alternativas de solución del módulo 3.	50
Tabla 2.17. Evaluación de peso específico de cada criterio.	51
Tabla 2.18. Evaluación del peso específico del criterio: precio.....	51
Tabla 2.19. Evaluación del peso específico del criterio: precisión.....	51
Tabla 2.20. Evaluación del peso específico del criterio: facilidad de implementación.	51
Tabla 2.21. Evaluación del peso específico del criterio: tiempo.....	52
Tabla 2.22. Conclusión de la evaluación de alternativas.	52
Tabla 2.23. Relación de longitud del barco y antebrazo de diferentes manipuladores.	53
Tabla 2.24. Propiedades físicas de los subcuerpos establecidos.....	56
Tabla 2.25. Reacciones y momentos obtenidos en Ansys.....	63
Tabla 2.26. Propiedades físicas de los subcuerpos establecidos.....	67
Tabla 2.27. Reacciones y momentos obtenidos en Ansys.....	71
Tabla 2.28. Variables para elegir los actuadores.	88
Tabla 2.29. Especificaciones técnicas del servo motor XM540-W270-R.	90
Tabla 2.30. Ganancias del controlador Dynamixel.	92
Tabla 3.1. Características técnicas del mecanismo.	96
Tabla 3.2. Costo de los componentes del brazo manipulador	114

RESUMEN

El presente documento detalla el proceso de diseño y simulación de un manipulador articulado para soldadura en carpintería metálica destinado al uso en las MiPymes manufactureras del país para ayudar a mejorar sus procesos de producción.

Primero se realizó el diseño mecánico del producto usando ingeniería concurrente. Como parte de este proceso se hizo el diseño conceptual, en el que se desarrolló el análisis funcional, definición de módulos y elección de las soluciones. Luego, en el diseño de materialización se realizaron cálculos y análisis estructurales de los componentes del manipulador y se establecieron las medidas del producto, todo esto materializado en un modelo CAD del robot.

El siguiente paso fue la creación de un modelo robótico. Usando Simulink, Simscape Multibody y Robotics System Toolbox se importó el modelo CAD del producto a Matlab y se creó un modelo robótico. Con este modelo se realizó un programa generador de trayectorias en el cual se generan los comandos de posiciones, velocidades, aceleraciones y torques para que el robot realice el proceso de soldadura en trayectorias rectangulares.

Luego, se eligieron los componentes del sistema mecatrónico del robot. Entre estos estuvieron los servomotores y controlador. Con estos elementos se seleccionó un sistema de control dinámico.

Finalmente se realizó una simulación en Simulink del comportamiento del manipulador robótico para evaluar su correcto funcionamiento.

Como resultado se obtuvo el diseño del manipulador que cumple con los requerimientos expresados en las especificaciones técnicas y con un costo menor al de los productos existentes en el mercado.

Palabras clave: Diseño, manipulador, mecatrónica, robótica, soldadura, trayectorias.

ABSTRACT

This document details the design and simulation process of an articulated manipulator for welding in metalwork meant to be used in the manufacturing MSMEs of the country to help improve their production processes.

First, the mechanical design of the product was performed using concurrent engineering. As part of this process, the conceptual design was carried out, in which the functional analysis, definition of modules and choice of solutions were developed. Then, in the materialization design, calculations and structural analyzes of the manipulator components were accomplished and the product measurements were established, all this materialized in a CAD model of the robot.

The next step was the creation of a robotic model. Using Simulink, Simscape Multibody and Robotics System Toolbox, the CAD model of the product was imported into the Matlab program and a robotic model was created. With this model, a trajectory generator program was created in which the positions, speeds, accelerations and torques commands are generated for the robot to carry out the welding process on rectangular trajectories.

Then the components of the robot's mechatronic system were chosen. Among these were the servo motors and controller. With these elements, a dynamic control system was selected.

Finally, a simulation in Simulink of the behavior of the robotic manipulator was performed to assess its correct operation.

As a result, a design of the manipulator that meets the requirements expressed in the technical specifications was obtained and with a lower cost than the existing products on the market.

Keywords: Design, manipulator, mechatronics, robotics, welding, trajectories.

DISEÑO Y SIMULACIÓN DE UN MANIPULADOR ROBÓTICO ARTICULADO PARA SOLDADURA POR ARCO PARA CARPINTERIA METÁLICA

INTRODUCCIÓN

Según [1], una de las principales actividades económicas a las que se dedican las MiPymes manufactureras del país es la fabricación de productos metálicos. En la actualidad, estas empresas son menos productivas que aquellas de países industrializados, debido al poco desarrollo tecnológico en nuestro medio. Esto se refleja en una industria menos competitiva.

Uno de los procesos con más importancia en la industria metalmecánica es la aplicación de soldadura para unir y modificar elementos metálicos. Este es un proceso indispensable que brinda aspectos esenciales para la calidad y el precio de sus procedimientos. Mediante una correcta aplicación de la soldadura se puede tener una relación calidad-precio excelente [2].

En base a lo anteriormente presentado, en la industria ecuatoriana es necesario que se adopten de nuevas tecnologías que permitan aumentar la productividad. Los brazos robóticos son una excelente alternativa para esta aplicación en las pequeñas empresas dedicadas a la metalmecánica. Sin embargo, los manipuladores industriales fabricados en el extranjero son muy costosos, por lo que es necesaria una opción que sea más asequible económicamente.

Por consiguiente, surge la necesidad de desarrollar un brazo robótico que sea de utilidad en las MiPymes de manufactura metalmecánica del Ecuador que ayude a mejorar los procesos productivos que emplean soldadura.

Objetivo general

Diseñar y simular un manipulador robótico articulado para soldadura por arco para carpintería metálica.

Objetivos específicos

- Analizar la cinemática directa e inversa del robot para determinar el volumen de trabajo.
- Analizar la dinámica inversa del manipulador para determinar los parámetros requeridos por los actuadores.
- Diseñar el sistema mecatrónico del manipulador para controlar su movimiento.
- Simular el comportamiento del brazo robótico al seguir una trayectoria especificada para validar el diseño propuesto.

1. MARCO TEÓRICO

1.1. Robótica

1.1.1. Origen y desarrollo de la robótica [3]

La historia de la automatización industrial está caracterizada por cambios rápidos en los métodos populares existentes. Estos períodos de cambio en la tecnología de automatización parecen estar estrechamente relacionados con la economía mundial. El uso de robots industriales, en conjunto con sistemas de diseño asistido por computadora (CAD) y sistemas de fabricación asistida por computadora (CAM) representan las últimas tendencias en tecnología. Estas tecnologías están liderando otra transformación de la automatización industrial, cuyo alcance aún se desconoce.

El decremento de costos es uno de los principales motivos del incremento del uso de robots industriales. La figura 1.1 muestra que, en la década de 1990, los precios de los robots cayeron mientras que los costos laborales aumentaron. Además, los robots no solo son cada vez más baratos, sino también más eficaces: más rápidos, más flexibles y precisos. Si incluimos estos ajustes de calidad en los números, el costo de usar robots caerá incluso más rápido que sus precios. A la par que los robots se vuelven más rentables en el trabajo y los costos laborales aumentan, más trabajos industriales se convierten en candidatos para la automatización usando robots.

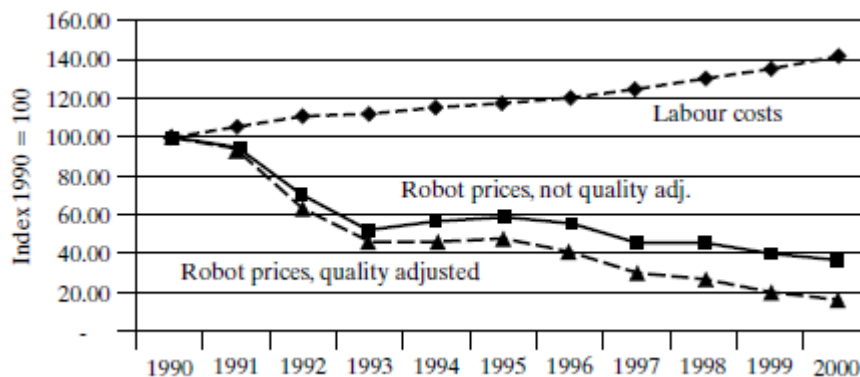


Figura 1.1. Precios de los robots comparados a los costos de mano de obra en la década de 1990. (Fuente: Craig, [3])

El estudio de los manipuladores en su parte mecánica y control no es una ciencia nueva, sino simplemente una colección de temas seleccionados de campos "clásicos". La ingeniería mecánica aporta metodologías para el análisis de máquinas en condiciones estáticas y dinámicas. Las matemáticas proporcionan herramientas para describir el movimiento espacial y otras propiedades de los manipuladores. La teoría de control brinda herramientas para el diseño y evaluación de algoritmos que permitan realizar movimientos especificados o aplicaciones de fuerza. La ingeniería eléctrica se emplea para diseñar

sensores e interfaces para robots industriales, y la informática proporciona una base para programar estos dispositivos para realizar las tareas especificadas.

1.1.2. Definición de robot manipulador industrial

La Asociación Internacional de Estándares (ISO) define al robot manipulador industrial en la norma ISO 8373, como:

Manipulador de tres o más ejes, controlado automáticamente, reprogramable, para múltiples aplicaciones, que puede ser móvil o fijo, para uso en aplicaciones de automatización industrial.

De esta forma, los manipuladores con control de tipo encendido/apagado (donde solo se pueden alcanzar posiciones discretas) serán excluidos de la definición de robot. En la práctica, estos sistemas corresponden a manipuladores controlados por un sistema lógico (por ejemplo, PLC), que tiene un pequeño número de posiciones estables discretas y el actuador se detiene en una posición fija. El cambio de estas posiciones significa cambiar la trayectoria del actuador a través de topes mecánicos o incluso rediseñar o cambiar su estructura.

1.1.3. Clasificación de los robots [4]

1.1.3.1. Clasificación según el área de aplicación

Desde la perspectiva del uso que se da al robot, estos se pueden clasificar según el sector económico en el que se incorporan o según el tipo de tarea o aplicación que realicen. La IFR (Federación Internacional de Robótica) define las aplicaciones que tiene los robots industriales.

Tabla 1.1. Clasificación de las aplicaciones de los robots industriales manipuladores

Sin especificar
Manipulación en fundición
Manipulación en moldeo de plásticos
Manipulación en tratamientos térmicos
Manipulación en la forja y estampación
Soldadura
Aplicación de materiales
Mecanizado
Otros procesos
Montaje
Paletización y etiquetado
Medición, inspección y control de calidad
Manipulación de materiales
Formación, enseñanza e investigación
Otros

(Fuente: Barrientos et al., [4])

1.1.3.2. Clasificación según la configuración

En casi la totalidad de los casos, los robots manipuladores son cadenas cinemáticas abiertas que poseen articulaciones giratorias o prismáticas. Dado que los grados de libertad del robot son generalmente los mismos que el número de articulaciones, es fácil encontrar este número.

El uso de combinaciones de articulaciones diferentes en el robot producirá varias configuraciones, cuyas características deben ser consideradas en el diseño y construcción de robots y sus aplicaciones. Para los robots con cadenas cinemáticas abiertas, las combinaciones más comunes son las que se observan en la Figura 1.2, que solo involucran las tres juntas iniciales del robot, siendo estas las más importantes al posicionar el extremo del robot en un punto espacial establecido.

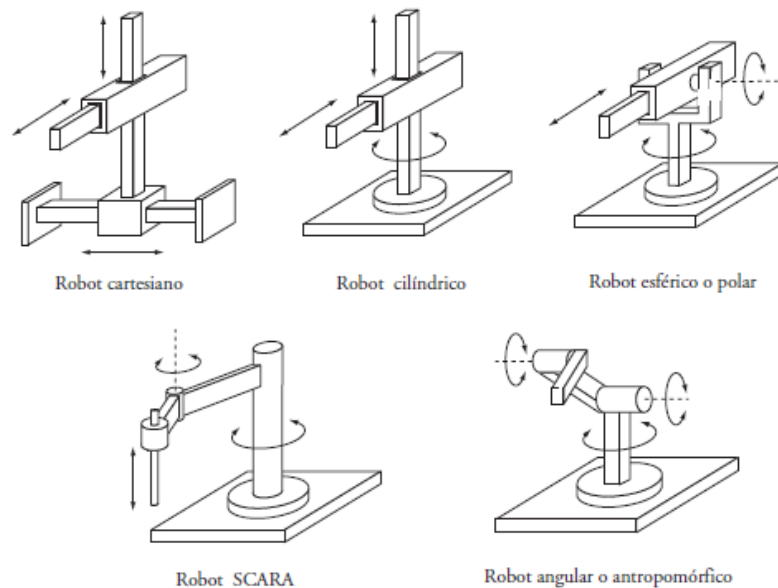


Figura 1.2. Configuraciones más frecuentes en robots industriales.
(Fuente: Barrientos et al., [4])

1.1.4. Morfología del robot [4]

Un robot está compuesto mecánicamente por una cadena de eslabones o enlaces conectados por juntas. Estas articulaciones facultan que cada par de enlaces consecutivos tengan un movimiento relativo. La mayoría de los robots industriales tienen estructuras físicas con semejanzas con la estructura anatómica del brazo humano, en muchas ocasiones se utiliza este como referencia para los diferentes componentes del manipulador.

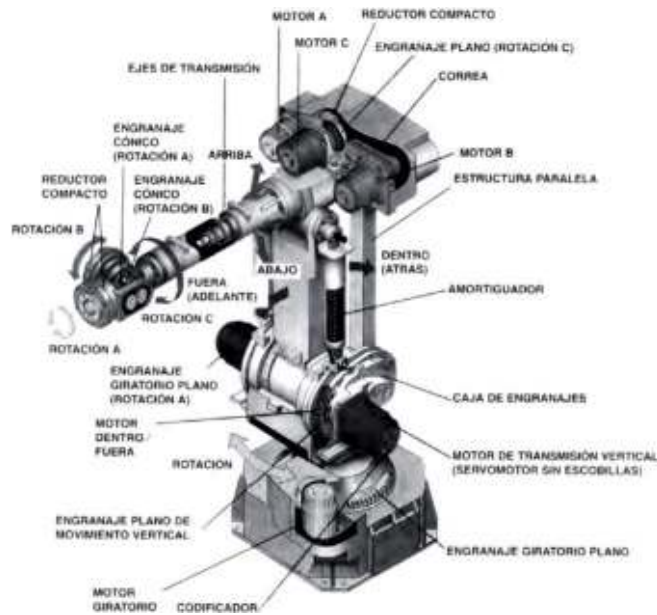


Figura 1.3. Ejemplo partes de un robot y su estructura mecánica.
(Fuente: Barrientos et al., [4])

Las articulaciones de los robots tienen capacidad para desplazarse, girar o tener un movimiento combinado de estos dos. Hay dos probables movimientos relativos entre estos cuerpos en contacto. El par inferior se refiere al par que hace contacto entre superficies. Por el contrario, si el contacto es en un punto o línea, se lo denomina par superior. En la Figura 1.4 se representan los seis posibles pares inferiores.

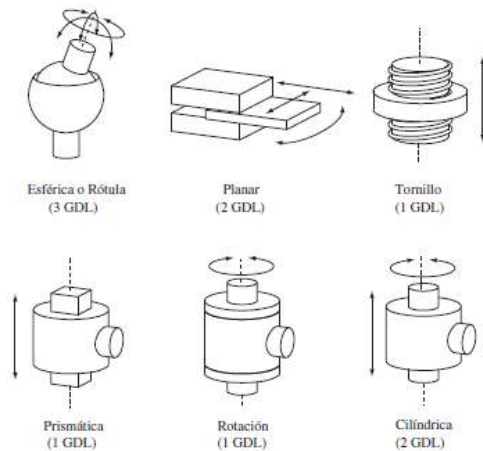


Figura 1.4. Los seis pares inferiores
(Fuente: Barrientos et al., [4])

Cada movimiento independiente que cada articulación puede realizar en relación con la articulación anterior se llama grado de libertad. La cantidad de grados de libertad de cada articulación correspondiente al par inferior se puede ver en la Figura 1.4. Se puede ver que solo tres clases de articulaciones poseen un grado de libertad único.

En robótica solo se utilizan juntas de revolución y prismáticas. Si una articulación del robot tiene varios grados de libertad, se puede suponer que son varias articulaciones diferentes y están conectadas por eslabones de longitud cero.

A la sucesión de cuerpos o eslabones ensamblados entre sí por articulaciones se le llama cadena cinemática. La estructura mecánica de los robots manipuladores se compone una cadena cinemática. Cuando se puede llegar a una cadena cinemática desde un eslabón cualquiera a otro a través de por lo menos dos trayectos, se puede decir que es una cadena cinemática cerrada. Cuando existe solo una ruta, estamos ante el caso una cadena cinemática abierta. Esto se observa en la Figura 1.5.

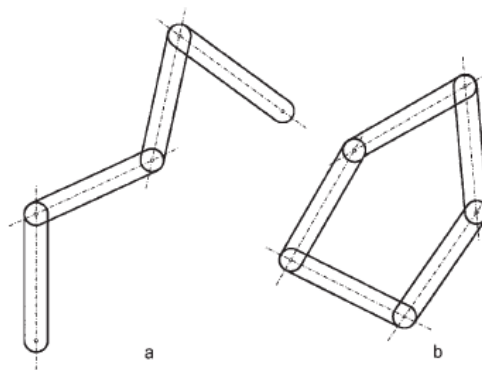


Figura 1.5. a) Cadena cinemática abierta, b) Cadena cinemática cerrada
(Fuente: Barrientos et al., [4])

1.1.4.1. Manipulador Articulado

A veces también denominado manipulador articulado, antropomórfico o de codo. Un manipulador de esta clase normalmente tiene dos articulaciones de "hombro" (una para la rotación alrededor de un eje vertical y otra para la elevación fuera del plano horizontal), una articulación de "codo" (generalmente con un eje paralelo a la articulación de elevación del hombro) y dos o tres articulaciones de muñeca en su extremo.

Los robots articulados reducen la posibilidad de que la estructura del manipulador invada el volumen de trabajo, permitiéndoles entrar en espacios estrechos. En comparación con los robots cartesianos, requieren una estructura general más pequeña, por lo que son menos costosos para aplicaciones que requieren un espacio de trabajo más pequeño [3].

1.1.5. Herramientas matemáticas para la ubicación espacial [4]

1.1.5.1. Representación de la posición

La posición de un cuerpo rígido en el espacio se debe detallar usando su posición y orientación al mismo tiempo. Ambos deben establecerse en relación con el sistema de referencia definido, y se pueden utilizar diferentes modos para definir qué relación existe entre la locación y la orientación del cuerpo y el sistema de referencia.

En un plano de dos dimensiones, se necesitan dos grados de libertad para obtener ubicación de un cuerpo rígido y con dos componentes independientes se define esta variable. Para tres dimensiones se aplica el mismo criterio, solamente se debe aumentar un componente independiente extra para la nueva dimensión.

1.1.5.1.1. Sistema cartesiano de referencia

El sistema de referencia suele estar definido por ejes que son perpendiculares entre sí y definidos a partir de un origen. Los sistemas cartesianos se usan cuando se trabaja en un plano, el sistema de referencia correspondiente OXY está definido con dos vectores perpendiculares entre sí de coordenadas OX y OY y O es su intersección común. Si trabajamos en el espacio, OXYZ es el sistema cartesiano formado por triples ortogonales de vectores unitarios OX, OY y OZ.

1.1.5.2. Representación de la orientación

Un punto se puede definir completamente en el espacio mediante información de su posición. No obstante, al tratar con cuerpos rígidos, la orientación con base al marco de referencia también se debe establecer. Adicionalmente, no es suficiente la especificación de la posición del extremo de los robots, sino que suele ser requerido definir su orientación. La orientación en el espacio de tres dimensiones está representada mediante tres grados de libertad independientes. Para definir fácilmente la orientación de un objeto en relación con el sistema de referencia, generalmente se asigna un nuevo sistema al objeto en conjunto, y luego se estudia la relación espacial entre ambos sistemas. Generalmente, esta relación estará dada por la ubicación y orientación del sistema asociado con el objeto, en relación con el objeto de referencia.

1.1.5.2.1. Ángulos Euler XYZ

Estas rotaciones alrededor de un eje fijo se denominan guiñada, cabeceo y balanceo. Esta es una representación comúnmente utilizada en la industria de la aviación. Este también es el método más común cuando se gira sobre el eje de un sistema fijo. Si comienza con los sistemas OXYZ y OUVW, se pueden seguir los pasos a continuación para colocar el sistema OUVW en cualquier orientación. Esto se puede ver en la Figura 1.6.

1. Rotar un ángulo ϕ el sistema con relación al eje OZ. Llamado guiñada o yaw.
2. Rotar un ángulo θ el sistema con relación al eje OY. Llamado cabeceo o pitch.
3. Rotar un ángulo ψ el sistema con relación al eje OX. Llamado alabeo o roll.

Generalmente, siempre que se conecten varios giros consecutivos, se debe tener en que esta no es una transformación conmutable y se debe seguir la secuencia de aplicación específica de estas transformaciones.

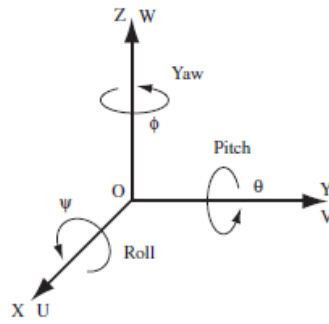


Figura 1.6. Ángulos de Euler XYZ.
(Fuente: Barrientos et al., [4])

1.1.5.3. Matrices de transformación homogénea

Hay muchas formas de expresar la localización o la orientación de un cuerpo en el espacio. Sin embargo, ninguno de estos métodos es capaz de definir la representación conjunta de ubicación y orientación por sí solos. Las matrices de transformación homogéneas permiten esta representación conjunta, y, juntamente con álgebra matricial se facilita su uso.

1.1.5.3.1. Coordenadas y matrices homogéneas

Por medio de coordenadas homogéneas se pueden representar y procesar de manera conjunta la ubicación y orientación de los cuerpos. Un componente del espacio n -dimensional se representa en coordenadas homogéneas $n + 1$ dimensionales. El vector $p(x, y, z)$ estará representado por $p(w_x, w_y, w_z, w)$, donde el factor de escala se representa con un valor w cualquiera. Generalmente, el vector $p = ai + bj + ck$, donde i, j y k son los vectores unitarios de los ejes OX, OY y OZ respectivamente del marco de referencia $OXYZ$, expresados como coordenadas homogéneas por vectores columna:

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} aw \\ bw \\ cw \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} \quad (1)$$

Los vectores vacíos se suelen representar usando $[0, 0, 0, n]^T$, aquí, n no es nulo. Cuando se requiere representar direcciones los vectores $[a, b, c, 0]^T$ son útiles, debido a que representan vectores infinitamente largos.

El concepto de matriz de transformación homogénea aparece en base a la definición de coordenadas homogéneas. La matriz de transformación homogénea T se define como una matriz de 4×4 , que simboliza la transformación de un vector de coordenadas homogéneo entre sistemas de coordenadas.

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} \quad (2)$$

La matriz homogénea se puede pensar como una composición de cuatro submatrices de diferentes tamaños: la submatriz $R_{3 \times 3}$ representa a la matriz de rotación; el vector de

traslación, definido por la submatriz $p_{3 \times 1}$; la submatriz $f_{1 \times 3}$ que corresponde a la transformación de perspectiva y la submatriz $w_{1 \times 1}$ que atañe el escalado global. En los robots, se considera los componentes de $f_{1 \times 3}$ como nulos y $w_{1 \times 1}$ como la unidad, y solo interesa conocer los valores de $R_{3 \times 3}$ y $p_{3 \times 1}$. Al ser una matriz (4×4) , los vectores sobre los que se apliquen deben tener cuatro dimensiones, que definen las coordenadas homogéneas del vector en tres dimensiones que se trata.

1.1.5.3.2. Aplicación de matrices homogéneas

Al tomar el escalado global unitario y la transformación de perspectiva nula, se tiene que la matriz homogénea T es:

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix} \quad (3)$$

Usando esta expresión, es posible establecer la posición y orientación de un marco $O'UVW$ generado al trasladar y girar el sistema base $OXYZ$ en base a $p_{3 \times 1}$ y $R_{3 \times 3}$. De manera similar, esta matriz se puede utilizar para saber la locación (r_x, r_y, r_z) del vector r en el marco $OXYZ$ en base a su ubicación (r_u, r_v, r_w) en el marco $O'UVW$:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = T \begin{bmatrix} r_u \\ r_v \\ r_w \\ 1 \end{bmatrix} \quad (4)$$

También se puede utilizar para representar la traslación y rotación de un vector con respecto al marco de referencia $OXYZ$ fijo, de modo que el vector r_{xyz} girado según $R_{3 \times 3}$ y desplazado según $p_{3 \times 1}$ se transforma en el vector r'_{xyz} definido mediante:

$$\begin{bmatrix} r'_u \\ r'_v \\ r'_w \\ 1 \end{bmatrix} = T \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} \quad (5)$$

1.1.6. Cinemática del robot [4]

La cinemática del robot requiere resolver dos problemas básicos; el primero de los cuales se llama problema cinemático directo, que permite calcular la locación y dirección del extremo final del robot en relación con el sistema de coordenadas utilizado como referencia y conocer los valores y parámetros geométricos de las articulaciones del robot; el siguiente método se denomina problema cinemático inverso, que calcula la configuración a adoptar del robot cuando se conoce su ubicación y orientación en el extremo. Estos problemas se representan en la Figura 1.7.

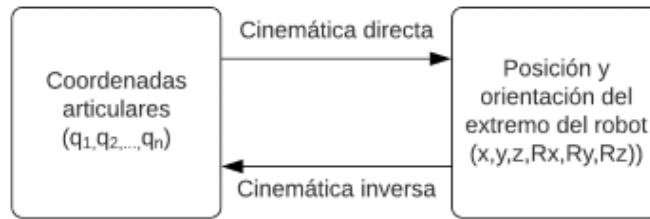


Figura 1.7. Relación entre la cinemática directa e inversa.
(Fuente: Barrientos et al., [4])

1.1.6.1. Cinemática directa

A través de la solución del problema de cinemática directa, es posible saber la ubicación y orientación en el extremo del robot cuando cada variable de la posición o dirección fija de la articulación toma un valor específico.

Dado que las variables de articulación se pueden leer de los sensores en la unidad de control del robot, el modelo de movimiento directo se utilizará para otros fines para proporcionar a los usuarios información sobre la posición final del robot.

Por lo tanto, si se seleccionan coordenadas cartesianas y ángulos de Euler para definir la ubicación y orientación del extremo del robot de seis grados de libertad, el problema de cinemática directa se resuelve con la siguiente relación:

$$(x, y, z, \phi, \theta, \psi) = f(q_1, q_2, q_3, q_4, q_5, q_6) \quad (6)$$

El modelo de cinemática directa se puede obtener mediante dos métodos diferentes, que se denominan métodos geométricos y métodos fundamentados en el cambio de marcos de referencia.

Estos primeros son aptos para situaciones sencillas, pero tienen el inconveniente de no ser sistemáticos, por este motivo su aplicación se limita a robots sin muchos grados de libertad. Los métodos fundamentados en la transformación de marcos de referencia logran obtener de forma sistemática el modelo de cinemática directa de robots con n grados de libertad, por lo que estos modelos son los más usados, especialmente aquellos que se basan en matrices de transformación homogénea.

1.1.6.1.1. Resolución del problema cinemático directo mediante matrices de transformación homogénea

En la base del robot es posible crear un marco de referencia fijo y especificar la ubicación de cada eslabón en relación con el sistema de referencia. De tal manera que el problema de la cinemática directa pasa a ser hallar una matriz de transformación homogénea T que vincule la ubicación y orientación del extremo del robot con respecto a un marco de referencia fijo en su base. La matriz T dependerá de coordenadas de las articulaciones.

Un robot con n grados de libertad consta de n enlaces y n uniones, por lo que un grado de libertad está conformado por cada par de elemento-articulación. Cada elemento se puede asignar a un sistema de referencia y, se puede representar la rotación y traslación relativa entre los diferentes cuerpos constituyentes el robot con la transformación homogénea. Por lo común, se denomina ${}^{i-1}A^i$ a la matriz de transformación homogénea que determina la ubicación y orientación relativa entre los marcos asignados a dos elementos conectados del robot. Para expresar de forma parcial o total la cadena cinemática que conforma el robot, se usan las matrices ${}^0A^k$ que se obtienen del producto de las matrices ${}^{i-1}A^i$ con i entre 1 y k .

A la matriz ${}^0A^n$ generalmente se le llama T al considerar todos los grados de libertad. Por tanto, la locación y dirección del extremo de un robot con seis grados de libertad, estará especificada por la matriz T :

$${}^0A^6 = T = {}^0A^1 {}^1A^2 {}^2A^3 {}^3A^4 {}^4A^5 {}^5A^6 \quad (7)$$

El cambio de base se representa con cada matriz ${}^{i-1}A^i$, esto permite pasar entre los marcos del elemento $i-1$ y el elemento i . Adicionalmente, las constantes geométricas vinculadas al eslabón y los grados de libertad q_i influyen en esta matriz.

La matriz de transformación homogénea T describe la correspondencia entre los sistemas de coordenadas de la base y del extremo, que es dependiente de las coordenadas articulares correspondientes a la posición establecida para el robot. Mediante esta igualdad se puede calcular la respuesta al problema cinemático directo.

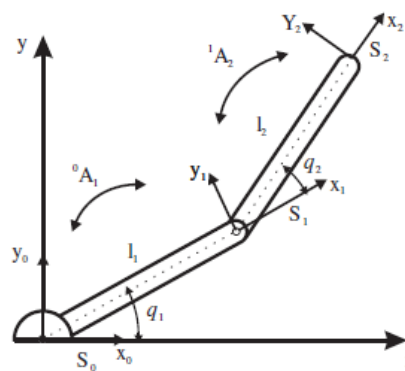


Figura 1.8. Asignación de marcos de referencia a cada eslabón del robot.
(Fuente: Barrientos et al., [4])

1.1.6.2. Cinemática inversa

El propósito del problema cinemático inverso es hallar las coordenadas articulares del robot $q = [q_1, q_2, \dots, q_n]^T$ que se debe adoptar para ubicar y orientar el extremo del robot de acuerdo con una localización espacial específica.

Al contrario del problema cinemático directo que se puede solucionar de una manera metódica con base en el uso de matrices de transformación homogéneas independientemente de la configuración del robot, lo mismo no es posible para abordar el problema cinemático inverso. El proceso de obtención de las ecuaciones depende en gran medida de la configuración del robot.

1.1.6.2.1. Resolución del problema cinemático directo mediante métodos numéricos [5]

La cinemática inversa mediante métodos numéricos son técnicas muy generales que pueden acomodar mecanismos de robot arbitrarios y espacios de tareas, sin ninguna necesidad de matemáticas específicas del mecanismo además de cinemática directa y cálculos jacobianos. Como resultado, son preferibles para su uso en paquetes de software que sean compatibles con múltiples tipos de robots. Sin embargo, tienen algunos inconvenientes: involucran mucho más cálculo que enfoques analíticos, solo brindan soluciones puntuales y pueden caer en los mínimos locales. Los profesionales de la robótica deben ser conscientes de estos inconvenientes al construir sistemas basados en técnicas numéricas de cinemática inversa.

Todos los enfoques numéricos comparten una estructura iterativa común. Primero, sea $f(q)$ el error de espacio de la tarea. Estos métodos reciben una configuración semilla q_0 y la mejoran iterativamente para obtener una secuencia q_1, q_2, \dots, q_k con valores de error decrecientes $\|f(q_i)\|$. La iteración puede terminar cuando el error disminuye por debajo de algún umbral, o cuando i excede un límite de iteración k .

1.1.6.3. Matriz Jacobiana

Un objetivo de hallar el modelo cinemático de un robot es encontrar la correspondencia que tienen la ubicación y orientación del extremo del robot con las variables articulares. Aquí, se ignoran las fuerzas o pares que actúan sobre el robot que pueden modificar sus condiciones de movimiento. No obstante, la cinemática del robot nos permite saber el efecto del movimiento diferencial de las variables de las articulaciones sobre las variables espaciales cartesianas. Dicha relación está basada en el modelo diferencial. El sistema de control del robot al ser usado puede determinar las velocidades con las que debe controlar a cada articulación para que el extremo trace una trayectoria en el tiempo requerida, como una línea recta con velocidad constante.

El modelo diferencial se define usando la matriz Jacobiana. Con esta matriz se puede asociar el vector de velocidades articulares $(\dot{q}_1, \dot{q}_2, \dot{q}_n)$ con otro de velocidades definido en un espacio diferente. Hay varias alternativas al momento de elegir este espacio.

La primera opción es tomar en cuenta la correspondencia con las velocidades de la ubicación del robot en su extremo, que es la ubicación y orientación representadas por sus coordenadas cartesianas y ángulos de Euler $(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$. Dicha relación en el manipulador viene definida por la matriz Jacobiana analítica.

Otra opción es asociar los vectores de velocidades lineales y angulares $(v_x, v_y, v_z, w_x, w_y, w_z)$ con que se mueve el extremo del robot con las velocidades de las articulaciones, estas velocidades definidas en un marco de referencia que podría ser el origen. La matriz Jacobiana geométrica del manipulador se usa para obtener la relación entre estas velocidades. En los dos casos, la matriz Jacobiana directa se usa para otorgar una relación entre las velocidades del final del robot en función a los componentes de las velocidades que tiene cada articulación.

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (8)$$

1.1.6.4. Jacobiana inversa

De la misma forma que se ha alcanzado la especificación de la relación directa entre velocidades, es posible tener la relación inversa tal que se puedan computar las velocidades de las articulaciones a partir de las del extremo. Se pueden emplear diferentes métodos para obtener esta relación.

Primero, suponiendo que se conoce la relación directa con la matriz Jacobiana, la relación inversa se puede tener invirtiendo la matriz de manera simbólica.

$$\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \dot{q}_n \end{bmatrix} = J^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \quad (9)$$

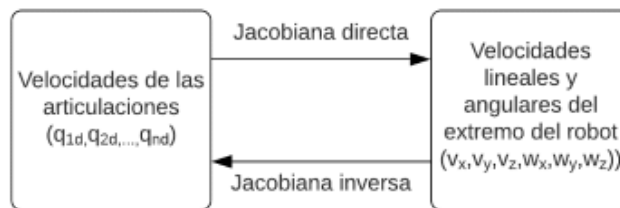


Figura 1.9. Diagrama de relación entra las Jacobianas geométricas directa e inversa (Fuente: Barrientos et al., [4])

En la práctica, esta alternativa es de difícil aplicación. Asumiendo que se tenga una matriz Jacobiana cuadrada 6×6 , la inversión simbólica de esta matriz, con elementos que son funciones generalmente trigonométricas, es muy complicada, por lo que este procedimiento no es factible.

Como segunda opción, se puede considerar evaluar numéricamente la matriz Jacobiana para una posición del robot específica (q_i), y luego invertir de forma numérica esta matriz para hallar la relación inversa correcta para esa configuración. En dicho caso, es necesario considerar que, a medida que se mueve el robot, el valor numérico de la matriz Jacobiana cambia, por lo que la matriz Jacobiana inversa debe ser recalculada continuamente.

1.1.7. Dinámica del robot [4]

La dinámica es la encargada de la correspondencia entre las fuerzas que se ejercen en un cuerpo y el movimiento que causa. Por ende, el objetivo del modelo dinámico de un robot es determinar qué relación existe entre el movimiento del robot y las fuerzas involucradas. Mediante el llamado modelo dinámico es posible obtener esta relación, que forma la relación matemática entre:

- a) La ubicación del robot establecida por las variables de sus articulaciones o mediante las coordenadas de ubicación de su extremo, y sus correspondientes derivadas: velocidad y aceleración.
- b) Las fuerzas y momentos ejercidos en las juntas del robot o en su extremo.
- c) Los parámetros del robot que miden sus dimensiones: longitudes, masas e inercias que tienen sus eslabones.

No es demasiado complicado obtener este modelo para un mecanismo de no más de dos grados de libertad, pero mientras aumenta el número de grados de libertad, la planificación y adquisición del modelo dinámico se vuelve extremadamente complicada. Por tanto, no siempre es factible desarrollar un modelo dinámico mediante un conjunto de ecuaciones, generalmente diferenciales de segundo orden, que al ser integradas permiten saber qué movimiento se producirá cuando se aplica una fuerza o qué fuerza se debe aplicar para tener un movimiento requerido. Por lo que, el modelo cinético debe resolverse iterativamente utilizando métodos numéricos.

Por tanto, una de las cuestiones más complicadas de la robótica es obtener el modelo dinámico de los robots, lo que ha causado que esto sea evitarlo en muchas situaciones. Empero, el modelo dinámico es fundamental para lograr los objetivos en los robots como:

1. Simular su movimiento.
2. Diseñar y evaluar su estructura mecánica.
3. Dimensionar sus actuadores.

4. Diseñar y evaluar el control dinámico.

El último objetivo es muy importante, porque la precisión y la velocidad del movimiento del robot dependen de la calidad del control dinámico establecido. La enorme complejidad que se ha mencionado en el momento de crear el modelo dinámico del robot ha provocado algunas simplificaciones, de tal manera que se pueda utilizar en el diseño de los controladores y también en el sistema de control según sea requerido por la técnica de control especificada.

En el modelo dinámico completo del robot no solo se incluye la dinámica de sus componentes, además, la dinámica de sus sistemas de transmisiones, actuadores y equipos de control electrónico se deben considerar. Estos elementos agregan nuevas inercias, fricciones y saturaciones de circuitos electrónicos en el modelo dinámico, entre otros efectos, lo que aumenta su complejidad aún más.

1.1.7.1. Modelo dinámico de la estructura mecánica de un robot rígido

Para obtener el modelo dinámico de una máquina o mecanismo, especialmente de robots, se usa principalmente el método de equilibrio de fuerzas definido en la segunda ley de Newton, o la ley de Euler que es su equivalente en el caso movimientos giratorios:

$$\begin{aligned}\sum F &= \frac{d}{dt}(mv) \\ \sum T &= \frac{d}{dt}(I\dot{\omega})\end{aligned}\tag{10}$$

Por lo tanto, en el caso inicial de un robot de una articulación como el que se tiene en la Figura 1.10, el equilibrio de fuerzas-momentos resultará en la siguiente ecuación:

$$\begin{aligned}\tau - mgL\cos(\theta) &= I\ddot{\theta} \\ \tau &= I\ddot{\theta} + mgL\cos(\theta)\end{aligned}\tag{11}$$

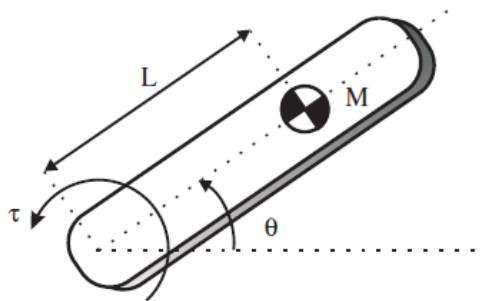


Figura 1.10. Modelo de un eslabón de masa concentrada.
(Fuente: Barrientos et al., [4])

Aquí se supone que la totalidad de la masa está concentrada en el centro de gravedad del cuerpo, Además, no existe fricción alguna ni se manipulan cargas.

Para un torque motor dado τ , al integrar la Ecuación (11) se obtiene la expresión de $\theta(t)$ y sus derivadas $\dot{\theta}(t)$ y $\ddot{\theta}(t)$, a partir de las cuales se puede conocer el cambio de las coordenadas articulares del robot y sus velocidades y aceleraciones.

Por el contrario, si se plantea que $\theta(t)$ evolucione en base a una determinada función del tiempo, se puede usar (11) para obtener el par $\tau(t)$ que debe aplicarse. Si en el extremo del robot se tuviera que ejercer alguna fuerza al manipular un peso o realizar el procesamiento de algún elemento, es suficiente incluir esta circunstancia en la Ecuación (11) y se procede de la misma forma.

Por tanto, los modelos dinámicos directos e inversos se obtienen mediante la formulación del equilibrio de fuerzas y momentos que actúan en el robot:

- Modelo dinámico directo: expresa cómo evoluciona en el tiempo las coordenadas articulares del robot en base a las fuerzas y momentos que actúan $\theta(t) = f(\tau(t))$.
- Modelo dinámico inverso: calcula las fuerzas y torques requeridos para lograr una evolución determinada de las coordenadas de las articulaciones $\tau(t) = g(\theta(t))$.

En un robot real de más de cuatro grados de libertad, la formulación del equilibrio de fuerzas es mucho más complejo que el de un solo eslabón. Debe considerarse que, además de las fuerzas de gravitatorias e inerciales, se presentan fuerzas centrípetas causadas por la rotación y fuerzas de Coriolis, esto por el movimiento relativo que hay entre los eslabones, que son función de la configuración del manipulador en todos los instantes.

El modelo dinámico del robot tiene la siguiente forma:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (12)$$

Donde:

q : vector de coordenadas articulares.

τ : vector de fuerzas o pares que se aplican en las articulaciones.

