



DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

PROYECTO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA: DISEÑO Y CONSTRUCCIÓN DE UN BRAZO
ROBÓTICO ANTROPOMÓRFICO DE SIETE GRADOS DE
LIBERTAD CON ANÁLISIS CINEMÁTICO Y DINÁMICO
MEDIANTE ALGORITMOS GENÉTICOS

AUTORES:

SÁNCHEZ FRÍAS, ALEXIS PATRICIO

TERÁN GORDILLO, ANDRÉS FRANCISCO

DIRECTOR:

ING. IBARRA JÁCOME, OSWALDO ALEXANDER MSC.

SANGOLQUÍ

2016



DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación “DISEÑO Y CONSTRUCCIÓN DE UN BRAZO ROBÓTICO ANTROPOMÓRFICO DE SIETE GRADOS DE LIBERTAD CON ANÁLISIS CINEMÁTICO Y DINÁMICO MEDIANTE ALGORITMOS GENÉTICOS” realizado por los señores ALEXIS PATRICIO SÁNCHEZ FRÍAS y ANDRÉS FRANCISCO TERÁN GORDILLO, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores ALEXIS PATRICIO SÁNCHEZ FRÍAS y ANDRÉS FRANCISCO TERÁN GORDILLO para que lo sustente públicamente.

Sangolquí, 03 de Febrero del 2016

Ing. Alexander Ibarra

DIRECTOR



DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, ALEXIS PATRICIO SÁNCHEZ FRÍAS, con cédula de identidad N° 1721251039 y ANDRÉS FRANCISCO TERÁN GORDILLO, con cédula de identidad N° 1723620629, declaramos que este trabajo de titulación “DISEÑO Y CONSTRUCCIÓN DE UN BRAZO ROBÓTICO ANTROPOMÓRFICO DE SIETE GRADOS DE LIBERTAD CON ANÁLISIS CINEMÁTICO Y DINÁMICO MEDIANTE ALGORITMOS GENÉTICOS” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 03 de Febrero del 2016

Alexis Patricio Sánchez Frías

C. I: 1721251039

Andrés Francisco Terán Gordillo

C. I: 1723620629



DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN

Nosotros: ALEXIS PATRICIO SÁNCHEZ FRÍAS y ANDRÉS FRANCISCO TERÁN GORDILLO, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “DISEÑO Y CONSTRUCCIÓN DE UN BRAZO ROBÓTICO ANTROPOMÓRFICO DE SIETE GRADOS DE LIBERTAD CON ANÁLISIS CINEMÁTICO Y DINÁMICO MEDIANTE ALGORITMOS GENÉTICOS”, cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Sangolquí, 03 de Febrero del 2016

Alexis Patricio Sánchez Frías

C. I: 1721251039

Andrés Francisco Terán Gordillo

C. I: 1723620629

ÍNDICE

CERTIFICACIÓN	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
ÍNDICE.....	v
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	xiii
RESUMEN	xiv
ABSTRACT	xv
CAPÍTULO 1	1
INTRODUCCIÓN.....	1
1.1 Definición del proyecto.....	1
1.2 Antecedentes	4
1.3 Justificación.....	5
1.3.1 Área de Influencia	5
1.4 Objetivos	5
1.4.1 Objetivo general	5
1.4.2 Objetivos específicos	6
1.5 Metodología	6
1.5.1 Metodología para el diseño mecánico.....	6
1.5.2 Metodología para el análisis cinemático y dinámico mediante algoritmos genéticos.....	7
CAPÍTULO 2.....	8
FUNDAMENTOS BÁSICOS	8
2.1 Introducción a la robótica industrial	8
2.1.1 Morfología de un robot industrial	9
2.1.2 Configuraciones básicas de robots industriales.....	10
2.1.3 Aplicación de robots industriales.....	15

2.2	Cinemática en Robótica Industrial.....	18
2.2.1	Cinemática Directa.....	18
2.2.2	Cinemática Inversa.....	19
2.3	Dinámica en Robótica Industrial.....	20
2.3.1	Dinámica directa	21
2.3.2	Dinámica inversa.....	22
2.3.3	Ley de Euler	22
2.3.4	Ley de Lagrange.....	23
2.4	Método de Algoritmos Genéticos	23
2.4.1	Utilización de algoritmos genéticos para el análisis cinemático y dinámico en robótica industrial.	27
CAPÍTULO 3.....		29
DISEÑO E IMPLEMENTACIÓN MECATRÓNICA.....		29
3.1	Diseño mecánico	29
3.1.1	Reconocimiento de la necesidad	29
3.1.2	Definición del problema.....	29
3.1.3	Síntesis	29
3.1.4	Análisis y Optimización.....	33
3.1.5	Evaluación.....	61
3.1.6	Implementación del diseño mecánico.	61
3.2	Implementación Electrónica.....	67
3.2.1	Fuente de alimentación	67
3.2.2	Controlador usb2Dynamixel	67
3.3	Sistema de control.....	70
3.3.1	Control de Posición Motores Dynamixel.....	70
3.4	Diseño del HMI.....	71
3.4.1	Guía Gedis.....	71
3.4.2	Pantalla de inicio	73
3.4.3	Pantalla de Simulación.....	73
3.4.4	Pantalla de Modo Manual	76
3.4.5	Pantalla de Programación.....	78

3.4.6	SimMechanics	82
3.4.7	Programación de Motores Dynamixel	86
CAPÍTULO 4		88
ANÁLISIS CINEMÁTICO Y DINÁMICO BASADO EN ALGORITMOS GENÉTICOS.		88
4.1	Análisis Cinemático.	88
4.1.1	Parámetros de Denavit Hatenberg.....	88
4.1.2	Solución de la cinemática inversa mediante un algoritmo genético.....	89
4.2	Generación de trayectorias.	97
4.3	Análisis dinámico.....	99
4.4	Ejemplo del análisis cinemático y dinámico.	106
CAPÍTULO 5		110
PRUEBAS Y RESULTADOS.		110
5.1	Pruebas.	110
5.1.1	Pruebas del algoritmo genético	110
5.1.2	Pruebas del robot.....	110
5.2	Resultados.	113
5.2.1	Resultado del algoritmo.	113
5.2.2	Exactitud de Posición.	113
5.2.3	Exactitud de orientación.....	114
5.2.4	Repetibilidad de posición.....	115
5.2.5	Repetibilidad de orientación.	115
CAPÍTULO 6		117
CONCLUSIONES Y RECOMENDACIONES.		117
6.1	Conclusiones.	117
6.2	Recomendaciones.....	118
6.3	Trabajos Futuros.....	118
REFERENCIAS		120

ÍNDICE DE FIGURAS

Figura 1. Configuración Robot redundante.....	1
Figura 2. Volumen de trabajo del Robot redundante, vista frontal, realizada esquemáticamente.	2
Figura 3. Volumen de trabajo del Robot redundante vista superior, realizada esquemáticamente.	2
Figura 4. Configuración Robot redundante realizada esquemáticamente con Matlab.	3
Figura 5. Esquema de metodología de diseño.....	6
Figura 6. Diagrama del método científico.....	7
Figura 7. Constitución física de un robot industrial.....	9
Figura 8. Tipo de articulaciones.	10
Figura 9. Configuraciones básicas de robots industriales.	11
Figura 10. Volumen de trabajo de robots con configuración básica.	11
Figura 11. Robot KUKA LBR iiwa 7 R800.	12
Figura 12. Volumen de trabajo del robot KUKA LBR iiwa 7 R800	12
Figura 13. Robot YASKAWA SIA5D	15
Figura 14. Volumen de trabajo del robot YASKAWA SIA5D.....	15
Figura 15. Robot MOTOMAN en un sistema de carga y descarga.	16
Figura 16. Robot FANUC pintando parachoques de autos.	16
Figura 17. Robot KUKA en un sistema de paletización.	17
Figura 18. Robot FANUC soldador.....	17
Figura 19. Diagrama de relación entre cinemática directa e inversa.	18
Figura 20. Relación entre dinámica directa y dinámica inversa	21
Figura 21. Cruzamiento basado en un punto.....	25
Figura 22. Operador de Mutación.	26
Figura 23. Algoritmo genético básico.	26
Figura 24. Base del robot	30
Figura 25. Eslabón 1	30
Figura 26. Eslabón 2	31
Figura 27. Eslabón 3	31
Figura 28. Eslabón 4	32
Figura 29. Eslabón 5	32
Figura 30. Eslabón 6	32

Figura 31. Aproximación de los eslabones, viga en voladizo.	37
Figura 32. Propiedades del material ABS.	39
Figura 33. Propiedades del soporte o mesa del robot.....	39
Figura 34. Propiedades del respaldo del Eslabón 2.....	40
Figura 35. Soporte de la base del robot.....	40
Figura 36. Sujeciones del soporte de la base para análisis de esfuerzos.....	41
Figura 37. Cargas aplicadas a soporte de la base para análisis de esfuerzos.	41
Figura 38. Tensiones von Mises de la placa de soporte para la base.	41
Figura 39. Desplazamientos de la placa de soporte para la base.	42
Figura 40. Deformación unitaria de la placa de soporte para la base.	42
Figura 41. Factor de seguridad de la placa de soporte para la base.	42
Figura 42. Base.....	43
Figura 43. Sujeciones de la base para análisis de esfuerzos.	43
Figura 44. Cargas aplicadas a la base para análisis de esfuerzos.....	43
Figura 45. Tensiones von Mises de la base.	44
Figura 46. Desplazamiento de la base.....	44
Figura 47. Deformación unitaria de la base.	44
Figura 48. Factor de seguridad en la base.....	45
Figura 49. Cargas aplicadas al eslabón 1 para el análisis de esfuerzos.....	45
Figura 50. Tensiones von Mises del eslabón 1.....	46
Figura 51. Desplazamiento del eslabón 1.	46
Figura 52. Deformación unitaria del eslabón 1.....	47
Figura 53. Factor de seguridad del eslabón 1.....	47
Figura 54. Sujeciones aplicadas al eslabón 2 para el análisis de esfuerzos.	48
Figura 55. Cargas aplicadas al eslabón 2 para análisis de esfuerzos.....	48
Figura 56. Tensiones von Mises en el eslabón 2.	49
Figura 57. Desplazamiento en el eslabón 2.....	49
Figura 58. Deformación unitaria en el eslabón 2.	50
Figura 59. Factor de seguridad del eslabón 2.....	50
Figura 60. Cargas y Sujeciones aplicadas a Eslabón 3.	51
Figura 61. Tensiones von Mises del eslabón 3.....	51
Figura 62. Desplazamiento del eslabón 3.	52
Figura 63. Deformación unitaria del eslabón 3.....	52
Figura 64. Factor de seguridad del eslabón 3.....	52

Figura 65. Cargas aplicadas al eslabón 4 para análisis de esfuerzos.....	53
Figura 66. Tensiones von Mises del eslabón 4.....	53
Figura 67. Desplazamientos del eslabón 4.....	54
Figura 68. Deformación unitaria del eslabón 4.....	54
Figura 69. Factor de seguridad del eslabón 4.....	54
Figura 70. Cargas aplicadas al eslabón 5 análisis de esfuerzos.	55
Figura 71. Tensiones von Mises del eslabón 5.....	55
Figura 72. Desplazamiento del eslabón 5.	56
Figura 73. Deformación unitaria del eslabón 5.....	56
Figura 74. Factor de seguridad del eslabón 5.....	56
Figura 75. Cargas aplicadas al eslabón 6 para análisis de esfuerzos.....	57
Figura 76. Tensiones von Mises del eslabón 6.....	57
Figura 77. Desplazamientos del eslabón 6.....	58
Figura 78. Deformación unitaria del eslabón 6.....	58
Figura 79. Factor de seguridad del eslabón 6.....	58
Figura 80. Cargas aplicadas a soporte del efector final para análisis de esfuerzos.	59
Figura 81. Tensiones von Mises del soporte para el efector final.....	59
Figura 82. Desplazamientos del soporte para el efector final.....	60
Figura 83. Deformación unitaria del soporte para el efector final.....	60
Figura 84. Factor de seguridad del soporte para el efector final.....	60
Figura 85. Máquina de prototipado rápido Makerbot Replicator x2.	62
Figura 86. Vista superior del prototipado del eslabón 2.....	62
Figura 87. Vista lateral del prototipado del eslabón 2.....	63
Figura 88. Eslabones y motores del robot.....	63
Figura 89. Gripper ensamblado al séptimo motor.....	64
Figura 90. Séptimo eslabón ensamblado al gripper.	64
Figura 91. Quinto, sexto y séptimo eslabón ensamblado.....	65
Figura 92. Robot ensamblado desde el tercer eslabón.....	65
Figura 93. Base y eslabón 1 ensamblados.....	66
Figura 94. Robot ensamblado.....	66
Figura 95. Controlador USB2Dynamixel.....	68
Figura 96. Diagrama de conexión PC/USB2Dynamixel/motores dynamixel.....	68
Figura 97. Diagrama de conexión de la fuente de poder a los motores.....	68

Figura 98. Status y selector de función del controlador USB2Dynamixel	69
Figura 99. Conectores del controlador USB3Dynamixel	69
Figura 100. Controlador PID interno de motores dynamixel.....	71
Figura 101. Diagrama de flujo Pantalla Inicio.	73
Figura 102. Pantalla de Inicio.	73
Figura 103. Pantalla Simulación.	74
Figura 104. Diagrama de flujo- botón home.....	74
Figura 105. Diagrama de flujo- botón Generar Trayectoria.....	75
Figura 106. Diagrama de flujo- botón Simular.	76
Figura 107. Diagrama de flujo- botón Teach.	77
Figura 108. Pantalla Modo Manual.....	78
Figura 109. Pantalla -Programación.....	79
Figura 110. Pantalla -Programación.....	79
Figura 111. Diagrama de flujo - mover	80
Figura 112. Pantalla – mover recta.....	81
Figura 113. Pantalla – Pausa.	81
Figura 114. Pantalla – Gripper.	82
Figura 115. Exportación de un ensamblaje CAD a SimMechanics.	82
Figura 116. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 1.	83
Figura 117. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 2.	83
Figura 118. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 3.	83
Figura 119. Parámetros del bloque Revolute.	84
Figura 120. Bloque Joint Actuator con constantes de posición, velocidad y aceleración.....	84
Figura 121. Parámetros del bloque Joint Actuator.	85
Figura 122. Ensamblaje CAD importado a Matlab.....	85
Figura 123. Paquete de transmisión de datos de Motores Dynamixel en Matlab.	86
Figura 124. Diagrama que representa los parámetros de Denavit Hatenberg.	88
Figura 125. Selección uniforme estocástica.	91
Figura 126. Trayectoria recta dividida en cuatro puntos.....	98
Figura 127. Masa, centros de gravedad y momentos de inercia del eslabón 1.	99