$M(q)$: matriz de inercias, con dimensión $(n \times n)$, cuyos elementos dependen de q .

$C(q, \dot{q})$: matriz $(n \times n)$ de fuerzas de Coriolis, función de q y \dot{q} .

$g(q)$: matriz $(n \times 1)$ de fuerzas de gravedad, función de q .

n : numero de grados de libertad del robot.

Es necesario tener en cuenta que el vector de torques τ , supone torques efectivos, por lo tanto, existen pares de perturbación o de rozamiento viscoso o seco, que se deben considerar, tal que:

$$\tau = \tau_{\text{motor}} - \tau_{\text{perturbador}} - \tau_{\text{rozamiento viscoso}} - \tau_{\text{rozamiento seco}} \quad (13)$$

Es y ha sido objeto de estudio e investigación la obtención del modelo dinámico de un manipulador. Para obtener modelos que puedan ser manejados por sistemas de cómputo

de una manera más eficiente, muchos investigadores han desarrollado fundamentalmente fórmulas alternas fundamentadas en teorías de mecánica Newtoniana y Lagrangiana.

1.1.7.2. Propiedades del modelo dinámico [6]

1.1.7.2.1. Matriz de inercia

La inercia del espacio articular es una matriz positiva definida y simétrica que es función de la configuración del manipulador. Los elementos diagonales M_{jj} describen la inercia experimentada por la articulación j , es decir, $Q_j = M_{jj}\ddot{q}_j$. Tenga en cuenta que los dos primeros elementos diagonales, correspondientes a las articulaciones de los hombros del robot, son grandes ya que el movimiento de estas articulaciones implica la rotación de los pesados enlaces de la parte superior e inferior del brazo. Los términos fuera de la diagonal $M_{ij} = M_{ji}, i \neq j$ son los productos de la inercia y representan el acoplamiento de la aceleración de la articulación j a la fuerza generalizada en la articulación i .

1.1.7.2.2. Matriz de fuerzas centrífugas y de Coriolis

La matriz de Coriolis C es una función de las coordenadas articulares y la velocidad articular. Los pares centrípetos son proporcionales a \dot{q}_j^2 , mientras que los torques de Coriolis son proporcionales a $\dot{q}_i\dot{q}_j$.

Los términos fuera de la diagonal $C_{i,j}$ representan el acoplamiento de la velocidad de la articulación j a la fuerza generalizada que actúa sobre la articulación i . Dado que los elementos de esta matriz representan un acoplamiento de la velocidad a la fuerza conjunta, tienen las mismas dimensiones que la fricción viscosa o la amortiguación, sin embargo, el signo puede ser positivo o negativo.

1.1.7.2.3. Matriz de torques gravitacionales

El término de gravedad g es una función de las coordenadas de las articulaciones y generalmente es el término dominante en la ecuación del modelo dinámico. Está presente incluso cuando el robot está parado o se mueve lentamente. Algunos robots usan contrapesos o incluso resortes para reducir el par de gravedad que deben proporcionar los motores; esto permite que los motores sean más pequeños y, por lo tanto, de menor costo. El torque ejercido en una articulación debido al efecto de la gravedad depende fuertemente de la pose del robot. Intuitivamente el torque en el hombro de un manipulador es mucho mayor cuando el brazo robótico está extendido horizontalmente.

El análisis de los torques requeridos para contrarrestar el efecto gravitatorio en las distintas poses posibles del manipulador es muy importante en el diseño robótico para determinar la capacidad de torque requerido en los actuadores.

1.1.7.2.4. Efecto de la carga útil

Cualquier robot real tiene una carga útil máxima especificada que viene dictada por dos efectos dinámicos. La primera es que una masa en el extremo del robot aumentará la inercia experimentada por los motores articulados y que reduce la aceleración y el rendimiento dinámico. La segunda es que la masa genera una fuerza de peso que todas las articulaciones necesitan soportar. En el peor de los casos, el componente de par de gravedad aumentado podría exceder la capacidad nominal de uno o más motores. Sin embargo, incluso si no se excede la clasificación, hay menos par disponible para la aceleración, lo que nuevamente reduce el rendimiento dinámico.

1.1.8. Control cinemático [4]

El control cinemático determina las trayectorias a seguir por cada articulación del robot en el tiempo para conseguir los objetivos marcados por el usuario, como son: punto de destino, trayectoria cartesiana del extremo del robot, tiempo de movimiento, etc. Hay que tener en cuenta las limitaciones físicas de los accionamientos para elegir estas trayectorias y a ciertos criterios de precisión o suavidad de la trayectoria.

1.1.8.1. Funciones de control cinemático

El funcionamiento del control cinemático se muestra esquemáticamente en la Figura 1.11, cuyas entradas son los datos de movimiento provenientes del programa de movimiento del robot hecho por el usuario (antes descritos), y establece las trayectorias para todas las articulaciones en función del tiempo según en el modelo cinemático del robot. Estas trayectorias se muestrean con un período T a elegir, y en cada instante kT se genera un vector de referencias articulares para ser usado por los sistemas de control dinámico.

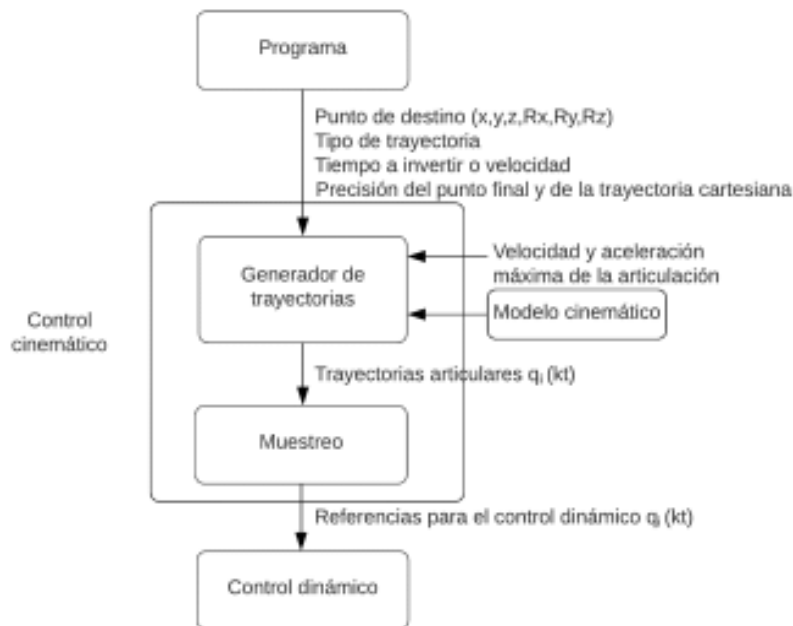


Figura 1.11. Funcionamiento del control cinemático.
(Fuente: Barrientos et al., [4])

Para solventar el control cinemático de un robot, es necesario saber las clases de trayectorias que se puede esperar que ejecute el robot, y cuales interpoladores son más eficaces para vincular los puntos de las articulaciones por los que se desea pasar. Del mismo modo, es importante determinar criterios con el fin de seleccionar cuántos puntos y cuáles se usarán en el muestreo de la trayectoria en el espacio cartesiano y con qué frecuencia se deberán actualizar las referencias articulares.

1.1.8.2. Espacio de las articulaciones [3]

El espacio de las articulaciones corresponde a los métodos de generación de trayectorias en los cuales las formas de trayectoria se especifican en términos de funciones de ángulos de articulación.

Cada punto de la trayectoria de la herramienta generalmente se especifica en términos de una posición y orientación requeridas en su marco, con relación al marco de la base. Al aplicar cinemática inversa, cada uno de estos puntos de trayectoria se transforma en un conjunto de ángulos de articulación requeridos. Luego, se halla una función suave para cada una de las n articulaciones que pasan por los puntos intermedios y terminan en el punto objetivo. El tiempo necesario para cada segmento es igual para todas las articulaciones, de modo que todas las articulaciones alcanzarán el punto de trayectoria de manera simultánea, dando como resultado la posición cartesiana establecida del extremo en cada punto intermedio. Además de especificar la misma duración para cada articulación,

la determinación de la función del ángulo de articulación requerida para una articulación en general es independiente de las demás articulaciones.

En consecuencia, los esquemas de espacio articular permiten conseguir la posición y orientación requeridas en los puntos intermedios. Entre los puntos de la trayectoria, la forma del camino es bastante simple en el espacio articular, pero compleja si se describe en el espacio cartesiano.

Los esquemas de espacio articular suelen ser los más sencillos de calcular y, debido a que no se hace una correlación constante entre los espacios articular y cartesiano, básicamente no hay problema con las singularidades del mecanismo.

1.1.8.3. Espacio cartesiano [3]

Las trayectorias calculadas en el espacio de las articulaciones pueden garantizar que se alcancen los puntos de vía y meta, incluso cuando estos puntos de trayectoria se especificaron usando coordenadas cartesianas. No obstante, la forma espacial de la trayectoria tomada por el efector final no es una línea recta a través del espacio; más bien, es una forma complicada que depende de la cinemática particular del manipulador que se utilice. En el espacio cartesiano se consideran métodos de generación de trayectorias en los que las formas de trayectoria se detallan en base a funciones que realizan el cálculo de la locación y orientación cartesianas como funciones temporales. De esta forma, se puede especificar la forma espacial de la ruta entre los puntos de la ruta también. La forma de trayectoria más común es una línea recta, pero se pueden utilizar formas de trayectoria circular, sinusoidal u otras.

Generalmente cada punto de la trayectoria se define en términos de una ubicación y orientación requeridas del marco de la herramienta en relación con el marco de la base. En los esquemas de generación de caminos basados en sistemas cartesianos, las funciones divididas se juntan para crear una trayectoria que son funciones de tiempo que caracterizan variables cartesianas. Estas trayectorias se pueden planificar directamente a partir de la definición del usuario de los puntos de ruta, que son especificaciones del extremo relativas a la base, sin calcular la cinemática inversa primero. Sin embargo, los esquemas cartesianos son más costosos desde el punto de vista computacional, porque la cinemática inversa debe resolverse en el tiempo de ejecución a la tasa de actualización de la trayectoria, es decir, después de generar la ruta en el espacio cartesiano; se realiza el cálculo de la cinemática inversa para calcular las configuraciones articulares del robot deseadas como último paso.

1.1.8.3.1. Comparación entre espacios [7]

Tabla 1.2. Comparación entre los espacios de las articulaciones y el cartesiano.

	Espacio cartesiano	Espacio de articulaciones
Ventajas	<ul style="list-style-type: none"> - El movimiento es predecible (Interpolación en el espacio de las tareas) - Mejor manejo de obstáculos y colisiones 	<ul style="list-style-type: none"> - Ejecución más rápida (Soluciona cinemática inversa en los puntos intermedios únicamente) - Movimiento de los actuadores es suave y más fácil de validar
Desventajas	<ul style="list-style-type: none"> - Ejecución más lenta (Solución de la cinemática inversa en cada paso de tiempo) - Movimiento del actuador no es necesariamente suave y es difícil de validar 	<ul style="list-style-type: none"> - No es posible garantizar que los puntos intermedios de la trayectoria respeten el límite de movimiento de las articulaciones y colisiones

(Fuente: https://blogs.mathworks.com/racing-lounge/2019/11/06/robot-manipulator-trajectory/?s_tid=blogs_rc_3)

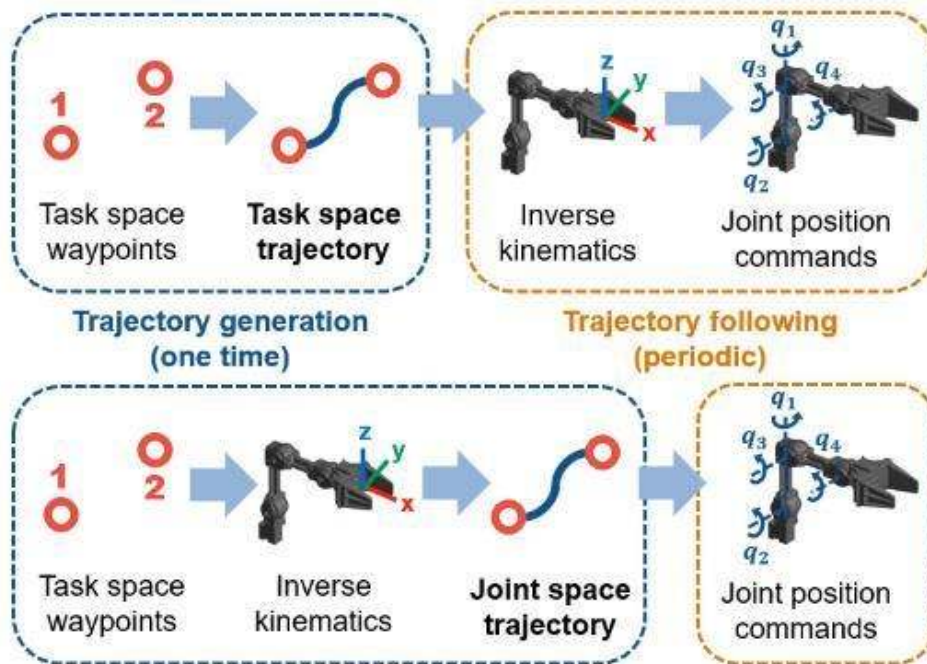


Figura 1.12. Generación de trayectorias en el espacio cartesiano (arriba) y articulado (abajo).
(Fuente: <https://blogs.mathworks.com/racing-lounge/2019/11/06/robot-manipulator-trajectory/>)

1.1.8.4. Tipos de trayectorias

El robot requiere trasladarse entre un punto inicial y uno final para realizar una tarea específica. Este movimiento se podría realizar en base a infinitas trayectorias en el espacio. Entre todas estas, algunas son las que se adoptan en los robots comerciales, ya sea por su facilidad de implementación o su utilidad y aplicación a varias tareas, su generación es dependiente del control cinemático. Así, se puede encontrar que los robots puedan realizar trayectorias continuas o trayectorias punto a punto.

1.1.8.4.1. Trayectorias punto a punto

En esta clase de trayectorias cada articulación se traslada entre su posición inicial y otra final sin tomar en cuenta el estado o movimiento de todas las otras articulaciones. Generalmente, cada actuador intenta mover la articulación que posee al punto objetivo en el más corto tiempo posible de modo que se pueda distinguir entre dos situaciones: movimiento eje a eje y movimiento simultáneo de ejes. Luego, en la segunda clase, los ejes son capaces de moverse independientemente o de forma coordinada.

1.1.8.4.2. Trayectorias continuas

Si desea que el usuario conozca la ruta en el espacio cartesiano por la que pasa el final del robot, es necesario calcular y controlar continuamente las trayectorias de las articulaciones, que suele presentar una evolución en el tiempo muy complicada.

Generalmente, el usuario final requiere que la trayectoria descrita por el robot sea una línea recta o una trayectoria de arco circular. El resultado obtenido es que cada articulación se mueve de forma aparentemente aleatoria, en el que la dirección y la velocidad pueden cambiar sin relación con el resto de las demás articulaciones. Pero, el extremo del robot mostrara la trayectoria requerida.

1.1.8.5. Generación de trayectorias

Por lo general, el usuario quien señala el movimiento que debe seguir el robot definiendo las posiciones en el espacio por la que pasa el extremo y otra información como tipos de trayectorias, tiempos de paso o velocidades. Por lo tanto, generalmente se especifica que el robot se debe mover desde el punto de inicio hacia el final a lo largo de una línea recta a una velocidad constante en el espacio cartesiano.

En otros casos, es necesario especificar una serie de ubicaciones por las que se necesita que se movilice el extremo del manipulador, por lo que requiere definir un interpolador entre las ubicaciones establecidas en el espacio cartesiano que resultará en una expresión del cambio de sus coordenadas. El caso de interpolación más sencillo implica la interpolación lineal entre 2 posiciones consecutivas, por lo que, cada coordenada en el espacio cartesiano cambia con velocidad constante comenzando en un valor inicial hasta el punto final en el intervalo entre cada dos posiciones.

Esta clase de evolución no es recomendable porque provoca discontinuidades importantes en la velocidad y aceleración en los puntos de trayectoria. Se pueden utilizar interpoladores a tramos para prevenir estas discontinuidades, en el caso de las variables articulares.

1.1.8.5.1. Evolución de la orientación

La posición en el espacio de cartesiano generalmente se especifica en coordenadas cartesianas, con pocas excepciones. A pesar de ello, se pueden utilizar diferentes

herramientas para completar la especificación de la orientación, como: par de rotación, ángulos de Euler, matrices de rotación o cuaternios. Esta alternativa única a la representación de posiciones indudablemente hace que el mejor proceso para conectar dos posiciones que están en el espacio cartesiano sea sin duda la interpolación de cada coordenada a través de un interpolador lineal u otros tipos de interpoladores. Esto produce movimientos simples que son fáciles de entender para los usuarios y tienen alta calidad.

No obstante, para la orientación, la interpolación entre los valores iniciales hacia los finales se puede explicar a partir de términos de las herramientas descritas anteriormente, dando como resultado trayectorias diferentes en cada caso.

El desarrollo más normal entre las orientaciones inicial y final podría ser la que gira gradualmente al efector final entre estas dos orientaciones alrededor de un eje de giro. Con lo cual, el uso cuaternios o el par de rotación, es la mejor forma de crear la trayectoria cartesiana de la orientación.

1.1.8.6. Muestreo de trayectorias

El tipo de trayectoria cartesiana más usada es la línea recta y, ocasionalmente, también se usan trayectorias circulares. Las dos pueden ser descritas fácilmente en función del tiempo de manera analítica, por lo que es sencillo establecer las coordenadas rectangulares por las que pasará el extremo del robot en un momento específico.

Generalmente, no es factible obtener la descripción analítica de la trayectoria cartesiana conjunta en base a la descripción analítica de esta, y solo es posible convertir la posición cartesiana en la configuración de las articulaciones a partir del modelo de cinemática directa. Por lo tanto, es necesario escoger puntos de trayectoria en el espacio cartesiano que serán convertirlos a su configuración de las articulaciones correspondiente, esta es tarea del control cinemático.

A primera vista parece que se deben tomar una gran cantidad de puntos, porque cuantos más instantes se muestreen de la trayectoria cartesiana, mejor será el resultado, porque la trayectoria final se ajustará al valor especificado a través de estos puntos con la mayor precisión posible. No obstante, hay que considerar que todos estos puntos se deben convertir a sus coordenadas articulares correspondientes, y luego usarlas en un interpolador que creará la trayectoria articular, que requiere más cálculos conforme aumentan los puntos. Este costo computacional tiene la limitación de que se necesita una respuesta en tiempo real que debe tener sistema de control, lo que evita el aumento arbitrario de puntos cartesianos.

Por otro lado, el modelo geométrico del robot y su no linealidad muestra que el error entre la trayectoria final de su extremo y la trayectoria requerida puede variar mucho según la

cinemática del robot y la posición instantánea considerada. Por tanto, es conviene decidir cuáles son los puntos requeridos para una trayectoria cartesiana optima, no solo el número de puntos de trayectoria. Es necesario alcanzar un balance entre la cantidad de puntos elegidos y el error entre la trayectoria obtenida y la requerida.

De hecho, elegir los puntos equidistantes más cercanos posibles suele ser suficiente para asegurar que la trayectoria conjunta se pueda generar antes del tiempo requerido para el control dinámico del robot.

1.1.8.7. Interpolación de trayectorias

Una vez obtenida la secuencia de configuraciones de las articulaciones por las que recorre el robot, es necesario que el control cinemático conecte estos puntos de unión consecutivos y asegurarse de que se cumplan las condiciones de configuración y tiempo de paso, así como las restricciones de velocidad y aceleración relacionadas con los actuadores que tiene el robot, para lograr una trayectoria factible y suave.

Es necesario elegir un determinado tipo de función, con coeficientes o parámetros que se ajusten dependiendo de las condiciones del movimiento como: posiciones, velocidades y aceleraciones. Al seleccionar dicha función, hay que considerar que debe realizarse en tiempo real el computo de sus parámetros, así como su operación para crear puntos de cálculo de control dinámico, por lo que es ideal una función que sea lo más simple posible. Hay varias funciones de interpolación que pueden conectar todos los puntos de la trayectoria, y se puede usar un conjunto de funciones simples para interpolar de forma local este conjunto de puntos, es decir, solo conectan unos cuantos puntos en el intervalo que sean consecutivos y se superponen entre sí para asegurar la continuidad.

1.1.8.7.1. Interpoladores trapezoidales

Implica descomponer la trayectoria que enlaza dos puntos q_0 , q_1 en tres secciones consecutivas. Utilizando un interpolador lineal en el tramo central, la velocidad permanece constante, y no se requiere conferir ninguna aceleración al actuador. Se usa un polinomio de segundo grado n para los tramos inicial y final, tal que en el primer tramo la velocidad sea lineal desde la velocidad de la trayectoria pasada a la presente, y en el tercer tramo sea lineal desde la velocidad de la trayectoria presente hasta la futura. Por lo tanto, en los tramos iniciales y finales la aceleración adquiere valores constantes diferentes de cero, y en el tramo de la mitad no existe aceleración.

Se puede observar que la distribución de velocidades toma la forma de un trapecioide, razón por la cual este interpolador se llama trapezoidal. Por su parte, existen discontinuidades en la aceleración.

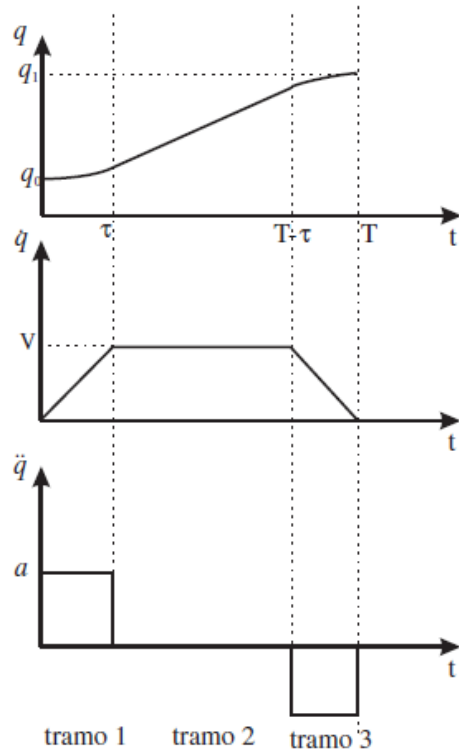


Figura 1.13. Perfiles de posición, velocidad y aceleración del interpolador trapezoidal.
(Fuente: Barrientos et al., [4])

1.1.9. Control dinámico [4]

La tarea del control dinámico es asegurar que la trayectoria seguida por el robot $q(t)$ sea lo más similares a las trayectorias establecidas con el control cinemático. Se usa el conocimiento del modelo dinámico del robot y herramientas de diseño y análisis que proporciona la teoría del servocontrol (estabilidad de Lyapunov, control adaptativo, control PID, etc.) para ello.

1.1.9.1. Control mono articular

La geometría, peso, tipos de actuadores, etc. del robot tienen una influencia importante en la clase de control a usar. Con la ecuación del modelo dinámico del robot:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v\dot{q} \quad (14)$$

A la que se incorpora la matriz diagonal constante F_v con los efectos del par debido al rozamiento viscoso, se puede entender cómo se afecta a la complejidad con la estructura cinemática del robot, que influye también en la estructura de las matrices M , C y g , y su correspondiente control.

Del mismo modo, la existencia de reductores de velocidad o elevados rozamientos viscosos pueden hacer que algunos de los términos de la ecuación se puedan despreciar.

1.1.9.1.1. Control PID [8]

Hoy en día, la mayoría de los manipuladores de robots industriales están controlados por controladores PID. El amplio uso de manipuladores de robots en aplicaciones cotidianas es un testimonio del rendimiento que se puede lograr en una gran variedad de aplicaciones cuando se utiliza el control PID. Sin embargo, a diferencia del control de PD, el procedimiento de ajuste para los controladores PID, es decir, el procedimiento para elegir las matrices definidas positivas apropiadas K_p , K_v y K_i , está lejos de ser trivial.

En la práctica, la puesta a punto de los controladores PID es más fácil para robots cuyo sistema de transmisión incluye mecanismos de reducción como engranajes o bandas. El uso de estas reducciones aumenta efectivamente el par o la fuerza producida por los actuadores y, por lo tanto, estos son capaces de impulsar eslabones de masas considerablemente grandes. En principio, esto tiene la consecuencia de que se pueden alcanzar grandes aceleraciones para enlaces "ligeros". No obstante, la presencia de mecanismos de reducción, como engranajes y bandas, puede introducir fenómenos físicos no deseados que obstaculicen el desempeño del robot en su tarea requerida. Entre estos fenómenos citamos vibraciones por holgura entre los dientes de los engranajes, errores de posicionamiento y desperdicio de energía por fricción en los engranajes, errores de posicionamiento causados por vibraciones y elasticidad de las bandas y por torsiones de engranajes.

A pesar de todo esto, el uso de mecanismos de reducción es común en la mayoría de los manipuladores de robots. Esto tiene un impacto positivo en la tarea de ajuste de los controladores y, más particularmente, de los controladores PID. De hecho, la dinámica completa del robot con transmisiones de alta relación de reducción se caracteriza básicamente por el modelo de los propios actuadores, que a menudo se modelan mediante ecuaciones diferenciales lineales. Así, en este escenario la ecuación diferencial que gobierna el comportamiento del sistema de lazo cerrado se vuelve lineal y, por lo tanto, la sintonización del controlador se vuelve relativamente simple.

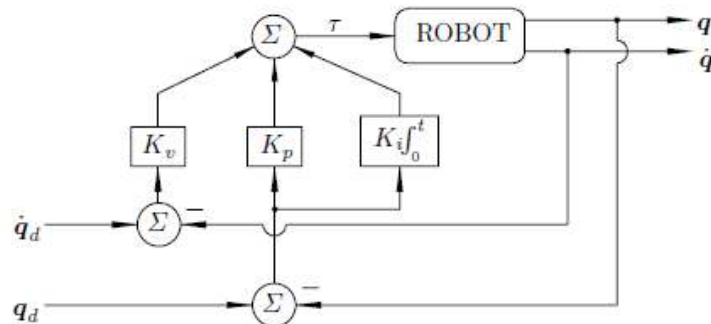


Figura 1.14. Diagrama de bloques del control PID.
(Fuente: Kelly et al., [8])

1.2. Mecatrónica [9]

1.2.1. Definición

Fue un ingeniero japonés el que “concibió” el término mecatrónica en 1969. Se utiliza para describir una doctrina en la ingeniería en la que hay una integración de las ramas de la mecánica, la electrónica y el control en el diseño y fabricación de productos y procesos. Como producto, las máquinas mecatrónicas reemplazan muchas funciones mecánicas por funciones electrónicas. Esto puede brindar una mayor flexibilidad, un rediseño y programación sencillos, y capacidades automáticas de recopilación de datos y realización de informes.

La mecatrónica implica métodos concurrentes de estas disciplinas, en lugar de métodos secuenciales: el desarrollo de sistemas mecánicos y luego el diseño de partes eléctricas y de microprocesadores. Por lo tanto, la mecatrónica es una doctrina de diseño y un enfoque integral de ingeniería. La mecatrónica junta campos técnicos que involucran sistemas de medición, actuación y microprocesadores, también incluye el análisis de sistemas de control y comportamiento de sistemas.

1.2.2. Sistema mecatrónico

Los sistemas mecatrónicos no son solo una combinación de sistemas eléctricos y mecánicos, y son más que simples sistemas de control. Es una integración completa de todas estas funciones, que utiliza un enfoque de diseño concurrente. En el diseño de robots, automóviles, máquinas herramienta, servomotores, cámaras entre otros, se utilizan cada vez más métodos de diseño de ingeniería integrados e interdisciplinarios. Si desea desarrollar un sistema más baratos, más confiables y flexibles, se deben integrar los límites pertinentes a la ingeniería mecánica, la eléctrica, la electrónica y el control al principio del proceso de diseño.

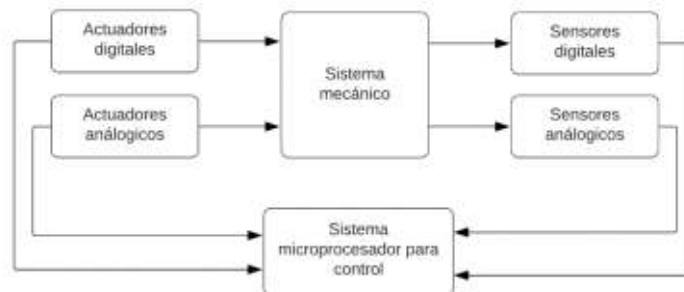


Figura 1.15. Elementos básicos de un sistema mecatrónico.
(Fuente: Bolton, [9])

1.2.3. Modelamiento de sistemas mecatrónicos

Uno de los pasos involucrados en el diseño de sistemas mecatrónicos, es crear un modelo del sistema para que su comportamiento pueda predecirse cuando se ocurran entradas.

Estos modelos implican graficar diagramas de bloques que representen al sistema. Puede pensarse en el sistema como un diagrama de bloques con entradas y salidas, en el que no tenemos que preocuparnos por lo que ocurre dentro de la caja, más bien solo por la relación entre la entrada y la salida.

Cuando usamos ecuaciones matemáticas para expresar el comportamiento de un sistema real, usamos el término "modelado", y las ecuaciones matemáticas expresan la relación entre las entradas y las salidas. Se puede pensar en un motor como un sistema que toma energía eléctrica como entrada y tiene la rotación de un eje como salida. Puede pensar en el sistema de medición como una caja para realizar mediciones. Tiene la cantidad medida como entrada y el valor de la cantidad como salida.

1.2.4. Sistemas de medición

Los sistemas de medición son muy valiosos en cualquier discusión sobre mecatrónica. Por lo regular, los sistemas de medición se pueden considerar que se componen de tres elementos básicos.

1. Un sensor responde a la variable que se mide, dando como salida una señal relacionada con esta cantidad.
2. Un acondicionador de señal recibe la señal del sensor y la convierte en una condición adecuada para su visualización o, en el caso de un sistema de control, para su uso para ejercer el control.
3. Un sistema de visualización muestra la salida del acondicionador de señal.

1.2.5. Sistemas de control

Los sistemas de control son interrelaciones de componentes de una configuración de sistema que generará respuestas requerida por el sistema. Los componentes o procesos para controlar se pueden representar mediante bloques. La relación causa-efecto del proceso se representa usando la relación entrada-salida, que también representa un procesamiento de la señal de entrada que brinda una señal de salida generalmente variable y con amplificación de potencia.

1.2.5.1. Sistemas en lazo abierto y cerrado [10]

1.2.5.1.1. Sistema de control en lazo abierto

Los sistemas de control de bucle abierto utilizan controladores y actuadores para conseguir la respuesta esperada. Este tipo de sistemas no tienen retroalimentación.



Figura 1.16. Diagrama de bloques de un sistema de control en lazo abierto.
(Fuente: Dorf & Bishop, [10])

1.2.5.1.2. Sistema de control en lazo cerrado

Los sistemas de control de bucle cerrado son los cuales procuran mantener una relación definida entre variables del sistema usando la comparación de funciones de dichas magnitudes y que usan como medio de control su diferencia. Los sistemas de control de lazo cerrado utilizan una medida adicional de las salidas reales para comparar la salida real con la respuesta de salida requerida. La señal de retroalimentación es la medida de la salida. Mediante sensores preciso, las salidas medidas son buenas aproximaciones a las salidas reales del sistema.

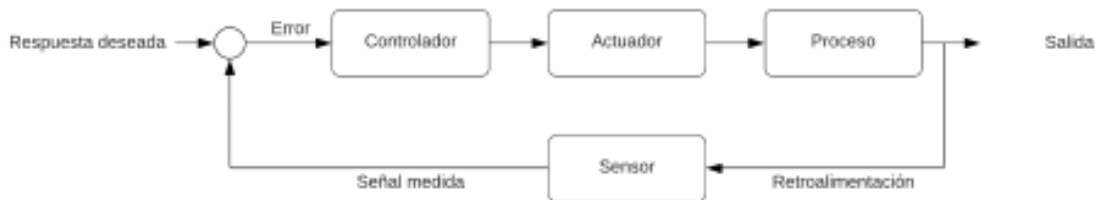


Figura 1.17. Diagrama de bloques de un sistema de control en lazo cerrado.
(Fuente: Dorf & Bishop, [10])

1.2.5.2. Sistemas de control analógico y digital

Con sistemas analógicos nos referimos a aquellos en los que sus señales son funciones continuas del tiempo y para medir las variables se usa el tamaño de la señal. Las señales digitales se pueden considerar como una sucesión de señales de orden de encendido/apagado, cuyo valor de la variable está definido por la sucesión de dichos pulsos.

Debido a que en la mayoría de las situaciones en los sistemas de control se trabaja con entradas y salidas de naturaleza analógica, una particularidad de los sistemas de control digitales es que las entradas analógicas provenientes del exterior se deben transformar en formas y salidas digitales y, estas de nuevo a formas analógicas del mundo exterior. Esto conlleva el uso de convertidores de señales analógicas a digitales (ADC) en las entradas y, por el contrario, convertidores de señales digitales a analógicas (DAC) en las salidas.

Las operaciones digitales se pueden controlar mediante programas, que son un conjunto de instrucciones almacenadas. El almacenamiento de esta información es más sencillo y la tiene una mayor precisión. Adicionalmente, Los circuitos digitales son menos propensos al ruido electromagnético y generalmente son más sencillos de diseñar.

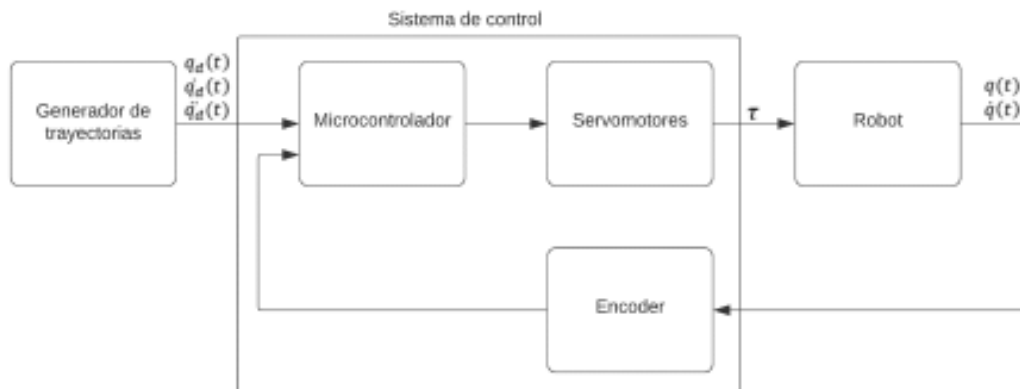


Figura 1.18. Elementos de un sistema de control digital en lazo cerrado.
(Fuente: Bolton, [9])

2. METODOLOGÍA

2.1. Definición del producto

2.1.1. Identificar y establecer necesidades

2.1.1.1. Requerimientos iniciales

El diseño presentado en el presente proyecto tiene la finalidad de satisfacer los requerimientos de un taller de fabricación de diversos productos de carpintería metálica, en el que el uso de soldadura es generalizado.

El manipulador robótico que se va a diseñar tiene como propósito ayudar a la renovación de la maquinaria existente en el taller, de tal manera que se pueda obtener un aumento en la productividad y calidad de los productos.

2.1.1.2. Casa de la calidad

La casa de la calidad es un instrumento del Desarrollo de Función de la Calidad (QFD), usado para traducir los requisitos de los usuarios, que es la voz del cliente, en requisitos técnicos del producto, que es la voz del ingeniero [11].

Para realizar esta matriz es necesario seguir una serie de pasos especificados en [11]. Los manipuladores TM5-900 de Tm Robot [12] y UR5 de Universal Robotics [13] se toman como base para la sección de análisis de la competencia, así como para el posterior diseño del manipulador a proponer. En el Anexo III se tienen algunas especificaciones de estos productos.

2.1.1.2.1. Requerimientos del usuario

De acuerdo con el criterio del usuario final, el manipulador debe tener las siguientes características. Esta es la voz del usuario:

El manipulador debe tener un movimiento preciso.

El robot debe tener la capacidad de realizar movimientos complejos.

Debe poder ser usado por tiempos prolongados.

El nivel de dificultad para su uso debe ser bajo.

Tiene que ser más barato que los productos de la competencia.

El alcance del extremo del robot tiene que ser considerable.

El peso debe ser reducido.

El montaje no debe ser complejo.

2.1.1.2.2. Características técnicas

Los requerimientos expresados anteriormente se manifiestan como una serie de características técnicas. Esta es la voz del ingeniero:

Precisión de los actuadores.

Morfología del manipulador.

Tipo de energía usada.

Programación de la interfaz entre el manipulador y el usuario.

Costo.

Volumen de trabajo.

Peso.

Modularidad.

2.1.1.2.3. Resultados

Con los requerimientos del usuario especificados y expresados en características técnicas, se procede a crear la casa de la calidad para determinar especificaciones que deberá tener el producto final.

(Ver ANEXO I)

2.1.1.2.4. Conclusiones de la casa de la calidad

Una vez realizada la matriz de la casa de la calidad se obtienen resultados que son aplicables en el desarrollo de las especificaciones técnicas. Las variables de diseño más importantes son:

- Precisión del movimiento

Se requiere que la precisión que consiga el extremo del manipulador sea de aproximadamente 1mm.

- Capacidad de movimientos complejos

El robot deberá poder ubicar el extremo de este en posiciones muy variadas para conseguir crear cordones de soldadura de geometrías muy variadas.

- Volumen de trabajo

El mecanismo deberá tener un alcance máximo de por lo menos 750 mm.

2.1.1.3. Especificaciones técnicas

Una vez realizada la casa de la calidad es posible determinar las especificaciones técnicas del producto a diseñar. En el siguiente cuadro se detalla el documento de especificaciones técnicas.

Tabla 2.1. Especificaciones técnicas del manipulador robótico

Empresa	Producto	Fecha inicial	
	MANIPULADOR ROBÓTICO	Última revisión	
Especificaciones			
Concepto	Propone	R/D	Descripción
Función	C	R	Manipulador robótico articulado que sigue trayectorias de soldadura.
	C	R	Ubicación del extremo del manipulador en todos los puntos de la trayectoria.
Dimensiones	C	D	Alcance máximo de 750 mm.
	C	R	Levantar la antorcha y el alambre de soldadura.
Movimientos	C	D	Precisión de movimiento de 1 mm.
	D	D	Seis grados de libertad.
	D	D	Rotación total del extremo mayor a 360°
	D	R	Movimiento del extremo en trayectorias rectangulares.
	D	D	Perfiles de velocidad trapezoidales.
Energía	C	R	Energía eléctrica.
Materiales	D	D	Estructura de aluminio.
	C	R	Resistente a ambientes nocivos.
Señales y control	D	D	Control dinámico de los elementos.
	D	D	Control realimentado.
	C	R	Señales enviadas desde una computadora al controlador de los actuadores.
Fabricación y montaje	D	D	Diseño modular.
	C	R	Partes estandarizadas.
	C	R	Fácil montaje.

(Fuente: Propia)

2.2. Diseño conceptual

2.2.1. Análisis funcional

La estructura funcional está representada en forma de diagrama de bloques, que define las funciones que el producto debe realizar, y los enlaces en él representan el flujo de energía, material y señal entre entradas, salidas y funciones.

La estructura funcional puede expresarse en el nivel funcional general del producto o sistema, o puede subdividirse en partes que contienen subfunciones menos complejas. En el proceso de diseño conceptual, las subfunciones son muy importantes [11].

2.2.1.1. Desarrollo del análisis funcional

Nivel 0

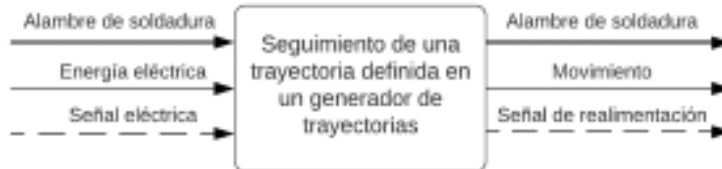


Figura 2.1. Diagrama funcional: nivel 0.
(Fuente: Propia)

Nivel 1

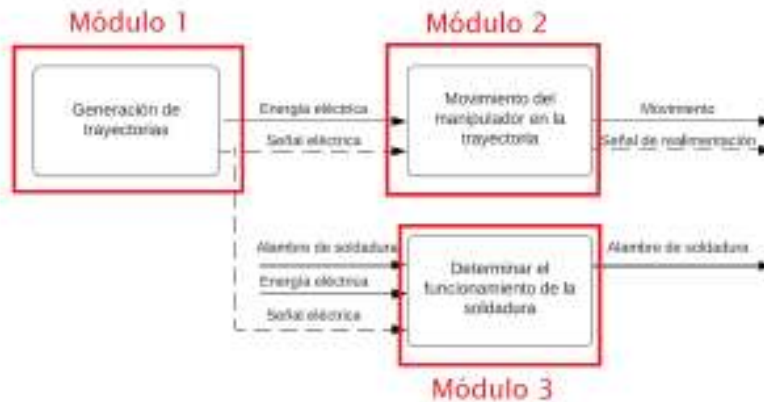


Figura 2.2. Diagrama funcional: nivel1.
(Fuente: Propia)

Nivel 2

(Ver ANEXO II)

2.2.1.2. Análisis de los diagramas funcionales

El nivel 0 representa la tarea global que realiza el manipulador. Esta tarea es seguir un movimiento definido en un programa generador de trayectorias.

El nivel 1 engloba los modos de operación principales:

1. Generar las trayectorias que debe seguir el manipulador.
2. Movimiento del manipulador en los puntos de trayectoria generados.
3. Determinar el funcionamiento de la máquina de soldadora.

En el nivel 2 se pueden ver el conjunto de acciones requeridas para cumplir con los modos de operación principales.

Para el primer modo de operación se observa que se requiere que se defina la trayectoria a seguir y esta debe ser programada de tal manera que controle el movimiento de los actuadores. Además, es necesario que exista una interfaz en la que se puedan ingresar los valores requeridos para crear las trayectorias.

En el segundo modo de operación es necesario primero definir la morfología del robot que es determinar las longitudes de los elementos, tipo de movimiento y actuadores idóneos para la función. Luego se requiere activar los actuadores del manipulador para poder controlar su movimiento. El siguiente paso es accionar el movimiento del manipulador de tal manera que su extremo se ubique en todos los puntos establecidos en el programa. Una vez terminado el movimiento requerido, se procede a regresar el manipulador a su posición inicial. Por último, se deben desactivar los actuadores.

Para el tercer modulo se tiene que activar el funcionamiento de la máquina de soldadura cuando se va a comenzar el movimiento del robot a través de la trayectoria programa. Al finalizar el seguimiento de la trayectoria, se debe desactivar el funcionamiento de la máquina de soldadura antes de regresar el aparato a su posición inicial.

2.2.2. Definición de los módulos

Al estudiar el diagrama de este producto se pueden establecer tres conjuntos principales de funciones, que se procede a dividir en módulos:

Módulo 1: Generación de trayectorias.

Módulo 2: Movimiento del manipulador.

Módulo 3: Soldadura.

2.2.3. Soluciones para cada módulo

2.2.3.1. Módulo 1: Generación de trayectorias

El módulo cumple las siguientes funciones:

- Definir la trayectoria del manipulador.
- Programar y controlar el movimiento de los actuadores.
- Generar interfaz entre la computadora y los actuadores.

2.2.3.1.1. Definir la trayectoria

La trayectoria del manipulador se establece con la ubicación de los puntos y la orientación del efector final. Existen algunas maneras de determinar estos parámetros:

Ubicar manualmente el efector final del robot y grabar la posición

Se mueve el efector final a los puntos intermedios de movimiento con la orientación deseada y se graban las posiciones de los actuadores en esos puntos.



Figura 2.3. Ubicación manual del efector en los puntos de trayectoria.
(Fuente: <https://www.youtube.com/watch?v=GCRX2ejG0fY&t=1s>)

Ventajas:

- Más sencillez al generar los puntos.
- Permite geometrías complejas.
- Más rápido.

Desventajas:

- Requiere tener el manipulador construido.
- La precisión depende de la habilidad del operador.

Definir las posiciones usando coordenadas cartesianas y ángulos de Euler

Se especifican las coordenadas de los puntos de la trayectoria y los ángulos de Euler del efector final en dichos puntos.

```

%% Definir trayectorias y orientaciones
% Para generar rectángulos se especifican las longitudes de los lados y la
% ubicación de los centros de estos

% Longitud de los lados de las geometrías
geomts = [0.1 0.1 0.1;
          0.1 0.1 0.1];

% Ubicación de los centros de las geometrías
centrs = [0.5 0.3 0;
          0.4 0 0.35];

% Numero de lados continuos en las trayectorias rectangulares
nunidad = 4;

% Orientaciones del efector final del manipulador en los vertices de las
% geometrías de las trayectorias
% Planos de trayectorias
plan = ['XI';
       'VI'];

% Dirección de movimiento
dir = ['AB';
      'CB'];

% Tipo de orientaciones
orien = ['OOT';
       'OOT'];

```

Figura 2.4. Puntos y orientaciones de la trayectoria deseada.
(Fuente: Propia)

Ventajas:

- Más útil para simulaciones.
- No requiere el robot construido.

Desventajas:

- Más complejo.
- Más lento.
- Susceptibilidad a equivocaciones de programación.

2.2.3.1.2. Programar el movimiento de los actuadores [14]

Para la planificación del movimiento y las posiciones de los actuadores en los puntos de la trayectoria se usa un lenguaje de programación cuya función será realizar los cálculos de cinemática inversa y la generación de trayectorias. Existen varias opciones para esto:

Matlab Versión 2020a

Matlab es un lenguaje de programación patentado y un entorno de computación numérica desarrollado por MathWorks.

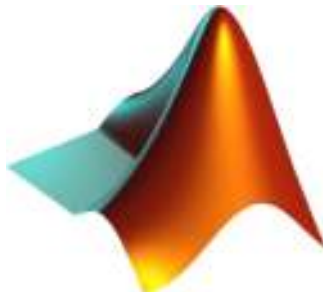


Figura 2.5. Logo de Matlab

(Fuente: https://upload.wikimedia.org/wikipedia/commons/2/21/Matlab_Logo.png)

Ventajas:

- Lenguaje robusto para análisis cinemático.
- Fácil de programar.
- Posee librerías especializadas en robótica.
- Permite simulaciones complejas.

Desventajas:

- Software propietario, por lo que es costoso.
- No está optimizado para la interfaz con hardware.
- No posee tantas librerías.

Python Versión 3.7

Python es un lenguaje de programación general de alto nivel interpretativo.



Figura 2.6. Logo de Python.

(Fuente: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/1024px-Python-logo-notext.svg.png>)

Ventajas:

- Tiene acceso a muchas librerías muy útiles.
- Fácil de programar.
- Fácil de aprender.

Desventajas:

- Dificultad para detectar errores.
- No se especializa en ninguna aplicación en particular.
- Códigos se pueden volver muy complejos.

C++ Versión 17

C ++ es un lenguaje de programación general que se concibió inicialmente como una extensión del lenguaje de programación C.



Figura 2.7. Logo de C++.

(Fuente: https://upload.wikimedia.org/wikipedia/commons/1/18/ISO_C%2B%2B_Logo.svg)

Ventajas:

- Acceso a muchas librerías útiles.
- Fácil implementación en hardware.
- Lenguaje de alto rendimiento (dependiendo del código).

Desventajas:

- Tiempo de aprendizaje elevado.

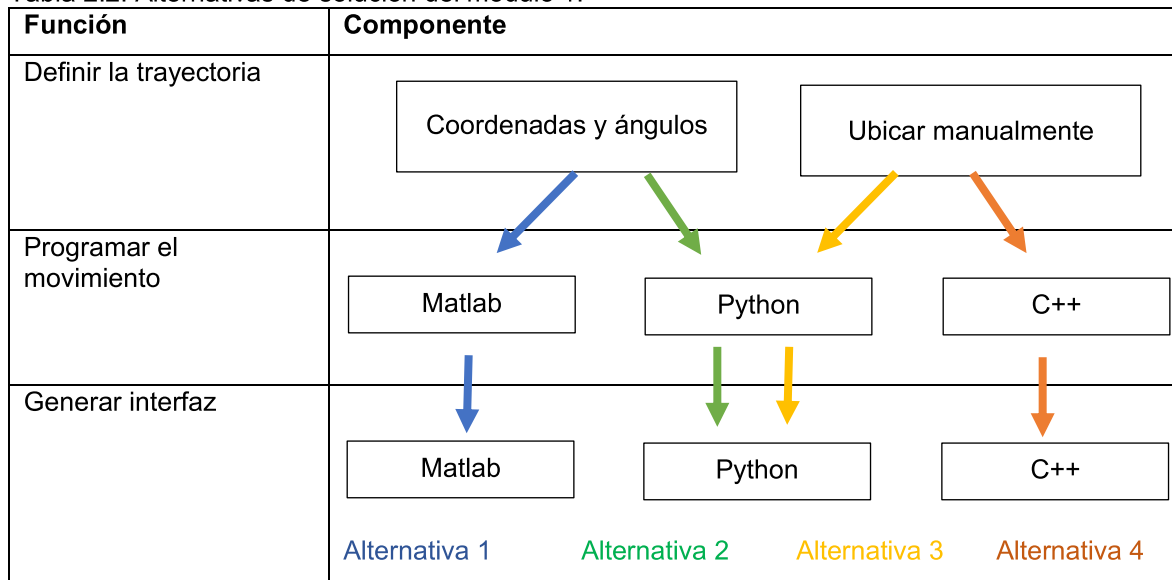
- Escribir programas toma mucho tiempo.
- Dificultad para usar algunas librerías.

2.2.3.1.3. Generar un interfaz de comunicación entre la computadora y los actuadores

Mediante la interfaz es posible interactuar con el manipulador usando el software de programación elegido.

2.2.3.1.4. Alternativas de solución del módulo

Tabla 2.2. Alternativas de solución del módulo 1.



(Fuente: Propia)

2.2.3.1.5. Evaluación y selección de solución del modulo

Para elegir la solución más idónea para este módulo se emplea el método ordinal corregido de criterios ponderados, descrito en [11].

Los criterios de evaluación más importantes son:

Facilidad de aplicación: El programa creado debe ser fácilmente aplicable para controlar los actuadores.

Precisión: Es importante que el cálculo del movimiento provea una solución muy parecida a la deseada.

Complejidad: Se desea que la realización del programa generador de trayectorias sea el menos complejo posible.

Facilidad de simulación: El software usado deberá brindar una manera de simular que se ajuste a las necesidades.

Tabla 2.3. Evaluación de peso específico de cada criterio.

Criterio	Facilidad de aplicación	Precisión	Complejidad	Facilidad de simulación	$\Sigma+1$	Ponderación
Facilidad de aplicación	-	0.5	1	0	2.5	0.25
Precisión	0.5	-	1	0.5	3	0.3
Complejidad	0	0	-	0	1	0.1
Facilidad de simulación	1	0.5	1	-	3.5	0.35
Suma					10	1
Facilidad de simulación > Precisión > Facilidad de aplicación > Complejidad						

(Fuente: Propia)

Evaluación del peso específico del criterio FACILIDAD DE SIMULACIÓN.

Tabla 2.4. Evaluación del peso específico: facilidad de simulación.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	1	1	1	4	0.4
Alternativa 2	0	-	1	1	3	0.3
Alternativa 3	0	0	-	0.5	1.5	0.15
Alternativa 4	0	0	0.5	-	1.5	0.15
Suma					10	1
Alternativa 1 > Alternativa 2 > Alternativa 3 = Alternativa 4						

(Fuente: Propia)

Evaluación del peso específico del criterio PRECISIÓN.

Tabla 2.5. Evaluación del peso específico: precisión.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0.5	1	1	3.5	0.35
Alternativa 2	0.5	-	1	1	3.5	0.35
Alternativa 3	0	0	-	0.5	1.5	0.15
Alternativa 4	0	0	0.5	-	1.5	0.15
Suma					10	1
Alternativa 1 = Alternativa 2 > Alternativa 3 = Alternativa 4						

(Fuente: Propia)

Evaluación del peso específico del criterio COMPLEJIDAD.

Tabla 2.6. Evaluación del peso específico: complejidad.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0.5	0	0	1.5	0.15
Alternativa 2	0.5	-	0	0	1.5	0.15
Alternativa 3	1	1	-	0.5	3.5	0.35
Alternativa 4	1	1	0.5	-	3.5	0.35
Suma					10	1
Alternativa 3 = Alternativa 4 > Alternativa 1 = Alternativa 2						

(Fuente: Propia)

Evaluación del peso específico del criterio FACILIDAD DE SIMULACIÓN.

Tabla 2.7. Evaluación del peso específico del criterio: facilidad de simulación.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	1	1	1	4	0.4
Alternativa 2	0	-	0.5	0.5	2	0.2
Alternativa 3	0	0.5	-	0.5	2	0.2
Alternativa 4	0	0.5	0.5	-	2	0.2
				Suma	10	1
Alternativa 1 > Alternativa 2 = Alternativa 3 = Alternativa 4						

(Fuente: Propia)

2.2.3.1.6. Conclusión de la evaluación de alternativas

Tabla 2.8. Conclusión de la evaluación de alternativas.

Conclusión	Facilidad de aplicación	Precisión	Complejidad	Facilidad de simulación	$\Sigma+1$	Prioridad
Alternativa 1	0.1	0.105	0.015	0.14	1.36	1
Alternativa 2	0.075	0.105	0.015	0.07	1.265	2
Alternativa 3	0.0375	0.045	0.035	0.07	1.1875	3=4
Alternativa 4	0.0375	0.045	0.035	0.07	1.1875	3=4
Alternativa 1 > Alternativa 2 > Alternativa 3 = Alternativa 4						

(Fuente: Propia)

De este análisis se tiene que la mejor solución es la ALTERNATIVA 1.

2.2.3.2. Módulo 2: Movimiento del manipulador

En este módulo se consiguen las siguientes funciones:

- Definir la morfología del robot.
- Activar y desactivar los actuadores del robot.
- Mover el manipulador a la posición requerida.

2.2.3.2.1. Definir la morfología del robot

Para definir la morfología del robot es necesario escoger la estructura mecánica, los sistemas de accionamiento, elementos terminales, sistema de potencia y control.

Estructura mecánica

El manipulador debe tener una morfología similar a la del brazo humano. Como especificación, para este manipulador se usará un robot de cadena cinemática abierta de tipo articulado de seis grados de libertad.



Figura 2.8. Estructura mecánica del manipulador robótico.
(Fuente: Propia)

Actuadores [15]

Para generar el movimiento de los eslabones se usarán actuadores eléctricos. Existen varios tipos de actuadores eléctricos, se consideran los siguientes:

Servomotor AC

Un servomotor AC está basado en diseño de un motor de inducción. Los servomotores tienen la característica de que funcionan en lazo cerrado, esto es posee un controlador y encoder para realizar el control del movimiento. El controlador maneja el funcionamiento del motor enviando señales de comando de velocidad o posición al amplificador. Un dispositivo de retroalimentación proporciona información sobre la posición del servomotor y velocidad, que es comparada por el controlador con el perfil de movimiento programado y es utilizado para ajustar su posición o velocidad [15].



Figura 2.9. Servomotor AC.

(Fuente: https://www.mitsubishielectric.co.jp/fa/products/faspec/download.do?kisyu=/servo&formNm=HK-ST_W_HK-ST172W_3&sub=cad&category=ex&id=spec)

Ventajas:

- Elevada potencia, torque y velocidad.

- Menor mantenimiento.
- Control preciso de posición, velocidad y torque.

Desventajas:

- Alto costo.
- Requiere acceso a tensiones elevadas.
- Gran tamaño y peso.

Servomotor DC

Este tipo de servomotor funciona de manera idéntica que un servomotor AC. La diferencia entre estos radica en que el servomotor DC está basado en el diseño de un motor DC.



Figura 2.10. Servomotor DC.

(Fuente: http://en.robotis.com/service/downloadpage.php?ca_id=70)

Ventajas:

- Tamaño compacto.
- Control preciso de posición, velocidad y torque.
- Construcción sencilla.

Desventajas:

- Alto costo.
- Potencia y torque limitados.
- Requiere mantenimiento periódico.

Motor paso a paso

En este tipo de motor el eje gira en incrementos discretos cuando se le aplican pulsos de comando eléctrico en la secuencia adecuada. Cada revolución está dividida en varios pasos y el motor recibe un pulso de voltaje para cada paso. La cantidad de rotación proporcional al número de pulsos y su velocidad de rotación está relacionada a la frecuencia de esos pulsos.

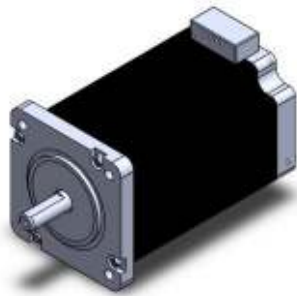


Figura 2.11. Motor a pasos.
(Fuente: <https://grabcad.com/library/stepper-motor-80>)

Ventajas:

- Bajo costo en comparación con las otras opciones.
- Construcción sencilla.
- Fácil de operar.

Desventajas:

- Potencia y torque limitados.
- No se puede controlar con precisión posición, velocidad y torque.
- Funcionamiento irregular a bajas velocidades.

Transmisión del movimiento [4]

Para transmitir el movimiento de los actuadores a las articulaciones existen varias opciones:

Reductor

Mediante el uso de sistemas de engranaje se conecta el eje del motor con el eje de las articulaciones. Su principal función es reducir la velocidad producida en los motores y aumentar el torque entregado.

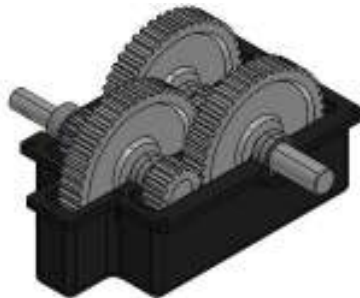


Figura 2.12. Reductor.
(Fuente: <https://grabcad.com/library/old-design-gear-speed-reducer-1>)

Ventajas:

- Permite elevar el torque producido por los actuadores.
- Gran variedad de opciones.

Desventajas:

- Aumenta el tamaño y peso del manipulador.
- Reduce la velocidad máxima del actuador.
- Produce un juego angular entre el actuador y la articulación.

Accionamiento directo

Esta solución consiste en conectar directamente el eje de los motores a los eslabones del manipulador.

Ventajas:

- Gran precisión en el posicionamiento.
- Alta velocidad de movimiento.
- Simplificación del sistema mecánico.

Desventajas:

- Requiere de actuadores de par elevado.
- Requiere de actuadores con codificador de posición muy precisos.

Tipo de efector final

Debido a que la función del manipulador es la de soldar, la herramienta terminal del robot debe ser un soplete para soldadura por arco.



Figura 2.13. Efecto final del manipulador.
(Fuente: Propia)

2.2.3.2.2. Activar y desactivar los actuadores

Los actuadores eléctricos poseen controladores específicos para cada uno que accionan y controlan su funcionamiento.



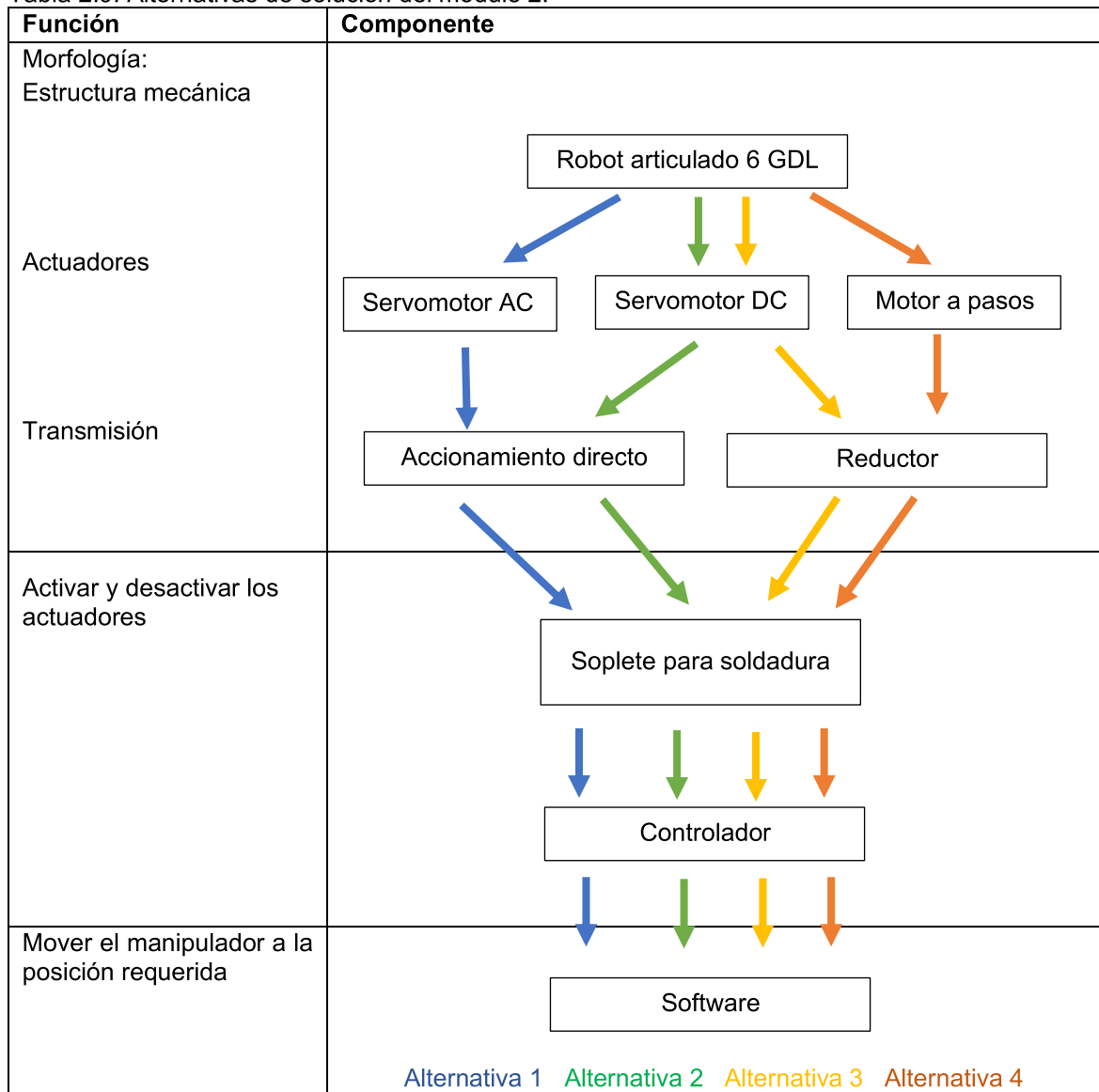
Figura 2.14. Controlador de motor eléctrico.
(Fuente: http://en.robotis.com/service/downloadpage.php?ca_id=70)

2.2.3.2.3. Mover el manipulador a la posición requerida

Mediante el software de programación elegido para el módulo 1 se definen las posiciones de todos los actuadores. De tal manera que solo se debe enviar la señal de movimiento al controlador del actuador para moverlo.

2.2.3.2.4. Alternativas de solución del modulo

Tabla 2.9. Alternativas de solución del módulo 2.



(Fuente: Propia)

2.2.3.2.5. Evaluación y selección de solución del modulo

Los criterios de evaluación más importantes son:

Precio: El costo debe ajustarse a lo que el cliente final está dispuesto a pagar.

Precisión: Se requiere que el movimiento producido por los actuadores produzca la trayectoria deseada con exactitud.

Tamaño: El volumen del manipulador en funcionamiento debe ser el menor posible para disminuir el peso y materiales.

Energía: La energía requerida para poner en funcionamiento el robot tiene que ser la más eficiente.

Tabla 2.10. Evaluación de peso específico de cada criterio.

Criterio	Precio	Precisión	Tamaño	Energía	$\Sigma+1$	Ponderación
Precio	-	0	1	0	2	0.2
Precisión	1	-	1	0.5	3.5	0.35
Tamaño	0	0	-	1	2	0.2
Energía	1	0.5	0	-	2.5	0.25
				Suma	10	1
Precisión > Energía > Precio = Tamaño						

(Fuente: Propia)

Evaluación del peso específico del criterio PRECIO.

Tabla 2.11. Evaluación del peso específico del criterio: precio.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0	0	0	1	0.1
Alternativa 2	1	-	1	0	3	0.3
Alternativa 3	1	0	-	0	2	0.2
Alternativa 4	1	1	1	-	4	0.4
				Suma	10	1
Alternativa 4 > Alternativa 2 > Alternativa 3 > Alternativa 1						

(Fuente: Propia)

Evaluación del peso específico del criterio PRECISIÓN.

Tabla 2.12. Evaluación del peso específico del criterio: precisión.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0.5	1	1	3.5	0.35
Alternativa 2	0.5	-	1	1	3.5	0.35
Alternativa 3	0	0	-	1	2	0.2
Alternativa 4	0	0	0	-	1	0.1
				Suma	10	1
Alternativa 1 = Alternativa 2 > Alternativa 3 > Alternativa 4						

(Fuente: Propia)

Evaluación del peso específico del criterio TAMAÑO.

Tabla 2.13. Evaluación del peso específico del criterio: tamaño.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0	0	0	1	0.1
Alternativa 2	1	-	1	1	4	0.4
Alternativa 3	1	0	-	0.5	2.5	0.25
Alternativa 4	1	0	0.5	-	2.5	0.25
				Suma	10	1
Alternativa 2 > Alternativa 3 = Alternativa 4 > Alternativa 1						

(Fuente: Propia)

Evaluación del peso específico del criterio ENERGÍA.

Tabla 2.14. Evaluación del peso específico del criterio: energía.

Criterio	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	$\Sigma+1$	Ponderación
Alternativa 1	-	0	0	0	1	0.1
Alternativa 2	1	-	1	1	4	0.4
Alternativa 3	1	0	-	1	3	0.3
Alternativa 4	1	0	0	-	2	0.2
				Suma	10	1
Alternativa 2 > Alternativa 3 > Alternativa 4 > Alternativa 1						

(Fuente: Propia)

2.2.3.2.6. Conclusión de la evaluación de alternativas.

Tabla 2.15. Conclusión de la evaluación de alternativas.

Conclusión	Facilidad de aplicación	Precisión	Complejidad	Facilidad de simulación	$\Sigma+1$	Prioridad
Alternativa 1	0.02	0.1225	0.02	0.025	1.1875	4
Alternativa 2	0.06	0.1225	0.08	0.1	1.3625	1
Alternativa 3	0.04	0.07	0.05	0.075	1.235	2
Alternativa 4	0.08	0.035	0.05	0.05	1.215	3
Alternativa 2 > Alternativa 3 > Alternativa 4 > Alternativa 1						

(Fuente: propia)

De este análisis se tiene que la mejor solución es la ALTERNATIVA 2.

2.2.3.3. Modulo 3: Soldadura

Este módulo tiene como función:

- Activar y desactivar el funcionamiento de la máquina de soldadura.

2.2.3.3.1. Determinar el funcionamiento de la máquina de soldadura

Existen varias alternativas para el proceso de soldadura por arco. Se consideran dos de las más comunes:

SMAW

Este es un tipo de soldadura por arco eléctrico entre un material base y un electrodo revestido. La temperatura producida por el arco en la punta del electrodo es de aproximadamente 3500°C, que es más alta que la requerida para fundir la mayoría de los metales. El material base y el electrodo revestido son fundidos por este calor. Así, se produce un baño de fusión, que se solidifica mientras el electrodo se mueve por toda la extensión de la junta [16].



Figura 2.15. Porta electrodo para soldadura SMAW.
(Fuente: Propia)

Ventajas:

- Proceso más barato que otros de soldadura.
- proceso sencillo, flexible y portátil.
- Capaz de soldar la mayoría de los metales ferrosos y no ferrosos.

Desventajas:

- Requiere cambio periódico del electrodo.
- Difícil control de posición del electrodo al consumirse.
- Bajo rendimiento en comparación con otros procesos de soldadura.

GMAW

Este proceso de soldadura genera un arco entre el alambre de soldadura alimentado continuamente y la pieza a soldar. Tiene una protección del arco que se logra mediante gas, generalmente inerte [17].



Figura 2.16. Soplete para soldadura de arco.
(Fuente: Propia)

Ventajas:

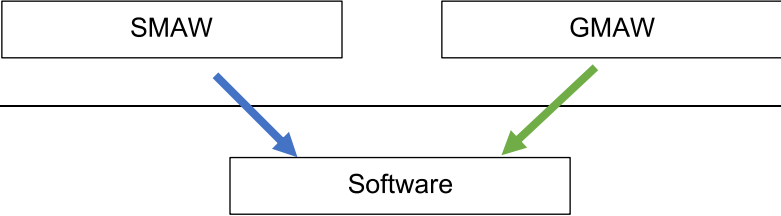

- Fácil automatización y control de posición.
- Soldadura continua en longitudes de soldadura largas.
- Soldadura que requiere mínima limpieza después de soldar.

Desventajas:

- Mayor costo que SMAW.
- Requiere más elementos para poder soldar.
- Dificultar para soldar juntas en ubicaciones de difícil llegada.

2.2.3.3.2. Alternativas de solución del modulo

Tabla 2.16. Alternativas de solución del módulo 3.

Función	Componente
Soldadura	
Accionamiento	

(Fuente: Propia)

2.2.3.3.3. Evaluación y selección de solución del modulo

Los criterios de valoración que se consideran más importantes son:

Precio: Es deseable que el precio sea económico.

Precisión: El proceso de soldadura tiene que permitir que el extremo del robot que suelda siga la trayectoria definida de manera fidedigna.

Facilidad de implementación: El proceso de soldadura a ser usado debe ser fácilmente aplicable al manipulador robótico.

Tiempo: El tiempo requerido para realizar la soldadura deber ser bajo para permitir un movimiento fluido.

Tabla 2.17. Evaluación de peso específico de cada criterio.

Criterio	Precio	Precisión	Facilidad de implementación	Tiempo	$\Sigma+1$	Ponderación
Precio	-	0	0	0	1	0.1
Precisión	1	-	1	1	4	0.4
Facilidad de implementación	1	0	-	1	3	0.3
Tiempo	1	0	0	-	2	0.2
				Suma	10	1
Precisión > Facilidad > Tiempo > Precio						

(Fuente: Propia)

Evaluación del peso específico del criterio PRECIO.

Tabla 2.18. Evaluación del peso específico del criterio: precio.

Criterio	Alternativa 1	Alternativa 2	$\Sigma+1$	Ponderación
Alternativa 1	-	1	2	0.67
Alternativa 2	0	-	1	0.33
Suma			3	1
Alternativa 1 > Alternativa 2				

(Fuente: Propia)

Evaluación del peso específico del criterio PRECISIÓN.

Tabla 2.19. Evaluación del peso específico del criterio: precisión.

Criterio	Alternativa 1	Alternativa 2	$\Sigma+1$	Ponderación
Alternativa 1	-	0	1	0.33
Alternativa 2	1	-	2	0.67
Suma			3	1
Alternativa 2 > Alternativa 1				

(Fuente: Propia)

Evaluación del peso específico del criterio FACILIDAD DE IMPLEMENTACIÓN.

Tabla 2.20. Evaluación del peso específico del criterio: facilidad de implementación.

Criterio	Alternativa 1	Alternativa 2	$\Sigma+1$	Ponderación
Alternativa 1	-	0	1	0.33
Alternativa 2	1	-	2	0.67
Suma			3	1
Alternativa 2 > Alternativa 1				

(Fuente: Propia)

Evaluación del peso específico del criterio TIEMPO.

Tabla 2.21. Evaluación del peso específico del criterio: tiempo.

Criterio	Alternativa 1	Alternativa 2	$\Sigma+1$	Ponderación
Alternativa 1	-	0	1	0.33
Alternativa 2	1	-	2	0.67
Suma			3	1
Alternativa 2 > Alternativa 1				

(Fuente: Propia)

2.2.3.3.4. Conclusión de la evaluación de alternativas.

Tabla 2.22. Conclusión de la evaluación de alternativas.

Conclusión	Precio	Precisión	Facilidad de implementación	Tiempo	$\Sigma+1$	Prioridad
Alternativa 1	0.07	0.13	0.10	0.07	1.37	2
Alternativa 2	0.03	0.27	0.20	0.13	1.63	1
Alternativa 2 > Alternativa1						

(Fuente: Propia)

De este análisis se tiene que la mejor solución es la ALTERNATIVA 2.

2.2.4. Solución conceptual

Tras el análisis de criterios ponderados se procede entonces al ensamble de los módulos físicos del manipulador robótico. Cabe recalcar que esta es tan solo una representación del esquema del manipulador y su diseño y dimensiones pueden variar en capítulos subsiguientes.

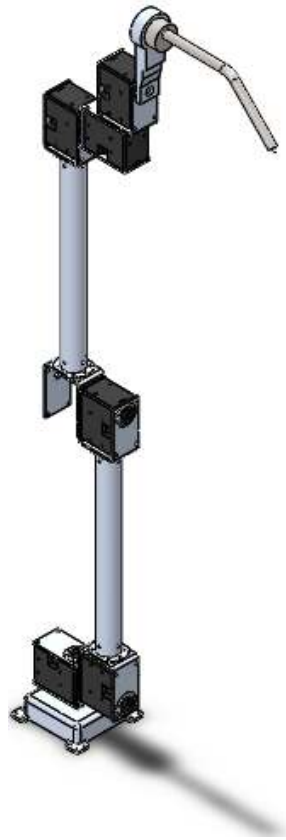


Figura 2.17. Solución conceptual del manipulador robótico.
(Fuente: Propia)

2.3. Diseño de materialización

2.3.1. Diseño del mecanismo

A partir de la tabla de especificaciones técnicas se obtuvo que el alcance máximo del brazo manipulador debería ser de por lo menos 750 mm. Por lo tanto, esta es la distancia normal entre el eje de giro del motor del hombro y el eje de giro del motor de la muñeca más extrema.

Las dimensiones de los eslabones se establecieron con base a las dimensiones de manipuladores existentes en el mercado. La relación de longitud del brazo y antebrazo es una de las características más importantes a tomar en cuenta. En la siguiente tabla se detalla la información de esta relación para los manipuladores comerciales TM5-900 y UR5.

Tabla 2.23. Relación de longitud del brazo y antebrazo de diferentes manipuladores.

Longitud del brazo	Longitud del antebrazo	Relación	Manipulador
429	410.5	1.05	Techman Robot TM5-900 [12].
425	392	1.08	Universal Robot UR5 [13].

(Fuente: Propia)

A partir de esta información, se procede a realizar un modelo CAD, usando Solidworks Versión 2018, del manipulador con una relación de 1.05, que es similar a los manipuladores

de referencia. Las dimensiones generales y las partes del brazo robótico se pueden observar en la siguiente figura.

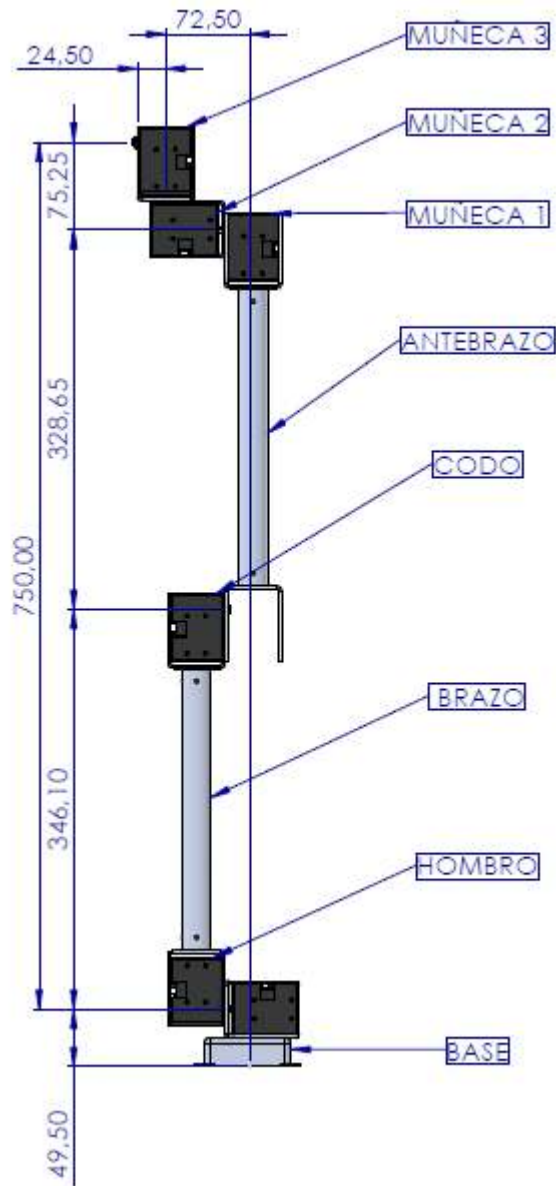


Figura 2.18. Partes y dimensiones del manipulador.
(Fuente: Propia)

2.3.1.1. Volumen de trabajo

Dado que el alcance mínimo requerido del manipulador es de 750 mm, se desarrolla un Volumen de trabajo en el que el manipulador puede maniobrar de manera óptima. Este espacio tiene forma de esfera en la que el centro de esta se ubica en el eje del motor sobre la base y su radio es el extremo del motor más extremo, se excluye de este espacio una forma cilíndrica que representa el cuerpo del robot. El volumen generado se puede ver en la siguiente imagen.

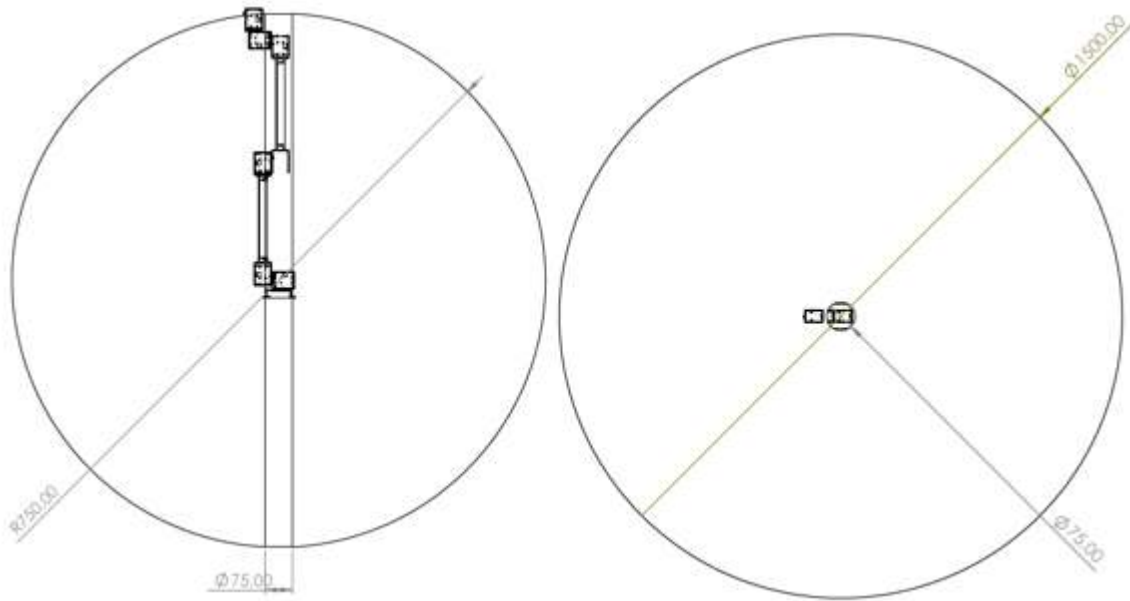


Figura 2.19. Volumen de trabajo el manipulador robótico.
(Fuente: Propia)

2.3.2. Cálculos estructurales

En [4] se desarrolló la ecuación de equilibrio de fuerzas y momentos para un robot mono articular. Como primera aproximación de los resultados se realizará primero este análisis para obtener valores referenciales de fuerza y momento. Luego se procederá a realizar un análisis más detallado usando elementos finitos.

2.3.2.1. Fuerzas y momentos

Para simplificar el análisis de este manipulador se dividirá el mismo en tres secciones: la primera sección comprenderá todos los componentes del hombro barco y codo, la segunda sección está constituida por los componentes del antebrazo y muñeca 1, la tercera sección a componen los miembros de la muñeca 2 y muñeca 3. Como se observa en la siguiente imagen. Las propiedades físicas se obtienen del modelo CAD y en base a esta información se calcularán las reacciones y momentos en cada articulación.

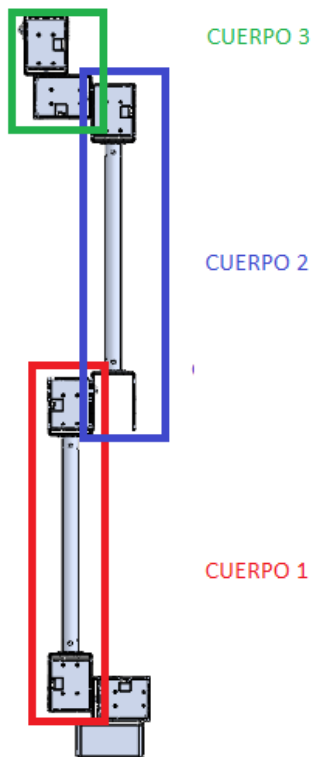


Figura 2.20. División del brazo manipulador en tres cuerpos distintos.
(Fuente: Propia)

Tabla 2.24. Propiedades físicas de los subcuerpos establecidos.

Cuerpo	Propiedad de masa	Valor
Cuerpo 1	Masa	573.45 g
	Ubicación del centro de masa	173.05 mm
	Momento de inercia alrededor del centro de gravedad	13339781.73 g · mm ²
Cuerpo 2	Masa	354.05 g
	Centro de masa	597.57 mm
	Momento de inercia alrededor del centro de gravedad	4747501.29 g · mm ²
Cuerpo 3	Masa	528.39 g
	Centro de masa	703.12 mm
	Momento de inercia alrededor del centro de gravedad	619584.32 g · mm ²

(Fuente: Propia)

2.3.2.1.1. Diagrama de cuerpo libre

El diagrama de cuerpo libre del manipulador se muestra en la siguiente figura. Se toma en cuenta el peso del efector final de este como una masa puntual de 400 g ubicada en el eje del motor más extremo del robot.