Figura 128. Masa, centros de gravedad y momentos de inercia del eslabón 2.	100
Figura 129. Masa, centros de gravedad y momentos de inercia del eslabón 3.	100
Figura 130. Masa, centros de gravedad y momentos de inercia del eslabón 4.	101
Figura 131. Masa, centros de gravedad y momentos de inercia del eslabón 5.	101
Figura 132. Masa, centros de gravedad y momentos de inercia del eslabón 6.	102
Figura 133. Masa, centros de gravedad y momentos de inercia del soporte del efector final.	102
Figura 134. Posición final del robot AR7D para el ejemplo en el plano X-Z.	107
Figura 135. Posición final del robot AR7D para el ejemplo en vista 3D.	107
Figura 136. Posición angular.	108
Figura 137. Velocidad angular.	108
Figura 138. Aceleración angular.	109
Figura 139. Torque requerido.	109
Figura 140. Robot AR7D realizando las pruebas de exactitud y repetibilidad.	111

ÍNDICE DE TABLAS

Tabla 1. Especificaciones básicas del robot KUKA LBR i 7 R800.	13
Tabla 2. Especificaciones del movimiento del robot KUKA LBR 7 R800.....	13
Tabla 3. Especificaciones técnicas robot YASKAWA SIA5D	14
Tabla 4. Expresiones utilizadas en genética y su equivalente en algoritmos genéticos.....	24
Tabla 5. Diferencias entre motores a pasos y servomotores DC	34
Tabla 6. Comparación motores dynamixel.....	35
Tabla 7. Peso y material de los elementos del robot manipulador	38
Tabla 8. Resultados del análisis estático de los eslabones	61
Tabla 9. Parámetros para el prototipado de los eslabones.....	62
Tabla 10. Corriente necesaria por los motores dynamixel del robot.....	67
Tabla 11. Características de la fuente de poder ALTEK ATX-750.....	67
Tabla 12. Partes del controlador USB2Dynamixel.	69
Tabla 13. Características del software del controlador USB2Dynamixel.....	70
Tabla 14. Parámetros de Denavit Hatenberg del robot AR7D	89
Tabla 15. Restricciones de las articulaciones del robot.....	90
Tabla 16. Variación de la constante C para sintonizar el error de orientación.	95
Tabla 17. Variación del tamaño de la población.....	96
Tabla 18. Variación del número de generaciones.	97
Tabla 19. Masas y coordenadas del centro de gravedad de los eslabones del robot AR7D.	105
Tabla 20. Momentos de inercia de los eslabones del robot AR7D.....	105
Tabla 21. Posición y orientación alcanzada por el algoritmo genético.	106
Tabla 22. Cinemática inversa resuelta por algoritmo genético.	106
Tabla 23. Exactitud del robot AR7D	115
Tabla 24. Repetibilidad del robot AR7D.....	116

RESUMEN

Un objetivo de investigación en la robótica industrial es brindar mayor flexibilidad a un robot, para ahorrar espacio en una fábrica, optimizar la producción, realizar tareas de difícil acceso o evitar colisiones. En el presente proyecto nos propusimos diseñar e implementar un robot de tipo antropomórfico que presente una mayor flexibilidad con respecto a los robots de configuraciones tradicionales. Una de las maneras de agregarle flexibilidad a un brazo robótico es incrementar los grados de libertad, Sin embargo incrementar grados de libertad dificulta el análisis y solución tanto de la cinemática como de la dinámica de un robot de estas características. El brazo robótico que se diseñó tiene siete grados de libertad y la herramienta para resolver el posicionamiento y orientación del robot es un algoritmo genético. Los algoritmos genéticos tienen un principio de optimización que se puede aplicar a un problema de ingeniería como en nuestro caso ayudarnos a resolver la cinemática y dinámica de un brazo robótico de siete grados de libertad.

Palabras clave:

- **ROBÓTICA**
- **ALGORITMOS GENÉTICOS**
- **GRADOS DE LIBERTAD.**

ABSTRACT

An objective of industrial robotics is the flexibility to a robot, for difficult tasks or avoid collisions. In this project, we design and implement an anthropomorphic robot more flexible than the traditional industrial robots. One way to add flexibility to a robotic arm is to increase the degrees of freedom; however increasing degrees of freedom makes difficult the kinematic analysis and the dynamic analysis. The robotic arm in this project has seven degrees of freedom, the positioning and orientation of the robot is going to be solved with a genetic algorithm. Genetic algorithms have an optimization principle that can be applied to an engineering problem and it can be useful to solve the kinematics and dynamics of a robotic arm with seven degrees of freedom.

Keywords:

- **ROBOTICS**
- **GENETIC ALGORITHMS**
- **DEGREES OF FREEDOM**

CAPÍTULO 1

INTRODUCCIÓN

1.1 Definición del proyecto

Un robot redundante es aquel que posee más grados de libertad que los que se usa convencionalmente para las tareas que puede realizar el mismo como manipulación, soldadura, pintura, etc. según (Ollero, 2001) el robot redundante posee más de seis grados de libertad y son usados para lugares de difícil acceso, mayor flexibilidad y seguridad industrial.

En estos últimos años los principales diseñadores de robots Industriales como KUKA, YASKAWA, etc. Están investigando nuevas configuraciones, las cuales en su gran mayoría son redundantes.

La configuración que se pretende implementar es una configuración más humanoide dándole mucha libertad de movimientos y flexibilidad como se lo puede visualizar en la Figura 1.

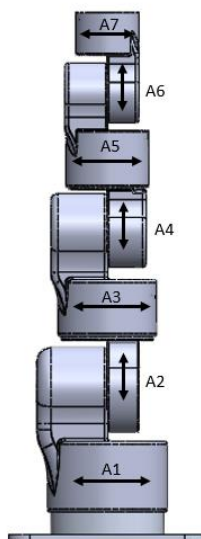


Figura 1. Configuración Robot redundante.

En la Figura 2 se muestra una vista frontal del robot manipulador, el espacio de color gris es la parte accesible para el robot y el espacio blanco corresponde al espacio no accesible del robot. Se forma una superficie no accesible semejante a una circunferencia.

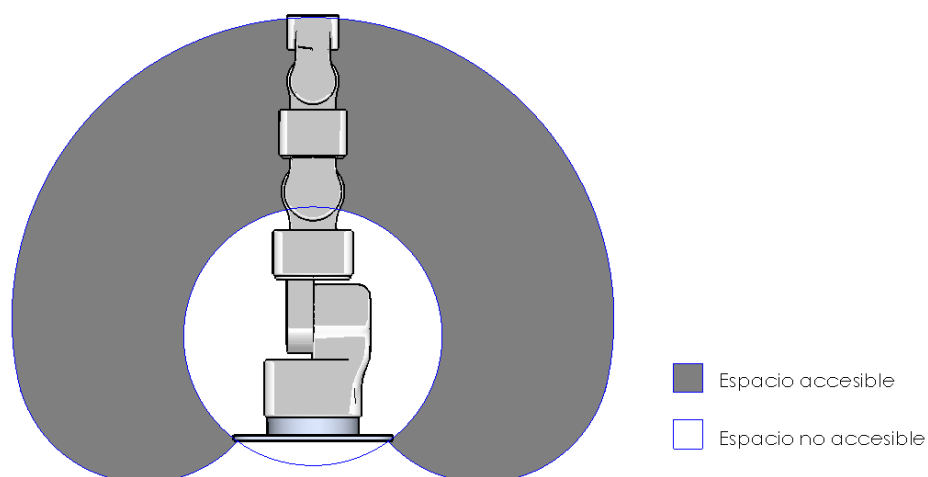


Figura 2. Volumen de trabajo del Robot redundante, vista frontal, realizada esquemáticamente.

En la Figura 3 se muestra una vista superior del robot manipulador, el espacio accesible es gris y el espacio no accesible es blanco. El volumen formado por la proyección de las dos áreas mostradas en la Figura 2 y la Figura 3 corresponde al volumen de trabajo del robot.

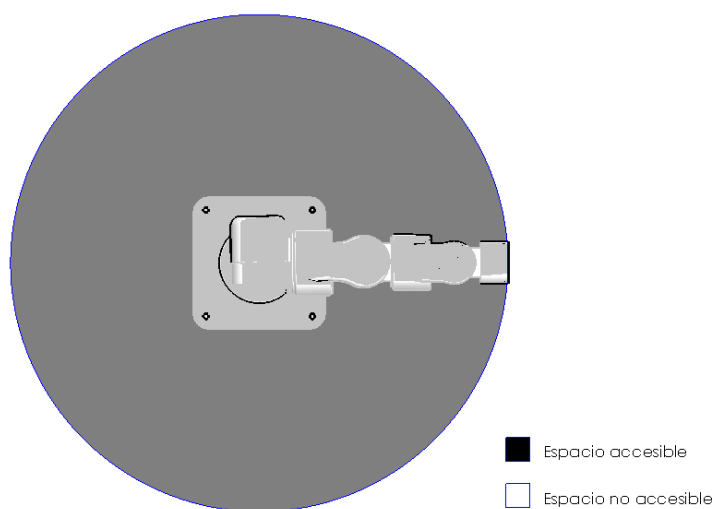


Figura 3. Volumen de trabajo del Robot redundante vista superior, realizada esquemáticamente.

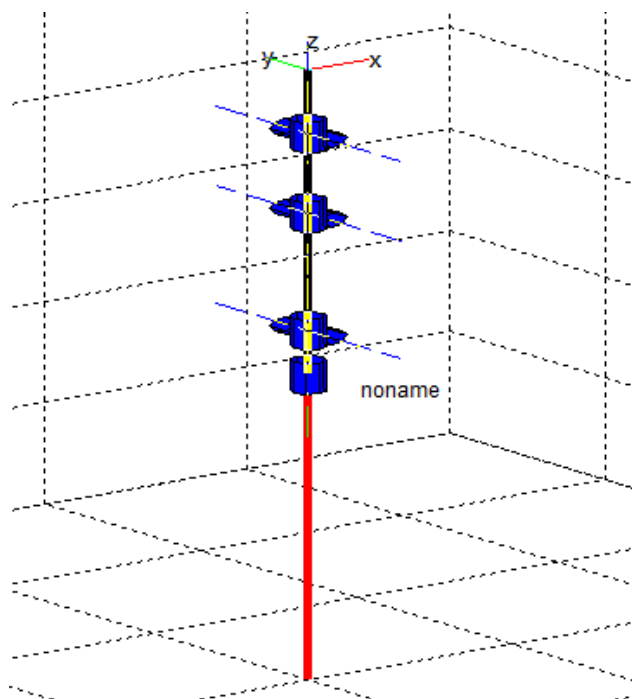


Figura 4. Configuración Robot redundante realizada esquemáticamente con Matlab.

La configuración elegida es de siete eslabones y siete articulaciones rotacionales alternadas una alrededor del eje Z y la siguiente alrededor del eje X y así sucesivamente hasta terminar con la última articulación girando con respecto al eje Z; debido a que permite una mayor flexibilidad de movimientos y es adecuada para demostrar el uso de posicionamiento en un robot redundante además posee una serie de aplicaciones industriales como son manipulación en espacios complicados, robots de asistencia, etc.



Figura 8. Vista Isométrica realizada esquemáticamente con RoboAnalyzer.

Una vez desarrollado el correcto diseño mecánico del robot manipulador redundante el siguiente aspecto es poder resolver su cinemática y su dinámica.

Debido a sus grados de libertad se vuelve complicado resolverlo por medio de geometría o el convencional método mediante la matriz homogénea. Nuevamente volvemos a los métodos utilizados por los grandes desarrolladores como KUKA y otros fabricantes que ya utilizan métodos evolutivos como algoritmos genéticos para resolver el problema cinemático inverso de dichos robots.

Esta investigación pretende demostrar que se puede aplicar el método de los algoritmos genéticos a la robótica industrial, en robots que son complejos en su cadena cinemática.

1.2 Antecedentes

Los manipuladores robóticos van cobrando mayor importancia en la industria ya que permiten realizar tareas pesadas, reiterativas, o peligrosas, este tipo de robot ya es ampliamente aplicado en el mundo sin embargo con el avance tecnológico aparecen nuevas necesidades, como por ejemplo acceder a lugares complicados, mayor flexibilidad, seguridad industrial, etc.

Para cumplir con estos requerimientos se necesita aumentar los posibles movimientos o grados de libertad que pueda realizar dicho robot, sin embargo esto aumenta la complejidad matemática al analizar el movimiento con los métodos tradicionales como Denavit-Hartenberg el cual permite la resolución del posicionamiento para robots.

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997) este método emplea cuatro submatrices (rotación, traslación, perspectiva y escalada) que componen una matriz principal denominada matriz de transformación homogénea.

Igualando los parámetros de traslación de la matriz de transformación se tendría tres ecuaciones correspondientes a los sistemas ortogonales X, Y, Z y n incógnitas correspondientes a los ángulos que debe moverse los eslabones por cada grado de libertad que se agrega al robot sin mencionar su orientación que generarían tres ecuaciones adicionales.

Al aumentar los grados de libertad a más de seis el método de Denavit-Hartenberg quedaría con más incógnitas que ecuaciones siendo muy complicada la resolución por este método.

Es por ello que el objetivo de esta investigación es diseñar un brazo robótico antropomórfico didáctico que permita demostrar el correcto posicionamiento por medio de métodos evolutivos como son los algoritmos genéticos.

1.3 Justificación

La Universidad de las Fuerzas Armadas ESPE podrá hacer uso del manipulador didáctico para demostrar el posicionamiento de un robot antropomórfico redundante de siete grados de libertad por medio del método de algoritmos genéticos, además emprender la investigación en nuevos métodos para resolver problemas en ingeniería.

Siendo la solución de la cinemática inversa mediante el método de algoritmos genéticos un tema nuevo de carácter no explorado antes en otros trabajos de tesis o investigación desarrollados en la Universidad de las Fuerzas Armadas ESPE.

La importancia de este tema radica en la incursión de métodos diferentes a los tradicionales permitiéndole ganar a la Universidad la primicia de explorar el tema de cinemática inversa y dinámica por algoritmos genéticos cuya investigación no ha sido desarrollada en otras instituciones del país.

1.3.1 Área de Influencia

El desarrollo de este proyecto aportará a la Universidad de las Fuerzas Armadas ESPE una importante investigación que ayudara al desarrollo del conocimiento en el área de robótica.

1.4 Objetivos

1.4.1 Objetivo general

- Diseñar y construir un brazo robótico antropomórfico de siete grados de libertad con análisis cinemático y dinámico mediante algoritmos genéticos.

1.4.2 Objetivos específicos

- Realizar una simulación del robot mediante software CAD para evaluar el diseño.
- Implementar el diseño del robot de manera adecuada siguiendo un proceso normativo para que sea seguro para el usuario u operario.
- Probar el robot mediante una serie de procedimientos diseñados para verificar su correcto funcionamiento.
- Analizar los resultados al utilizar algoritmos genéticos para el posicionamiento y generación de trayectorias para el robot comprobando el error y la precisión del mismo.

1.5 Metodología

1.5.1 Metodología para el diseño mecánico

Se seguirá la metodología de diseño propuesta en el libro Diseño en ingeniería de Shigley, octava edición. Recomendado para diseño mecánico. En la figura 5 se puede analizar el diagrama de flujo de la metodología de diseño.

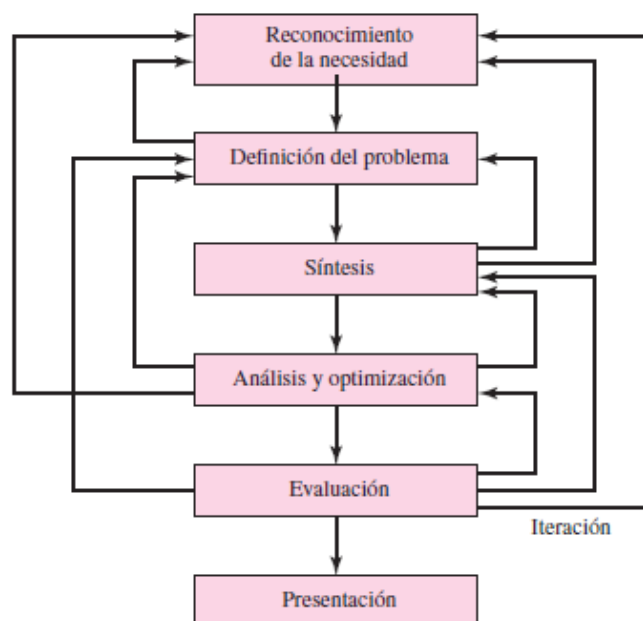


Figura 5. Esquema de metodología de diseño

Fuente: (Budynas & Nisbett, 2008)

1.5.2 Metodología para el análisis cinemático y dinámico mediante algoritmos genéticos

Para el algoritmo genético se utilizará el método científico, ideal para investigación, el diagrama de flujo del método científico se observa en la Figura 6.

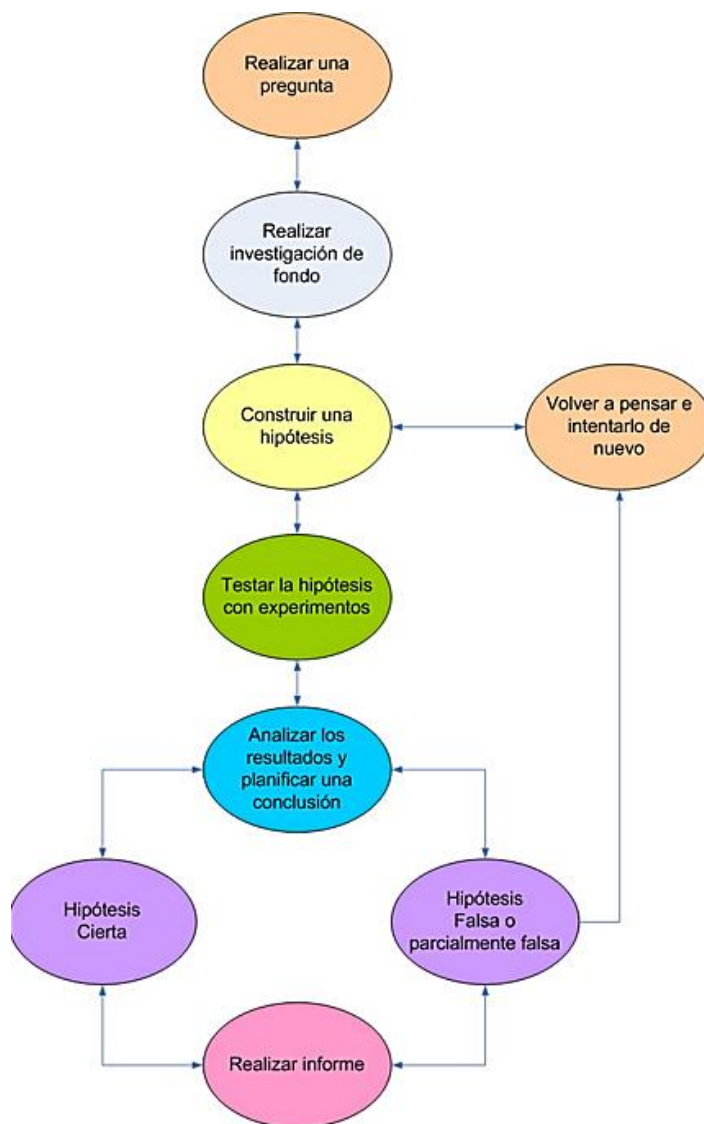


Figura 6. Diagrama del método científico.

Fuente: (Wikipedia, 2015)

CAPÍTULO 2

FUNDAMENTOS BÁSICOS

2.1 Introducción a la robótica industrial

Los robots aparecieron por primera vez en películas de ciencia ficción como máquinas mecánicas con la forma de un ser humano pero destinado a realizar las actividades engorrosas para ellos.

Se desarrollaron robots controlados a distancia como lo describe (Barrientos, Peñín, Balaguer, & Aracil, 1997) en 1948 R.C. Goertz del Argonne National Laboratory desarrolló el primer robot telemanipulador de elementos radioactivos, un dispositivo mecánico maestro-esclavo, el maestro era movido por el operador y el esclavo reproducía los movimientos.

Según (Craig, 2006) y (Barrientos, Peñín, Balaguer, & Aracil, 1997) aparecieron los primeros robots manipuladores en Estados Unidos por los años 1960, la empresa Unimation utilizó robots esféricos nombrados Unimate para la automatización de procesos de manufactura en la fábrica de General Motors de Trenton, Nueva Jersey, gracias a la interacción del diseño asistido por computador (CAD) y la manufactura asistida por computador (CAM) como una solución en la automatización de procesos de manufactura.

El avance de la robótica en Japón se desarrolló en la misma época con la firma entre la marca Kawasaki y Unimation para la producción de robots tipo Unimate. En 1973 en Europa la firma sueca ASEA construyó el primer robot antropomórfico con accionamiento totalmente eléctrico llamado IRb6. En 1982, el profesor Makito de la Universidad Yamanashi desarrolló el robot SCARA (Selective Compliance Assembly Robot Arm).

Los siguientes robots fueron mejorando su diseño para un mayor acceso a lugares difíciles, flexibilidad, movilidad, repetitividad, facilidad de uso, entre otras características.

Algunos fabricantes aumentaron el número de eslabones, usan el modelo de un torso humano (dos brazos) y configuraciones especiales para manipulación o rapidez. En otras áreas se desarrollan humanoides que son robots con su fisionomía muy parecida a la de los humanos para asistencia de adultos mayores, robots exploradores

para buscar vida en otros planetas, drones, robots para desarmar bombas, entre otros. A continuación algunas marcas de robots industriales: KUKA Roboter GmbH, Yaskawa Electric Corporation, Fanuc Robotics, ABB (Asea Brown Boveri), Universal Robots.

2.1.1 Morfología de un robot industrial

Los robots industriales poseen una configuración semejante a la de un brazo humano, de una mano o la de un torso (dos brazos) como se muestra en la Figura 7.

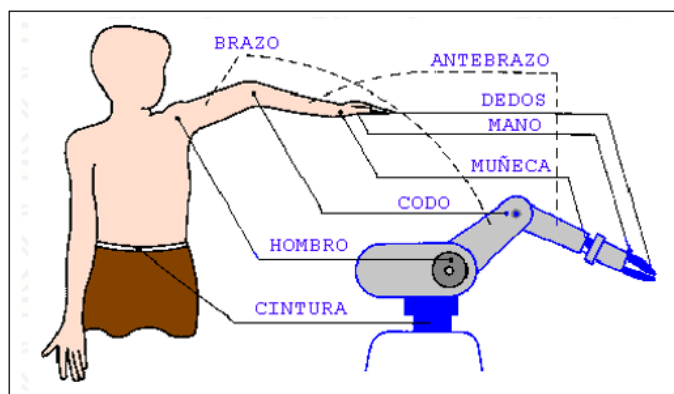


Figura 7. Constitución física de un robot industrial

Fuente: (González, 2002)

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997) un robot está constituido por los siguientes partes:

- Estructura mecánica: eslabones conectados mediante articulaciones que determinan el número de grados de libertad del robot.
- Transmisiones: permiten transferir el movimiento, pueden aumentar o disminuir el torque mecánico teniendo un efecto inverso en la velocidad del movimiento.
- Sistema de accionamiento: actuadores mecánicos (motores), hidráulicos y neumáticos (pistones).
- Sistema sensorial: existen sensores internos y sensores externos. Los sensores internos detectan el estado del mismo principalmente la posición de las articulaciones; temperatura, torque y velocidad, de los actuadores. Los sensores externos registran variables del entorno del robot.
- Sistema de control: gobierna el funcionamiento del robot, puede contar o no de un teach pendant y un ordenador.

- Elementos terminales: pinzas, herramientas, dispositivos de sujeción, etc.

Un robot que se desenvuelve en un entorno de producción industrial de forma automática, posee tres o más grados de libertad y actuadores finales, ejecuta programas previamente cargados en su controlador, es reprogramable y algunos son reconfigurables físicamente debido a la adición o reducción de eslabones, es considerado un robot industrial.

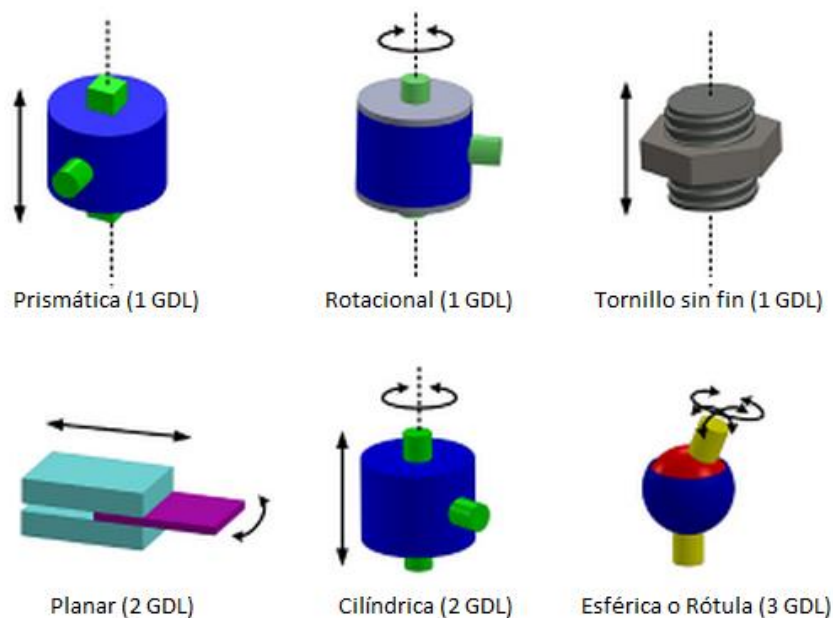


Figura 8. Tipo de articulaciones.

Fuente: (Hernández Ordoñez, Ortiz Moctezuma, Calles Arriaga, & Rodríguez Portillo, 2015)

2.1.2 Configuraciones básicas de robots industriales

Dependiendo del tipo y ubicación de articulaciones empleadas, se puede obtener las configuraciones básicas que se ilustran en la Figura 9, estos robots poseen tres articulaciones. El robot cartesiano (PPP) también conocido como prismático posee tres articulaciones prismáticas o lineales.

El robot cilíndrico (TLL) posee una articulación rotacional de torsión y dos articulaciones lineales. El robot esférico (TRL) o polar tiene una articulación rotacional de torsión, una rotacional y una lineal o prismática. El robot antropomórfico (TRR), articulado vertical o angular, posee una articulación rotacional de torsión y dos

articulaciones rotacionales. El robot SCARA (TRL) o articulado horizontal posee una articulación rotacional de torsión, una rotacional y una lineal o prismática.

El espacio de trabajo o volumen de trabajo que describe el movimiento de cada robot se puede observar en la Figura 10. El robot cartesiano tiene un espacio de trabajo prismático con base rectangular, el cilíndrico un cilindro, el esférico una esfera, el antropomórfico una esfera con limitaciones en la proximidad de su base y el SCARA el espacio comprendido entre dos cilindros.

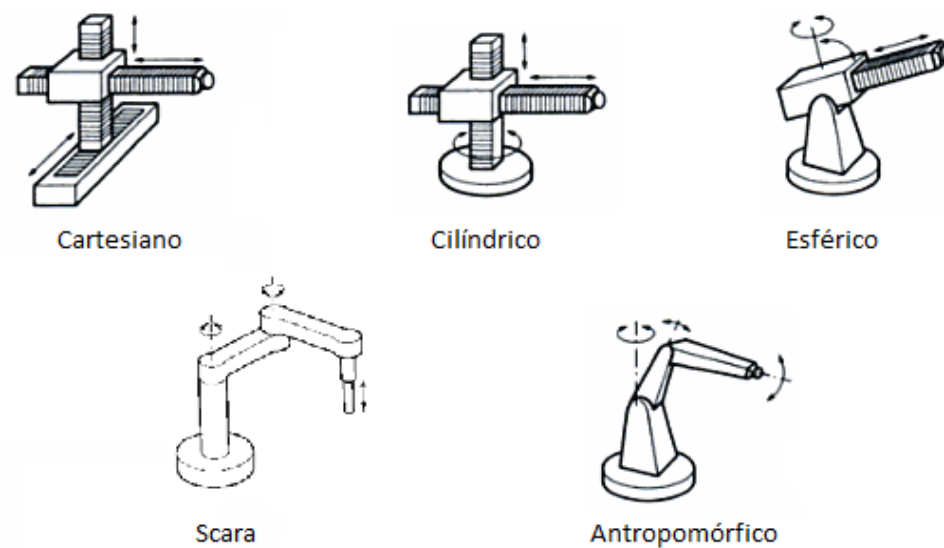


Figura 9. Configuraciones básicas de robots industriales.