Se considerará que los cuerpos se comportan como vigas en voladizo empotradas en la base, esto debido al momento de reacción que producen los actuadores.

Usando la ecuación (11) se tiene que el máximo momento requerido por los actuadores se da con los eslabones ubicados en un ángulo de 0° con respecto al eje X, en base a esto se determinará los requerimientos de torque máximo de los motores.

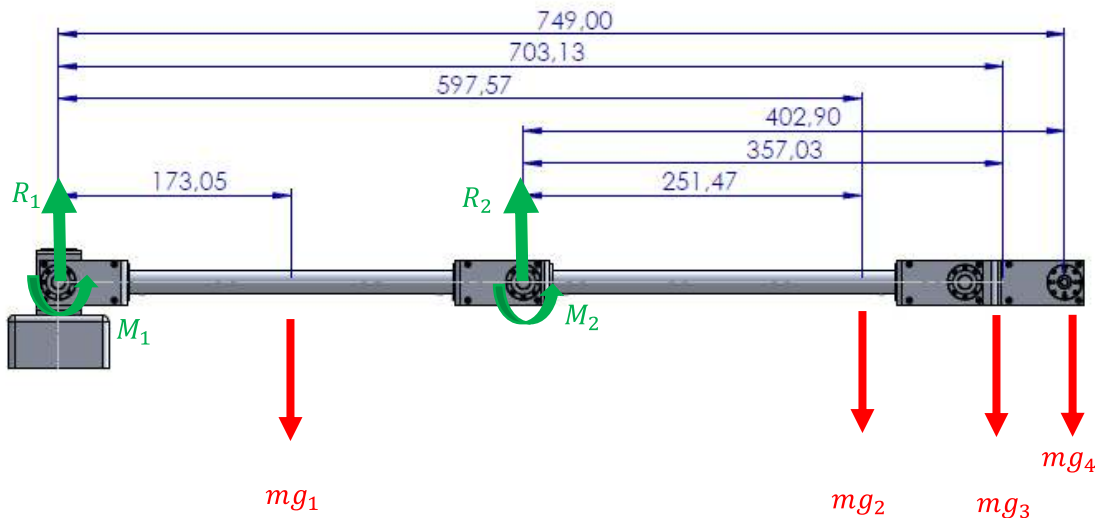


Figura 2.21. Diagrama de cuerpo libre del mecanismo.
(Fuente: Propia)

Cuerpo 1

Reacciones

$$\Sigma F_{Y_1} = 0$$

$$R_1 - m_1g - m_2g - m_3g - m_4g = 0$$

$$M_1 = 0.57345 \cdot 9.81 + 0.35405 \cdot 9.81 + 0.52839 \cdot 9.81 + 0.4 \cdot 9.81$$

$$R_1 = 18.27 \text{ N}$$

Momentos

$$\Sigma M_1 = 0$$

$$M_1 - m_1g \cdot L_1 - m_2g \cdot L_2 - m_3g \cdot L_3 - m_4g \cdot L_4 = 0$$

$$M_1 = 0.57345 \cdot 9.81 \cdot 0.17305 + 0.35405 \cdot 9.81 \cdot 0.59757 + 0.52839 \cdot 9.81 \cdot 0.70312 + 0.4 \cdot 9.81 \cdot 0.75$$

$$M_1 = 9.69 \text{ Nm}$$



Figura 2.22. Diagrama de cuerpo libre del cuerpo 1.
(Fuente: Propia)

2.3.2.1.2. Diagramas de momentos y cortantes del cuerpo 1

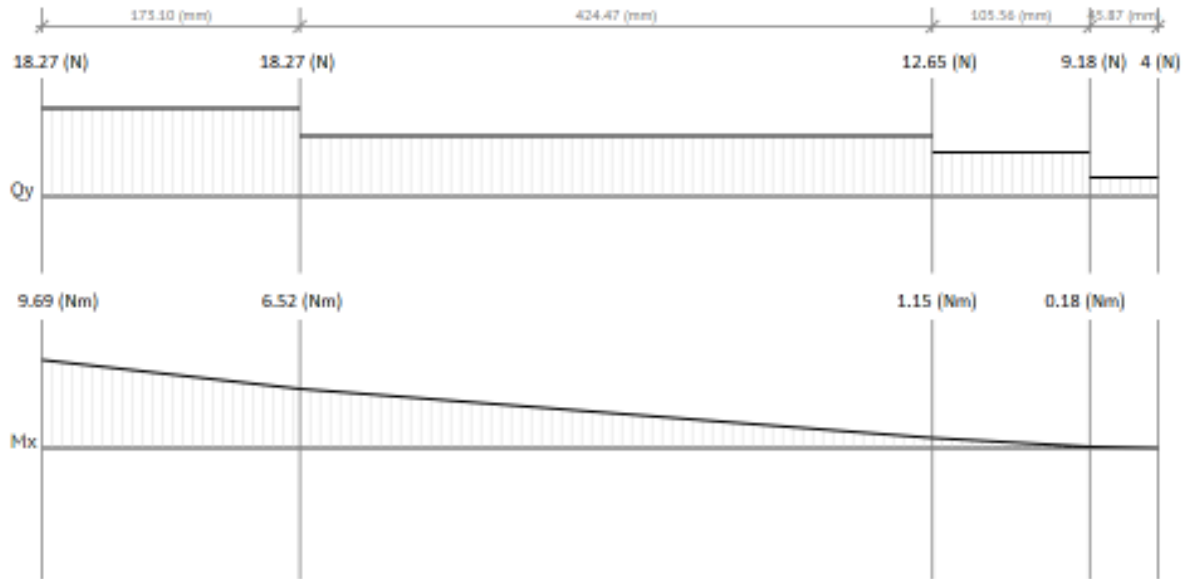


Figura 2.23. Diagramas de fuerza cortante y momento flector del cuerpo 1.
(Fuente: Propia)

Cuerpo 2

Reacciones

$$\Sigma F Y_2 = 0$$

$$R_2 - m_2 g - m_3 g - m_4 g = 0$$

$$M_1 = 0.35405 \cdot 9.81 + 0.52839 \cdot 9.81 + 0.4 \cdot 9.81$$

$$R_1 = 12.65 \text{ N}$$

Momentos

$$\Sigma M_2 = 0$$

$$M_2 - m g_2 \cdot L_2 - m g_3 \cdot L_3 - m g_4 \cdot L_4 = 0$$

$$M_2 = 0.35405 \cdot 9.81 \cdot 0.25147 + 0.52839 \cdot 9.81 \cdot 0.35703 + 0.4 \cdot 9.81 \cdot 0.4029$$

$$M_2 = 4.33 \text{ Nm}$$

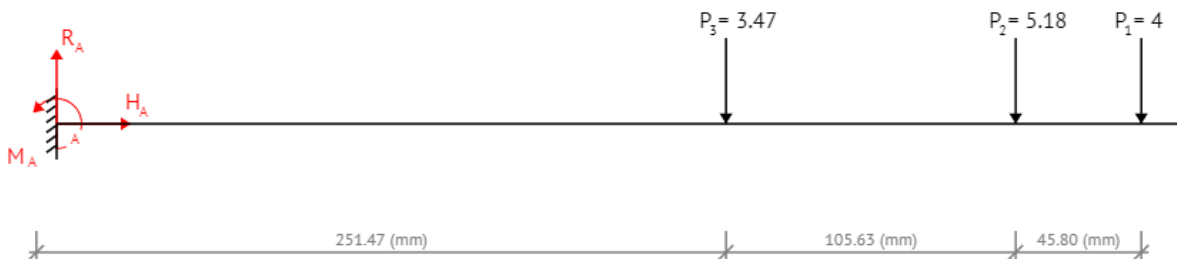


Figura 2.24. Diagrama de cuerpo libre del cuerpo 2.
(Fuente: Propia)

2.3.2.1.3. Diagramas de momentos y cortantes del cuerpo 2

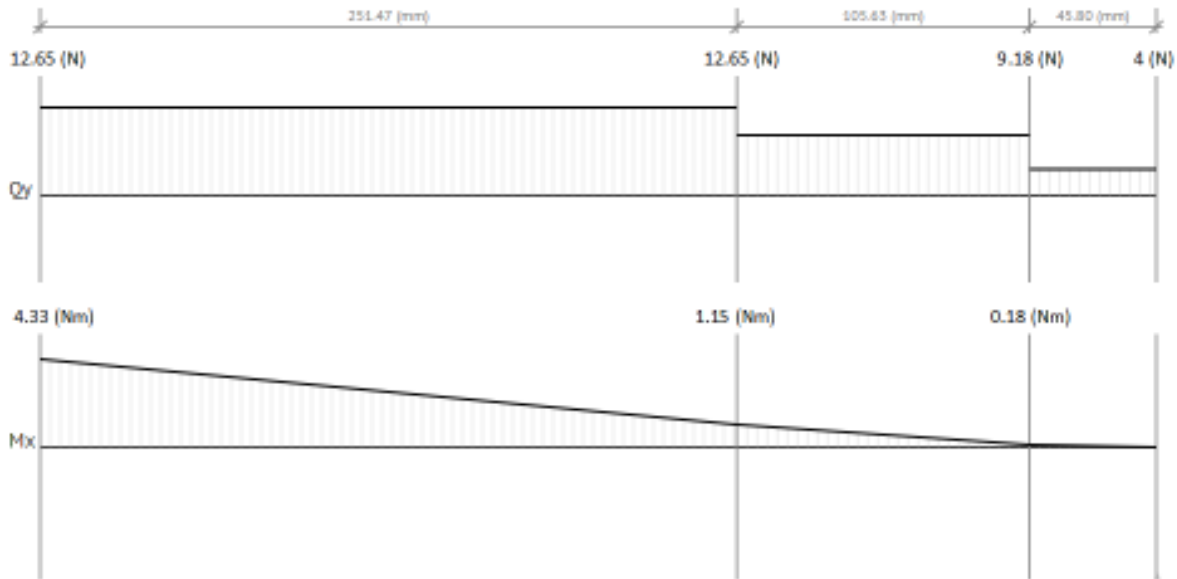


Figura 2.25. Diagramas de fuerza cortante y momento flector del cuerpo 2.
(Fuente: Propia)

2.3.2.2. Selección del perfil

En esta parte se trabajará con perfiles tubulares de aluminio, debido a la facilidad de conseguir este producto en el mercado local.

En la sección de diseño de viga prismática desarrollada en [18], se especifica que en la selección de un perfil estructural se debe considerar un diseño basado en la flexión y de ser necesario se comprueba con la resistencia a cortante. El esfuerzo en ambos casos no deberá exceder el esfuerzo de fluencia del material.

$$\sigma_{m\acute{a}x} \leq \sigma_{adm}$$

$$\sigma_{adm} = \frac{S_y}{FS}$$

Donde, S_y es el esfuerzo de fluencia del material, FS es el factor de seguridad.

Usando el procedimiento especificado se procederá a elegir el perfil tubular usando el diseño basado en flexión, de tal manera que se tenga un rendimiento idóneo para la aplicación deseada.

El esfuerzo máximo de flexión se define como:

$$\sigma_{m\acute{a}x} = \frac{M_{m\acute{a}x} \cdot c}{I}$$

Donde, $M_{m\acute{a}x}$ es el momento máximo en el cuerpo, c es el diámetro externo del perfil e I es el segundo momento de área.

Reemplazando en la ecuación de resistencia a la flexión, se tiene:

$$\sigma_{m\acute{a}x} = \frac{M_{m\acute{a}x} \cdot c}{I} = \frac{M_{m\acute{a}x}}{S}$$

Donde, S es el m3dulo de secci3n del perfil.

$$\frac{M_{m\acute{a}x}}{S} \leq \frac{S_y}{FS}$$

$$S \geq \frac{M_{m\acute{a}x} \cdot FS}{S_y}$$

Para el c3lculo del m3dulo de secci3n S requerido se usar3 un momento m3ximo de 10 N, un factor de seguridad m3nimo de 2 y un valor de limite el3stico del aluminio de 248 MPa [19].

$$S \geq \frac{10 \text{ Nm} \cdot 2}{2.4E + 8 \text{ Pa}}$$

$$S \geq 8.33E - 8 \text{ m}^3$$

Usando el cat3logo de perfiles normalizados de Cedral se procede a elegir el perfil de 17.9 mm de di3metro y 1.07 mm de espesor. Este perfil posee un m3dulo de secci3n de $1.12E - 7 \text{ m}^3$ de tal manera que:

$$1.12E - 7 \text{ m}^3 > 8.33E - 8 \text{ m}^3$$


		M (Nm)	10			
		Fs	2			
		Sy(Pa)	2.40E+08			
REFERENCIA	Di3metro (D)		Espesor (E) mm	Peso Kg/m	I m ⁴	S m ³
	Milimetros	Pulgadas				
1012	50.80	2	1.27	0.536	6.06E-08	1.19E-06
1013	76.20	3	1.27	0.810	2.10E-07	2.75E-06
1136	90.00	3 35/64	6.00	4.291	1.40E-06	1.56E-05
1148	9.52	3/8	1.14	0.081	2.68E-10	2.82E-08
1149	7.92	5/16	1.20	0.070	1.48E-10	1.86E-08
1157	17.90	45/64	1.07	0.153	2.01E-09	1.12E-07
1158	23.40	59/64	1.24	0.234	5.32E-09	2.27E-07

Figura 2.26. Cat3logo de tubos redondos de aluminio de la marca Cedral. (Fuente: http://www.cedral.com.ec/uploads/product/17_cedral-tubos-redondos.pdf)

2.3.3. An3lisis estructural por elementos finitos

En esta secci3n se realizar3 el an3lisis estructural est3tico por elementos finitos del manipulador con la finalidad de comprobar si las dimensiones de los elementos y materiales usados proveer3n resistencia suficiente para soportar las fuerzas y momentos. Este an3lisis se realizar3 usando el programa ANSYS Versi3n 18.

2.3.3.1. Procedimiento:

El procedimiento se realiza siguiendo los lineamientos en [20] y el manual de usuario incluido en el software.

2.3.3.1.1. Crear el sistema de análisis

Este análisis se realizará al ensamble completo del manipulador en posición horizontal. Los motores no se considerarán para los cálculos de esfuerzos y deformaciones y solo contribuyen con sus propiedades físicas

2.3.3.1.2. Definir datos de ingeniería

En esta sección es necesario definir los datos de rigidez requeridos. En este caso se selecciona el material aleación de aluminio de la biblioteca de materiales.

2.3.3.1.3. Adjuntar geometría

La geometría se importa desde el software CAD Solidworks.

La primera aproximación del diseño consiste en placas de aluminio de 2 mm de espesor y tubos redondos de aluminio de diámetro 17.9 mm y espesor de 1 mm.

2.3.3.1.4. Definir conexiones

Los contactos, las juntas, los resortes, las vigas, las conexiones de malla son válidas en un análisis estructural estático.

2.3.3.1.5. Crear malla

Se debe proporcionar una densidad de malla adecuada en las superficies de contacto para permitir que las tensiones de contacto se distribuyan de manera suave. Asimismo, hay que tener una densidad de malla adecuada para resolver tensiones; las áreas donde las tensiones o deformaciones son de interés requieren una malla relativamente fina en comparación con la necesaria para la resolución de desplazamiento o no linealidad.

Aplicando el criterio de calidad de malla Jacobiano (Jacobian ratio MAPDL), que es una métrica de la desviación de un elemento cualquiera de un elemento con forma ideal, donde 1 es representa un elemento de forma perfecta. Se obtiene un valor promedio de 0.99. Esto significa que la malla generada es de excelente calidad.

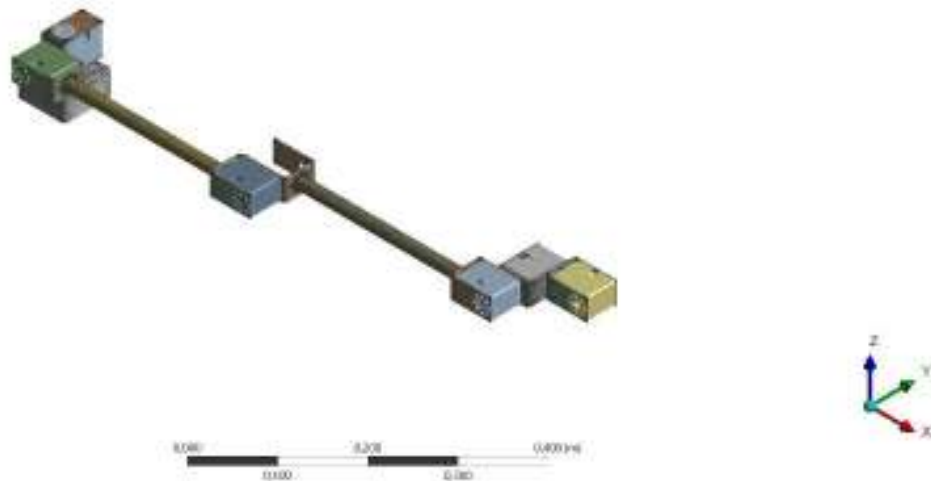


Figura 2.27. Malla generada en ANSYS.
(Fuente: Propia)

2.3.3.1.6. Establecer la configuración de análisis

Para análisis estáticos lineales simples, normalmente no es necesario cambiar esta configuración.

2.3.3.1.7. Aplicar cargas y soportes

Para un análisis estructural estático, las cargas aplicables son todas inerciales, estructurales, importadas y de interacción, y los soportes aplicables son todos los soportes estructurales.

Las restricciones se definen de tal manera que se asemeje a las condiciones impuestas en el cálculo manual. En la siguiente imagen se observan las restricciones impuestas.

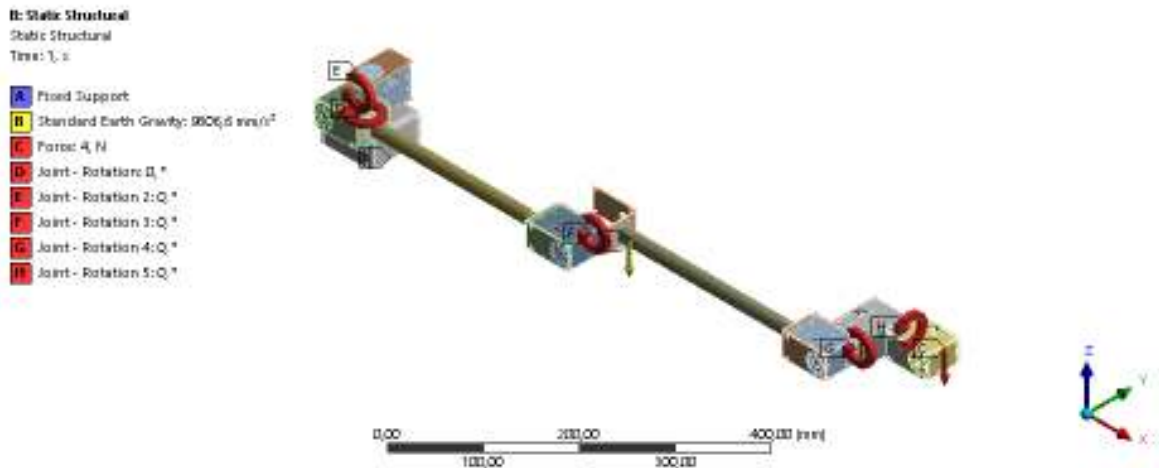


Figura 2.28. Cargas y soportes del modelo de estudio.
(Fuente: Propia)

2.3.3.2. Solución y resultados

En el paso Resolver, Mechanical pasa sus datos al kernel de resolución de MAPDL apropiado, según el tipo de análisis que se va a realizar. Ese kernel luego devuelve los datos de la solución a Mechanical, donde puede ver los resultados.

Los resultados más relevantes para el análisis estructural estático del manipulador se muestran a continuación.

2.3.3.2.1. Reacciones y momentos

Tabla 2.25. Reacciones y momentos obtenidos en Ansys.

Articulación	Momento Nm	Reacción N
1	4.52E-07	20.469
2	9.5446	17.927
3	4.2986	12.388
4	0.45151	9.083
5	0.090436	6.5415

(Fuente: Propia)

2.3.3.2.2. Esfuerzo equivalente Von Mises

La tensión equivalente de Von Mises se utiliza a menudo en el diseño mecánico porque permite representar cualquier estado de tensión tridimensional arbitrario como un único valor de tensión positivo. La tensión equivalente es parte de la teoría de rotura por tensión equivalente máxima utilizada para predecir la fluencia en un material dúctil [21].

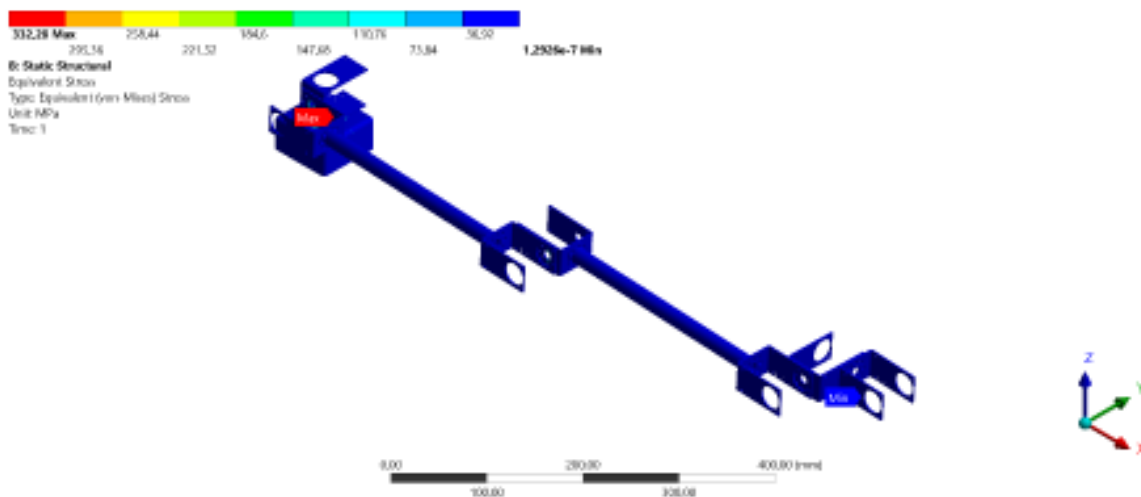


Figura 2.29. Distribución de las tensiones de Von Mises.
(Fuente: Propia)

Esfuerzo máximo: 332.28 MPa

Ubicación: Base

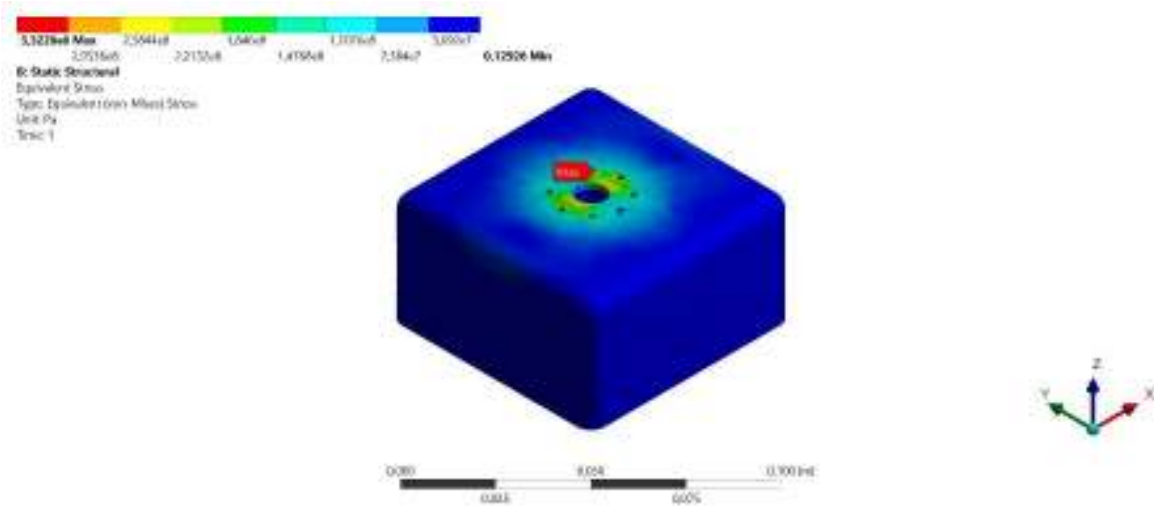


Figura 2.30. Distribución de los esfuerzos de Von Mises en el cuerpo crítico.
 (Fuente: Propia)

2.3.3.2.3. Factor de seguridad

La herramienta seguridad por esfuerzo máximo equivalente se basa en la teoría de falla por esfuerzo equivalente máximo para materiales dúctiles. La teoría especifica que una combinación cualquiera de tensiones principales provoca una falla si la tensión máxima equivalente en material es igual o supera el límite de tensión específico [21].

$$\frac{\sigma_y}{\sigma_e} < 1$$

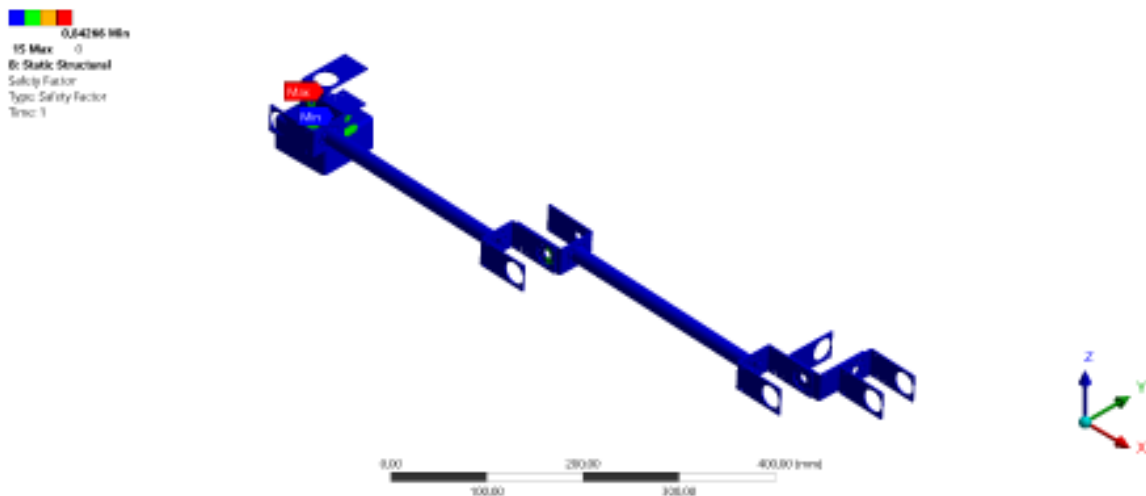


Figura 2.31. Distribución de factores de seguridad.
 (Fuente: Propia)

Valor mínimo: 0.84

Ubicación: Base

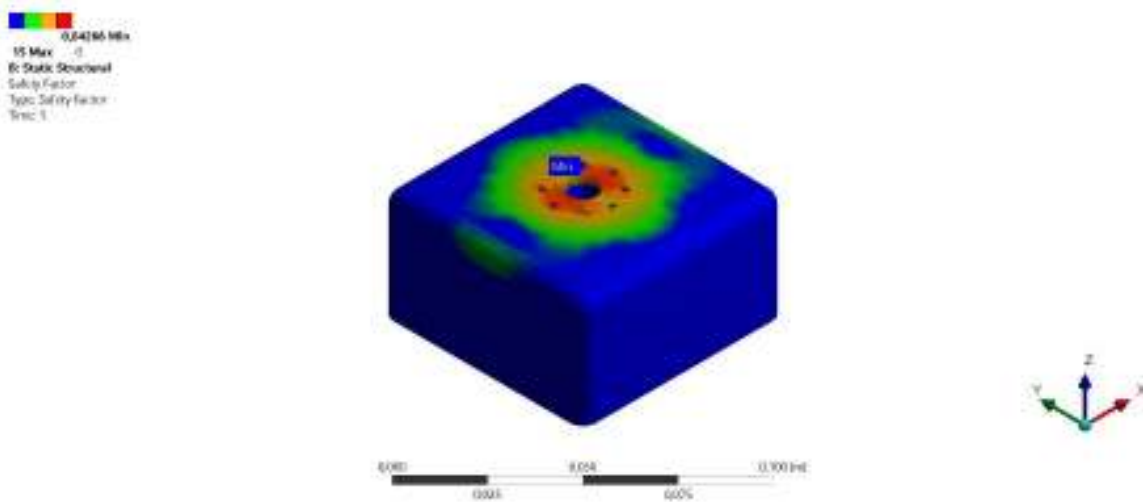


Figura 2.32. Distribución de factores de seguridad en el cuerpo crítico.
(Fuente: Propia)

2.3.3.2.4. Deformación direccional

La deformación direccional permite estudiar las deformaciones en un área seleccionada o delimitada de una pieza o un ensamblaje. En este caso se usa para determinar la flecha máxima de la estructura.

Cuerpo 1

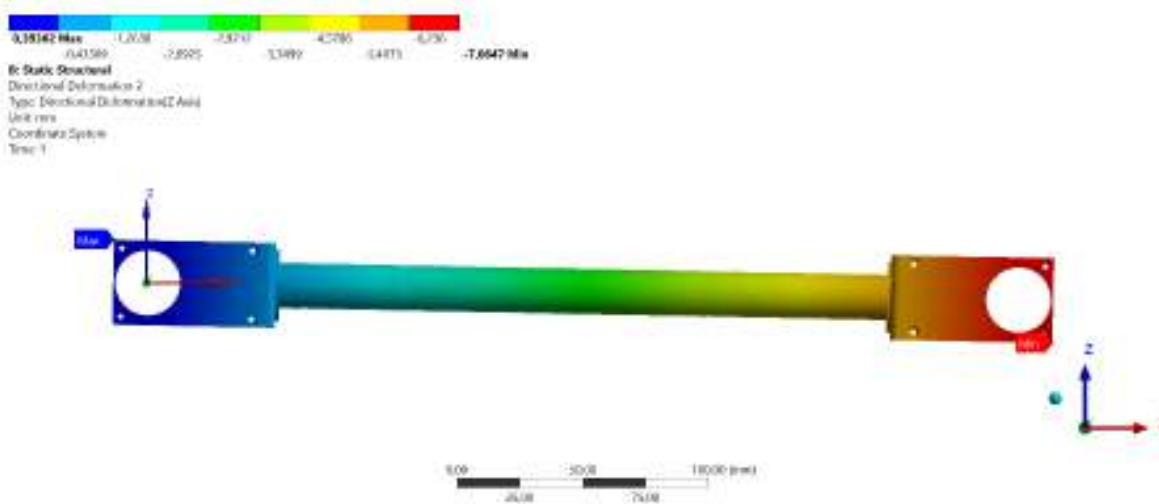


Figura 2.33. Deformación direccional en el cuerpo 1.
(Fuente: Propia)

Deformación máxima: 7.1 mm

Cuerpo 2

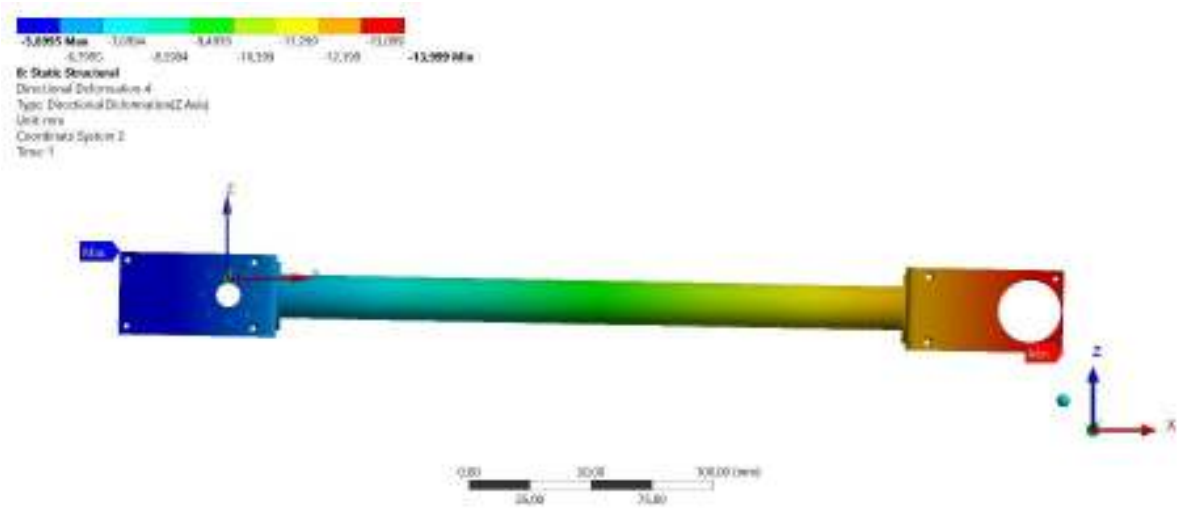


Figura 2.34. Deformación direccional del cuerpo 2.
(Fuente: Propia)

Deformación máxima: 9 mm

2.3.3.3. Análisis de resultados

En los resultados obtenidos usando ANSYS se observa que las reacciones calculadas no son exactamente iguales a las obtenidas en la simulación. Esto se debe a que en el cálculo solamente se toma en cuenta una componente cartesiana para las reacciones y momentos; en tanto que el software calcula las tres componentes de estas.

Se observa además que el esfuerzo equivalente máximo supera al valor del límite elástico del material en la base del manipulador, lo que significa que entra en la región plástica; lo que no es recomendable. Esto se puede comprobar debido a que el valor mínimo obtenido del factor de seguridad es 0.84.

Usando un criterio de deflexión máxima de $\delta = \frac{L}{250}$ establecido en [22], para los eslabones individuales se tiene:

$$\delta_1 = \frac{346.10}{250} = 1.38 \text{ mm}$$

$$\delta_2 = \frac{328.65}{250} = 1.31 \text{ mm}$$

Por lo que ninguno de los eslabones cumple el requerimiento.

Por este motivo es necesario cambiar la geometría del modelo y realizar el análisis hasta conseguir resultados satisfactorios.

2.3.4. Modificación del diseño

Las principales modificaciones realizadas al modelo CAD serán aumentar los espesores de las placas de aluminio a uno de 2.5 mm y la selección de un perfil de mayor diámetro.

En esta parte se repetirá el procedimiento con los cambios realizados y se comprobara si cumplen con los requerimientos.

2.3.4.1. Cálculos estructurales estáticos

2.3.4.1.1. Fuerzas y momentos

Tabla 2.26. Propiedades físicas de los subcuerpos establecidos.

Cuerpo	Propiedad de masa	Valor
Cuerpo 1	Masa	634.31 g
	Ubicación del centro de masa	173.05 mm
	Momento de inercia alrededor del centro de gravedad	13474169.56 g · mm ²
Cuerpo 2	Masa	439.00 g
	Centro de masa	584.17 mm
	Momento de inercia alrededor del centro de gravedad	5352256.78 g · mm ²
Cuerpo 3	Masa	517.75 g
	Centro de masa	704.19 mm
	Momento de inercia alrededor del centro de gravedad	619584.32 g · mm ²

(Fuente: Propia)

Cuerpo 1

Reacciones:

$$\Sigma FY_1 = 0$$

$$R_1 - m_1g - m_2g - m_3g - m_4g = 0$$

$$M_1 = 0.63431 \cdot 9.81 + 0.439 \cdot 9.81 + 0.51775 \cdot 9.81 + 0.4 \cdot 9.81$$

$$R_1 = 19.25 \text{ N}$$

Momentos:

$$\Sigma M_1 = 0$$

$$M_1 - m_1g \cdot L_1 - m_2g \cdot L_2 - m_3g \cdot L_3 - m_4g \cdot L_4 = 0$$

$$M_1 = 0.63431 \cdot 9.81 \cdot 0.17305 + 0.439 \cdot 9.81 \cdot 0.59757 + 0.51775 \cdot 9.81 \cdot 0.70312 + 0.4 \cdot 9.81 \cdot 0.749$$

$$M_1 = 10.04 \text{ Nm}$$



Figura 2.35. Diagrama de cuerpo libre del cuerpo 1.
(Fuente: Propia)

Diagrama de momentos y cortantes del cuerpo 1

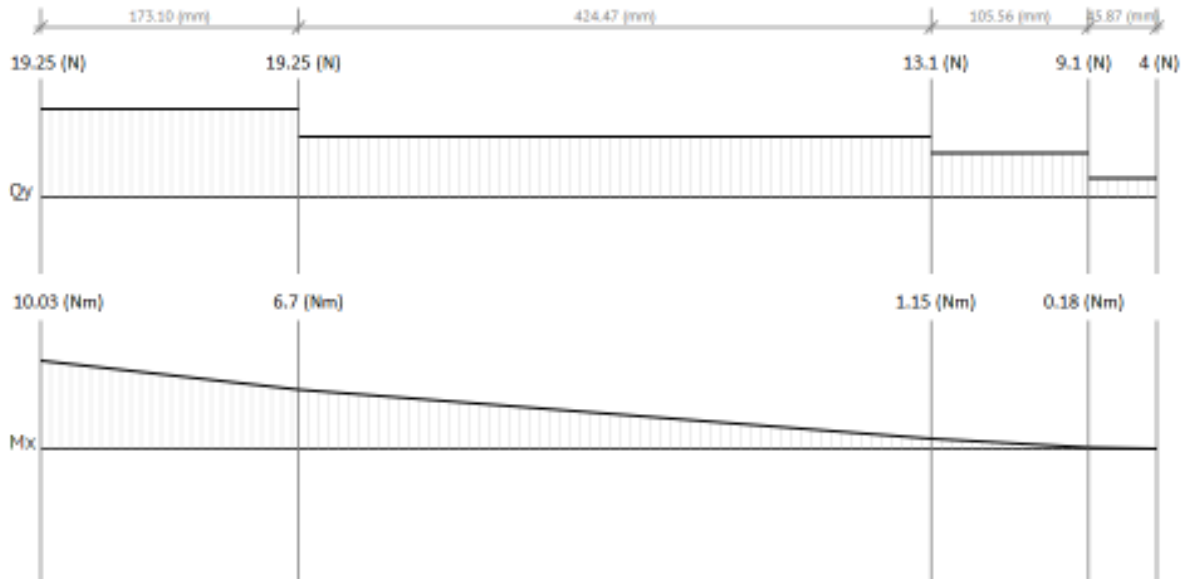


Figura 2.36. Diagramas de fuerza cortante y momento flector del cuerpo 1.
(Fuente: Propia)

Cuerpo 2

Reacciones:

$$\sum F_{Y_2} = 0$$

$$R_2 - m_2g - m_3g - m_4g = 0$$

$$M_1 = 0.439 \cdot 9.81 + 0.51775 \cdot 9.81 + 0.4 \cdot 9.81$$

$$R_1 = 13.1 \text{ N}$$

$$\sum M_2 = 0$$

$$M_2 - mg_2 \cdot L_2 - mg_3 \cdot L_3 - mg_4 \cdot L_4 = 0$$

$$M_2 = 0.439 \cdot 9.81 \cdot 0.25147 + 0.51775 \cdot 9.81 \cdot 0.35703 + 0.4 \cdot 9.81 \cdot 0.4029$$

$$M_2 = 4.44 \text{ Nm}$$

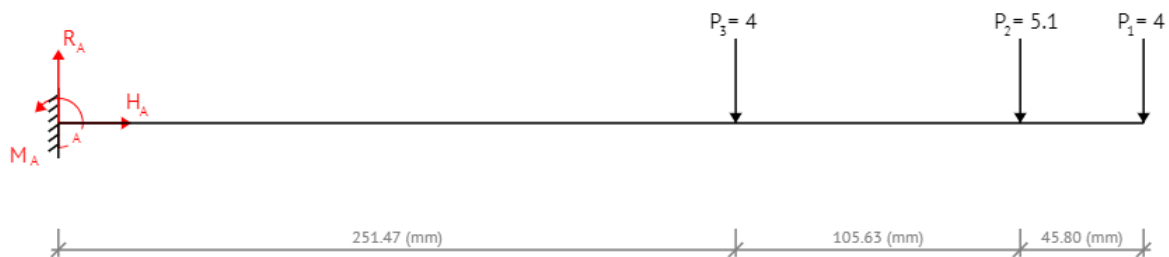


Figura 2.37. Diagrama de cuerpo libre del cuerpo 2.
(Fuente: Propia)

Diagramas de momentos y cortantes del cuerpo 2

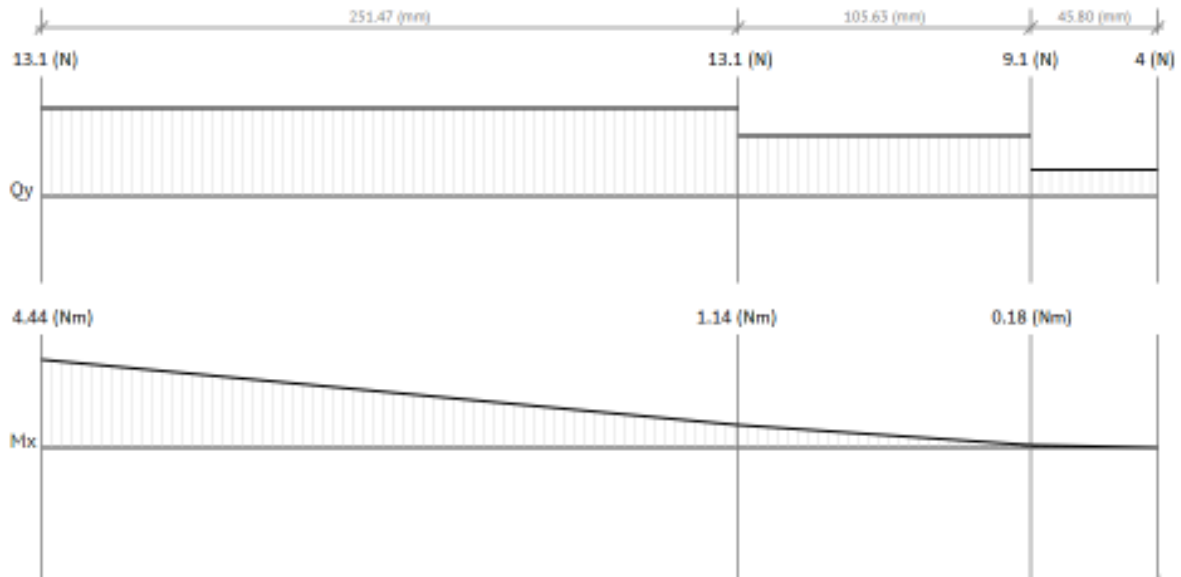


Figura 2.38. Diagramas de fuerza cortante y momento flector del cuerpo 2.
(Fuente: Propia)

2.3.4.2. Selección de perfil:

Nuevamente se usa la tabla de perfiles disponibles en Cedal y esta vez se elige un perfil de 32 mm de diámetro y 2 mm de espesor. Usando los mismos valores para el módulo de sección requerido se tiene que el módulo calculado es mayor que el requerido para la aplicación.

$$6.66E - 7 \text{ m}^3 > 8.33E - 8 \text{ m}^3$$

REFERENCIA	Diámetro (D)		Espesor (E) mm	Peso Kg/m	I m ⁴	S m ³
	Milímetros	Pulgadas				
1012	50.80	2	1.27	0.536	6.06E-08	1.19E-06
1013	76.20	3	1.27	0.810	2.10E-07	2.75E-06
1136	90.00	3 35/64	6.00	4.291	1.40E-06	1.56E-05
1148	9.52	3/8	1.14	0.081	2.68E-10	2.82E-08
1149	7.92	5/16	1.20	0.070	1.48E-10	1.86E-08
1157	17.90	45/64	1.07	0.153	2.01E-09	1.12E-07
1158	23.40	59/64	1.24	0.234	5.32E-09	2.27E-07
1167	101.60	4	1.27	1.085	5.04E-07	4.96E-06
1168	127.00	5	1.50	1.603	1.16E-06	9.17E-06
1169	152.40	6	1.65	2.12	2.22E-06	1.46E-05
1204	25.40	1	2.00	0.38	1.01E-08	3.99E-07
1254	32.00	1 17/64	2.00	0.511	2.13E-08	6.66E-07
1482	90.00	3 35/64	4.50	3.720	1.11E-06	1.23E-05

Figura 2.39. Catálogo de tubos redondos de aluminio de la marca Cedal.
(Fuente: http://www.cedal.com.ec/uploads/product/17_cedal-tubos-redondos.pdf)

2.3.4.3. Análisis estructural estático por elementos finitos

2.3.4.3.1. Procedimiento:

El procedimiento para el análisis del modelo CAD modificado es el mismo que en la sección anterior.