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, 1997)

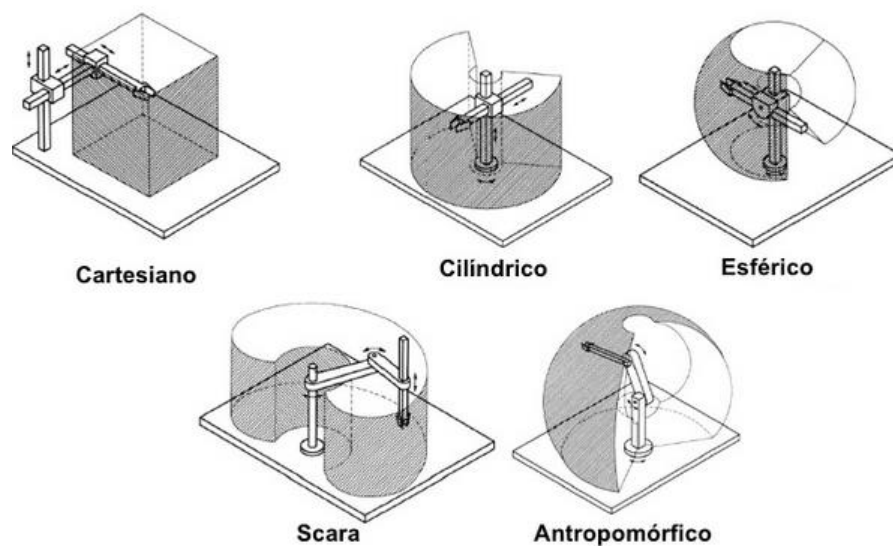


Figura 10. Volumen de trabajo de robots con configuración básica.

Fuente: (Sánchez, 2008)

2.1.2.1 Configuraciones de robots antropomórficos redundantes

Las empresas KUKA Roboter GmbH, YASKAWA Motoman Robotics desarrollan modelos redundantes. Los dos fabricantes desarrollaron una configuración diferente ocupando 7 grados de libertad.

Ambos diseños permiten alcanzar lugares difíciles, el robot desarrollado por la empresa YASKAWA posee un mayor rango de movimiento de sus articulaciones y una velocidad mayor en sus actuadores.

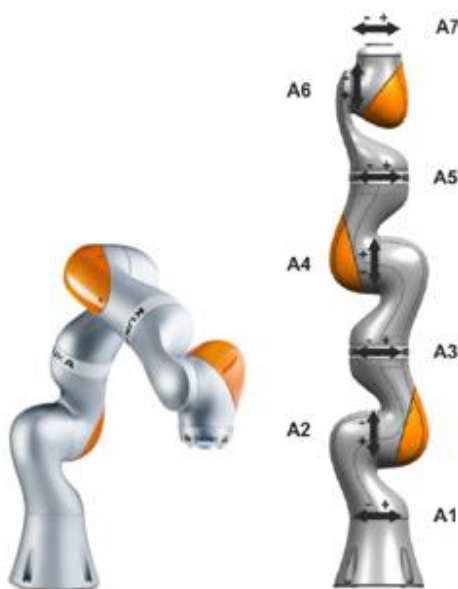


Figura 11. Robot KUKA LBR iiwa 7 R800.

Fuente: (Roboter GmbH, 2015)

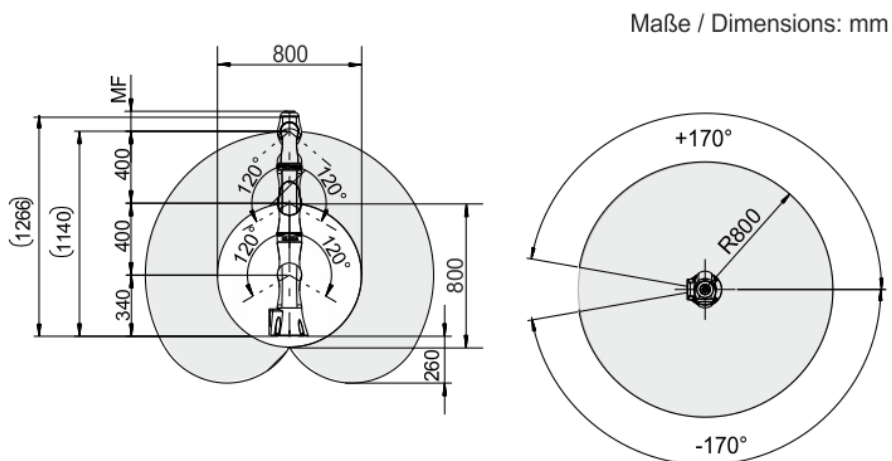


Figura 12. Volumen de trabajo del robot KUKA LBR iiwa 7 R800

Fuente: (Roboter GmbH, 2015)

Tabla 1.

Especificaciones básicas del robot KUKA LBR iiwa 7 R800

Parámetros	Especificaciones
Número de ejes	7
Número de ejes controlados	7
Volumen de espacio de trabajo	1.7 m ³
Pose repetibilidad (ISO 9283)	± 0.1 mm
Peso aproximado	23.9 kg
Carga útil nominal	7 kg
Alcance máximo	800 mm
Grado de protección	IP 54
Grado de protección, muñeca en línea	IP 54

Fuente: (Roboter GmbH, 2015)

Tabla 2.

Especificaciones del movimiento del robot KUKA LBR iiwa 7 R800

Rango de movimiento	
A1	±170 °
A2	±120 °
A3	±170 °
A4	±120 °
A5	±170 °
A6	±120 °
A7	±175 °
Velocidad con carga útil nominal	
A1	98 °/s
A2	98 °/s
A3	100 °/s
A4	130 °/s
A5	140 °/s
A6	180 °/s
A7	180 °/s

Fuente: (Roboter GmbH, 2015)

El robot iiwa de KUKA Robotics (Figura 11) tiene un volumen de trabajo más completo y su diseño es un poco más agradable visualmente en comparación al robot SIA5D de YASKAWA (Figura13).

En videos de demostración se puede notar la versatilidad del robot iiwa al jugar ping pong, en comparación el SIA5D realizar actividades de pick and place de excelente manera evadiendo obstáculos en lugares con obstáculos y de difícil acceso

Tabla 3.

Especificaciones técnicas robot YASKAWA SIA5D

Parámetros		Especificaciones
Estructura		Articulada
Montaje		Piso, pared o techo
Ejes controlados		7
Carga útil		5 kg (11 lb)
Alcance vertical		1.007 mm (39.6")
Alcance horizontal		559 mm (22")
Repetibilidad		± 0.06 mm (± 0.002 ")
Máximo rango de movimiento	S-Axis (Turning/Sweep)	$\pm 180^\circ$
	L-Axis (Lower Arm)	$\pm 110^\circ$
	E-Axis (Elbow)	$\pm 170^\circ$
	U-Axis (Upper Arm)	$+115^\circ/-90^\circ$
	R-Axis (Wrist Roll)	$\pm 180^\circ$
	B-Axis (Bend/Pitch/Yaw)	$\pm 110^\circ$
	T-Axis (Wrist Twist)	$\pm 180^\circ$
Velocidad máxima	S-Axis	200°/s
	L-Axis	200°/s
	E-Axis	200°/s
	U-Axis	200°/s
	R-Axis	200°/s
	B-Axis	230°/s
	T-Axis	350°/s
Masa aproximada		30 kg (66.2 lb)
Potencia nominal		1 kVA
Momento admisible	R-Axis	14.7 N • m
	B-Axis	14.7 N • m
	T-Axis	7.35 N • m
Momento de inercia admisible	R-Axis	0.45 kg • m ²
	B-Axis	0.45 kg • m ²
	T-Axis	0.11 kg • m ²
Clase de protección	Estándar	Sin clasificar
	Versión XP* (opción)	N/A

* XP Versión: Paquete de protección extra de Yaskawa Motoman

Fuente: (Motoman Robotics, 2014)

Comparando las especificaciones del robot KUKA LBR iiwa 7 R800 (tabla 1) con las del robot YASKAWA SIA5D, se puede observar que la repetibilidad es mayor en el modelo de la compañía YASKAWA, a su favor el Robot de la empresa KUKA tiene una mayor capacidad de carga.



Figura 13. Robot YASKAWA SIA5D

Fuente: (Motoman Robotics, 2014)

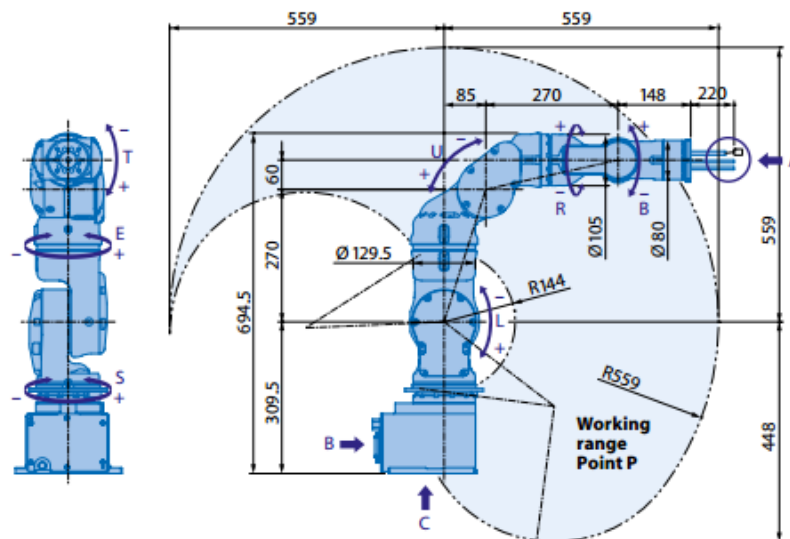


Figura 14. Volumen de trabajo del robot YASKAWA SIA5D

Fuente: (Motoman Robotics, 2014)

2.1.3 Aplicación de robots industriales

Los robots industriales son útiles en tareas peligrosas para los seres humanos o en procesos que se pueden automatizar.

Los robots industriales hoy en día cumplen muchas funciones en el área de manufacturera, alimenticia, farmacéutica, agrícola, científica, electrónica, automovilismo, química, etc.

Dentro de la manufactura hay trabajos como: fundición donde los hornos pueden alcanzar temperaturas superiores a los 1000 °C, siendo útiles para tomar y colocar piezas o hasta verter hierro fundido; soldadura de chasises o de partes metálicas, hasta en posiciones incómodas como la 5G para tubería; pintura de partes de un automóvil con una homogeneidad para ahorrar pintura y a la vez obtener un buen acabado; en operaciones de maquinado como torneado, colocando y posicionando piezas.



Figura 15. Robot MOTOMAN en un sistema de carga y descarga.

Fuente: (Rivas, 2010)



Figura 16. Robot FANUC pintando parachoques de autos.

Fuente: (Fanuc, 2015)



Figura 17. Robot KUKA en un sistema de paletización.

Fuente: (Rivas, 2010)



Figura 18. Robot FANUC soldador.

Fuente: (Mecalux, 2015)

2.2 Cinemática en Robótica Industrial

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997) la cinemática estudia el movimiento del robot con respecto a un sistema de referencia, nos interesa principalmente las relaciones que existen entre la orientación y la posición del efector final del robot con los valores de sus coordenadas articulares.

Dependiendo del sentido en el que se tome en cuenta la relación antes mencionada se genera dos problemas o tipos de cinemática, el problema cinemático directo y el problema cinemático inverso.

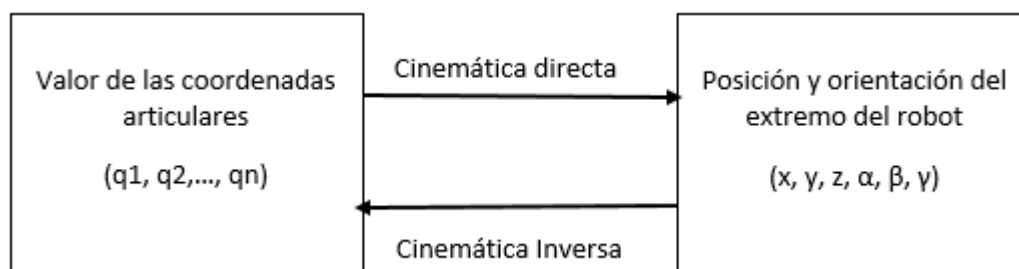


Figura 19. Diagrama de relación entre cinemática directa e inversa.

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, 1997)

2.2.1 Cinemática Directa

El problema cinemático directo consiste en encontrar la posición y orientación del efector final teniendo como datos las coordenadas articulares.

Para resolver el problema cinemático directo existen varios métodos como son: Matrices de transformación homogénea, Consideraciones geométricas, Parámetros de Denavit-Hartenberg o mediante el uso de cuaternios.

Para nuestro diseño utilizaremos los parámetros de Denavit-Hartenberg para obtener el modelo cinemático directo del robot manipulador. (Denavit, Hartenberg, 1955) propusieron un algoritmo que permite resolver el problema cinemático directo mediante un método matricial y sistemático, encontrando las relaciones que permiten conocer la posición del efector final teniendo las coordenadas articulares.

El algoritmo de Denavit- Hartenberg para obtener el modelo cinemático directo se encuentra en (Barrientos, Peñín, Balaguer, & Aracil, 1997, págs. 97-100) y menciona que a partir de cuatro matrices de transformación básicas que son: Rotación alrededor

del eje \mathbf{z} un ángulo θ , translación a lo largo de \mathbf{z} una distancia \mathbf{d} , translación a lo largo de \mathbf{x} una distancia \mathbf{a} y rotación alrededor de \mathbf{x} un ángulo α .

Luego se procede a multiplicar las matrices de rotación y translación obteniendo la representación de cada eslabón según Denavit- Hartenberg, de la siguiente manera:

$${}^{i-1}A_i = T(z, \theta_i)T(0,0, d_i)T(a_i, 0,0)T(x, \alpha_i)$$

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

De forma expandida la matriz de transformación ${}^{i-1}A_i$, es:

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

A continuación se obtiene la matriz de transformación ${}^{i-1}A_i$ de cada eslabón y se multiplican entre si ordenadamente para obtener la matriz T:

$$T = {}^0A_1, {}^1A_2, \dots, {}^{n-1}A_n$$

La matriz T define la orientación y posición del robot en función de las coordenadas articulares es decir la solución a la cinemática directa.

2.2.2 Cinemática Inversa

El problema cinemático inverso es lo opuesto al problema cinemático directo, es decir que consiste en encontrar las coordenadas articulares del robot teniendo como datos los posición y orientación deseadas en el efector final.

Para resolver el problema cinemático inverso también existen varios métodos los más utilizados se mencionan en (Barrientos, Peñín, Balaguer, & Aracil, 1997, págs. 110-128), y son:

Métodos geométricos: Consiste en utilizar relaciones geométricas entre las coordenadas articulares, las coordenadas que se desean en el efector final y las dimensiones de sus eslabones, este método es recomendado para robots de pocos grados de libertad.

A partir de la matriz de transformación homogénea: consiste en obtener las coordenadas articulares a partir del modelo cinemático directo mediante la matriz de transformación homogénea, e igualando los parámetros de posición y orientación a las respectivas coordenadas que se desean alcanzar con el efector final, sin embargo se considera un método complejo.

Desacoplo cinemático: Consiste en separar el problema de posicionamiento y el problema de orientación, por lo general se aplica a robots de seis grados de libertad donde es importante la orientación del robot, donde los tres primeros eslabones posicionan el robot y a partir de la muñeca los otros se encargan de orientar al robot.

Sin embargo el problema cinemático inverso se complica al incrementar articulaciones ya que solo podemos tener como máximo seis relaciones, tres para posicionamiento y tres para orientación.

Para el robot antropomórfico de siete grados de libertad estos métodos se vuelven ineficientes y no tendrá una solución matemática explícita, para ello se utilizará una metodología mediante algoritmos genéticos.

2.3 Dinámica en Robótica Industrial

Según (Craig, 2006) la dinámica de un robot trata las fuerzas que se requieren para producir el movimiento y el torque necesario en las articulaciones. El modelo dinámico del robot sirve para dimensionar los actuadores de las articulaciones, simular el movimiento del robot, comprobar la estructura mecánica del robot y verificar el comportamiento del control dinámico que se emplea.

En (Kelly & Santibáñez, 2003) considera a un robot como un sistema de n entradas correspondientes al número de articulaciones y $2n$ salidas teniendo la posición y velocidad en cada articulación, pudiendo expresarse el modelo dinámico del robot mediante ecuaciones diferenciales, utilizando las técnicas:

- Analítica, mediante ecuaciones físicas que representan el modelo matemático del robot, se emplean técnicas como: Euler y Lagrange.
- Experimental, dando entradas conocidas al robot y analizando el comportamiento del mismo, obteniendo datos que describen su modelo, este método es más preciso que el analítico pero se requiere tener físicamente el robot.

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997), con cualquiera de las dos técnicas se puede obtener un modelo dinámico directo o inverso, dependiendo de la relación que se desee encontrar como se muestra en la Figura 20.

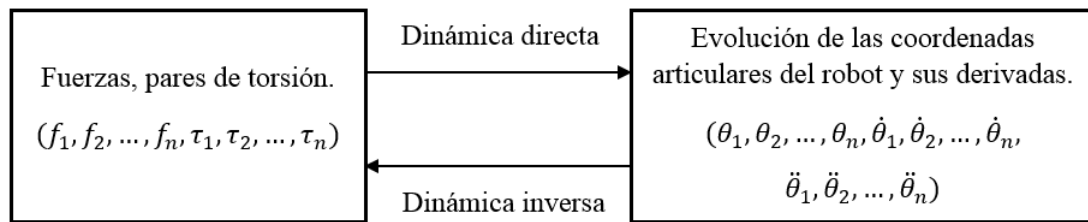


Figura 20. Relación entre dinámica directa y dinámica inversa

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, 1997)

Dependiendo del modelo dinámico, directo o inverso que se desee encontrar es necesario conocer las siguientes relaciones:

- La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- Las fuerzas y pares aplicados en las articulaciones (o en el extremo del robot), teniendo en cuenta si se ocupan o no sistemas de transmisión de potencia mecánica como: poleas, correas, engranes, etc.
- Los parámetros dimensionales del robot, como longitud, masa e inercias de sus extremos.

2.3.1 Dinámica directa

Se tiene como datos las fuerzas y pares de torsión inmersos en el movimiento y se pretende encontrar la evolución de las coordenadas articulares (posición) del robot y sus derivadas (velocidad y aceleración), se puede encontrar el modelo dinámico directo del robot mediante la técnica analítica o experimental.

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997) es necesario conocer las siguientes relaciones:

- Las fuerzas y pares aplicados en las articulaciones (o en el extremo del robot), teniendo en cuenta si se ocupan o no sistemas de transmisión de potencia mecánica como: poleas, correas, engranes, etc.

- Los parámetros dimensionales del robot, como longitud, masa e inercias de sus extremos.

2.3.2 Dinámica inversa

Se conoce la evolución de las coordenadas articulares (posición) y sus derivadas (velocidad y aceleración) y se pretende encontrar las fuerzas y pares de torsión en las articulaciones, se puede encontrar el modelo dinámico directo del robot mediante la técnica analítica o experimental.

Según (Barrientos, Peñín, Balaguer, & Aracil, 1997) es necesario conocer las siguientes relaciones:

- La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- Los parámetros dimensionales del robot, como longitud, masa e inercias de sus extremos.

2.3.3 Ley de Euler

Se basa en el equilibrio de fuerzas de la segunda ley de Newton:

$$\sum F = m\dot{v} \quad (1)$$

Donde:

$\sum F$: Sumatoria de fuerzas.

m : Masa.

\dot{v} : Aceleración (derivada de la velocidad).

Y su equivalente en movimientos de rotación que es la ley de Euler:

$$\sum T = I\dot{\omega} + \omega \times (I\omega) \quad (2)$$

Donde:

I : Tensor de inercia.

ω : Velocidad angular.

$\dot{\omega}$: Aceleración angular.

$\sum T$: Sumatoria de momentos T

2.3.4 Ley de Lagrange

Mediante la consideración de energías tenemos:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau \quad (3)$$

$$\mathcal{L} = \kappa - U \quad (4)$$

Donde:

q_i : Coordenadas generalizadas (coordenadas articulares).

τ : Vector de fuerzas y pares de torsión aplicados en las q_i .

\mathcal{L} : Función Lagrangiana.

κ : Energía cinética.

U : Energía potencial.

2.4 Método de Algoritmos Genéticos

Los algoritmos genéticos son una rama de la inteligencia artificial y son métodos computacionales diseñados para resolver problemas complejos, robustos, de aprendizaje y de optimización.

Según (Ponce Cruz, 2010) Un algoritmo genético es una búsqueda iterativa que está inspirada en los principios de selección natural y que son aplicados a técnicas de optimización, es decir que una función objetivo alcance un máximo o un mínimo con el pasar de generaciones de padres e hijos, aproximándose cada vez más al objetivo de búsqueda, el método funciona debido a la competencia por los recursos y los rasgos favorables que se heredan.

(Holland, 1975) profesor de la Universidad de Michigan considerado el padre de los algoritmos genéticos quien a su vez fue inspirado por una previa Tesis de (Bagley, 1967), denominada “El funcionamiento de los sistemas adaptables empleando algoritmos genéticos y correlativos”. Afirma que los organismos que mejor se adaptan sobreviven y de igual manera sucede en un algoritmo genético.

Un algoritmo genético consta de operaciones básicas: selección, cruzamiento, mutación y reemplazo, que se realizan en una población inicialmente creada y codificada es decir transformada a una cadena binaria.

En cada individuo se realizan las operaciones de selección, cruzamiento, mutación y reemplazo, siendo un individuo una cadena finita binaria, es decir antes de proceder a evaluar cada una de estas operaciones es necesario codificar los individuos y convertirlos en una cadena binaria a este proceso se lo denomina codificación.

Aunque se basan en los mismos principios que en la genética, algunas de las definiciones tienen su equivalente en el ámbito computacional como se observa en la Tabla 4.

Tabla 4.

Expresiones utilizadas en genética y su equivalente en algoritmos genéticos

<i>Evolución natural</i>	<i>Algoritmo genético</i>
Genotipo	Código de cadena
Fenotipo	Punto sin codificar
Cromosoma	Cadena
Gen	Posición de cadena
Alelo	Valor en una posición determinada
Función de aptitud o aptitud	Valor de la función objetivo

Fuente: (Ponce Cruz, 2010, pág. 290)

Un genotipo es la estructura o código genético de un organismo, es equivalente a un código de cadena que son los valores binarios que posee una cadena, a su vez una cadena equivale a un cromosoma que es una molécula de ADN que posee información única de cada individuo.

Fenotipo se refiere a las características físicas y de conductas de un organismo, gen es un carácter heredado y corresponde a una posición de la cadena binaria, un alelo es el valor que tiene el gen y es análogo al valor en una posición determinada de la cadena.

La función de aptitud cuantifica la optimalidad de un individuo y corresponde al valor que toma la función objetivo, es más óptimo mientras más se aproxime a un máximo o a un mínimo.

Selección: Son algoritmos que clasifican a los individuos en los aptos para la reproducción y en los no aptos para la reproducción utilizando por lo general una distribución de probabilidad, donde se evalúa la aptitud (valor que toma la función objetivo).

Cruzamiento: Dos progenitores previamente seleccionados con características aptas se fusionan produciendo así hijos o descendientes que heredan las mejores características de sus padres.

La forma de cruzamiento más simple es el operador de cruce basado en un punto en el cual se divide la cadena cinemática de los padres en un punto y se cruzan las subcadenas resultantes como se puede apreciar en la Figura 21.

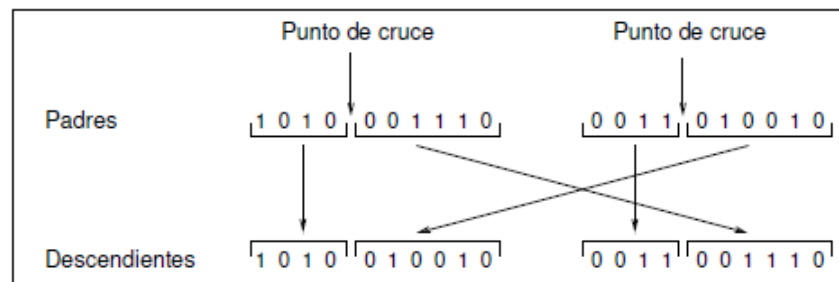


Figura 21. Cruzamiento basado en un punto.

Fuente: (Moujahid, Inza, & Larrañaga, 2014)

Además existen otros métodos de cruzamiento como se menciona en (Ponce Cruz, 2010, pág. 294) son:

- Cruzamiento de n puntos: en donde se elige varios puntos de ruptura al azar, los puntos de ruptura no pueden variar.
- Cruzamiento segmentado: puede variar el número de puntos de ruptura.
- Cruzamiento aleatorio: Se realiza algoritmos de permutación y se aplica el cruzamiento de n puntos.
- Cruzamiento uniforme: Se decide al azar las posiciones intercambiadas.

Mutación: En algoritmos genéticos se producen cambios aleatorios en los valores de una determinada posición de la cadena de bits o su equivalente término genético alelo en ciertos individuos como por ejemplo en la Figura 22, esta mutación puede ser favorable o no, pero su importancia radica en introducir ciertas características de las que carecen sus padres.

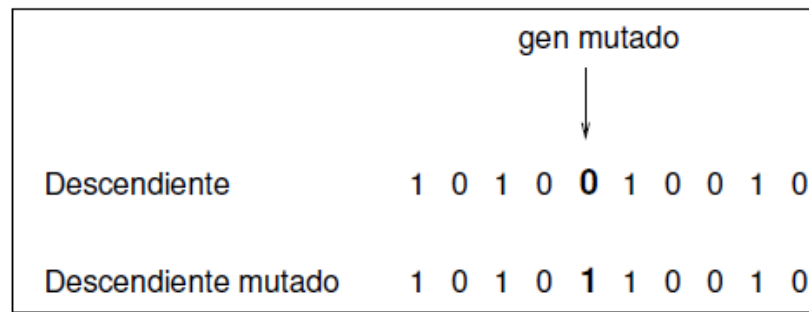


Figura 22. Operador de Mutación.

Fuente: (Moujahid, Inza, & Larrañaga, 2014)

Reemplazo: Consiste en cambiar a un individuo por otro después de la operación de cruzamiento y obviar a los individuos no seleccionados para la operación de cruzamiento.

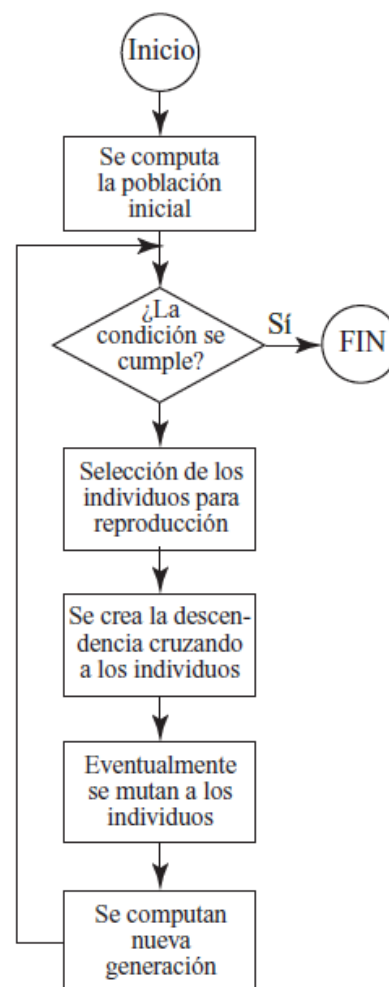


Figura 23. Algoritmo genético básico.

Fuente: (Ponce Cruz, 2010, pág. 313)

Se procederá a realizar las operaciones de una manera iterativa hasta que se cumpla una convergencia o condición de parada para visualizar mejor el método se puede apreciar el diagrama básico de un algoritmo genético que se muestra en la Figura 23, aunque cabe mencionar que hay varias maneras de mejorar un algoritmo genético como por ejemplo por el método de ruleta o generar nuevas poblaciones a partir de las operaciones de cruzamiento y mutación.

2.4.1 Utilización de algoritmos genéticos para el análisis cinemático y dinámico en robótica industrial.

El uso de algoritmos genéticos para la solución de la cinemática inversa, así como planeación de trayectorias y análisis de parámetros dinámicos se ha visto en trabajos de investigación y tesis como:

En la universidad nacional de Colombia (Giraldo, Delgado, & Castellanos, 2006, págs. 29-34), investigan la cinemática inversa de un Brazo robot utilizando algoritmos genéticos.

En este artículo se describe cómo se puede resolver la cinemática inversa partiendo de la solución de cinemática directa por Denavit-Hartenberg, luego se introduce un algoritmo genético básico, variando la función de aptitud.