2.3.4.3.2. Generar malla

Aplicando el criterio de calidad de malla Jacobiano (Jacobian ratio MAPDL) para medir la métrica de malla se obtiene un valor promedio de 1.17. Esto significa que la malla generada es de muy buena calidad.



Figura 2.40. Malla generada en Ansys.
(Fuente: Propia)

2.3.4.3.3. Aplicar cargas y soportes:

Las cargas y soportes para este análisis son los mismos que en el estudio anterior.

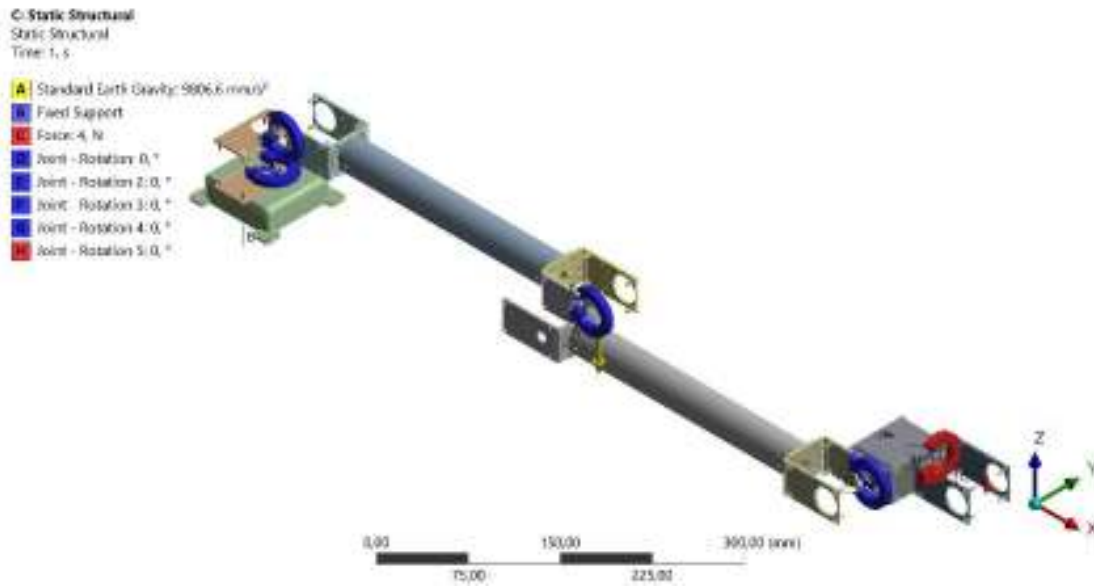


Figura 2.41. Cargas y soportes del modelo de estudio.
(Fuente: Propia)

2.3.4.3.4. Solución y resultados

Los resultados obtenidos en el análisis de este modelo del manipulador se muestran a continuación.

Reacciones y momentos

Tabla 2.27. Reacciones y momentos obtenidos en Ansys.

Articulación	Momento Nm	Reacción N
1	1.46E-03	6.2377
2	9.9289	27.776
3	4.2215	21.734
4	0.2765	18.943
5	6.11E-02	28.566

(Fuente: Propia)

Esfuerzo equivalente Von Mises

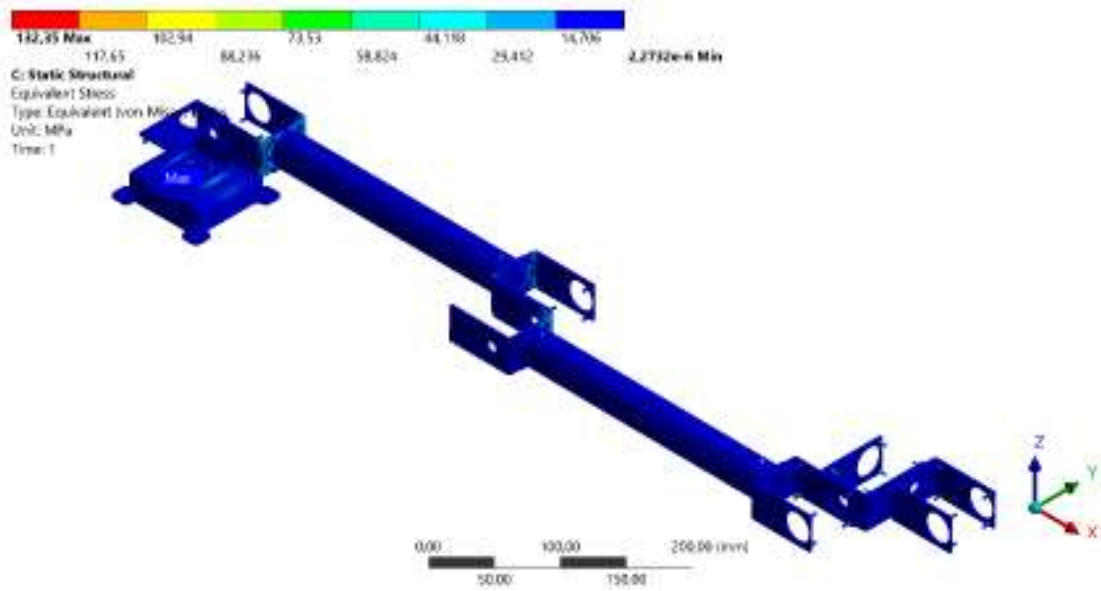


Figura 2.42. Distribución de los esfuerzos de Von Mises.
(Fuente: Propia)

Esfuerzo máximo: 132.35 Mpa

Ubicación: Base

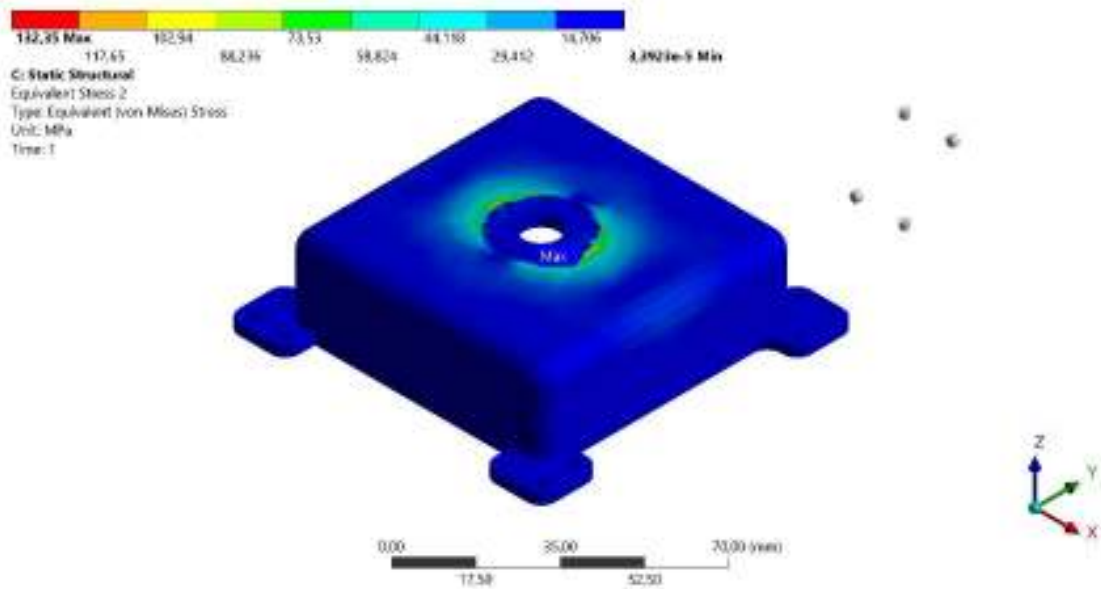


Figura 2.43. Distribución de esfuerzos de Von Mises en el cuerpo crítico.
(Fuente: Propia)

Factor de seguridad

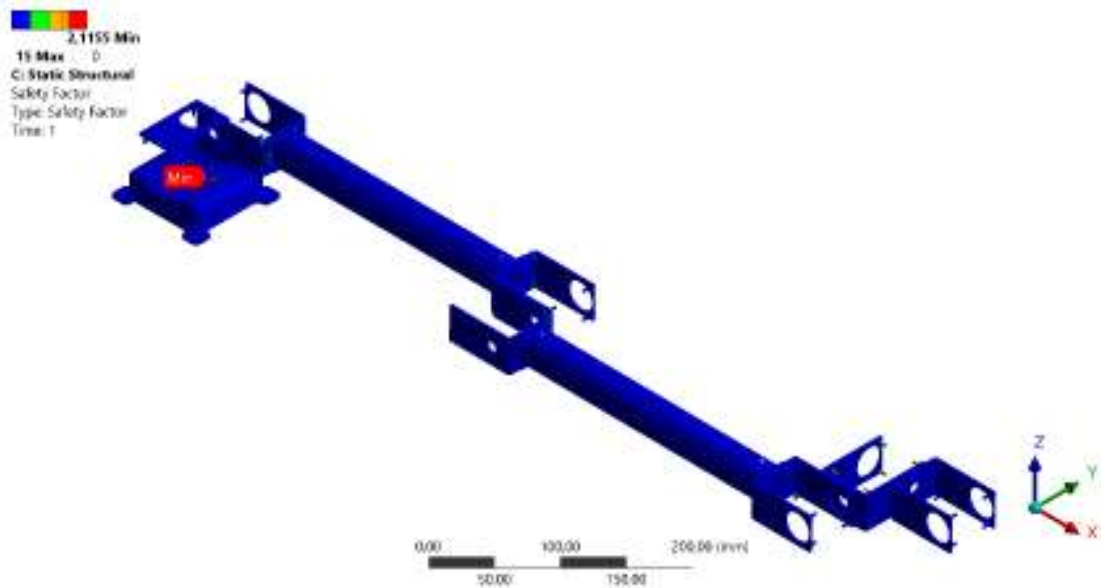


Figura 2.44. Distribución de factores de seguridad.
(Fuente: Propia)

Valor mínimo: 2.11

Ubicación: Base

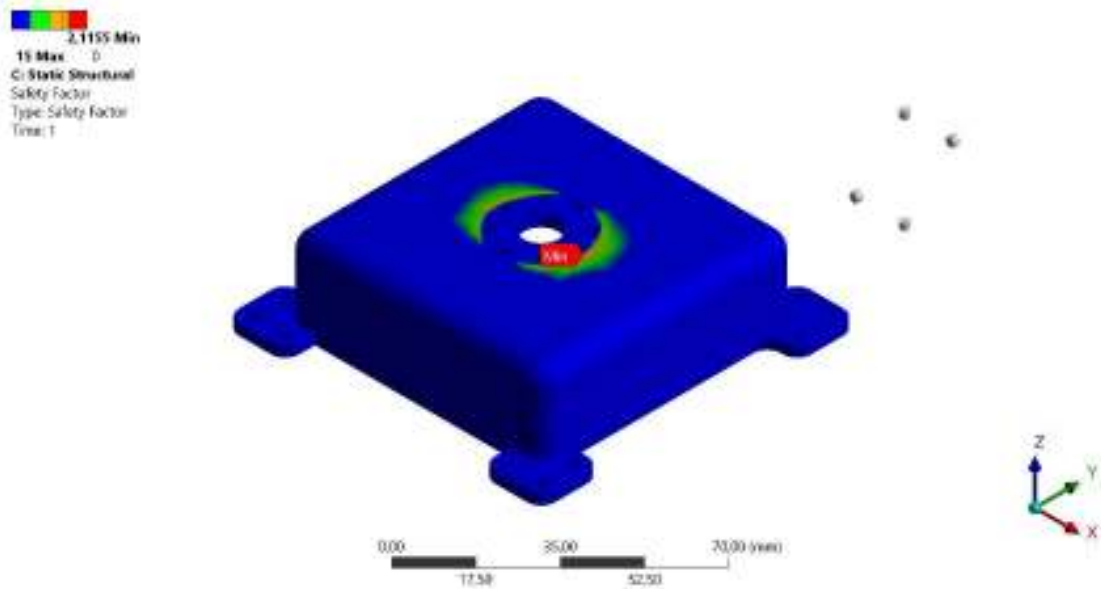


Figura 2.45. Distribución de factores de seguridad del cuerpo crítico.
(Fuente: Propia)

Deformación direccional

Cuerpo 1

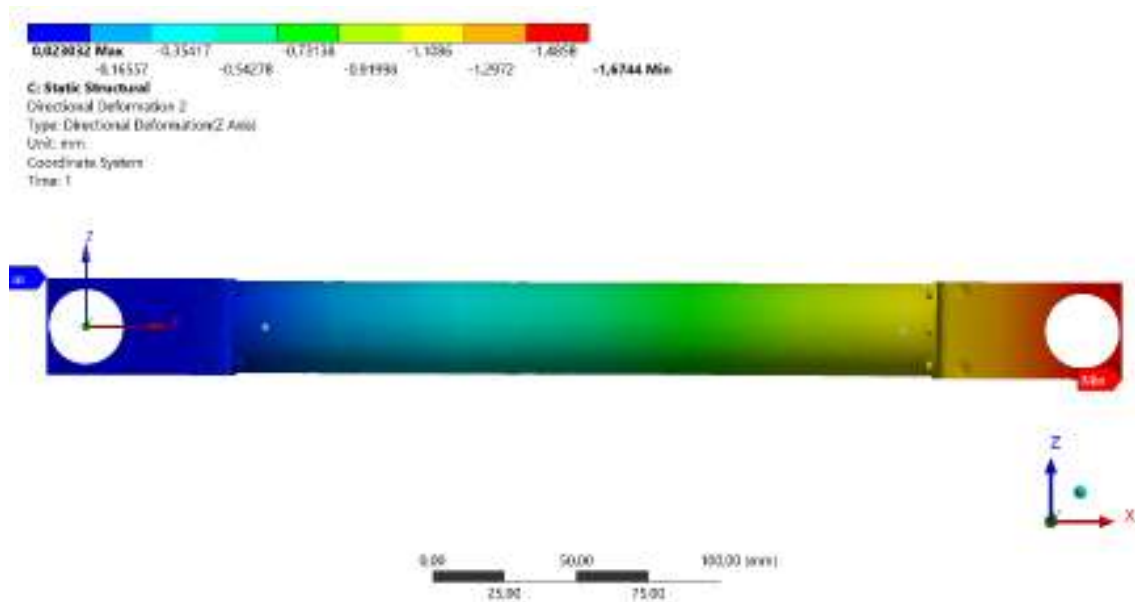


Figura 2.46. Deformación direccional del cuerpo 1.
(Fuente: Propia)

Deformación máxima: 1.67 mm

Cuerpo 2

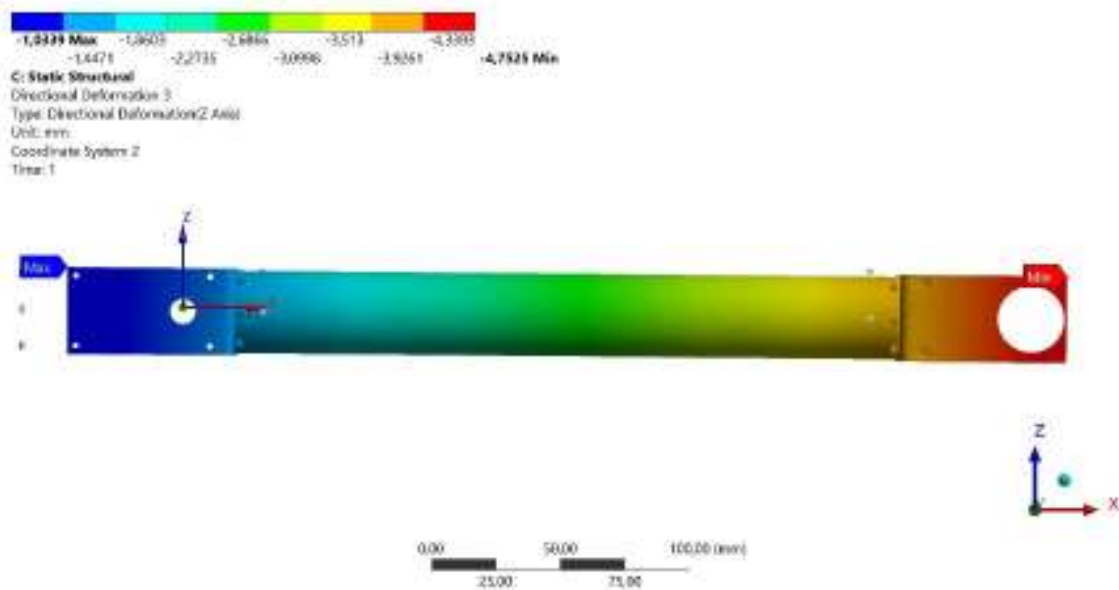


Figura 2.47. Deformación direccional del cuerpo 2.
(Fuente: Propia)

Deformación máxima: 3.08 mm

Análisis de resultados

En los resultados que se obtuvieron usando ANSYS se observa que las reacciones calculadas siguen sin ser exactamente iguales a las obtenidas en la simulación, pero son muy similares por lo que se consideran correctas.

Se observa además que ahora el esfuerzo equivalente máximo ya no supera al valor del límite elástico del material en la base del manipulador, esto quiere decir que permanece en la región elástica. Los valores obtenidos en el factor de seguridad, el valor mínimo es 2.11 que se puede considerar como satisfactorio.

La deformación direccional máxima obtenida es de 3.07 mm en el eslabón del antebrazo, por lo que habría que tomarlo en cuenta al momento de generar el movimiento del manipulador y corregirlo. El eslabón del brazo permanece en el rango requerido.

2.3.5. Solución del diseño de materialización

El diseño estructural final se puede observar en la siguiente imagen. Una vez conseguida una solución satisfactoria para el manipulador, se puede continuar con el sistema control del robot para cumplir el objetivo de seguir trayectorias en las que realice cordones de soldadura.

Las dimensiones de los componentes y la forma de ensamble del manipulador se encuentran en el ANEXO VI.

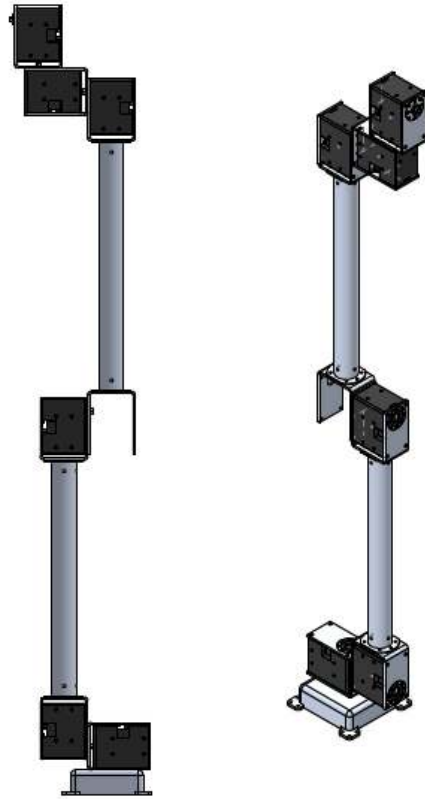


Figura 2.48. Modelo final del manipulador robótico.
(Fuente: Propia)

2.4. Modelo robótico

2.4.1. Importar modelo CAD a Simulink

El importar el modelo permite representar las piezas CAD, denominadas cuerpos en el software Simscape Multibody [23], utilizando subsistemas Simulink que comprenden múltiples bloques de transformación rígida y sólida. Los bloques sólidos proporcionan las geometrías, inercias y colores del cuerpo. Los bloques Rigid Transform proporcionan a los marcos las poses necesarias para la conexión entre cuerpos.

La traducción de un modelo CAD se consigue en dos pasos: exportar e importar.

1. El paso de exportación convierte el modelo de ensamblaje CAD en un archivo de descripción XML multicuerpo y un conjunto de archivos de geometría de pieza STEP o STL.
2. El paso de importación convierte la descripción multicuerpo y los archivos de geometría de la pieza en un modelo SLX Simscape Multibody y un archivo de datos m. El modelo obtiene todas las entradas de parámetros de bloque del archivo de datos.

2.4.1.1. Exportar CAD

Para esta sección es necesario que el manipulador tenga el efector final requerido para cumplir su función, por lo que se aumenta una antorcha de soldadura GMAW. Para esta parte se tomará un modelo genérico de antorcha para mostrar su funcionalidad. En la imagen siguiente se observa el manipulador en Solidworks ensamblado y listo para ser exportado a Matlab.



Figura 2.49. Modelo CAD final del manipulador y la antorcha de soldadura.
(Fuente: Propia)

2.4.1.2. Modelo importado a Simulink

Siguiendo los procedimientos descritos anteriormente se obtiene el modelo exportado a Simulink usando Simscape Multibody, en las siguientes imágenes se observa el modelo de bloques creado en Simulink y una vista del modelo en Matlab.

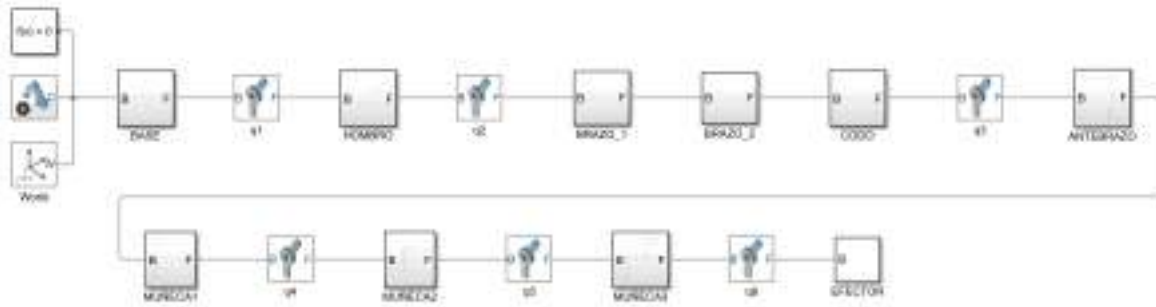


Figura 2.50. Diagrama de bloques del manipulador importado a Simulink.
(Fuente: Propia)

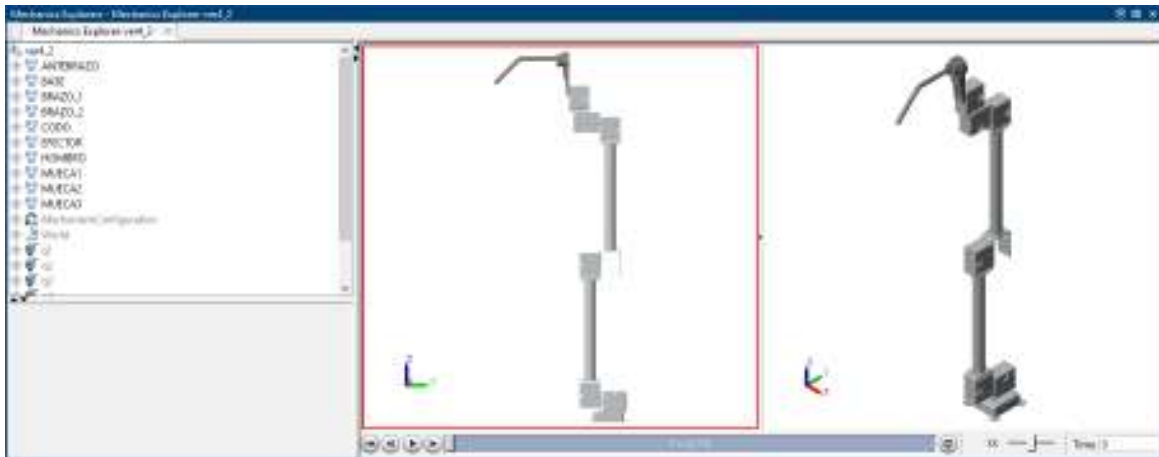


Figura 2.51. Vista en Matlab del manipulador importado.
(Fuente: Propia)

2.4.2. Importar diagrama de bloques a Matlab

2.4.2.1. Sistemas de coordenadas de referencia

La convención del sistema de coordenadas se tomará del existente en el Toolbox Robotics System [24] como el que se muestra en la siguiente imagen, donde:

El eje X es de color rojo y positivo en la dirección que sale del papel.

El eje Y es de color verde y positivo hacia la derecha.

El eje Z es de color azul y positivo hacia arriba.

Todas las articulaciones girarán alrededor del eje Z y siguen la convención de la mano derecha.

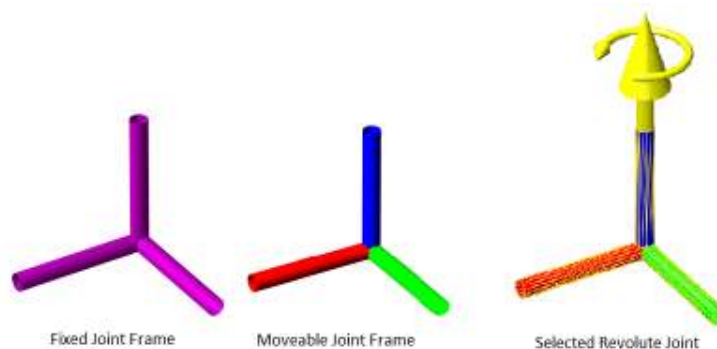


Figura 2.52. Marcos de referencia para los eslabones del manipulador.
(Fuente: <https://la.mathworks.com/help/robotics/ref/rigidbodytree.show.html>)

2.4.2.2. Modelo robótico con Toolbox Robotics System Toolbox

Usando la función `importrobot` de Robotics System Toolbox de Matlab es posible importar el modelo de árbol de cuerpo rígido (`rigidBodyTree`) desde el modelo Simscape Multibody.

Mediante esta función se crea el modelo robótico en Matlab a partir del cual se trabajará para crear los modelos cinemático y dinámico. también se usará este Toolbox para generar las trayectorias y sistemas de control de movimiento del manipulador.

Al importar el modelo de bloques de Simulink se obtiene un modelo robótico en el que se encuentran todos los datos geométricos y físicos útiles para los cálculos posteriores.

Robot: (7 bodies)

Idx	Body Name	Simulink Source Blocks	Joint Name	Simulink Source Blocks	Joint Type	Parent Name (Idx)	Children Name (s)
1	Body1	Info List Highlight	Joint1	Info List Highlight	revolute	Base (0)	Body2 (2)
2	Body2	Info List Highlight	Joint2	Info List Highlight	revolute	Body1 (1)	Body3 (3)
3	Body3	Info List Highlight	Joint3	Info List Highlight	revolute	Body2 (2)	Body4 (4)
4	Body4	Info List Highlight	Joint4	Info List Highlight	revolute	Body3 (3)	Body5 (5)
5	Body5	Info List Highlight	Joint5	Info List Highlight	revolute	Body4 (4)	Body6 (6)
6	Body6	Info List Highlight	Joint6	Info List Highlight	revolute	Body5 (5)	Body7 (7)
7	Body7	Info List Highlight	Joint7	Info List Highlight	fixed	Body6 (6)	

Figura 2.53. Relación entre el modelo importado y el modelo robótico.
(Fuente: Propia)

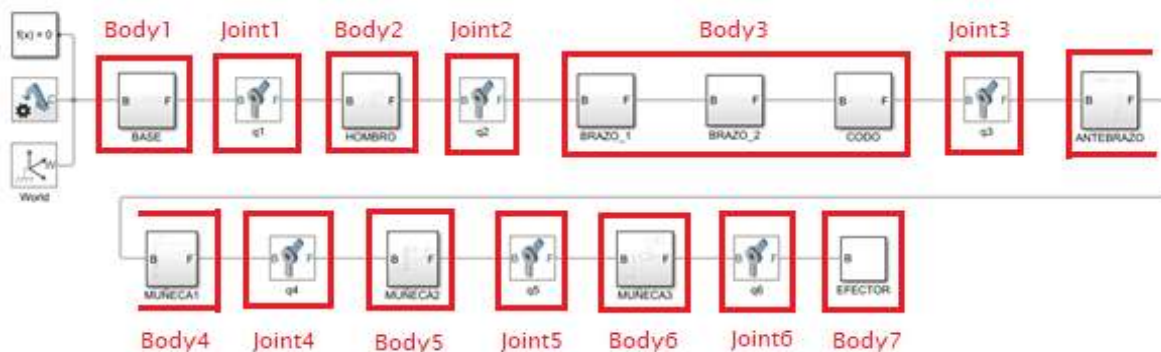


Figura 2.54. Bloques pertenecientes a cada elemento del modelo robótico.
(Fuente: Propia)

Cada cuerpo cuenta con un sistema de coordenadas ubicado en su eje de giro. Además, para el efector final del robot incluye un marco de referencia fijo que servirá de referencia para hallar la ubicación y orientación del extremo del manipulador. En la imagen siguiente se visualiza el modelo robótico importado.

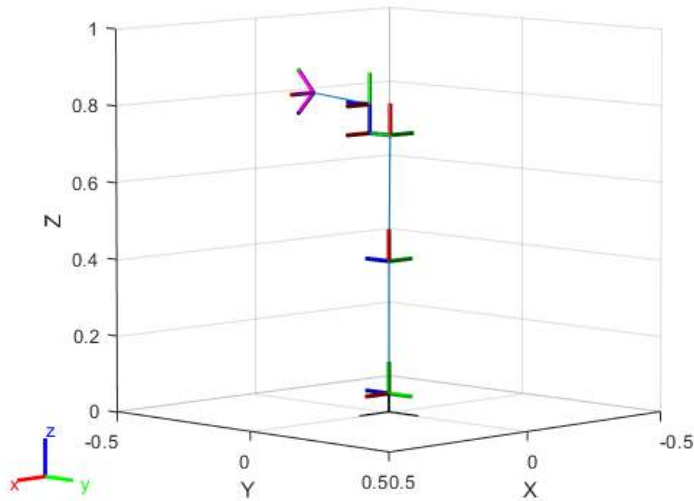


Figura 2.55. Sistemas de coordenadas locales de cada cuerpo del robot.
(Fuente: Propia)

2.5. Análisis cinemático y dinámico del manipulador

Para realizar el análisis cinemático y dinámico del manipulador robótico se usarán funciones definidas en el Toolbox Robotics System. De esta manera no es necesario realizar soluciones analíticas y se aprovechan las herramientas de esta librería.

A continuación, se describen las funciones disponibles en el Toolbox que resultan más idóneas en cada situación.

2.5.1. Análisis cinemático

2.5.1.1. Solución al problema cinemático directo

2.5.1.1.1. Función getTransform [24]

Usando la función getTransform es posible obtener la matriz de transformación homogénea entre dos marcos de referencia.

Su sintaxis es de la forma:

$$\text{transform} = \text{getTransform}(\text{robot}, \text{configuration}, \text{sourcebody}, \text{targetbody})$$

donde, robot es el modelo robótico, configuration son las posiciones articulares, sourcebody es el cuerpo de referencia y targetbody es el cuerpo objetivo del que se obtendrá la transformación homogénea.

2.5.1.2. Solución al problema cinemático inverso

2.5.1.2.1. Función `generalizedInverseKinematics` [24]

Esta función crea un solucionador de cinemática inversa con múltiples restricciones cinemáticas para calcular una configuración de las articulaciones del modelo robótico. Esta función usa un solucionador no lineal para alcanzar la mejor aproximación a la solución

Para solucionar un problema cinemático inverso se requiere:

- a) Crear la función `generalizedInverseKinematics` y definir sus propiedades
- b) Llamar a la función con argumentos.

Existen las siguientes restricciones:

`constraintAiming`: Orienta el extremo del robot hacia un punto en el espacio.

`constraintCartesianBounds`: Restringe el movimiento de un cuerpo del robot a un rango cartesiano.

`constraintJointBounds`: Limita el rango de movimiento de una articulación.

`constraintOrientationTarget`: Determina la orientación de un cuerpo.

`constraintPoseTarget`: Calcula la cinemática del manipulador hacia una transformación homogénea.

`constraintPositionTarget`: Mueve el manipulador a una posición cartesiana especificada.

Crear la función tiene la forma:

$$\text{gik} = \text{generalizedInverseKinematics}$$

Se llama a la función y realizar el cálculo de la cinemática inversa se hace con la sintaxis:

$$[\text{configSol}, \text{solInfo}] = \text{gik}(\text{initialguess}, \text{constraintObj}, \dots, \text{constraintObjN})$$

Donde, `initialguess` es la estimación inicial para el cálculo y las demás entradas son las restricciones aplicadas.

2.5.1.3. Matriz Jacobiana

2.5.1.3.1. Función `geometricJacobian` [24]

Esta función computa el Jacobiano geométrico para una configuración especificada del manipulador.

Su sintaxis es:

$$\text{jacobian} = \text{geometricJacobian}(\text{robot}, \text{configuration}, \text{endeffectorname})$$

donde, `robot` es el modelo robótico, `configuration` son los ángulos de las articulaciones y `endeffectorname` es el cuerpo del que se obtiene el Jacobiano.

En el caso de la matriz Jacobiana inversa solo es necesario realizar una inversión de la matriz Jacobiana usando funciones existentes en Matlab.

2.5.2. Análisis dinámico

2.5.2.1. Solución al problema dinámico directo

2.5.2.1.1. Función forwardDynamics [24]

Esta función calcula las aceleraciones de las articulaciones en base a los torques de las articulaciones.

La sintaxis de esta función es de la forma:

$$\text{jointAccel} = \text{forwardDynamics}(\text{robot}, \text{configuration}, \text{jointVel}, \text{jointTorq}, \text{fext})$$

donde, robot es el modelo robótico, configuration son las posiciones articulares, jointVel son las velocidades de las articulaciones, jointTorq son los torques aplicados a los eslabones y fex son fuerzas externas aplicadas a cada articulación.

2.5.2.2. Solución al problema dinámico inverso

2.5.2.2.1. Función inverseDynamics [24]

inverseDynamics calcula los torques requeridos para unas condiciones de movimiento especificadas.

Tiene la sintaxis:

$$\text{jointTorq} = \text{inverseDynamics}(\text{robot}, \text{configuration}, \text{jointVel}, \text{jointAccel}, \text{fext})$$

donde, robot es el modelo robótico, configuration son las posiciones articulares, jointVel son las velocidades de las articulaciones, jointAccel son las aceleraciones de los eslabones y fex son fuerzas externas aplicadas a cada articulación.

2.6. Control

2.6.1. Control cinemático

2.6.1.1. Tareas del control cinemático

El funcionamiento del control cinemático manipulador establece las trayectorias para todas las articulaciones del manipulador como función del tiempo. Esto se consigue a partir de las entradas de la información del movimiento definidos por el usuario [4].

Las tareas que debe cumplir el robot manipulador se establecieron en el análisis funcional del robot, desarrollado en una sección anterior y desglosan de la siguiente manera:

- a) Definir la geometría a seguir para conseguir el cordón de soldadura.
- b) Crear los puntos en el espacio por el que pasa el extremo del manipulador, así como su orientación.
- c) Posicionar el manipulador en el punto y orientación inicial de la trayectoria a seguir.
- d) Ubicar el efector final del robot en cada uno de los puntos generados de la trayectoria definida.

- e) En caso de que exista más de una trayectoria definida, es necesario mover el manipulador desde el punto final de una trayectoria al punto inicial de la siguiente.
- f) Finalmente, regresar el manipulador a su posición inicial desde el punto final de la última trayectoria deseada.

2.6.1.2. Generación de trayectorias

2.6.1.2.1. Selección del espacio de movimiento

El espacio de movimiento establece los métodos de generación de trayectorias en los que la forma de los puntos de recorrido se describe en términos de funciones de ángulos de las articulaciones [3].

El movimiento del manipulador en las trayectorias definidas requiere un control de posición muy preciso en el sistema cartesiano debido a que las trayectorias definidas son en puntos cartesianos y orientaciones de Euler en XYZ.

Por este motivo, para el movimiento del extremo final del robot en la trayectoria cartesiana se usará el espacio cartesiano.

El movimiento del manipulador entre trayectorias y en el posicionamiento inicial y final no es muy estricto en cuanto al seguimiento del movimiento establecido entre dichos puntos. Por lo tanto, para estos segmentos de movimiento el uso del espacio de las articulaciones es suficiente.

2.6.1.2.2. Selección del tipo de trayectoria

La trayectoria describe la evolución temporal de la posición, velocidad y aceleración de cada grado de libertad [3].

Debido a que el espacio en el que trabaja el manipulador al seguir trayectorias cartesianas, el tipo de trayectoria requerida es trayectoria continua.

Del mismo modo, para el resto de las secciones de movimiento, el tipo de trayectoria más útil es trayectoria punto a punto coordinada.

Función transformtraj [24]

Esta función genera una trayectoria que interpola entre dos transformaciones homogéneas de 4 por 4, con puntos basados en el intervalo de tiempo y muestras de tiempo dadas.

Su sintaxis es:

$$[tforms, vel, acc] = transformtraj(T0, TF, tInterval, tSamples, Name, Value)$$

Donde, T_0 es la transformación inicial, T_f es la transformación final, $t_{samples}$ son los puntos de cálculo, $name$ y $value$ son parámetros adicionales.

2.6.1.2.3. Selección del interpolador de trayectoria

El interpolador de trayectoria realiza la función de conectar esta serie de puntos de trayectoria para asegurar que se respeten las limitaciones de velocidad y aceleración de los actuadores del robot, a fin de obtener una trayectoria factible y suficientemente suave [4].

Dado que es un requerimiento de diseño, y también por las ventajas que ofrece en comparación a los demás tipos de interpoladores, el uso de un interpolador trapezoidal es el mejor para los requerimientos establecidos.