El robot que utilizan para este experimento es el Scorbobot VR plus, con cinco grados de libertad que posee la Universidad Nacional de Colombia en Manizales, los resultados obtenidos manifiestan que después de realizar el marco experimental se demostró que el algoritmo es efectivo y eficiente para resolver el problema, no se han puesto restricciones al algoritmo genético lo que hace que tenga varias soluciones al encontrar la cinemática inversa.

En el instituto politécnico nacional de México (Becerra Pedraza, 2008), realiza un estudio acerca de robots modulares auto configurables, el robot simulado es un manipulador planar con n grados de libertad, su objetivo es el auto ensamble de sus eslabones y para ello utiliza el método de algoritmos genéticos para resolver la cinemática inversa y planifica la trayectoria, este trabajo demuestra que el aumento de grados de libertad no son un problema para el método de algoritmos genéticos, además

se utiliza el mismo algoritmo genético sin importar los grados de libertad que se aumenten como reto el autor propone realizar físicamente el robot.

En la Universidad Carlos III de Madrid (Bravo Sancho & González Uzcátegui, 2012), analizan técnicas evolutivas muy relacionadas con los algoritmos genéticos para resolver el problema cinemático inverso en robots de 3 y 6 grados de libertad.

La investigación parte de la solución de la cinemática directa de la misma manera que los trabajos antes mencionados, aquí se varían los parámetros del algoritmo evolutivo como son tamaño de población, factor de escala de mutación entre otros.

Además introduce al algoritmo evolutivo el mecanismo discarding que consiste en acelerar la convergencia del algoritmo mediante el reemplazo de los peores individuos por nuevos más cercano al objetivo.

En la Escuela superior de Ingeniería Mecánica y Eléctrica del Instituto politécnico Nacional en México (Ramírez Gordillo, 2010), investiga la planeación de trayectorias en sistemas manipuladores robóticos múltiples.

Para este estudio se emplea cuaternios para resolver la cinemática directa y a continuación se plantea varios métodos computacionales entre ellos algoritmos genéticos, además se analiza el comportamiento dinámico del robot obteniendo resultados muy favorables y positivos.

El método de algoritmos genéticos gana importancia en el ámbito de la robótica y se ha obtenido buenos resultados, en esta investigación a diferencia de los trabajos mencionados, se diseñara y construirá un robot de siete grados de libertad debido a sus amplias aplicaciones y flexibilidad, se obtendrá la solución de su posicionamiento y orientación que le permitan realizar aplicaciones de manipulación flexibles.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN MECATRÓNICA

3.1 Diseño mecánico

3.1.1 Reconocimiento de la necesidad

Se requiere diseñar y construir un robot antropomórfico manipulador que tenga siete grados de libertad de características didácticas para realizar el análisis cinemático y dinámico basándose en algoritmos genéticos y que pueda ser capaz de realizar varias tareas programadas mediante planeación de trayectorias.

Se pretende además implementar un efector final tipo pinza que permita operaciones de manipulación de objetos.

3.1.2 Definición del problema

Los requisitos de diseño mecánico del robot antropomórfico son:

- Debe tener 7 grados de libertad.
- La distribución de sus ejes será como se muestra en la Figura 1.
- Capacidad de carga pequeña porque es solo de fines didácticos.
- Debe utilizar actuadores eléctricos.
- Su alcance máximo debe ser de mediano (40 a 60) cm
- Debe realizar actividades similares a un brazo humano
- El efector final será una pinza que permita manipular objetos

3.1.3 Síntesis

3.1.3.1 Geometría del robot

Para conocer una aproximación de la forma del robot se realiza un diseño geométrico de las piezas que conforman cada eslabón.

Las partes que conforman el robot antropomórfico de siete grados de libertad son:

- Base (Figura 24): parte inferior que anclará al robot al sitio de trabajo.
- Eslabón 1 (Figura 25): Eslabón rotacional que se conecta con el eslabón base.
- Eslabón 2 (Figura 26): Eslabón rotacional que se conecta con el eslabón 1.
- Eslabón 3 (Figura 27): Eslabón rotacional que se conecta con el eslabón 2.
- Eslabón 4 (Figura 28): Eslabón rotacional que se conecta con el eslabón 3.

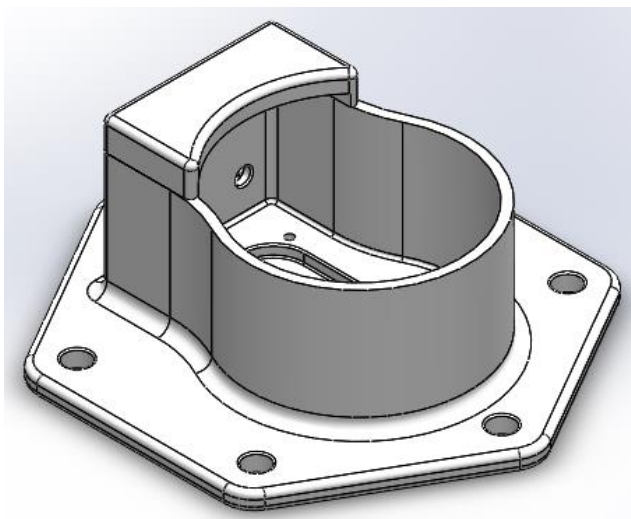


Figura 24. Base del robot

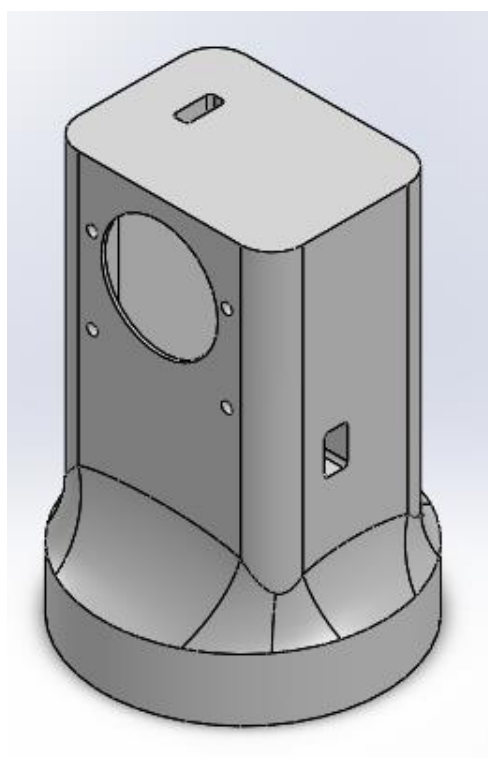


Figura 25. Eslabón 1

- Eslabón 5 (Figura 29): Eslabón rotacional que se conecta con el eslabón 4.
- Eslabón 6 (Figura 30): Eslabón rotacional que se conecta con el eslabón 5.
- Soporte Gripper (Figura 31): Eslabón rotacional que une al robot con el efector final.
- Efector final tipo Gripper (Figura 32): Efector final que funciona como una pinza, destinada a manipulación de objetos.

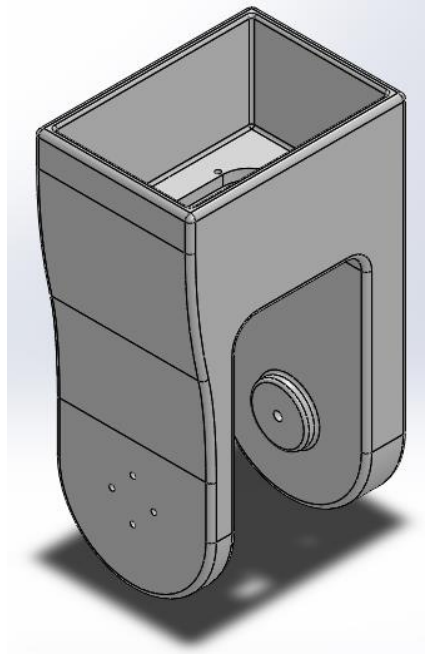


Figura 26. Eslabón 2

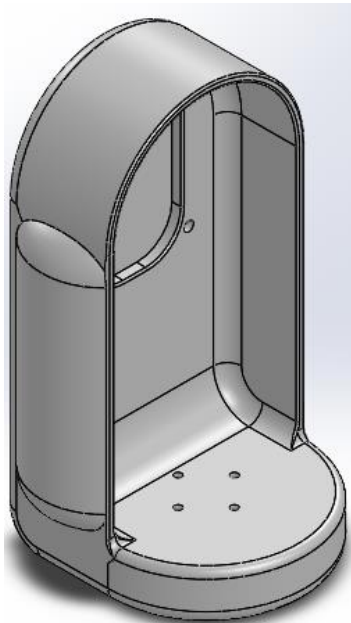


Figura 27. Eslabón 3

La geometría final del robot se puede observar en la Figura 33, pretende ser agradable a la vista y sobretodo funcional, pensando en un correcto desempeño del robot dándole al robot una buena flexibilidad.

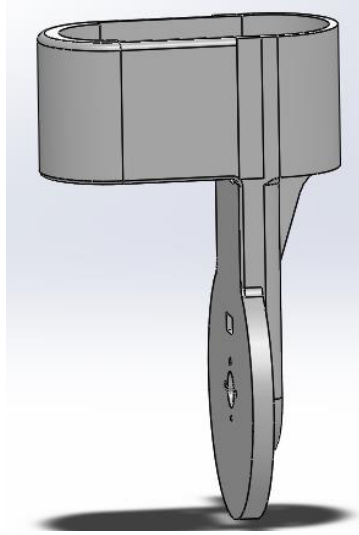


Figura 28. Eslabón 4

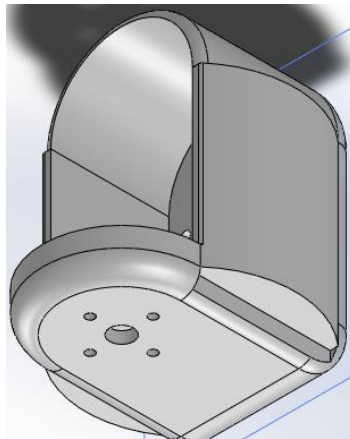


Figura 29. Eslabón 5

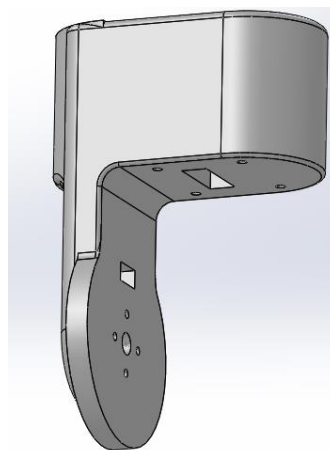


Figura 30. Eslabón 6

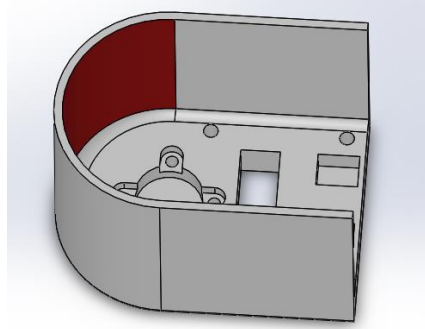


Figura 31. Soporte de gripper

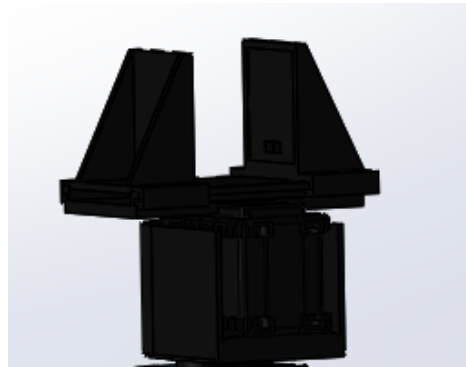


Figura 32. Efecto final tipo gripper.

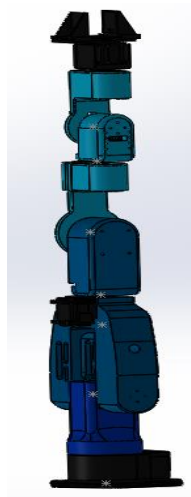


Figura 33. Geometría del robot manipulador.

3.1.4 Análisis y Optimización

3.1.4.1 Selección de actuadores.

En la Tabla 5 se muestra algunas diferencias entre los motores a pasos y los servomotores, existen servomotores accionados con corriente alterna (AC) y con corriente directa (DC).

Se tomaron en consideración las características de los servomotores DC. Elegimos los servomotores sobre los motores a pasos por su menor tamaño, facilidad de implementación y la opción de conseguir servomotores con sensores de torque, corriente y posición incluidos.

Tabla 5.

Diferencias entre motores a pasos y servomotores DC

<i>Motores a pasos</i>	<i>Servomotores DC</i>
Control en lazo abierto	Control en lazo cerrado (potenciómetros, codificadores, generadores tacométricos, etc.)
Pobre relación potencia-volumen, son más grandes.	Buena relación potencia-volumen.
Robustos, envejecen muy lentamente.	Presentan envejecimiento de las escobillas.
Buenas características de bloqueo.	Para el bloqueo necesitan frenos extra (frenos mecánicos).
Tienen bajo costo, baja inercia, escaso peso y funcionamiento silencioso.	Costo más elevado que los motores a pasos.
Está constituido de un estator y un rotor.	Está constituido por un motor DC, una caja reductora y un circuito de control.
Para dar pasos necesita una serie de pulsos en sus bobinados.	Para dar giros requiere una señal de pulsos.
Cada paso hace girar al rotor un ángulo determinado.	Cada ancho de pulsos hace girar al motor un ángulo determinado.
Tiene una resolución 1.8° por paso con un driver de micro pasos llega hasta 0.0144° por paso.	Tienen una resolución de 0.72° por ancho de pulso, hasta 0.045° por ancho de pulso.
Pueden dar varias vueltas.	Algunos pueden girar 180° , 360° y 3 vueltas.
Posee un par de mantenimiento sin excitación en la configuración de imanes permanentes y en la híbrida (no en reluctancia variable).	No posee un par de mantenimiento sin excitación, requiere frenos magnéticos para mantener la posición del motor.

Fuente: (Conti, 2005)

Los servomotores dynamixel desarrollados por la empresa coreana Robotis se ilustran en la Figura 34, en la Tabla 6 se muestran algunas características de los servos AX-12A, MX-28T y MX-64T.

Los más sencillos son los AX-12A con un par de funcionamiento bajo y velocidad baja. Los MX-28T y los MX-64T poseen comunicación TTL al igual que los AX-12A.

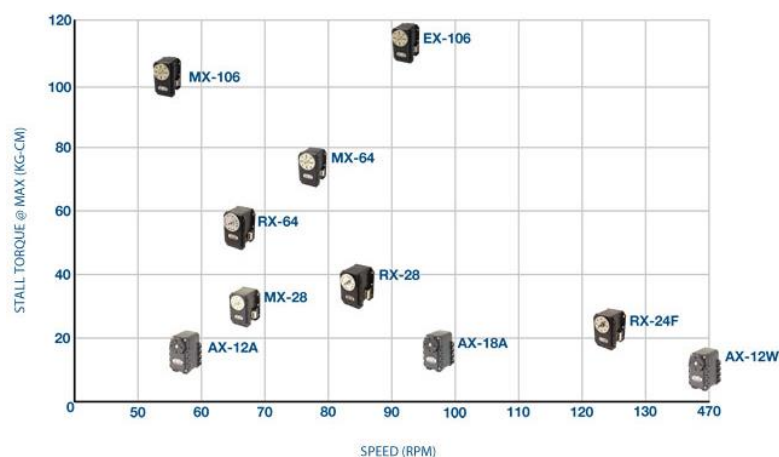


Figura34. Comparación de motores dynamixel torque vs velocidad

Fuente: (Trossen Robotis, 2015)

Tabla 6.

Comparación motores dynamixel.

Especificaciones	AX-12A	MX-28T/MX-28R	MX-64T / MX-64R
Stall torque @ Max	1,5 Nm / 16,5 kg-cm	3.1Nm (31.6 kg-cm)	7.3Nm / 74.4 kg-cm
Voltaje			
Velocidad (RPM)	59	97	78
Tensión de funcionamiento	9 ~ 12V (Rec. Voltaje 11.1V)	10 ~ 14.8V (Rec. Voltaje 12V)	10 ~ 14.8V (Rec. Voltaje 12V)
Puesto de Consumo de corriente	1.5A	1.7A	5.2A
Dimensiones	32mm x 50mm x 40mm	35.6mm x 50.6mm x 35.5mm	40.2mm x 61.1mm x 41mm
Peso	54.6g	72g	126g
Resolución	0.29 °	0,088 °	0,088 °
Ángulo de funcionamiento	300 °	360 °	360 °
Reducción engranes	254: 1	193: 1	200: 1
Tren de engranajes de materiales	Plástico	Acero endurecido	Acero endurecido
CPU a bordo	ATMega8 (ATMEGA8-16AU @ 16MHZ, 8 bits)	Cortex M3 (STM32F103C8 @ 72MHz, 32 Bit)	Cortex M3 (STM32F103C8 @ 72MHz, 32 Bit)
Sensor de Posición	Potenciómetro	Codificador magnético	Codificador magnético
Protocolo de Comunicación	TTL	TTL / RS-485	TTL / RS-485
Velocidad de comunicación	1Mbps	3Mbps	3Mbps
Cumplimiento / PID	Cumplimiento	PID	PID

Fuente: (Trossen Robotis, 2015)

3.1.4.2 Selección de materiales.

Plásticos

Según (Askeland & Phulé, 2004), los plásticos son materiales compuestos de polímeros que son la unión de química de moléculas conocidas como monómeros (una sola unidad) o oligómeros (Varias unidades), y forman largas cadenas, este tipo de material es ampliamente utilizado en juguetería, electrodomésticos, elementos estructurales, pinturas, adhesivos, etc. Se pueden clasificar en tres tipos:

Termoplásticos: O también conocidos como material de ingeniería por sus aplicaciones orientadas a chasis, etc. Están formados por cadenas largas y se puede moldear al aplicar calor

Termoestables: Son similares a un manojo de hilos por su unión en cadenas ramificadas o lineales, son más resistentes pero más frágiles y difíciles de reutilizar que los termoplásticos.

Elastómeros: Conocidos como hules con una alta deformación elástica.

ABS (Acrilonitrilo–butadieno–estireno)

El material para elaborar los eslabones de la cadena cinemática es un termoplástico denominado ABS (Acrilonitrilo–butadieno–estireno), es el termoplástico técnico más usado debido a que posee buenas propiedades mecánicas, además se puede realizar las piezas por impresión 3D en este material.

Este proceso resulta muy bueno para piezas de formas relativamente complejas como las mencionadas en la síntesis, se logra excelentes tolerancias, es menos costoso en piezas de geometría compleja, y se puede lograr un modelo más agradable estéticamente, las propiedades del termoplástico ABS se observan en la Tabla 5:

Tabla 5.

Propiedades del termoplástico ABS

<i>Nombre</i>	<i>Su</i> <i>Kpsi</i>	<i>E</i> <i>Mpsi</i>	<i>Dureza</i> <i>Rockwell</i>	<i>Elongación</i> <i>%</i>	<i>Estabilidad</i> <i>dimensional</i>
Grupo					
ABS	2-8	0.10-0.37	60-110R	3-50	Buena

Fuente: (Budynas & Nisbett, 2008)

Posición crítica del robot.

Se trata de la posición donde los actuadores requieren el mayor torque para mover a cada eslabón. En un robot manipulador la posición crítica más utilizada para el diseño de análisis de los eslabones es la aproximación a una viga en voladizo.

Esta aproximación consiste en empotrar el eslabón en su extremo, considerar el peso del eslabón en conjunto con el peso del actuador, y en extremo del robot aplicar una carga con el peso de los eslabones siguientes y en algunos casos para obtener resultados más aproximados se pone un momento producido por el motor al final del eslabón.

Para este análisis se utilizará un software CAD que realiza el análisis mediante el método de elementos finitos.

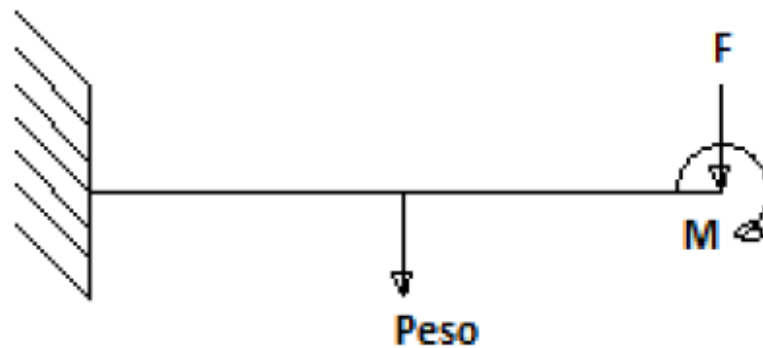


Figura 31. Aproximación de los eslabones, viga en voladizo.

Fuente: (Vaca Jiménez, 2012, pág. 51)

3.1.4.3 Análisis de carga y esfuerzo.

El análisis será realizado con el peso y las propiedades de cada elemento que conforma el robot especificadas en la Tabla 7.

Se pretende diseñar el robot para que pueda manipular objetos de una carga máxima de 200 gramos, es una carga liviana pero es suficiente para un prototipo didáctico, cabe mencionar que el robot no se utilizará en un ámbito industrial y el principal objetivo es posicionar y orientar el robot.

La capacidad de carga no es prioridad pero el robot debe ser diseñado para soportar una carga, si bien ya nos propusimos un límite de 200 gramos, para asegurar el diseño

hemos decidido diseñar los eslabones para que no fallen con una carga aplicada de 1Kg en el efector final.

Tabla 7.

Peso y material de los elementos del robot manipulador

<i>Elementos</i>	<i>Peso (g)</i>	<i>Material</i>
Soporte	7536.86	Acero AISI 1010
Base	113.34	ABS
Motor 1 (Dynamixel Mx-64T)	126.00	Varios
Eslabón 1	194.94	ABS
Motor 2 (Dynamixel Mx-64T)	126.00	Varios
Eslabón 2	503.71	ABS
Motor 3 (Dynamixel Mx-64T)	126.00	Varios
Eslabón 3	105.14	ABS
Motor 4 (Dynamixel Mx-28T)	72.00	Varios
Eslabón 4	83.56	ABS
Motor 5 (Dynamixel Ax-12A)	54.60	Varios
Eslabón 5	62.53	ABS
Motor 6 (Dynamixel Ax-12A)	54.60	Varios
Eslabón 6	65.54	ABS
Motor 7(Dynamixel Ax-12A)	54.60	Varios
Soporte Motor 8	17.44	ABS
Motor 8(Dynamixel Ax-12A)	54.60	Varios
Gripper	43.23g	ABS

En el programa de diseño CAD, usamos el material ABS con las propiedades que se muestran en la Figura 32, estas propiedades son las que se obtienen mediante fabricación por impresión 3D.

Tipo de modelo:	Isotrópico elástico lineal	▼
Unidades:	SI - N/mm ² (MPa)	▼
Categoría:	Plástico	
Nombre:	Absa	
Criterio de fallos predeterminado:	Tensión de von Mises máx.	▼
Descripción:	-impresion 3d	
Origen:		
Sostenibilidad:	No definido	Seleccionar...

Propiedad	Valor	Unidades
Módulo elástico	2000	N/mm ²
Coefficiente de Poisson	0.394	N/D
Módulo cortante	318.9	N/mm ²
Densidad de masa	1020	kg/m ³
Límite de tracción	30	N/mm ²
Límite de compresión	15	N/mm ²
Límite elástico	30	N/mm ²
Coefficiente de expansión térmica		/K
Conductividad térmica	0.2256	W/(m·K)
Calor específico	1386	J/(kg·K)
Cociente de amortiguamiento del material		N/D

Figura 32. Propiedades del material ABS.

El soporte del robot está construido con el acero AISI 1010 cuyas propiedades mecánicas y térmicas se ilustran en la Figura 33.

Tipo de modelo:	Isotrópico elástico lineal	▼
Unidades:	SI - N/mm ² (MPa)	▼
Categoría:	Acero	
Nombre:	AISI 1010 Barra de acero laminac	
Criterio de fallos predeterminado:	Tensión de von Mises máx.	▼
Descripción:		
Origen:		
Sostenibilidad:	Definido	

Propiedad	Valor	Unidades
Módulo elástico	200000	N/mm ²
Coefficiente de Poisson	0.29	N/D
Módulo cortante	80000	N/mm ²
Densidad de masa	7870	kg/m ³
Límite de tracción	325	N/mm ²
Límite de compresión		N/mm ²
Límite elástico	180	N/mm ²
Coefficiente de expansión térmica	1.22e-005	/K
Conductividad térmica	51.9	W/(m·K)
Calor específico	448	J/(kg·K)
Cociente de amortiguamiento del material		N/D

Figura 33. Propiedades del soporte o mesa del robot.

El respaldo realizado para el eslabón número dos tiene como material el acero AISI 321, un acero inoxidable cuyas propiedades se encuentran representadas en la Figura 34.

Tipo de modelo:	Plasticidad - von Mises	
Unidades:	SI - N/mm ² (MPa)	
Categoría:	Acero	Crear curva Esfuerzo-Tensión
Nombre:	AISI 321 Acero inoxidable recoci	
Criterio de fallos predeterminado:	Tensión de von Mises máx.	
Descripción:		
Origen:		
Sostenibilidad:	Definido	

Propiedad	Valor	Unidades
Módulo elástico	193000	N/mm ²
Coefficiente de Poisson	0.27	N/D
Límite de tracción	620	N/mm ²
Límite elástico	234.42	N/mm ²
Módulo tangente		N/mm ²
Coefficiente de expansión térmica	1.7e-005	/K
Densidad de masa	8000	kg/m ³
Factor de endurecimiento	0.85	N/D

Figura 34. Propiedades del respaldo del Eslabón 2.

Soporte de la base: Mediante software CAD se diseña el soporte de la Figura 35, en la simulación se consideró que está sujeto por cuatro tornillos ubicados cercanos a las esquinas del mismo como se ilustra en la figura 36. El peso del robot a su máxima capacidad es de 4.83 Kgf y también produce un torque de 6.7 Nm como se representa en la figura 37. Considerando todos los parámetros descritos se realiza el análisis estático y los resultados se indican en las Figuras 38, 39, 40 y 41.

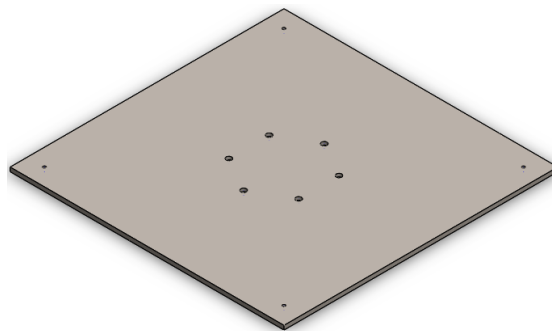


Figura 35. Soporte de la base del robot.

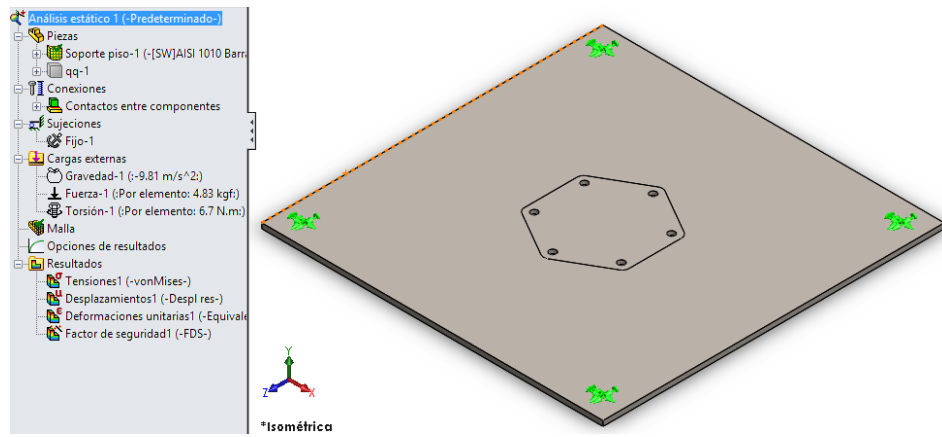


Figura 36. Sujeciones del soporte de la base para análisis de esfuerzos.