Función trapveltraj [24]

Esta función genera una trayectoria con perfil de velocidad trapezoidal a través de un conjunto dado de puntos de trayectoria. La función genera posiciones, velocidades y aceleraciones en muestras de tiempo, basándose en el número especificado de muestras, numSamples.

$$[q, qd, qdd, tSamples, pp] = \text{trapveltraj}(\text{wayPoints}, \text{numSamples})$$

Donde, wayPoints son los puntos de trayectoria para el perfil y numSamples son los puntos de muestra para el cálculo.

2.6.1.3. Programa generador de trayectorias

Las tareas requeridas para el control cinemático ya se han establecido. El siguiente paso es crear el programa generador de trayectorias usando Matlab y el Toolbox Robotics System.

En [7] se detalla el proceso de planificación de trayectorias de movimiento para manipuladores robóticos usando el Toolbox Robotics System de Matlab. Teniendo como base este trabajo se procederá a crear el programa generador de trayectorias.

En el siguiente diagrama se puede observar el proceso a seguir y los resultados que se esperan obtener.

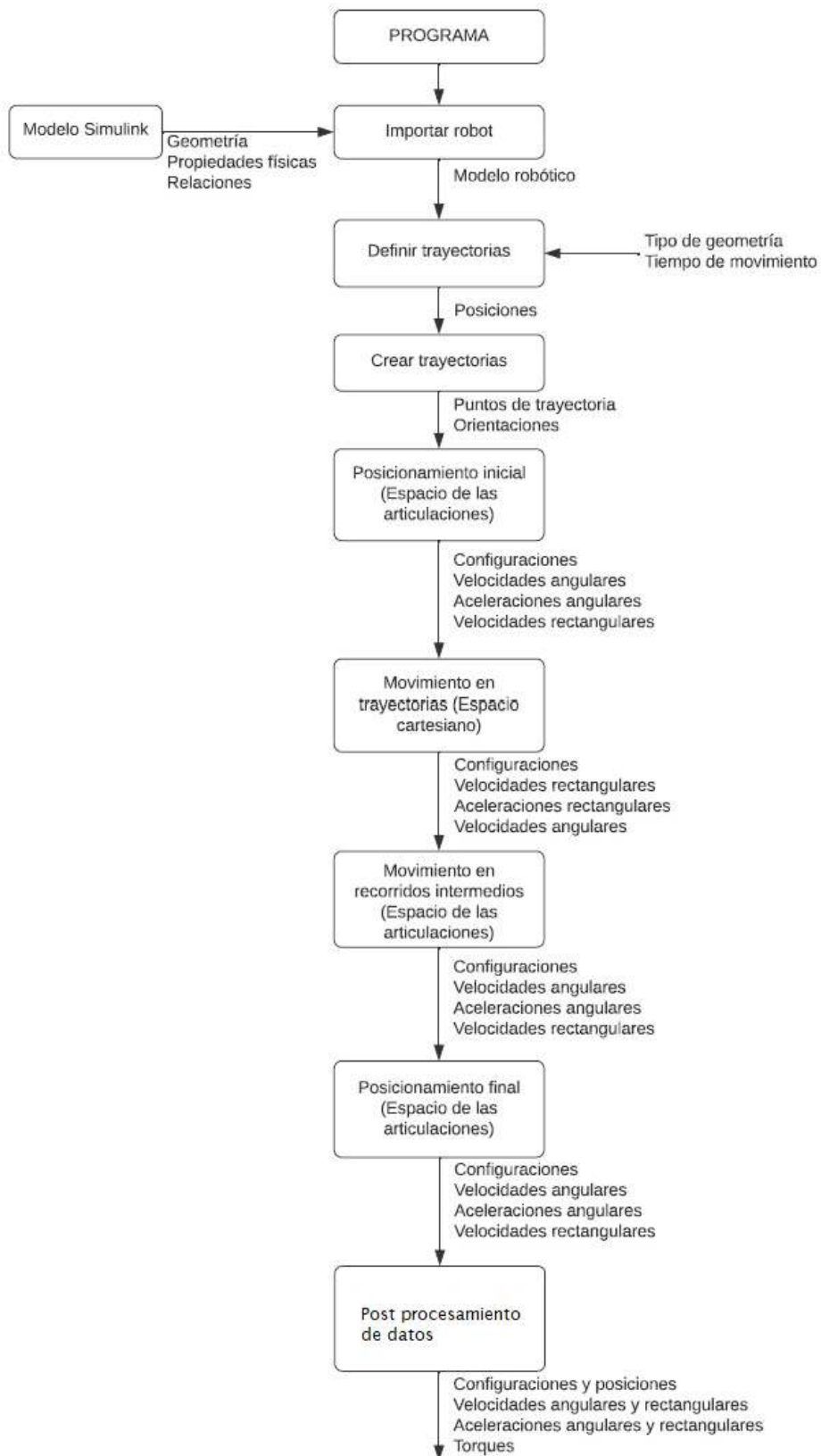


Figura 2.56. Programa generador de trayectorias para el control cinemático.
(Fuente: propia)

2.6.1.3.1. Definir trayectorias

Las trayectorias rectangulares que seguir se definen mediante las dimensiones de los lados, ubicación del punto central de las geometrías, el número de lados consecutivos de movimiento, el plano en el que se están los recorridos, el sentido de movimiento y la orientación del efector final del manipulador. Además, es necesario definir el tiempo que necesitara el manipulador para realizar cada parte de la trayectoria.

2.6.1.3.2. Crear trayectorias

Las trayectorias se crean y establecen las coordenadas cartesianas de los vértices de las trayectorias rectangulares, así como las orientaciones del extremo del robot usando ángulos de Euler. Finalmente se designan las matrices de transformación homogénea de los puntos iniciales de cada trayectoria especificada. Todos estos datos se usan en el cálculo de la cinemática inversa. Todo esto se realiza con una función personalizada programada en Matlab.

2.6.1.3.3. Posicionamiento inicial (Espacio de las articulaciones)

Usando la configuración inicial del manipulador se calcula la configuración en el primer punto del recorrido rectangular usando cinemática inversa. Seguidamente, obtenidas las dos configuraciones se genera la trayectoria en el espacio de las articulaciones con perfiles de velocidad trapezoidales.

Para esta parte se crea una función en Matlab en la que usan las funciones `trapveltraj` para generar la trayectoria de movimiento con perfiles de velocidades trapezoidales y `generalizedInverseKinematics` para calcular la configuración del robot entre los puntos de movimiento.

2.6.1.3.4. Movimiento en trayectorias (Espacio cartesiano)

Usando las coordenadas de los vértices de los caminos y las orientaciones del extremo del manipulador se crean matrices de transformación homogénea. Luego, se generan trayectorias en el espacio cartesiano entre dichas matrices con perfiles de velocidad trapezoidal. Finalmente, en cada punto de la trayectoria realizada se realiza el cálculo de la configuración del manipulador usando cinemática inversa.

Con una función personalizada se realiza esta sección usando las funciones propias de robótica: `transformtraj` que crea trayectorias de movimiento con perfiles de velocidad trapezoidales entre las transformaciones homogéneas correspondientes a los vértices de las trayectorias rectangulares. Así como la función `generalizedInverseKinematics`, con restricciones `constraintCartesianBounds` para limitar el área de movimiento del manipulador robótico, `constraintJointBounds` para que el movimiento de las articulaciones sea suave y

constraintPoseTarget de tal manera que el cálculo de la cinemática inversa sea entre transformaciones homogéneas.

2.6.1.3.5. Movimiento en recorridos intermedios (Espacio de las articulaciones)

El procedimiento para estos recorridos y las configuraciones del robot en los recorridos es el mismo que aquel del posicionamiento inicial. Varían únicamente las configuraciones requeridas para el cálculo.

En este punto se usa la misma función que en el posicionamiento inicial.

2.6.1.3.6. Posicionamiento final (Espacio de las articulaciones)

En este paso del proceso se realiza el procedimiento inverso al del posicionamiento inicial. Con la configuración del punto final de la trayectoria especificada se realiza el recorrido en el espacio de las articulaciones hasta la configuración inicial del robot.

Igualmente, aquí se usa la misma función que en el posicionamiento inicial.

2.6.1.3.7. Post procesamiento

Usando todas las configuraciones obtenidas en el recorrido total del manipulador se realiza la cinemática directa para obtener las matrices de transformación homogénea en cada punto del recorrido y usando estas matrices se extraen las coordenadas del extremo del manipulador. Además, se calcula el torque requerido en cada articulación usando dinámica inversa.

En Matlab se procede a reordenar los resultados obtenidos en las secciones anteriores y además se usan las funciones getTransform para obtener las matrices de transformación homogénea y la función inverseDynamics para el cálculo de la dinámica inversa.

2.7. Sistema mecatrónico del manipulador

2.7.1. Sistema mecatrónico

El sistema mecatrónico del manipulador robótico incluye la estructura mecánica, los componentes eléctricos y electrónicos y el programa generador de trayectorias. En el diagrama de bloques a continuación se muestra la relación que tienen cada uno de estos componentes en este robot.

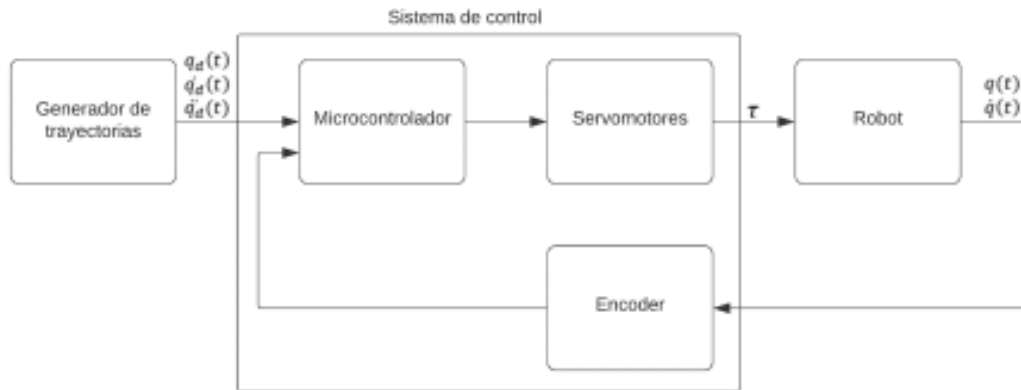


Figura 2.57. Elementos constituyentes del sistema mecatrónico del manipulador.
(Fuente: Bolton, [9])

Seguidamente se procede a escoger los elementos de este sistema. El robot y el generador de trayectorias ya se han especificado, por lo cual solo queda definir los componentes del sistema de control.

2.7.2. Seleccionar componentes

2.7.2.1. Requerimientos

En [15] se presenta los criterios requeridos a la hora de escoger un motor eléctrico. Al momento de elegir un motor para una aplicación requerida, se deben tener en cuenta factores que incluyen:

Relación de inercia: El momento de inercia de la carga debe ser similar al del motor para obtener la máxima transferencia de potencia.

Torque: No hay que exceder el valor de par de bloqueo para funcionamiento continuo. Este es el valor de par máximo al que no se producirá un sobrecalentamiento. Son posibles pares mayores para uso intermitente.

Potencia: El motor debe poder funcionar a la velocidad máxima requerida sin sobrecalentarse.

Para la selección de los servomotores de este robot, se tomarán en consideración el torque y velocidad máximas requeridos. En el siguiente cuadro se presentan los valores máximos requeridos para el correcto funcionamiento del manipulador. Usando estos datos se procederá a escoger los servomotores idóneos.

Tabla 2.28. Variables para elegir los actuadores.

Variable	Valor
Torque máx. (Nm)	10.044
Velocidad máx. (rad/s)	2.36 (22.44 RPM)

(Fuente: Propia)

Después de una búsqueda de las opciones disponibles en el mercado, se ha llegado a la conclusión de que los servomotores Dynamixel de la empresa Robotis son los más idóneos para ser aplicados en este manipulador.

Dynamixel son actuadores inteligentes para robots con motor DC, engranes de reducción, controlador totalmente integrados en un servomódulo DC [25].

La serie DYNAMIXEL-XH / XM ofrece una variedad de técnicas de control basadas en algoritmos de detección de corriente de alto rendimiento [26].

Por este motivo se escogerán dentro de la gama de productos de esta empresa los actuadores que mejor cumplan los requerimientos expresados.

2.7.2.2. Comparación de rendimiento de las opciones

En la siguiente imagen se puede ver un esquema en el que se comparan los distintos motores para el torque de bloqueo y velocidad sin carga.

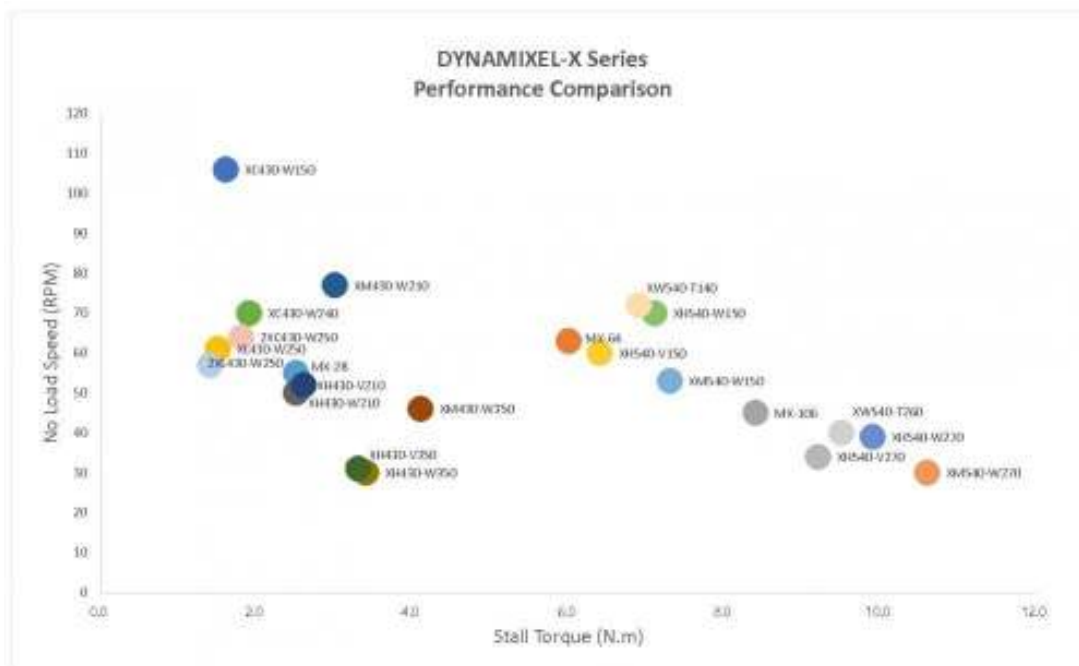


Figura 2.58. Comparación de rendimiento de los actuadores Dynamixel.
(Fuente: http://en.robotis.com/shop_en/item.php?it_id=902-0133-000#sit_rel)

2.7.2.3. Selección del servomotor:

Debido a que el torque máximo requerido es alto y la velocidad máxima es relativamente baja, basado en los actuadores disponibles en la marca selecciona, se elige el servomotor XM540-W270-R que cumple con las necesidades de torque y velocidad. En la imagen y cuadro siguiente se puede ver la geometría del actuador en conjunto con sus principales características.

2.7.2.3.1. Servomotor XM540-W270-R [26]



Figura 2.59. Servomotor XM540-W270-R.
(Fuente: http://en.robotis.com/shop_en/item.php?it_id=902-0133-000#sit_rel)

Tabla 2.29. Especificaciones técnicas del servo motor XM540-W270-R.

Nombre del modelo		XM540-W270-R
Voltaje de entrada	Min. [V]	10.0
	Recomendado [V]	12.0
	Max. [V]	14.8
Características de rendimiento	Voltaje [V]	12.0
	Torque de bloqueo [Nm]	10.60
	Corriente de bloqueo [A]	4.4
	Velocidad sin carga [rpm]	30.0
	Corriente sin carga [A]	0.07
Sensor de posición		Enconder absoluto sin contacto
Algoritmo de control		PID
Tipo de engrane / Material		Recto / Metal
Relación de transmisión		272.5:1
Materiales de cubierta		Metal (Frontal, Medio), plástico ingenieril (Posterior)
Dimensiones (W x H x D) [mm]		33.5 X 58.5 X 44
Peso [g]		165.00
Realimentación		Posición, velocidad, corriente, trayectoria, temperatura, voltaje, puerto externo, etc.

(Fuente: http://en.robotis.com/shop_en/item.php?it_id=902-0133-000#sit_rel)

2.7.2.4. Selección del encoder

El encoder de tipo absoluto sin contacto para el servomotor viene incluido en el modelo XM540-W270-R.

2.7.2.5. Selección del controlador

El controlador requerido para el motor elegido viene especificado. A continuación, se detalla más acerca de las características de este.

2.7.2.5.1. Controlador OpenCR 1.0

OpenCR1.0 es un controlador basado en software libre que tiene un potente MCU incluido de la línea ARM Cortex-M7. Admite RS-485 y TTL para controlar los servomotores Dynamixel. Soporta hasta 3 servomotores [27].

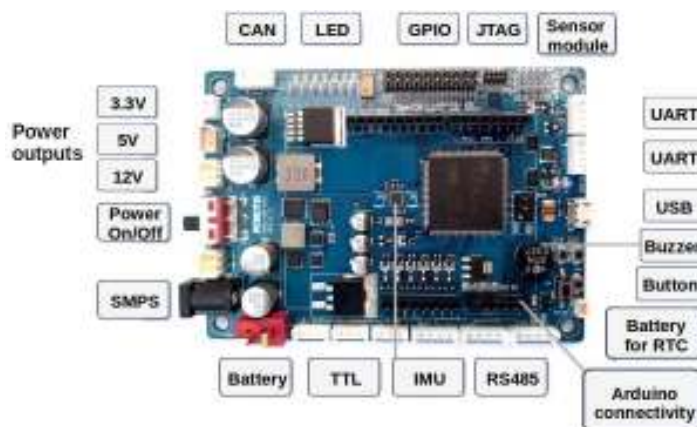


Figura 2.60. Controlador OpenCR 1.0 y sus partes.
(Fuente: <https://www.robotis.us/opencr1-0/>)

Placa de expansión OpenCm 485

La placa de expansión OpenCm 485 se utiliza para conectar motores Dynamixel que admiten RS-485 y TTL. Tiene un amplio rango de voltaje de entrada para conectar varios motores Dynamixel. Soporta hasta 4 servomotores [28].



Figura 2.61. Placa de expansión OpenCm 485.
(Fuente: <https://www.robotis.us/opencm-485-expansion-board/>)

2.7.3. Control dinámico

La tarea del control dinámico es asegurar que el movimiento seguida por el extremo final del robot $q(t)$ sea, en lo posible, lo más similar a la trayectoria especificada por el control cinemático [4].

En el diagrama del sistema mecatrónico se tiene que para lograr el movimiento preciso del manipulador es necesario controlar el funcionamiento de los servomotores asociados al brazo. El controlador OpenCR 1.0 es el que posibilita esto, para lo cual requiere controlar la posición y velocidad de los motores mediante una señal de retroalimentación obtenida del encoder.

Al momento de seleccionar el tipo de control dinámico del manipulador se debe tomar en cuenta que el motor XM540-W270-R posee una relación de reducción de 272.5:1. Esto causa que, al controlar el movimiento de un eslabón, la influencia de los demás cuerpos se minimiza en gran medida, como se explica en [4]. También, el controlador OpenCR viene incluido con un sistema de control PID. Por lo tanto, es posible usar un control mono articular PID para este robot.

Debido a que se pueden sintonizar las ganancias del PID con el software incluido con los servomotores, no es necesario desarrollar un sistema dinámico del brazo robot en este caso en particular y en su lugar se explicará con más detalle el funcionamiento del control de posición PID de los servomotores.

2.7.3.1. Control mono articular PID del manipulador [29]

Estas ganancias se utilizan en el modo de control de posición y en el modo de control de posición extendido. Las ganancias del controlador interno de Dynamixel se pueden calcular a partir de las ganancias de la tabla de control como se muestra a continuación. La constante en cada ecuación incluye el tiempo de muestreo. Posición P de la ganancia del controlador interno de Dynamixel se abrevia como $K_p P$ y la de la Tabla de control se abrevia como $K_p P_{TBL}$.

Tabla 2.30. Ganancias del controlador Dynamixel.

Ganancia	Ecuaciones de conversión	Rango	Descripción	
Position D (80)	$K_p D$	$K_p D = K_p D_{TBL}/16$	0 ~ 16,383	Ganancia D
Posición I (82)	$K_p I$	$K_p I = K_p I_{TBL}/65,536$	0 ~ 16,383	Ganancia I
Posición P (84)	$K_p P$	$K_p P = K_p P_{TBL}/128$	0 ~ 16,383	Ganancia P
2da Prealimentación (Feedforward) (88)	K_{FF2nd}	$K_{FF2nd}(TBL)/4$	0 ~ 16,383	Ganancia de prealimentación de aceleración
1ra Prealimentación (90)	K_{FF1st}	$K_{FF1st}(TBL)/4$	0 ~ 16,383	Ganancia de prealimentación de velocidad

(Fuente: <https://emanual.robotis.com/docs/en/dxl/x/xm540-w270/>)

La siguiente figura es un diagrama de bloques que describe el controlador de posición en el modo de control de posición y el modo de control de posición extendido. Cuando DYNAMIXEL recibe la instrucción del usuario, realiza los siguientes pasos hasta hacer mover el motor.

- a) Una instrucción del usuario se transmite a través del bus DYNAMIXEL y luego se registra en la posición de meta (116).
- b) La posición de meta (116) se convierte en la trayectoria de posición deseada y la trayectoria de velocidad deseada mediante la velocidad del perfil (112) y la aceleración del perfil (108).
- c) La trayectoria de posición deseada y la trayectoria de velocidad deseada se almacenan en Trayectoria de posición (140) y Trayectoria de velocidad (136) respectivamente.
- d) El controlador Feedforward y PID calculan la salida PWM para el motor en función de las trayectorias deseadas.
- e) El objetivo PWM (100) establece un límite en la salida PWM calculada y decide el valor final de PWM.
- f) El valor final de PWM se aplica al motor a través de un convertidor y se activa el eje del motor.
- g) Los resultados se almacenan en Posición actual (132), Velocidad actual (128), PWM actual (124) y Corriente actual (126).

K_a es una ganancia Anti-windup que el usuario no puede modificar.

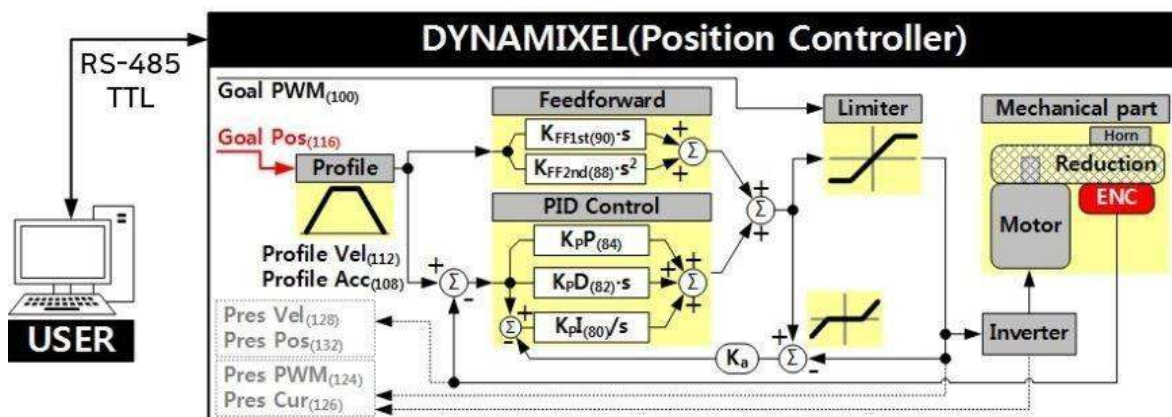


Figura 2.62. Diagrama bloques del controlador PID del servomotor Dynamixel.
(Fuente: <https://emmanual.robotis.com/docs/en/dxl/xm540-w270/>)

2.8. Simulación

2.8.1. Generar interfaz grafica

En esta parte se requiere modificar el diagrama de bloques del modelo CAD importado por defecto (Figura 2.50) para adaptarlo a los comandos especificados en el control cinemático y lograr que tenga el funcionamiento deseado. Para conseguir el movimiento se realizará un accionamiento directo de las articulaciones, es decir, los parámetros de entrada serán las posiciones, velocidades y aceleraciones articulares calculadas en el generador de trayectorias.

Al modificar el diagrama de bloques del modelo CAD importado a Simulink para hacer posible el movimiento de las articulaciones del robot, se tienen que agregar entradas para las posiciones, velocidades y aceleraciones a los bloques de junta de revolución del modelo mecánico. Adicionalmente, se incluyen sensores para estos parámetros y un sensor para los torques requeridos.

Para ver los valores obtenidos por los sensores se incluyen también ventanas de visualización (Scopes) para los valores obtenidos.

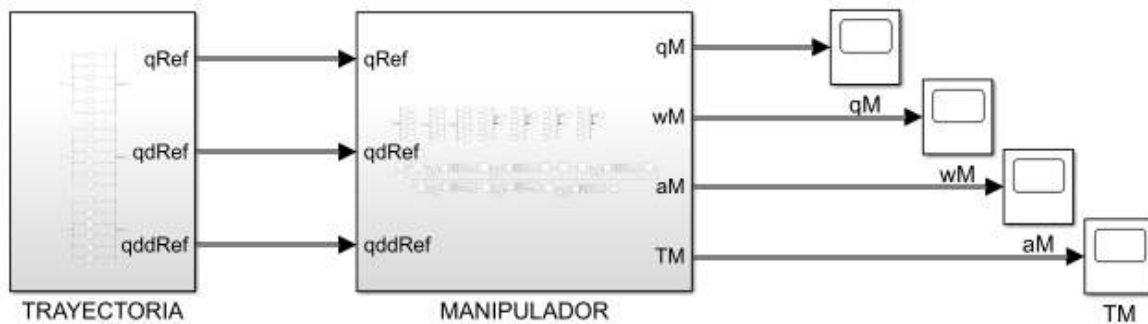


Figura 2.63. Diagrama de bloques del manipulador realizado en Simulink.
(Fuente: Propia)

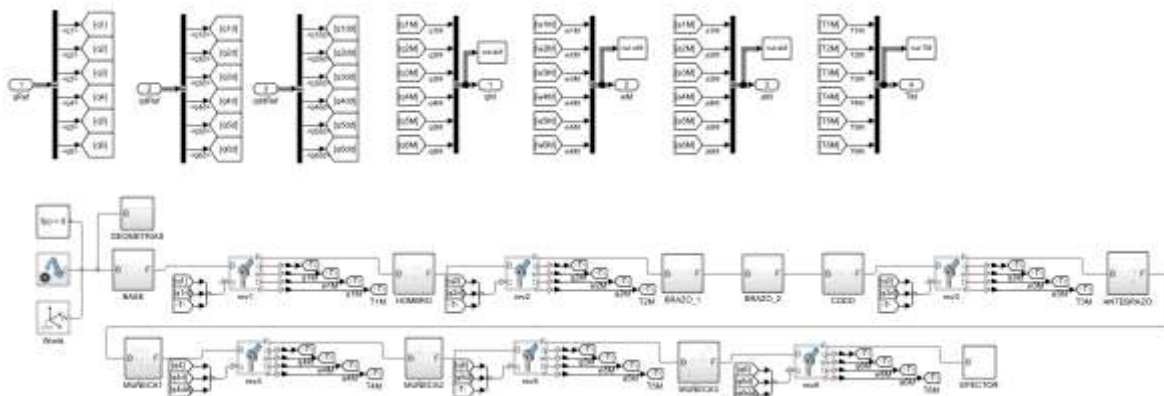


Figura 2.64. Conexiones de las entradas y salidas de los elementos del manipulador.
(Fuente: Propia)

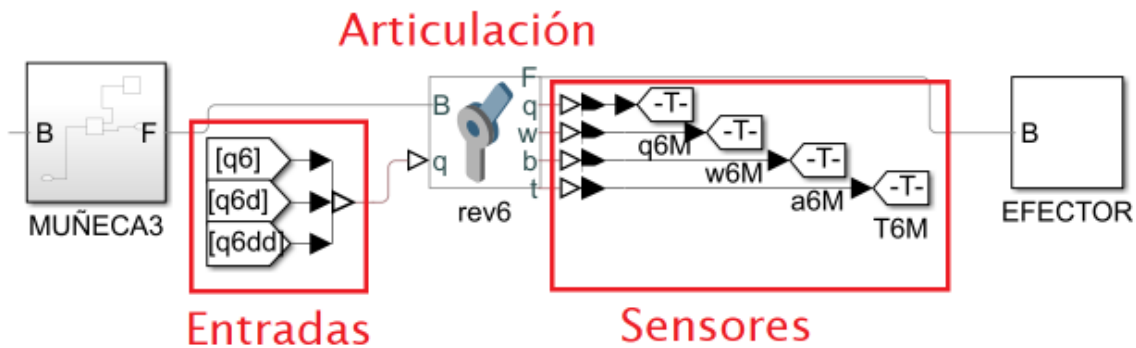


Figura 2.65. Entradas y sensores ubicados en cada articulación del robot.
(Fuente: Propia)

2.8.2. Simular funciones del manipulador

En Simulink se realiza la simulación del funcionamiento del manipulador. Usando las configuraciones obtenidas en el programa se accionan las articulaciones del robot y se observa su funcionamiento. Finalmente, se comprueba que el comportamiento sea adecuado a los requerimientos y las coordenadas rectangulares del programa se comparan con los obtenidos con el programa realizado en Matlab, así como los ángulos, velocidades, aceleraciones y torques.

3. RESULTADOS Y DISCUSIÓN

3.1. Sistema mecatrónico

Una vez escogidos todos los elementos del sistema mecatrónico, se presentan en el siguiente diagrama.

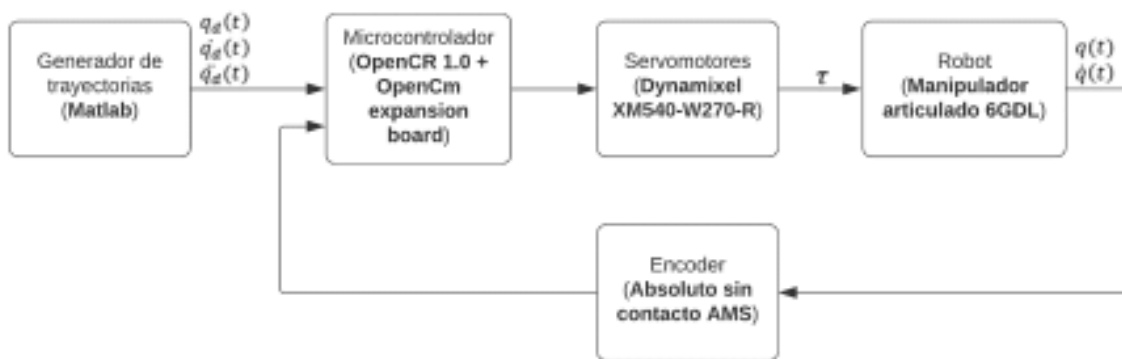


Figura 3.1. Elementos del Sistema mecatrónico.
(Fuente: Propia)

3.2. Especificaciones técnicas del manipulador

La siguiente tabla muestra algunas características técnicas del manipulador diseñado.

Tabla 3.1. Características técnicas del mecanismo.

Característica	Valor
Peso	2 kg
Carga útil	400 g
Alcance	750 mm
Rango de acción de las articulaciones	$\pm 360^\circ$
Grados de libertad	6
Velocidad máxima de giro de las articulaciones	30 RPM
Área de base	$\phi 75$ mm
Programación	Matlab
Fuente de alimentación	12 V 40 A
Fuente de potencia	120 V 60 Hz
Materiales	Aluminio, acero
Temperatura de funcionamiento	-5°, 80°

(Fuente: Propia)



Figura 3.2. Manipulador robótico articulado de seis grados de libertad.
(Fuente: Propia)

3.3. Funcionamiento del manipulador al moverse en una trayectoria

3.3.1. Resultados obtenidos usando Robotics System Toolbox

3.3.1.1. Volumen de trabajo del manipulador

En la siguiente figura se puede observar el modelo del manipulador usando el Toolbox Robotics System y una semiesfera sombreada que representa el volumen de trabajo del robot. Las dimensiones coinciden con las dimensiones mostradas en la figura 2.19, que era la estimación inicial del volumen de trabajo en base al alcance requerido.

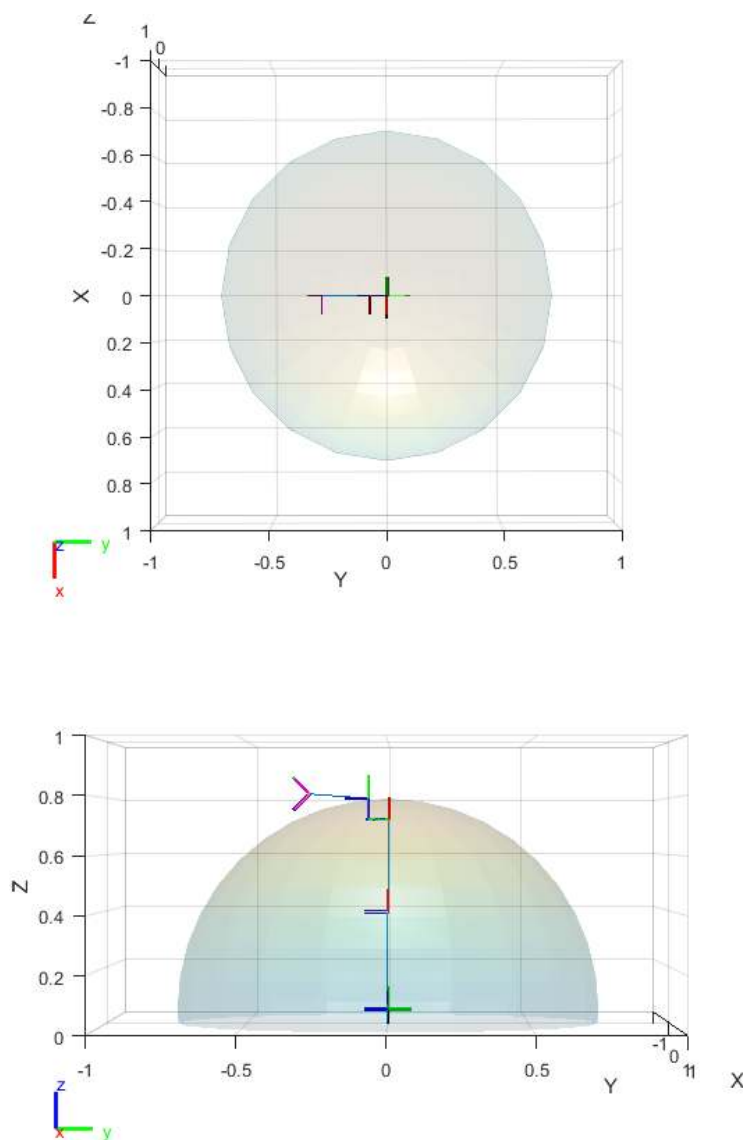


Figura 3.3. Volumen de trabajo del manipulador robótico usando Robotics System (Fuente: Propia)

3.3.1.2. Programa generador de trayectorias

A continuación, se realizará un análisis del movimiento en dos trayectorias cuadradas ubicadas en planos y sentido de movimiento distintos. El código del programa generador de trayectorias en Matlab se puede ver en el ANEXO IV.

3.3.1.2.1. Definir trayectorias

Las dos geometrías deseadas son cuadrados de longitud de arista de 100 mm, los centros de estos están en las coordenadas $(0.5, 0.3, 0)m$ y $(0.4, 0, 0.35)m$.

El primer cuadrado está ubicado en el plano XY y el movimiento será en sentido antihorario. El segundo en cuadrado está en el plano YZ y tiene movimiento en sentido horario.

El extremo de la antorcha de soldadura tendrá una orientación tal que apunte siempre al centro del cuadrado con un ángulo de 45° .

Los tiempos de movimiento de la posicionamiento inicial y final son 3 segundos para ambos casos.

El tiempo que el extremo del robot se demorará en recorrer cada lado de la primera geometría es de 4 segundos y 5 segundos para el caso de la segunda trayectoria.

El tiempo que tardará el manipulador en moverse entre los dos trayectos es 4 segundos.

El tiempo de muestreo para los cálculos requeridos será de 0.1 segundo y se usará un interpolador para refinar estos datos hasta 0.01 segundos.

```
%% Definir trayectorias y orientaciones
% Para generar rectangulos se especifican las longitudes de los lados y la
% ubicación de los centros de estos

% Longitud de los lados de las geometrías
geoms = [0.1 0.1 0.1;
         0.1 0.1 0.1];

% Ubicación de los centros de las geometrías
cents = [0.5 0.3 0;
         0.4 0 0.35];

% Numero de lados continuos en las trayectorias rectangulares
numlad = 4;

% Orientaciones del efector final del manipulador en los vertices de las
% geometrías de las trayectorias
% Planos de trayectorias
plan = ['XY';
        'YZ'];

% Dirección de movimiento
dir = ['AB';
        'BA'];

% Tipo de orientaciones
orien = ['OAT';
         'OAT'];
```

Figura 3.4. Definición de las geometrías rectangulares de movimiento.
(Fuente: Propia)

```

%% Tiempos de movimiento en trayectorias

% Tiempos de movimiento al punto inicial de la trayectoria y del punto final
% de las trayectorias a la ubicación inicial (ESPACIO ARTICULADO)
tmov = [3 3];

% Tiempo de recorrido en cada lado de las geometrias definidas
% (ESPACIO CARTESIANO)
ttraj = [4 5];

% Tiempos de colocación del manipulador entre trayectorias cuadradas
% (ESPACIO ARTICULADO)
tcol = [4];

% Tiempo de muestreo para los cálculos
deltat = 0.01;

```

Figura 3.5. Tiempos establecidos para el movimiento del manipulador.
(Fuente: Propia)

3.3.1.2.2. Crear trayectorias

A continuación, se observa el manipulador en su posición inicial y las dos trayectorias establecidas.

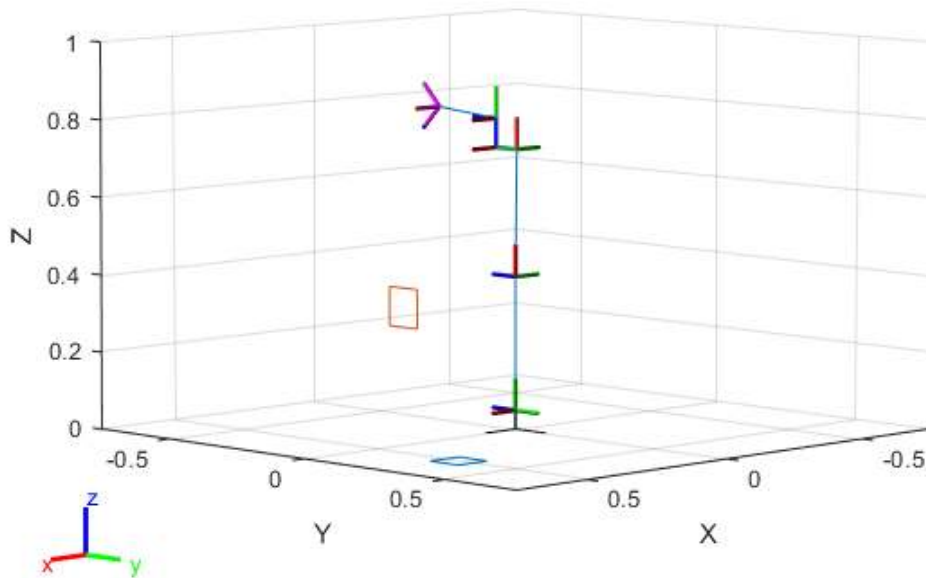


Figura 3.6. Robot manipulador y trayectorias cartesianas.
(Fuente: Propia)

3.3.1.2.3. Posicionamiento inicial (Espacio de las articulaciones)

En la figura siguiente se tiene el manipulador en la configuración que lo ubica en el primer punto de la trayectoria.

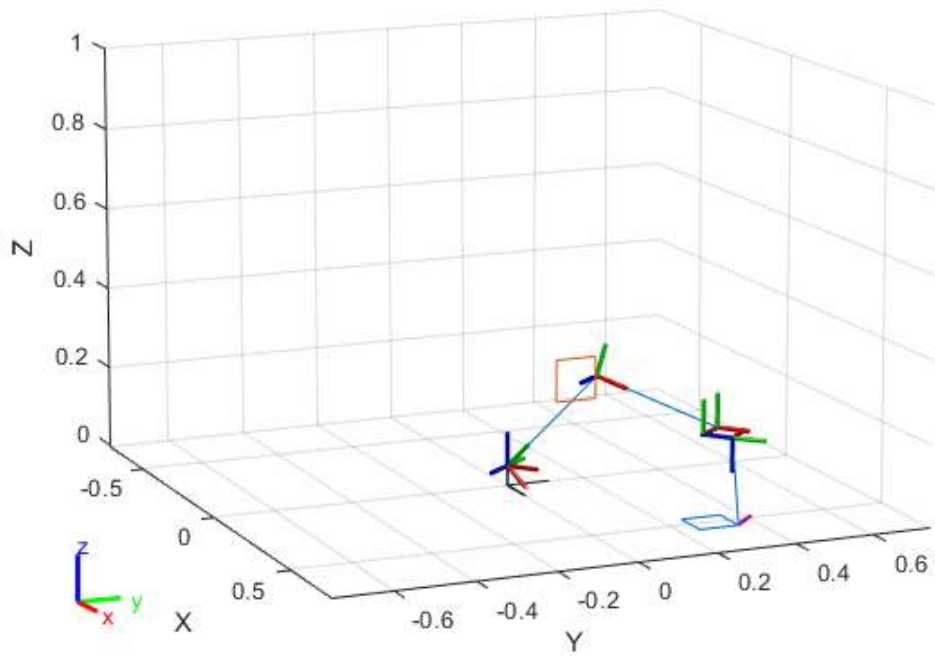


Figura 3.7. Manipulador ubicado en el primer punto de la trayectoria.
(Fuente: Propia)

3.3.1.2.4. Movimiento en trayectorias (Espacio cartesiano)

En la siguiente imagen se muestra el manipulador en una posición aleatoria de las dos trayectorias especificadas.

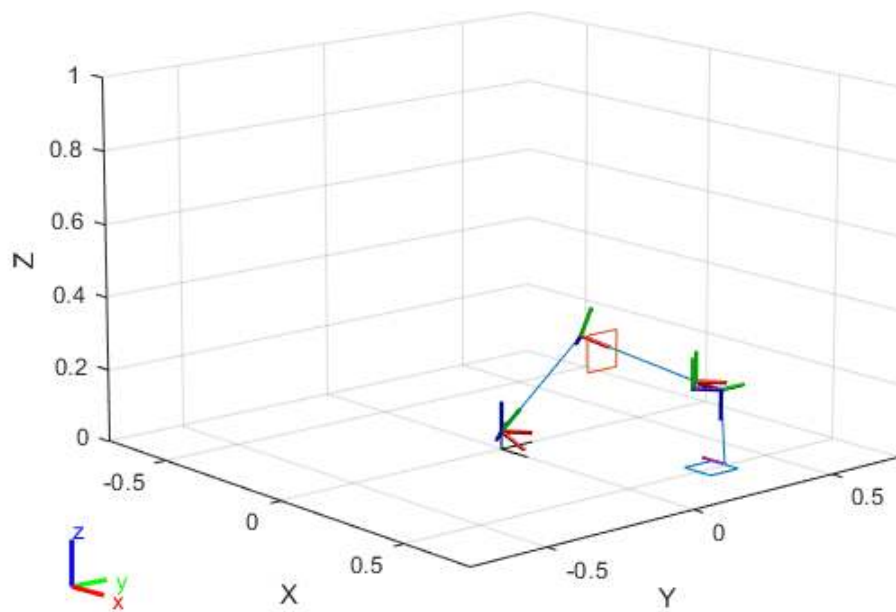


Figura 3.8. Movimiento del manipulador en la primera trayectoria cuadrada.
(Fuente: Propia)

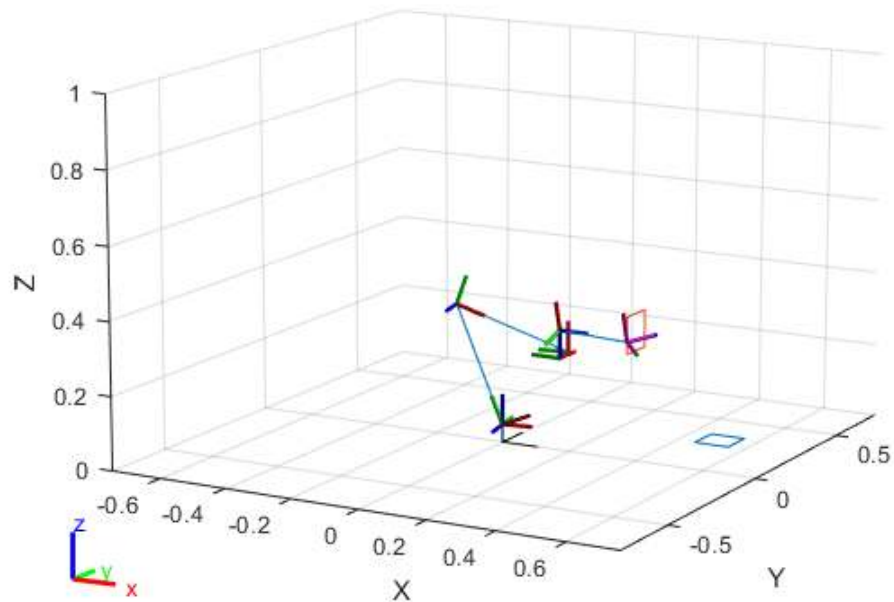


Figura 3.9. Movimiento del manipulador en la segunda trayectoria cuadrada.
(Fuente: Propia)

3.3.1.2.5. Movimiento en recorridos intermedios (Espacio de las articulaciones)

En la imagen se puede ver el movimiento del manipulador entre las dos trayectorias.

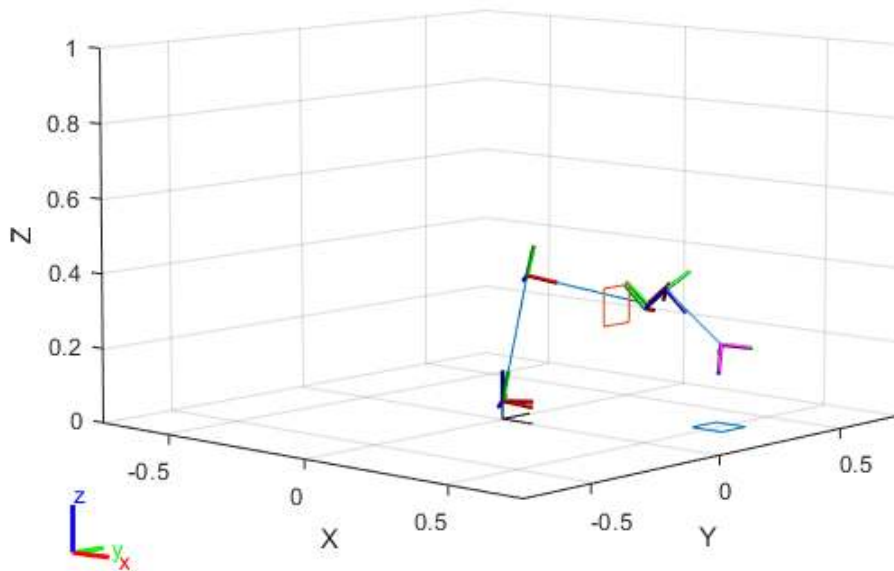


Figura 3.10. Movimiento del manipulador entre trayectorias especificadas.
(Fuente: Propia)

3.3.1.2.6. Posicionamiento final (Espacio de las articulaciones)

El último paso es regresar a la configuración inicial del robot, como se muestra a continuación.

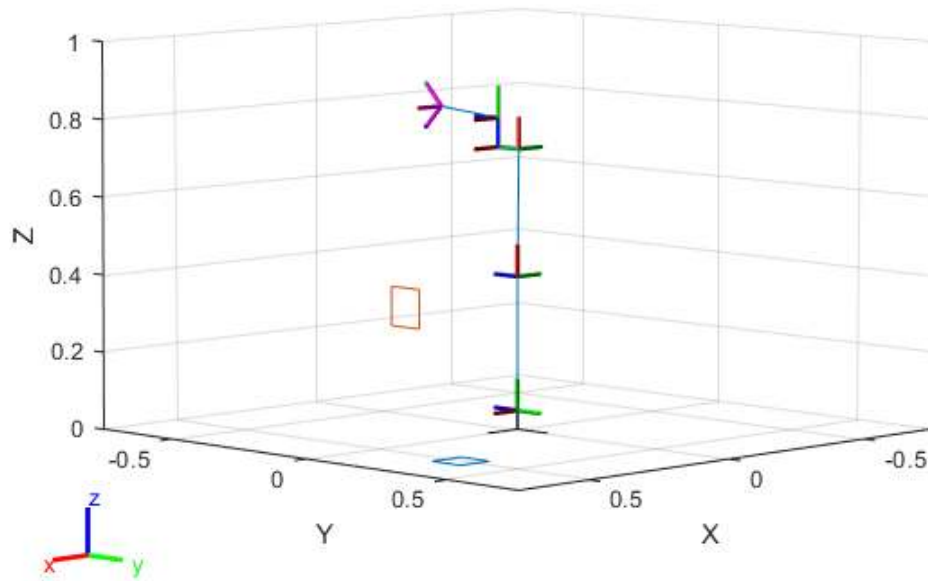


Figura 3.11. Configuración del robot en su posición inicial.
(Fuente: Propia)

3.3.1.2.7. Post procesamiento

A continuación, se observan algunos resultados en cuanto a la evolución temporal de las variables articulares y cartesianas del manipulador. Igualmente se observa los torques requeridos para conseguir el movimiento.

Evolución del movimiento del manipulador en el espacio articulado

En la siguiente imagen se observa como varia el comportamiento de los actuadores acoplados en las articulaciones del manipulador robótico. En la gráfica Ángulo Vs Tiempo se observa un comportamiento que no cambia de manera repentina en todo el tramo de movimiento, la variación del ángulo de la articulación q_6 que mueve la antorcha de soldadura presenta el mayor movimiento, consistente con el giro continuo del efector final alrededor de las geometrías de recorrido. La grafica de velocidad angular Vs Tiempo tiene perfiles trapezoidales tal como se deseaba, donde cada forma trapezoidal corresponde a cada tramo de recorrido. Del mismo modo, La gráfica aceleración angular Vs Tiempo tiene forma de rectángulos correspondientes a cada tramo de aceleración y desaceleración en los perfiles trapezoidales.

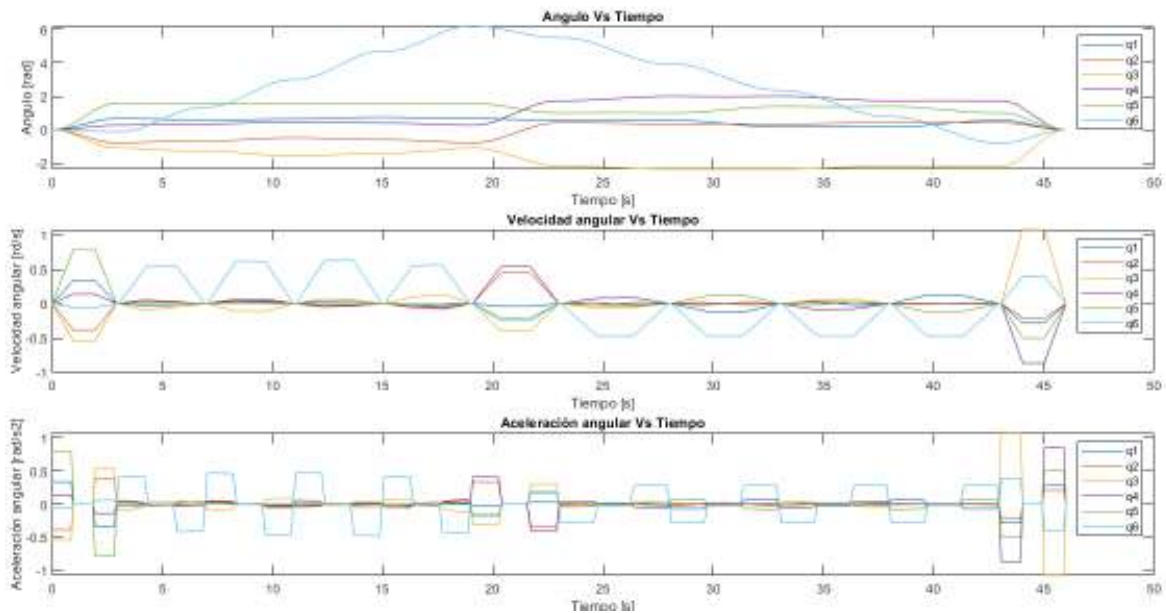


Figura 3.12. Movimiento del manipulador en el espacio articulado.
(Fuente: Propia)

Evolución del movimiento del manipulador en el espacio cartesiano

La siguiente figura muestra como varia el movimiento del extremo de la antorcha de soldadura acoplada al manipulador robótico. En la gráfica Posición Vs Tiempo se puede ver el movimiento del efector final en coordenadas cartesianas, aquí se observa un movimiento sin cambios brusco y los segmentos de línea horizontales permiten identificar el plano en el que se está moviendo la antorcha, por ejemplo, para el primer trayecto rectangular, se tiene que la posición en Z se mantiene constante en 0 m durante el primer tramo de movimiento, que indica que el manipulador solo se traslada en el plano XY. En la gráfica Velocidad Vs Tiempo se observan los perfiles trapezoidales en los segmentos pertenecientes al recorrido por las geometrías rectangulares, donde se requiere un movimiento controlado y estable. Del mismo modo, en la gráfica aceleración Vs Tiempo se tiene los segmentos de aceleración rectangulares en los movimientos por las trayectorias rectangulares.

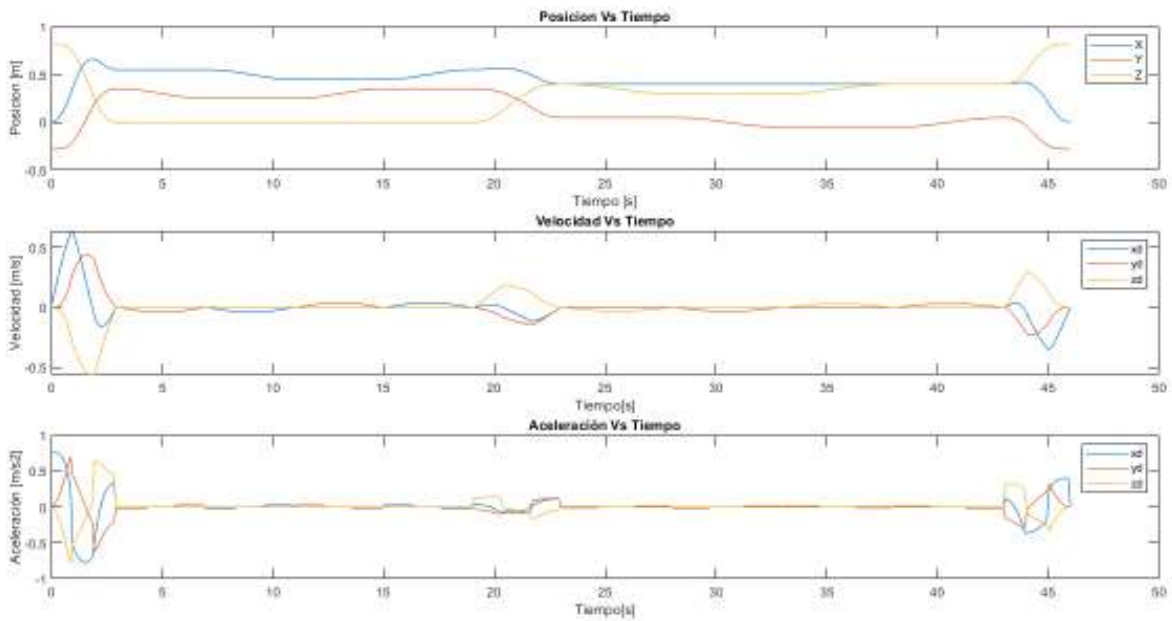


Figura 3.13. Movimiento del manipulador en el espacio cartesiano.
(Fuente: Propia)

Torques requeridos por los actuadores

En la gráfica Torque Vs Tiempo se observa que el actuador dos es el que tiene el mayor requerimiento de todos, especialmente para el recorrido en la primera trayectoria rectangular. Sin embargo, el torque máximo requerido para este movimiento (7 Nm) es menor al obtenido en el análisis estático (10 Nm). Esto se debe a que, en aplicaciones prácticas el manipulador no requiere colocarse en posición totalmente horizontal. El actuador tres es el segundo con mayores requerimientos de torque, en tanto que los demás actuadores tiene requerimientos de torque mínimos.

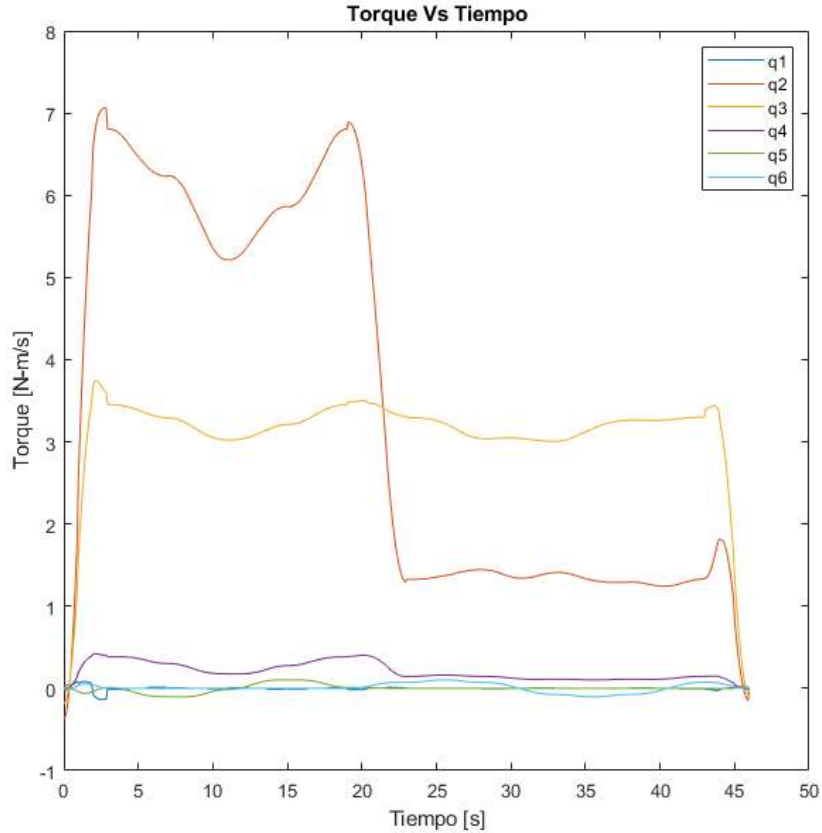


Figura 3.14. Torques requeridos para el movimiento en la trayectoria definida.
(Fuente: Propia)

Recorrido realizado

El recorrido de la antorcha de soldadura sigue los lineamientos requeridos en el generador de trayectorias. Se distinguen los pasos establecidos en el programa. Los movimientos en el espacio de las articulaciones siguen trayectorias curvas, mientras que los movimientos en el espacio cartesiano siguen trayectorias rectas. Se observa que el recorrido realizado cumple con el especificado, por lo que se puede considerar que el programa generador de trayectorias se desempeña de manera idónea.

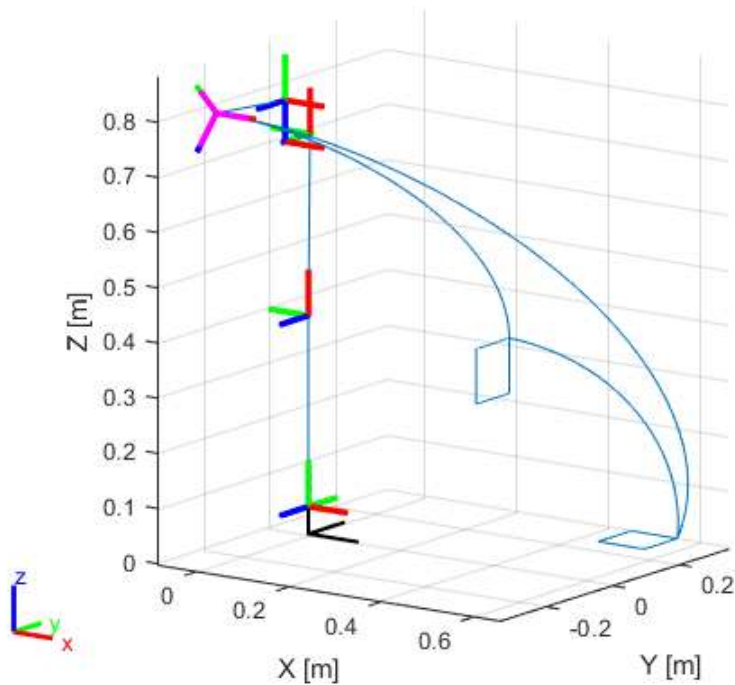


Figura 3.15. Recorrido realizado por el extremo del manipulador.
(Fuente: Propia)

3.3.2. Simulación del manipulador usando Simulink

3.3.2.1. Simulación de seguimiento de trayectorias

Una vez obtenidos los resultados del programa generador de trayectorias, es posible realizar una simulación con el objetivo de observar si los parámetros de movimiento generados son replicables en un entorno virtual.

3.3.2.1.1. Representación del manipulador

Por medio de Simulink y Simscape Multibody es posible representar el entorno virtual en el que se observará el comportamiento del robot al realizar las tareas especificadas. Este entorno posee el manipulador y las dos geometrías requeridas ubicadas en su posición de soldadura.

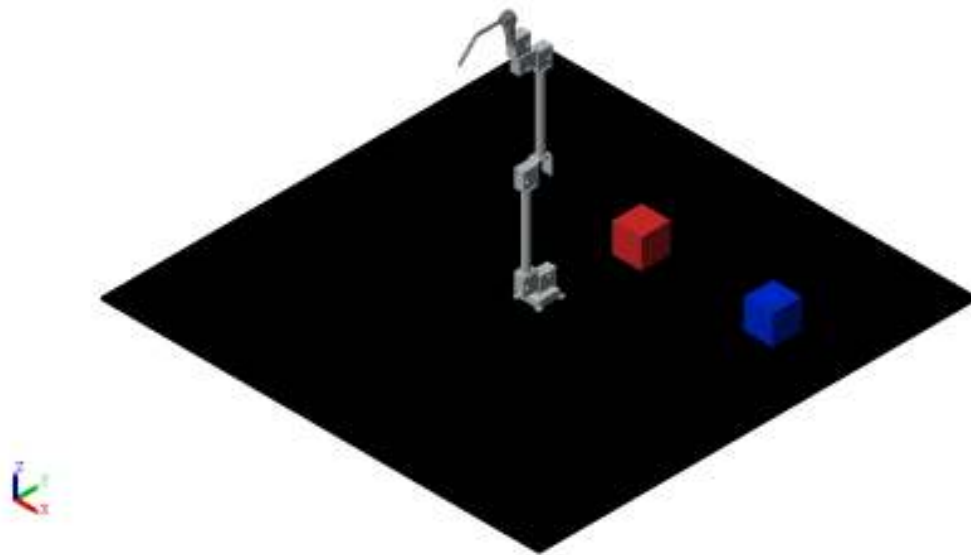


Figura 3.16. Vista del brazo manipulador generado con Simscape con las trayectorias cuadradas.
(Fuente: Propia)

3.3.2.1.2. Movimiento del brazo robótico

Se observa que el movimiento sigue las formas geométricas con la antorcha de soldadura apuntando siempre hacia el centro de estas. A continuación, se puede ver imágenes del proceso.

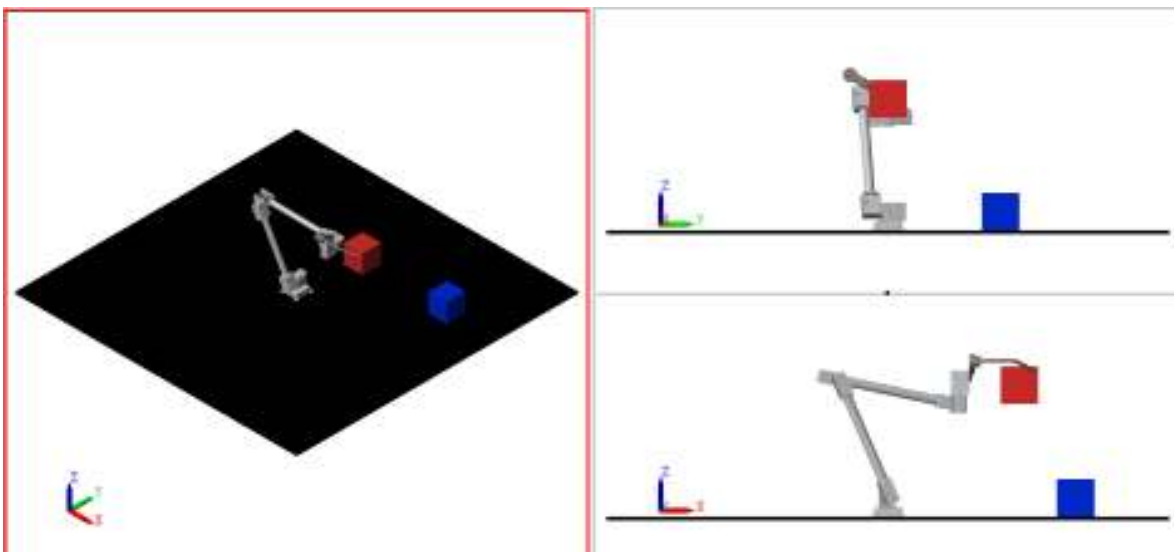


Figura 3.17. Funcionamiento del manipulador robótico.
(Fuente: Propia)

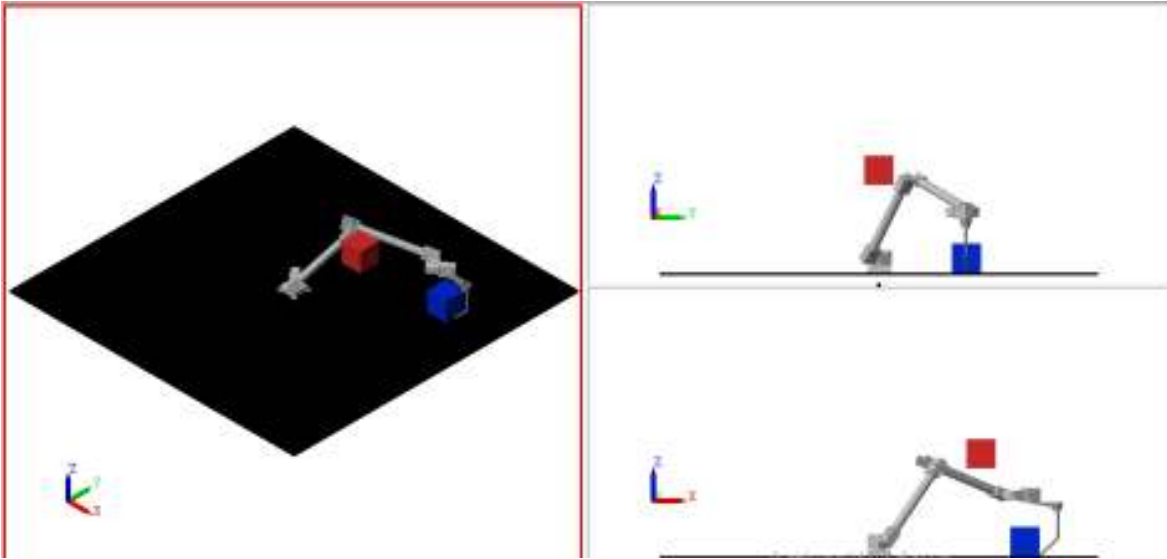


Figura 3.18. Funcionamiento del manipulador robótico.
(Fuente: Propia)

3.3.2.1.3. Datos obtenidos

Usando los sensores de las articulaciones y los cuerpos se pueden medir los mismos parámetros calculados usando el Toolbox Robotics System ya partir de esto realizar una comparación de resultados.

Evolución del movimiento en el espacio articulado

Se observa en la siguiente gráfica que la variación de los ángulos, velocidades y aceleraciones de las articulaciones medidas con los sensores de los motores son idénticas a las obtenidas en el programa generador de trayectorias (Figura 3.11).

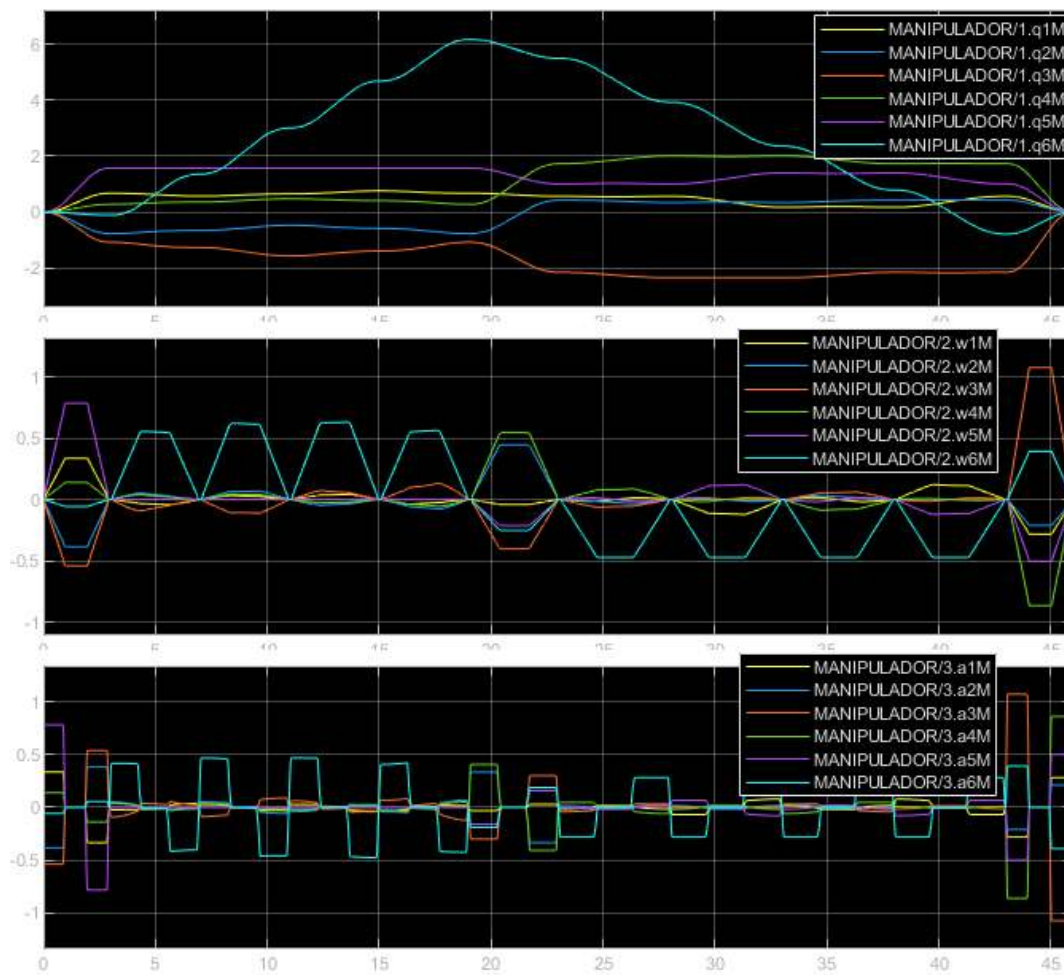


Figura 3.19. Movimiento del manipulador en el espacio articulado producido en Simulink.
(Fuente: Propia)

Evolución del movimiento en el espacio cartesiano

Igualmente, los sensores de movimiento ubicados en el extremo de la antorcha de soldadura crean los mismos perfiles de movimiento para las posiciones, velocidades y aceleraciones que los obtenidos en el generador de trayectorias (Figura 3.12).

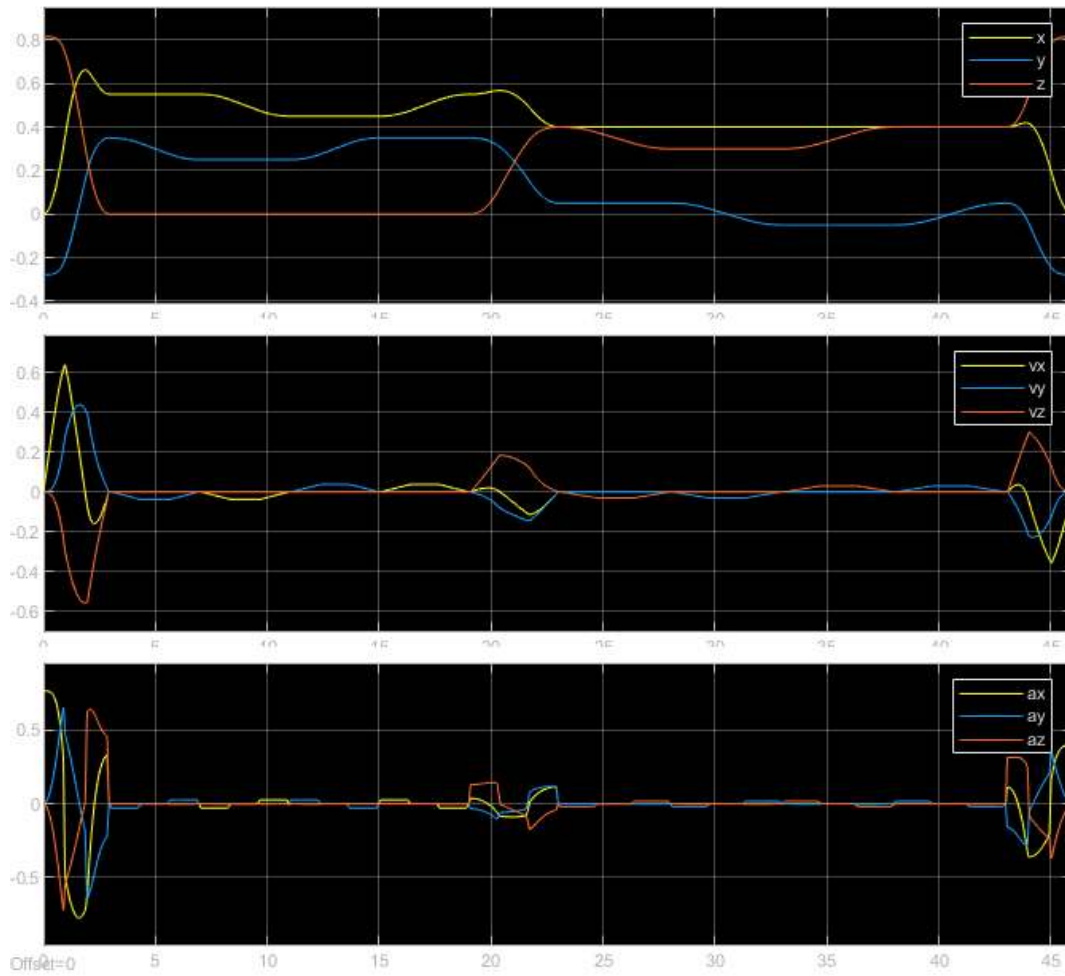


Figura 3.20. Movimiento del manipulador en el espacio cartesiano producido en Simulink.
(Fuente: Propia)

Torques requeridos por los actuadores

Los sensores de torque realizan la misma medición que la obtenida en el programa (Figura 3.13), como se puede ver en la siguiente imagen.

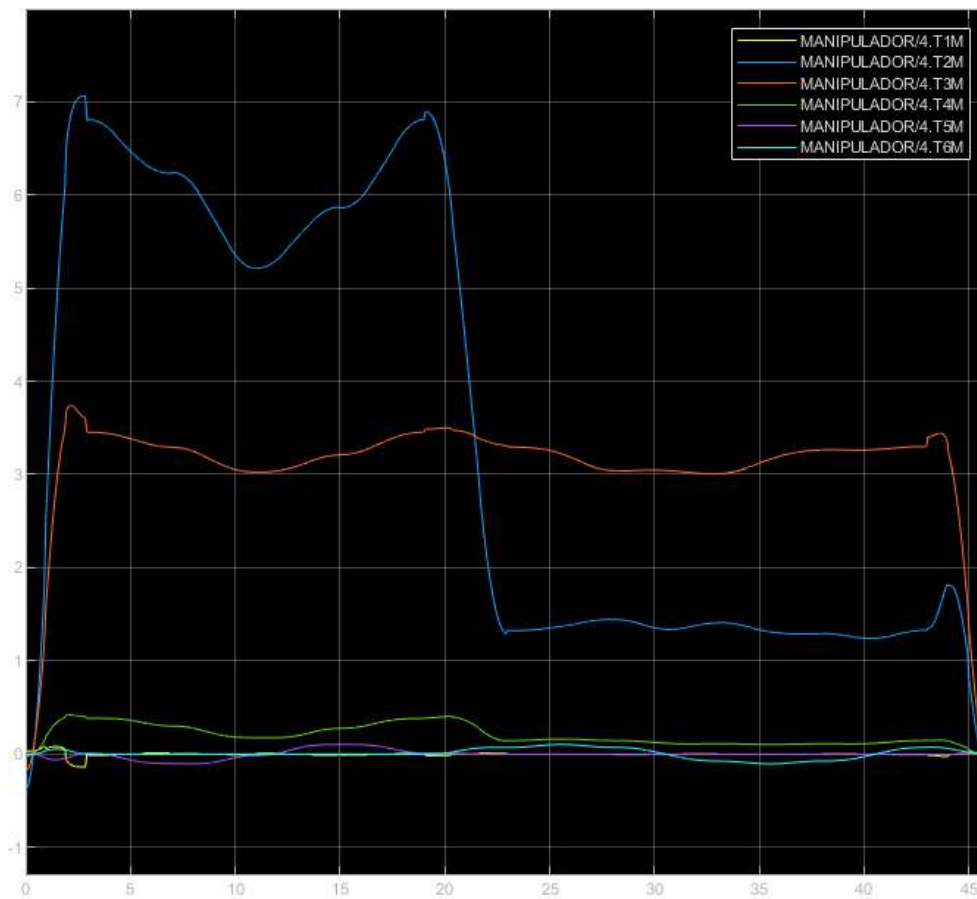


Figura 3.21. Torques requeridos para el movimiento producido en Simulink.
(Fuente: Propia)

Recorrido realizado por el manipulador

En la imagen siguiente se puede observar que el recorrido obtenido con la simulación en Simulink posee el mismo comportamiento que la observada en el generador de trayectorias (Figura 3.14).

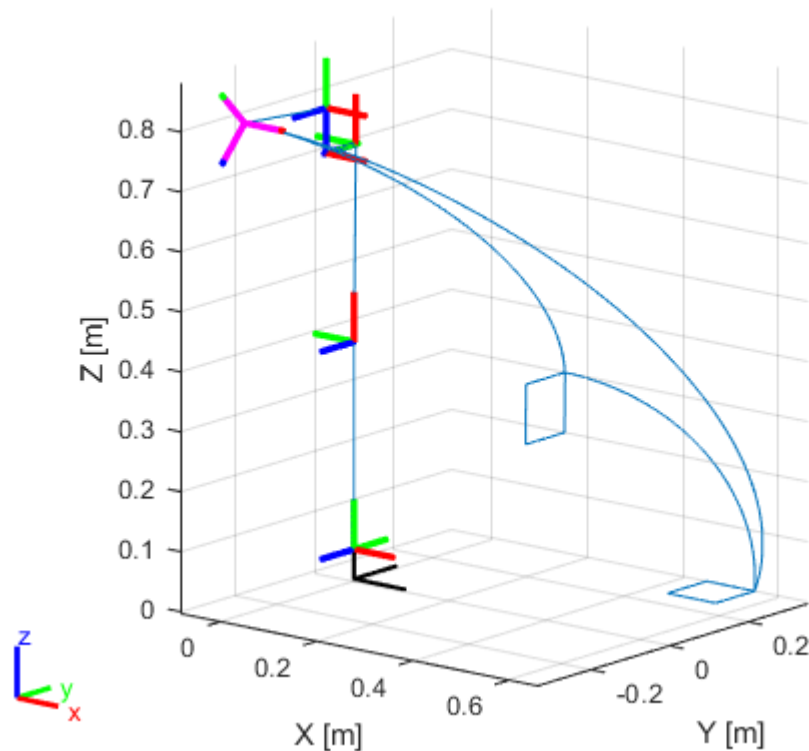


Figura 3.22. Recorrido realizado por el manipulador.
(Fuente: Propia)

3.3.3. Comparación del movimiento realizado

Por último, se representan los recorridos obtenidos tanto con el programa generador de trayectorias como con la simulación en un mismo para compararlos y se obtiene que los perfiles de movimiento son los mismos.

Teniendo en cuenta los gráficos de la evolución temporal del movimiento del robot en los espacios rectangular y articulado y el recorrido obtenido con el generador de trayectorias y en Simulink, se puede decir que el programa generador de trayectorias produce movimientos útiles para ser aplicados en soldadura GMAW.

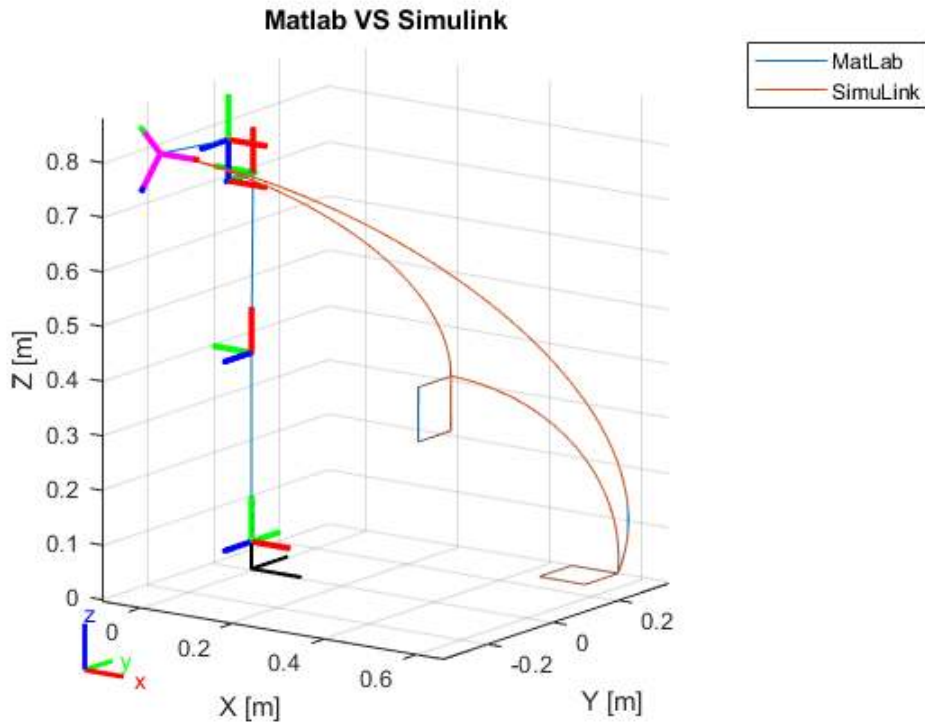


Figura 3.23. Recorrido del manipulador generado en Matlab (azul) y obtenido en Simulink (rojo).
(Fuente: Propia)

3.4. Costos

En la siguiente tabla se detallan los costos de los distintos componentes del manipulador robótico. Cabe recalcar que los valores de componentes electrónicos se obtienen de la página web del fabricante y no se toman en cuenta los costos de transporte ni tampoco los impuestos a los que son susceptibles al ser importados al país.

Tabla 3.2. Costo de los componentes del brazo manipulador

Denominación	Cantidad	Coste unitario	Coste total
Servomotor XM540-W270-R	6	\$359.99	\$2,159.94
Controlador OpenCR1.0	1	\$179.90	\$179.90
Tarjeta de expansión OpenCM 485	1	\$29.90	\$29.90
Tubo de aluminio 32x2 mm x 6 m	2	\$10.00	\$20.00
Pletina de aluminio 37.7x2.4 mm	7	\$6.00	\$42.00
Tornillo cabeza plana de estrella M2.5x0.4x4	48	\$0.05	\$2.40
Perno cilíndrico de hueco hexagonal M2x0.4x4	80	\$0.10	\$8.00
Tuerca hexagonal M2x0.4	24	\$0.10	\$2.40
			\$2,444.54

(Fuente: Propia)

Al comparar este costo referencial con el de los robots ya existentes tomados en cuenta para este trabajo (ANEXO III), se obtiene que el costo es mucho menor para este manipulador diseñado.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

El manipulador robótico diseñado cumple con los requerimientos especificados para ser usado en soldadura para carpintería mecánica.

Se definieron y establecieron los requerimientos de funcionamiento del brazo robótico. Este debe ser capaz de manipular una antorcha de soldadura de tal manera que siga trayectorias rectangulares de soldadura con una orientación que va cambiando en función de la ubicación en la que se encuentra y con una velocidad de avance estable durante su movimiento.

Usando el procedimiento de diseño usando ingeniería concurrente se consiguió avanzar hasta la etapa de diseño de materialización. Entre las herramientas de ingeniería concurrente usadas están la especificación del producto, el análisis funcional, definición de módulos, uso de herramientas computaciones para diseño, entre otros.

Se estableció la geometría del manipulador, así como los materiales requeridos y conexiones entre los distintos componentes del producto diseñado. Todo esto documentado en un archivo CAD en Solidworks, planos de conjunto y taller en los que se indican además los elementos normalizados requeridos para la fabricación y ensamble del manipulador.

Usando teoría de estática, dinámica, resistencia de materiales y análisis por elementos finitos se consiguió dimensionar los distintos elementos mecánicos del manipulador robótico para que respondan de manera fiable a los requerimientos de fuerzas y momentos a los que será expuesto el producto.

Con Simulink y Simscape Multibody fue posible convertir el modelo CAD de Solidworks a un modelo mecánico de diagrama de bloques y crear un entorno de simulación para el manipulador robótico que posteriormente se pudo importar a Matlab y utilizar todos los Toolboxes disponibles en este programa.

Usando el Toolbox Robotics System de Matlab fue posible adaptar el modelo CAD para convertirlo en un modelo robótico y a partir de ahí realizar el análisis cinemático y dinámico del robot. Así como crear un programa generador de trayectorias requerido para el control cinemático del manipulador.

Además de los elementos mecánicos constituyentes del brazo robótico, se definieron los componentes electrónicos y, junto con el programa generador de trayectorias, conforman el sistema mecatrónico del producto final.

4.2. RECOMENDACIONES

Tomar en cuenta la posición y orientación del manipulador al momento de elegir las trayectorias de movimiento. Debido a la forma de la antorcha de soldadura, no todas las posiciones de soldadura son alcanzables por el brazo robótico de manera que se consiga un seguimiento exacto del cordón de soldadura deseado.

Se recomienda realizar a futuro un programa de pruebas para comprobar el correcto funcionamiento del manipulador una vez que se ensamble y se comience a usar.

Como continuación a este estudio, se podría añadir a este diseño nuevos módulos que permitan su uso en aplicaciones diferentes para la cual fue diseñado este manipulador, por ejemplo, pintura, manipulación de objetos, inspección, entre otros.

Es importante a futuro crear una aplicación para poder especificar los puntos y orientaciones del manipulador en las trayectorias de soldadura de una manera más amigable e interactiva que usando líneas de código de programación.

Al ser los servomotores el factor limitante en el peso máximo de la carga útil que soporta este manipulador, se recomienda realizar una investigación posterior de servomotores existentes en el mercado con mejores características que los elegidos para este proyecto.

Se recomienda continuar con investigaciones futuras acerca de temas de diseño de manipuladores robóticos en la facultad. En especial se debería profundizar en temas relacionados al control dinámico, uso de softwares de simulación y la implementación de este tipo de robots en diferentes aplicaciones industriales requeridas en el país.

Visitar los sitios web y usar el menú de ayuda de los distintos softwares de diseño es de mucha utilidad para comprender el funcionamiento de los programas de diseño y también para aprender a usarlos de manera eficiente.

5. BIBLIOGRAFÍA

- [1] S. Astudillo y A. Briozzo, «Innovación en las Mipymes Manufactureras de Ecuador y Argentina,» *Semestre Económico*, vol. 19, nº 40, pp. 117-144, 2016.
- [2] Exel Air Gas & Welding Products, «La importancia de la soldadura en la industria metalúrgica,» 2020. [En línea]. Available: <https://exelair.com.mx/blog/la-importancia-de-la-soldadura-en-la-industria-metal%C3%BArgica>. [Último acceso: 26 Agosto 2020].
- [3] J. Craig, *Introduction to Robotics: Mechanics and Control*, Third ed., Harlow: Pearson Education Limited, 2014.
- [4] A. Barrientos, L. Peñin, C. Balaguer y R. Aracil, *Fundamentos de robótica*, Segunda ed., Madrid: McGraw-Hill/Interamericana de España, S.A.U., 2007.
- [5] K. Hausser, «Inverse Kinematics: Duke University,» Duke University, 3 October 2018. [En línea]. Available: <http://motion.pratt.duke.edu/RoboticSystems/InverseKinematics.html#Chapter-6.-Inverse-Kinematics>. [Último acceso: 13 Octubre 2020].
- [6] P. Corke, *Robotics, Vision and Control. Fundamental Algorithms in MATLAB*, Second ed., Cham: Springer Tracts in Advanced Robotics, 2017.
- [7] S. Castro, «Trajectory Planning for Robot Manipulators: Mathworks,» The Mathworks Inc, 6 November 2019. [En línea]. Available: <https://blogs.mathworks.com/racing-lounge/2019/11/06/robot-manipulator-trajectory/>. [Último acceso: 20 Octubre 2020].
- [8] R. Kelly, V. Santibañez y A. Loría, *Control of Robot Manipulators in Joint Space*, London: Springer-Verlag, 2005.
- [9] W. Bolton, *Mechatronics, Electronics Control Systems in Mechanical and Electrical Engineering*, Sixth ed., Harlow: Pearson Education Limited, 2015.
- [10] R. Dorf and R. Bishop, *Modern Control Systems*, Twelfth ed., New Jersey: Prentice Hall, 2011.
- [11] C. Riba, *Diseño concurrente*, Barcelona: Edicions UPC, 2002.
- [12] TM Robot, «Techman Robot TM5-900,» 2020. [En línea]. Available: <https://www.tm-robot.com/en/regular-payload/>. [Último acceso: 16 Octubre 2020].

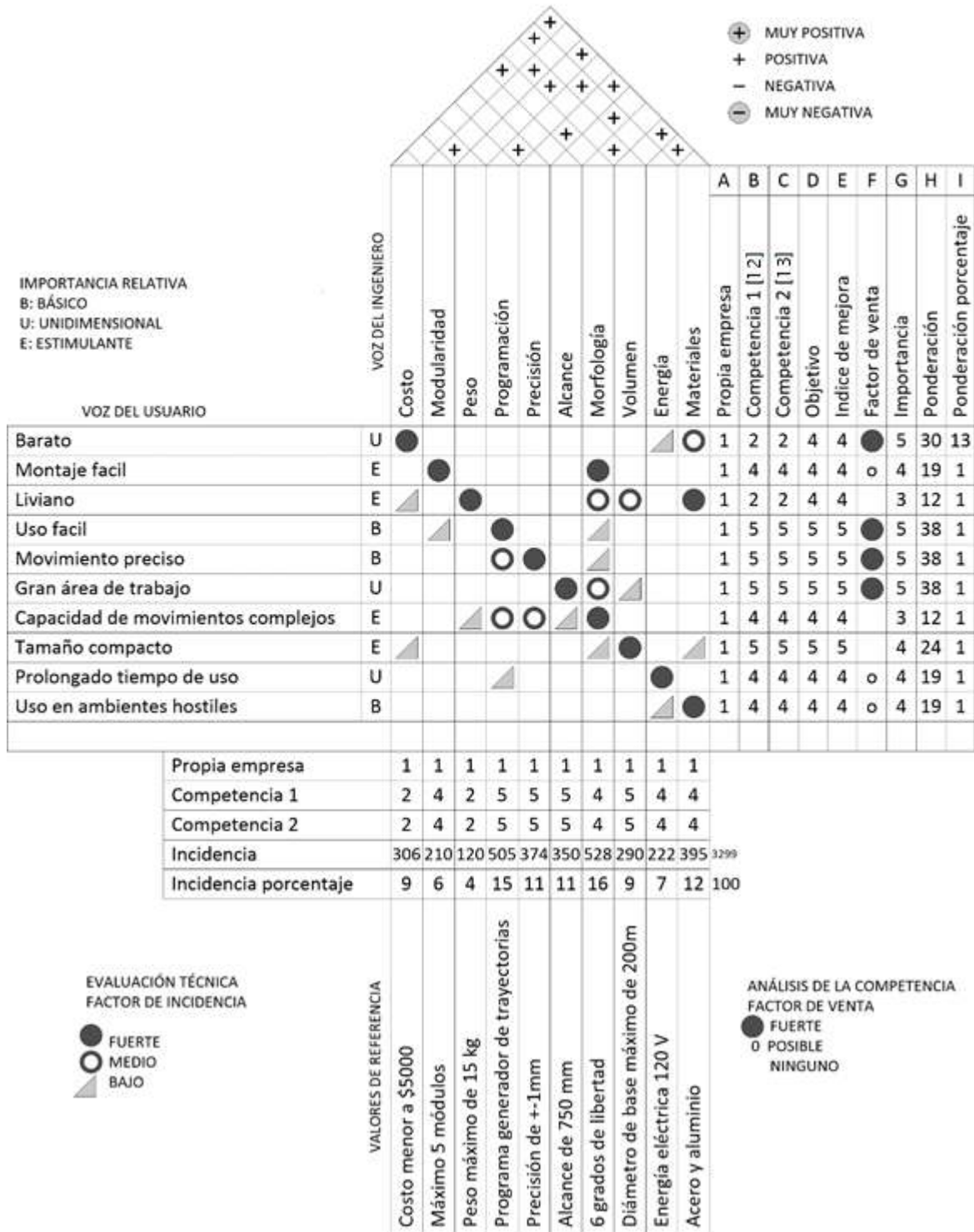
- [13] Universal Robots, «Universal Robot UR5e,» 2020. [En línea]. Available: <https://www.universal-robots.com/es/productos/robot-ur5/>. [Último acceso: 16 Octubre 2020].
- [14] A. Owen-Hill, «Python vs C++ vs C# vs MATLAB: Which Robot Language is Best?,» RoboDk, 31 Julio 2018. [En línea]. Available: <https://robodk.com/blog/robot-programming-language/>.
- [15] F. Petruzella, Electric Motors and Control Systems, Second ed., New York: McGraw-Hill Education, 2016.
- [16] M. Rodriguez, «Proceso de Soldadura - SMAW: Centro de conocimiento Esab,» [En línea]. Available: <https://www.esab.com.ar/ar/sp/education/blog/proceso-soldadura-smaw.cfm>. [Último acceso: 12 Noviembre 2020].
- [17] Esab, «Proceso de Soldadura - GMAW: Centro de conocimiento Esab,» [En línea]. Available: https://www.esab.com.ar/ar/sp/education/blog/proceso_soldadura_gmaw.cfm. [Último acceso: 12 Noviembre 2020].
- [18] R. Hibbeler, Mecánica de materiales, Octava ed., México: Pearson Educación, 2011.
- [19] Servicio Ecuatoriano de Normalización, *NTE INEN 2 250:2000 - Aluminio. Perfiles, barras, varillas y tubos extruidos. Requisitos e inspección*, Quito: INEN, 2000.
- [20] E. Madenci and I. Guven, The Finite Element Method and Applications in Engineering Using Ansys, Second ed., New York: Springer International Publishing, 2015.
- [21] R. Budynas y K. Nisbett, Diseño en ingeniería mecánica de Shigley, Novena ed., México, D.F.: McGraw-Hill/Interamericana de editores, S.A. de C.V., 2012.
- [22] M. Adoniadis, «The Mathematics of Simple Beam Deflection,» The Royal Academy of Engineering, [En línea]. Available: <https://www.raeng.org.uk/publications/other/15-beam-deflection#:~:text=Typically%2C%20the%20maximum%20deflection%20is,as%2020mm%20without%20adverse%20effect..> [Último acceso: 23 Octubre 2020].
- [23] The Mathworks Inc, «Simscape Multibody,» 2020. [En línea]. Available: <https://www.mathworks.com/products/simmechanics.html>. [Último acceso: 15 Noviembre 2020].

- [24] The Mathworks Inc, «Robotics System Toolbox,» 2020. [En línea]. Available: https://www.mathworks.com/help/robotics/index.html?s_tid=CRUX_lftnav. [Último acceso: 17 Noviembre 2020].
- [25] Robotis, «Dynamixel: Robotis,» 2020. [En línea]. Available: <https://www.robotis.us/dynamixel/>. [Último acceso: 24 Noviembre 2020].
- [26] Robotis, «XM540-W270-R: Robotis,» 2020. [En línea]. Available: http://en.robotis.com/shop_en/item.php?it_id=902-0133-000. [Último acceso: 24 Noviembre 2020].
- [27] Robotis, «OpenCr 1.0: Robotis,» 2020. [En línea]. Available: <https://www.robotis.us/opencr1-0/>. [Último acceso: 24 Noviembre 2020].
- [28] Robotis, «OpenCM 485 Expansion Board: Robotis,» 2020. [En línea]. Available: <https://www.robotis.us/opencm-485-expansion-board/>. [Último acceso: 24 Noviembre 2020].
- [29] Robotis, «XM540-W270-T/R e-Manual: Robotis,» 2020. [En línea]. Available: <https://emanual.robotis.com/docs/en/dxl/x/xm540-w270/>. [Último acceso: 24 Noviembre 2020].

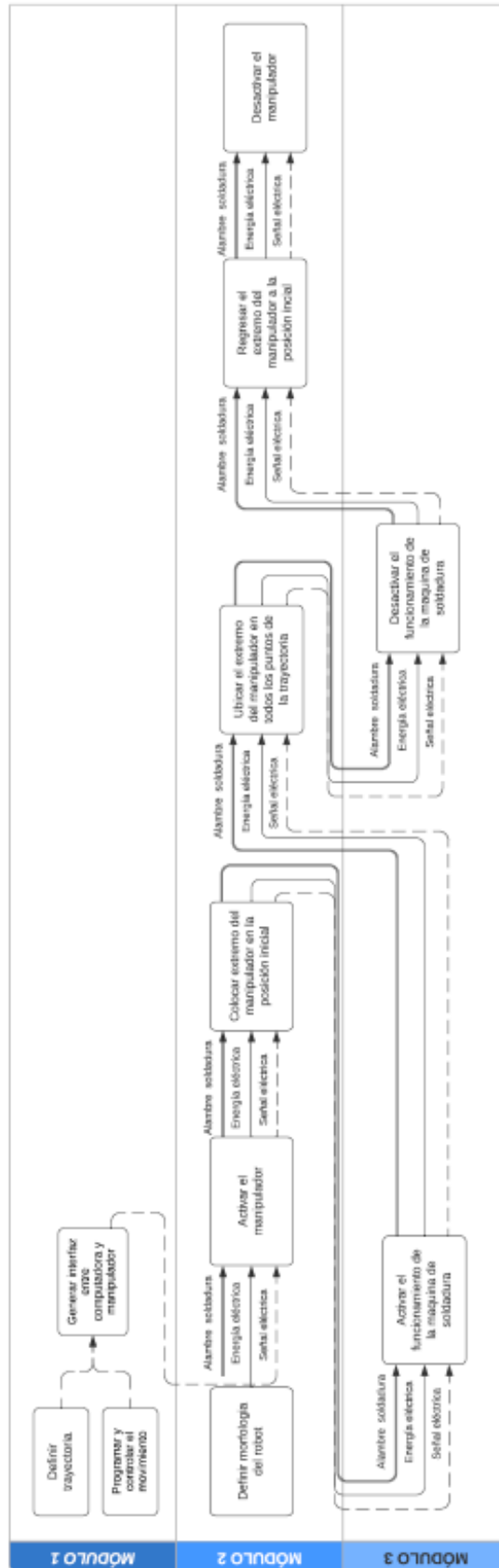
ANEXOS

ANEXO I. CASA DE LA CALIDAD

CASA DE LA CALIDAD



ANEXO II. DIAGRAMA FUNCIONAL, NIVEL 2



ANEXO III. ESPECIFICACIONES TÉCNICAS DE LOS MANIPULADORES UR5 Y TM5-900

Característica	UR5	TM5-900
Peso	18.4 kg	22.6 kg
Carga útil	5 kg	4 kg
Alcance	850 mm	900 mm
Rango de acción de las articulaciones	360°	q1, q6 = 270° q2, q4, q5 = 180° q3 = 155°
Grados de libertad	6	6
Velocidad máxima de giro de las articulaciones	30 RPM	q1, q2, q3 = 30 RPM q4, q5, q6 = 37.5 RPM
Área de base	149 mm	149 mm
Programación	Desarrollada por el fabricante	Desarrollada por el fabricante
Fuente de alimentación	24 V 2 A / 12 V 600 mA	24 V 2 A / 12 V 600 mA
Fuente de potencia	120-240 V 50-60 Hz	120-240 V 50-60 Hz
Materiales	Aluminio, plástico	-
Temperatura de funcionamiento	0°, 50°	0°, 55°
Costo referencial	\$35,000.00	\$37,500.00

ANEXO IV. CODIGO DEL PROGRAMA GENERADOR DE TRAYECTORIAS

```
clear
close all
clc
format shortg

%% Inicialización del programa
%Valores para inicializar el modelo en SIMULINK

load full.mat
load tT.mat
load full_vel.mat
load full_acc.mat

%% Importar robot y condiciones iniciales

importarRobot;

%% Definir trayectorias y orientaciones
% Para generar rectangulos se especifican las longitudes de los lados y
la
% ubicación de los centros de estos

% Longitud de los lados de las geometrías
geoms =[0.1 0.1 0.1;
        0.1 0.1 0.1];

% Ubicación de los centros de las geometrías
cents = [0.5 0.3 0;
        0.4 0 0.35];

% Numero de lados continuos en las trayectorias rectangulares
numlad = 4;

% Orientaciones del efector final del manipulador en los vertices de las
% geometrías de las trayectorias
% Planos de trayectorias
plan = ["XY";
        "YZ"];

% Direccion de movimiento
dir = ["AH";
        "SH"];

% Tipo de orientaciones
orien = ["OUT";
        "OUT"];

%% Tiempos de movimiento en trayectorias

% Tiempos de movimiento al punto inicial de la trayectoria y del punto
final
% de las trayectorias a la ubicación inicial (ESPACIO ARTICULADO)
tmov = [3 3];

% Tiempo de recorrido en cada lado de las geometrías definidas
```

```

% (ESPACIO CARTESIANO)
ttraj = [4 5];

% Tiempos de colocación del manipulador entre trayectorias cuadradas
% (ESPACIO ARTICULADO)
tcol = [4];

% Tiempo de muestreo para los cálculos
deltat = 0.1;

%% Generador de trayectorias
Robotics_fun(robot,tmov,ttraj,tcol,deltat,geoms,cents,numlad,plan,dir,orien)

function
Robotics_fun(robot,tmov,ttraj,tcol,deltat,geoms,cents,numlad,plan,dir,orien)
% importarRobot;
% Area de trabajo
homeConfig = homeConfiguration(robot);
eeName = 'Body7';
numJoints = robot.NumBodies-1;

workCent = tform2trvec(getTransform(robot,homeConfig,'Body2','Base'));
workRad = tform2trvec(getTransform(robot,homeConfig,'Body6','Body2'));
Wr = workRad(2);
[wx,wy,wz] = sphere;
wx = wx*Wr;
wy = wy*Wr;
wz = wz*Wr;

%% Crear Trayectorias
orientacioneplanos
waypoints = zeros(numlad+1,3,numel(geoms(:,1)));
numWaypoints = zeros(1,numel(geoms(:,1)));
Path = zeros(4,4,numel(geoms(:,1)));
orientations = zeros(numlad+1,3,numel(geoms(:,1)));

for i=1: numel(geoms(:,1))
    [waypoints(:, :, i), numWaypoints(i), Path(:, :, i), orientations(:, :, i)] =
    createWaypoints(numlad,geoms(i, :), cents(i, :), plan(i), dir(i), orien(i));
end

figure
show(robot);
hold on
axis equal
axis([-1 1 -1 1 0 1]);
surf(wx,wy+workCent(2),wz+workCent(3), 'FaceAlpha',0.15, 'LineStyle', 'none'
)
for i=1:numel(geoms(:,1))
    plot3(waypoints(:,1,i), waypoints(:,2,i), waypoints(:,3,i))
end
legend('robot','surf','traj1','traj2')

%% Posicionamiento inicial (ARTICULACIONES)

```

```

% GIK Inicial
% Calcula la cinematica inversa para llegar al primer punto

% Define IK
ikInitGuess1 = homeConfig;
height1 = [-inf, inf; ...
           -inf, inf; ...
           0, inf];
body1 = 'Body3';
maxJointChange1 = deg2rad(360);

[config1,info1] =
GIK(robot,ikInitGuess1,Path(:, :, 1),body1,maxJointChange1,height1);

if info1.Status ~= "success"
    error('Geometria no valida')
end

% Mostrar el resultado de IK
% figure
% show(robot,config1);
% hold on
% % plot3(waypoints1(:,1),waypoints1(:,2),waypoints1(:,3))
%
surf(wx,wy+workCent(2),wz+workCent(3),'FaceAlpha',0.15,'LineStyle','none'
)
% for i=1:numel(geoms(:,1))
%     plot3(waypoints(:,1,i),waypoints(:,2,i),waypoints(:,3,i))
% end
% axis([-1 1 -1 1 0 1]);

% Crea la trayectoria al primer punto de la trayectoria

t1 = tmov(1);
cof11 = homeConfig;
cof12 = config1;

[q1,q1d,q1dd,jacl,vell,tvec1]=articfun(robot,eeName,t1,deltat,cof11,cof12
);

%% Movimiento en trayectorias (CARTESIANO)
% Calcula la cinemática inversa para las trayectorias especificadas

%Valores incialización
maxJointChange = deg2rad(10);
height1 = [-inf, inf; ...
           -inf, inf; ...
           0, inf];
height2 = [-inf, inf; ...
           -inf, inf; ...
           0, inf];

heights = cat(3,height1,height2);

ikInitGuess2 = q1(:,numel(tvec1))';

waypos = zeros(numel(geoms(:,1)),3);
oriens = zeros(numel(geoms(:,1)),3);

```

```

for i=2:numel(geoms(:,1))
    waypos(i-1,:) = waypoints(1,:,i);
    oriens(i-1,:) = orientations(1,:,i);
end

% Numero de pasos de cada trayectoria
stvec2 = zeros(1,numel(geoms(:,1)));
for i=1:numel(geoms(:,1))
    stvec2(i) = numel(0:deltat:ttraj(i));
end
%Numero maximo de passo de trayectoria
aa = max(stvec2);

%Matrices de inicialización
q2 = zeros(aa,6,numlad,numel(geoms(:,1)));
vel2 = zeros(6,aa,numlad,numel(geoms(:,1)));
acc2 = zeros(6,aa,numlad,numel(geoms(:,1)));
q2d = zeros(6,aa,numlad,numel(geoms(:,1)));
stat2 = zeros(1,aa,numlad,numel(geoms(:,1)));
tvec2 = zeros(numel(geoms(:,1)),aa);
config2 = zeros(numel(geoms(:,1)),6);
ttr = zeros(1,numel(geoms(:,1)));

%Iteración para el numero de trayectorias
for i=1:numel(geoms(:,1))
    % Calcula cinemtica inversa para una trayectoria
    [q,vel,acc,jac,qd,stat,tvec] =
cartefun(robot,eeName,numJoints,maxJointChange,height1,ttraj(i),...
        deltat,waypoints(:, :, i),orientations(:, :, i),ikInitGuess2);

    q2temp = vertcat(q,zeros(aa-numel(tvec),numJoints,numWaypoints(i)-
1));
    vel2temp = horzcat(vel,zeros(numJoints,aa-
numel(tvec),numWaypoints(i)-1));
    acc2temp = horzcat(acc,zeros(numJoints,aa-
numel(tvec),numWaypoints(i)-1));
    q2dtemp = horzcat(qd,zeros(numJoints,aa-numel(tvec),numWaypoints(i)-
1));
    stat2temp = horzcat(stat,zeros(1,aa-numel(tvec),numWaypoints(i)-1));
    tvec2temp = horzcat(tvec,zeros(1,aa-numel(tvec)));

    % Angulos, velocidad lineal, aceleracion lineal, velocidad angular
    q2(:, :, :, i) = q2temp;
    vel2(:, :, :, i) = vel2temp;
    acc2(:, :, :, i) = acc2temp;
    %     jac2(:, :, :, i) = jac2temp;
    q2d(:, :, :, i) = q2dtemp;
    stat2(:, :, :, i) = stat2temp;
    tvec2(i, :) = tvec2temp;

    % Punto inicial para la siguiente trayectoria
    ikInitGuess3 = q2(stvec2(i), :, numWaypoints(i)-1);
    %Transformada del punto inicial de la siguiente trayectoria
    tgtPose3 = trvec2tform(waypos(i, :)) * eul2tform(oriens(i, :));

    maxJointChange3 = deg2rad(145);
    %Calcula la configuración inicial de la siguiente trayectoria

```

```

    [config2(i,:),info2] =
GIK(robot,ikInitGuess3,tgtPose3,"Body3",maxJointChange3,height1);
    ttr(i) = ttraj(i)*(numWaypoints(i)-1);
    ikInitGuess2 = config2(i,:);

end

%% Trayectorias intermedias (ARTICULACIONES)
% Calcula cinemática inversa entre el punto final de una trayectoria y
% el inicial de la siguiente

stvec3 = zeros(1,numel(geoms(:,1))-1);

if (numel(geoms(:,1))-1)>0
    %Puntos totales de calculo en cada movimiento
    for i=1:numel(geoms(:,1))-1
        stvec3(i) = numel(0:deltat:tcol(i));
    end
    %Numero maximo de pasos de calculo
    bb = max(stvec3);

    %Matrices de inicialización
    q3 = zeros(6,bb,numel(geoms(:,1))-1);
    q3d = zeros(6,bb,numel(geoms(:,1))-1);
    q3dd = zeros(6,bb,numel(geoms(:,1))-1);
    vel3 = zeros(6,bb,numel(geoms(:,1))-1);
    tvec3 = zeros(numel(geoms(:,1))-1,bb);

    % Calculo en cada recorrido intermedio
    for i=1:numel(geoms(:,1))-1
        cof11 = q2(stvec2(i),:,numlad,i);
        cof12 = q2(1,,:,1,i+1);

[q,qd,qdd,jac,velxyz,tvec]=articfun(robot,eeName,tcol(i),deltat,cof11,cof
12);

        q3temp = horzcat(q,zeros(numJoints,bb-numel(tvec)));
        q3dtemp = horzcat(qd,zeros(numJoints,bb-numel(tvec)));
        q3ddtemp = horzcat(qdd,zeros(numJoints,bb-numel(tvec)));
        vel3temp = horzcat(velxyz,zeros(numJoints,bb-numel(tvec)));
        tvec3temp = horzcat(tvec,zeros(1,bb-numel(tvec)));

        q3(:, :, i) = q3temp;
        q3d(:, :, i) = q3dtemp;
        q3dd(:, :, i) = q3ddtemp;
        vel3(:, :, i) = vel3temp;
        tvec3(i, :) = tvec3temp;

    end

end

%% Posicionamiento final (ARTICULACIONES)
% Recorrido del punto final de la última trayectoria al punto inicial del
% recorrido

% Tiempo
t1 = tmov(end);
%Configuraciones inicial y final

```

```

cof13 = q2(stvec2(end), :, end, end);
cof14 = homeConfig;

%Calculo
[q4,q4d,q4dd,jac4,vel4,tvec4]=articfun(robot,eeName,t1,deltat,cof13,cof14
);

%% Reordenamiento de matrices

% 1. Reorganizar matrices de trayectorias RECTANGULARES
% Matrices de incicalización
q2r = zeros(aa*numlad,numJoints);
vel2r = zeros(aa*numlad,numJoints);
acc2r = zeros(aa*numlad,numJoints);
q2dr = zeros(aa*numlad,numJoints);

for i=1:numel(geoms(:,1))
    % Permutar matriz de trayectoria
    q2per = permute(q2(:, :, :, i), [1 3 2]);
    % Reorganizar matriz de trayectoria para unir
    q2r(:, :, i) = reshape(q2per, [aa*numlad, numJoints]);

    vel2per = permute(vel2(:, :, :, i), [2 3 1]);
    vel2r(:, :, i) = reshape(vel2per, [aa*numlad, numJoints]);

    acc2per = permute(acc2(:, :, :, i), [2 3 1]);
    acc2r(:, :, i) = reshape(acc2per, [aa*numlad, numJoints]);

    q2dper = permute(q2d(:, :, :, i), [2 3 1]);
    q2dr(:, :, i) = reshape(q2dper, [aa*numlad, numJoints]);

end

% 2. Unir matrices de trayectorias RECTANGULARES con recorridos
intermedios
% ARTICULACIONES

if (numel(geoms(:,1))-1)>0
    % Matrices de inicialización
    q23 = zeros(aa*numlad+bb,6,numel(geoms(:,1))-1);
    q23d = zeros(aa*numlad+bb,6,numel(geoms(:,1))-1);
    vel23 = zeros(aa*numlad+bb,6,numel(geoms(:,1))-1);

    % Calculos
    for i=1:numel(geoms(:,1))-1
        % Unir matrices de trayectoria y posicionamiento
        % Angulos
        q23(:, :, i) = vertcat(q2r(:, :, i), q3(:, :, i)');
        %Velocidades angulares
        q23d(:, :, i) = vertcat(q2dr(:, :, i), q3d(:, :, i)');
        % Velocidades cartesiana
        vel23(:, :, i) = vertcat(vel2r(:, :, i), vel3(:, :, i)');
    end

    %Angulos

```



```

% Unir matrices con movimiento final
%Permutar matriz unida anterior
q23per = permute(q23,[1 3 2]);
%Reorganizar matriz permutada
q23r1 = reshape(q23per,[numel(q23(:,1,1))*(numel(geoms(:,1))-
1),numJoints]);
%Unir matriz para movimiento total
q232 = vertcat(q23r1,q2r(:, :,end));

%Velocidades angulares
% Unir matrices con movimiento final
%Permutar matriz unida anterior
q23dper = permute(q23d,[1 3 2]);
%Reorganizar matriz permutada
q23dr1 = reshape(q23dper,[numel(q23d(:,1,1))*(numel(geoms(:,1))-
1),numJoints]);
%Unir matriz para movimiento total
q23d2 = vertcat(q23dr1,q2dr(:, :,end));

%Velocidad cartesiana
% Unir matrices con movimiento final
%Permutar matriz unida anterior
vel23per = permute(vel23,[1 3 2]);
%Reorganizar matriz permutada
vel23r1 = reshape(vel23per,[numel(q23(:,1,1))*(numel(geoms(:,1))-
1),numJoints]);
%Unir matriz para movimiento total
vel232 = vertcat(vel23r1,vel2r(:, :,end));

% Matrices sin ceros
[row, ~,~] = find(q232(:,1));

q232nz = zeros(numel(row),6);
q23d2nz = zeros(numel(row),6);
vel232nz = zeros(numel(row),6);

end
%% Trayectoria final generada
if (numel(geoms(:,1))-1)>0

    for i=1:numel(row)
        q232nz(i,:) = q232(row(i),:);
        q23d2nz(i,:) = q23d2(row(i),:);
        vel232nz(i,:) = vel232(row(i),:);
    end
else
    q232nz = q2r;
    q23d2nz = q2dr;
    vel232nz = vel2r;
end

%Posiciones, velocidades y aceleraciones articulares
qtot = [q1';q232nz;q4'];
qdtot = [q1d';q23d2nz;q4d'];
qddtot = [diff(qdtot);zeros(1,6)]*10;

%Velocidades y aceleraciones cartesianas
veltot = [vel1';vel232nz;vel4'];

```

```

acctot = [diff(veltot);zeros(1,6)]*10;

%Tiempo total de movimiento
ttot = sum(numlad.*ttraj)+sum(tcol)+sum(tmov);
tT = ttot;

%Pasos totales de calculo
stepsTot = numel(qtot(:,1));

%% Torques y vectores posición

jointTorqu = zeros(stepsTot,6);
posc = zeros(stepsTot,3);

for i = 1:stepsTot
    %Calculo de torques en los actuadores en cada paso de movimiento
    jointTorqu(i,:) =
inverseDynamics(robot,qtot(i,:),qdtot(i,:),qddtot(i,:));
    %Ubicacion del efector en cada paso de movimiento
    posc(i,:) =
tform2rvec(getTransform(robot,qtot(i,:), 'Body7', 'Base'));
end

%% Interpolación

tti = linspace(0,tT,stepsTot);
ttm = 0:0.01:tT;
qtoti = pchip(tti,qtot',ttm);
qdtoti = pchip(tti,qdtot',ttm);
qddtoti = pchip(tti,qddtot',ttm);
posci = pchip(tti,posc',ttm);

% %% Graficas
tTotal = linspace(0,ttot,stepsTot);

% Graficas para ARTICULACIONES
%Grafica Angulos VS Tiempo
figure
subplot(3,1,1)
plot(tTotal,qtot)
title('Angulo Vs Tiempo')
ylabel('Angulo [rad]')
xlabel('Tiempo [s]')
legend('q1','q2','q3','q4','q5','q6')
%Grafica Velocidades angulares VS Tiempo
subplot(3,1,2)
plot(tTotal,qdtot)
title('Velocidad angular Vs Tiempo')
ylabel('Velocidad angular [rd/s]')
xlabel('Tiempo [s]')
legend('q1','q2','q3','q4','q5','q6')
subplot(3,1,3)
plot(tTotal,qddtot)
title('Aceleración angular Vs Tiempo')
ylabel('Aceleración angular [rad/s^2]')
xlabel('Tiempo [s]')
legend('q1','q2','q3','q4','q5','q6')

```

```

% Graficas para CARTESIANO
%Grafica Posicion VS Tiempo
figure
subplot(3,1,1)
plot(tTotal, posc)
title('Posicion Vs Tiempo')
ylabel('Posicion [m]')
xlabel('Tiempo [s]')
legend('X', 'Y', 'Z')
%Grafica Velocidad VS Tiempo
subplot(3,1,2)
plot(tTotal, veltot(:,4:6))
title('Velocidad Vs Tiempo')
ylabel('Velocidad [m/s]')
xlabel('Tiempo[s]')
legend('xd', 'yd', 'zd')
subplot(3,1,3)
plot(tTotal, acctot(:,4:6))
title('Aceleración Vs Tiempo')
ylabel('Aceleración [m/s2]')
xlabel('Tiempo[s]')
legend('xd', 'yd', 'zd')

%%Grafica Torques VS Tiempo
figure
plot(tTotal, jointTorqu)
title('Torque Vs Tiempo')
ylabel('Torque [N-m/s]')
xlabel('Tiempo [s]')
legend('q1', 'q2', 'q3', 'q4', 'q5', 'q6')

%% Matrices para simulink
full_motion = [ttm;qtoti]';
full_vel = [ttm;qdtoti]';
full_acc = [ttm;qddtoti]';

%% Simulación de Simulink
save tT.mat tT
save full_mot.mat full_motion
save full_vel.mat full_vel
save full_acc.mat full_acc

% Corre simulacion en Simulink
outs =
sim('ver4_2_1.slx', 'SaveOutput', 'on', 'OutputSaveName', 'yout', 'SaveFormat',
, 'Array');

% Comparar resultados de Matlab Simulink
figure
plot3(posci(1,:), posci(2,:), posci(3,:))
hold on
plot3(outs.Cartesian(:,1), outs.Cartesian(:,2), outs.Cartesian(:,3))
show(robot);
axis equal
grid on
title('Matlab VS Simulink')
legend('MatLab', 'SimuLink')
xlabel('X [m]')
ylabel('Y [m]')

```

```
zlabel('Z [m]')
```

```
end
```

ANEXO V. ESPECIFICACIONES TÉCNICAS DE MATERIALES ELECTRICOS

Especificaciones del controlador OpenCR 1.0

Microcontrolador	STM32F746ZGT6 / 32-bit ARM Cortex®-M7 con FPU (216MHz, 462DMIPS)
Sensores	Giroscopio 3Axis, Acelerómetro 3Axis, Magnetómetro 3 Ejes (MPU9250)
Programación	ARM Cortex 10pin JTAG/SWD conector USB Actualización del firmware del dispositivo (DFU) Serial
Pines de extensión	32 pines (L 14, R 18) * conectividad con Arduino Módulo de sensor x 4 pines Conector de extensiones x 18 pines
Circuitos de comunicación	USB (Micro-B USB conector/USB 2.0/Host/Peripheral/OTG) TTL (B3B-EH-A / Dynamixel) RS485 (B4B-EH-A / Dynamixel) UART x 2 (20010WS-04) CAN (20010WS-04)
LEDs y botones	LD2 (rojo/verde): Comunicación USB LEDs de usuario x 4: LD3 (Rojo), LD4 (Verde), LD5 (Azul) Botón de usuario x 2
Potencia	Fuente de entrada externa 5 V (USB VBUS), 7-24 V (Batería o SMPS) Batería predeterminada: LI-PO 11.1V 1,800mAh 19.98Wh SMPS predeterminada: 12V 5A Fuente de salida externa 12V@1A(SMW250-02), 5V@4A(5267-02A), 3.3V@800mA(20010WS-02) Puerto de batería externa para RTC (Real Time Clock) (Molex 53047-0210) LED de alimentación: LD1 (red, 3.3 V power on) Botón de reinicio x 1 (para el restablecimiento de potencia de la placa) Interruptor de alimentación on/off x 1
Dimensiones	105(W) X 75(D) mm
Masa	60g

Especificaciones de la placa de expansión OpenCm 485

Tamaño	68mm x 66.5mm x 16mm
Peso	32g
Voltaje de entrada	5 ~ 30V
Alimentación	SMPS, LIPO Battery, 24V (Dynamixel P)
Interruptor de alimentación	1
Puerto Dynamixel	4Pin x 5 3Pin x 5
Botones	2
LED	5

ANEXO VI. PLANOS