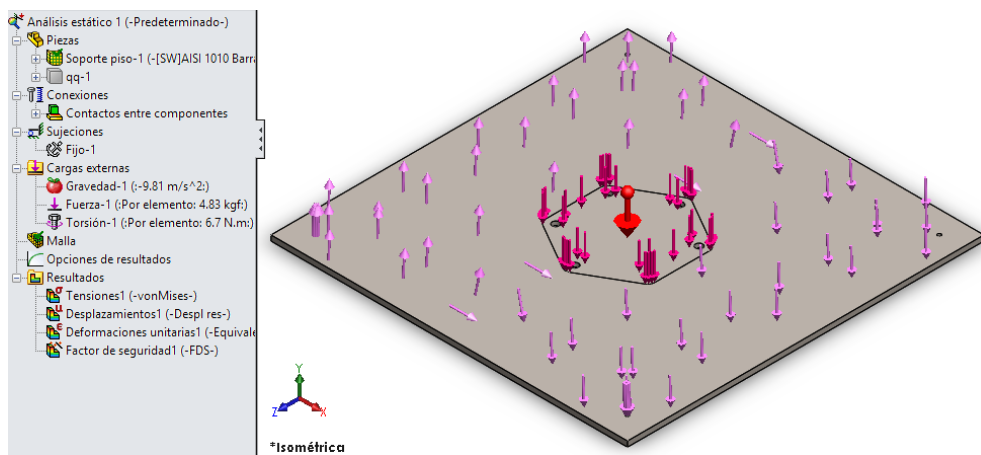


Figura 37. Cargas aplicadas a soporte de la base para análisis de esfuerzos.

Nombre del modelo: Prueba soporte
 Nombre de estudio: Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Análisis estático tensión nodal Tensiones1
 Escala de deformación: 846,586

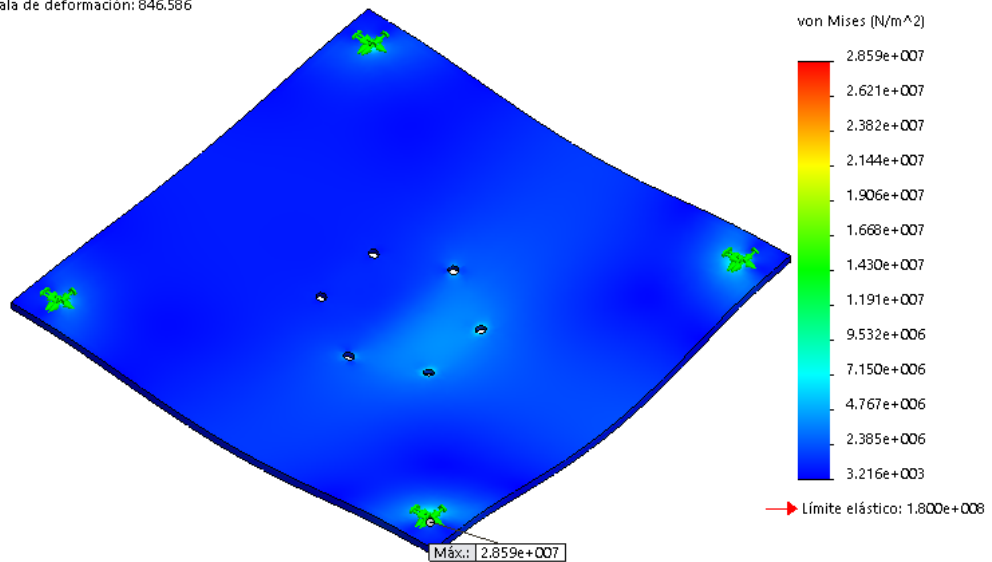


Figura 38. Tensiones von Mises de la placa de soporte para la base.

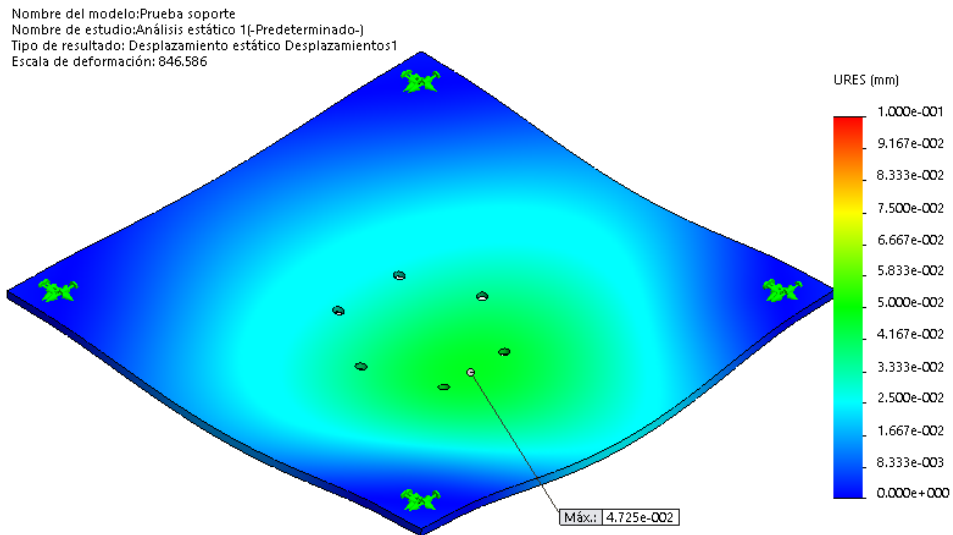


Figura 39. Desplazamientos de la placa de soporte para la base.

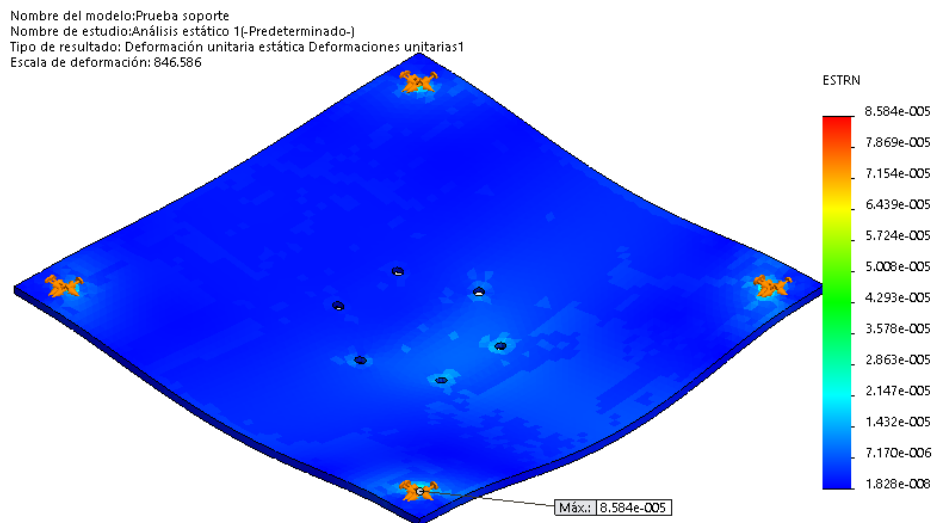


Figura 40. Deformación unitaria de la placa de soporte para la base.

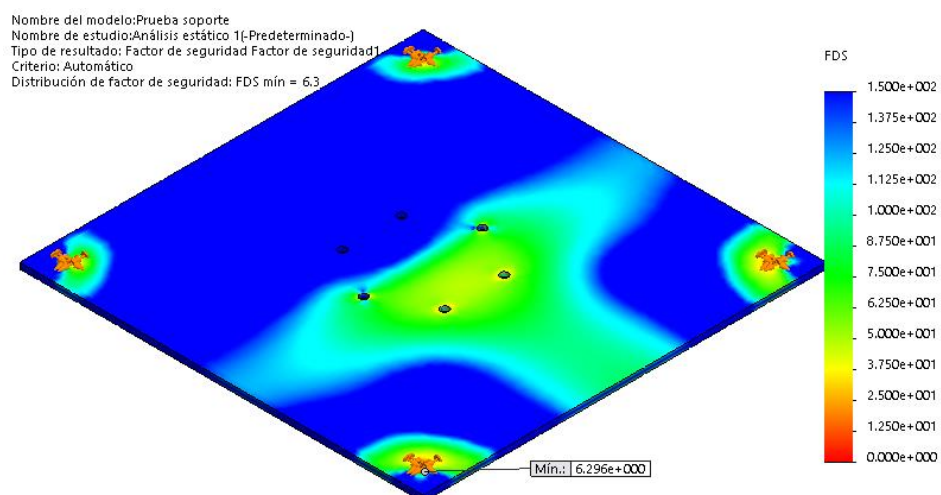


Figura 41. Factor de seguridad de la placa de soporte para la base.

Base: El análisis estático de este elemento se lo realiza considerando que está construido con ABS, sujeto mediante pernos en su parte inferior y apoyado sobre un soporte, la representación de esta sujeción se ilustra en la Figura 43. Se aplica una carga de 2.6 Kgf sobre el eje del motor y se realiza un torque de 6N.m sobre la brida del motor como se indica en la Figura 44. Los resultados del análisis estático se muestran en las Figuras 45, 46, 47 y 48.

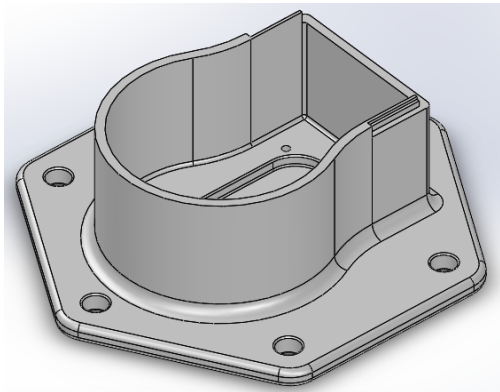


Figura 42. Base.

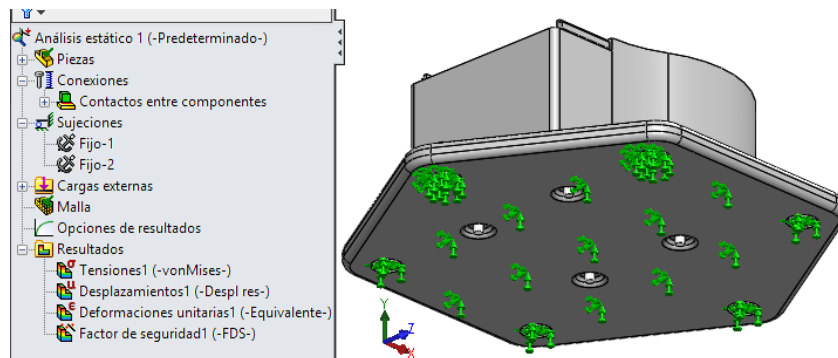


Figura 43. Sujeciones de la base para análisis de esfuerzos.

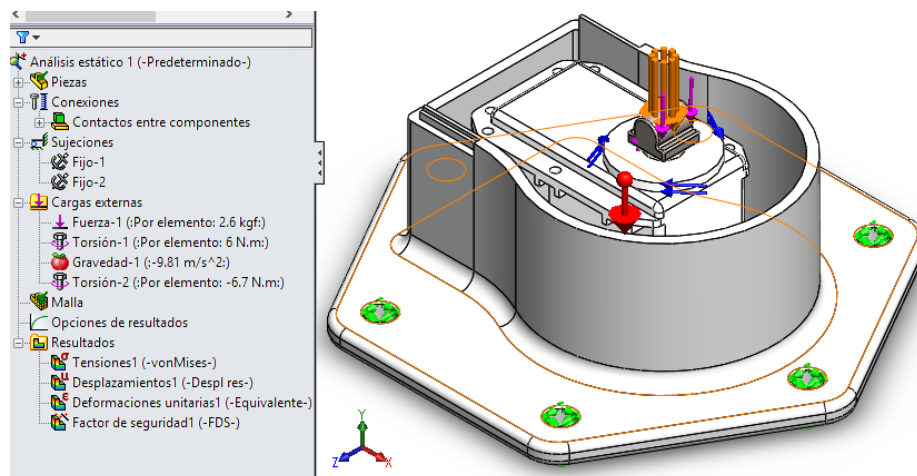


Figura 44. Cargas aplicadas a la base para análisis de esfuerzos.

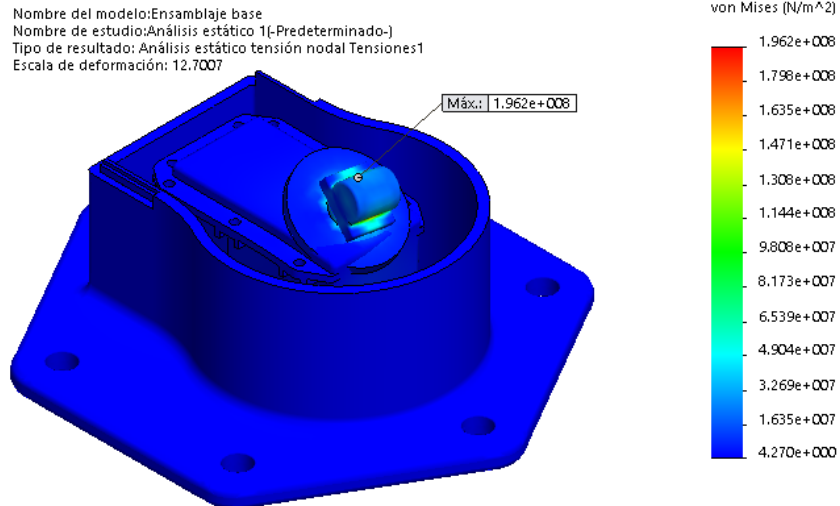


Figura 45. Tensiones von Mises de la base.

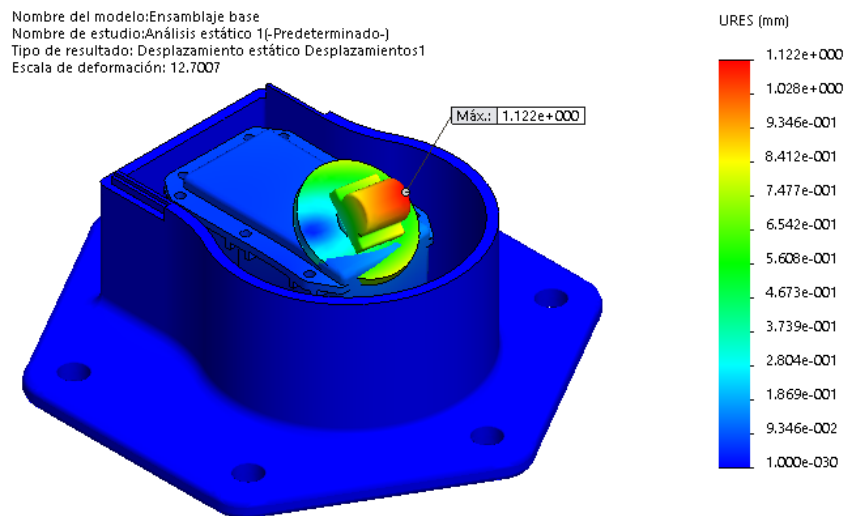


Figura 46. Desplazamiento de la base.

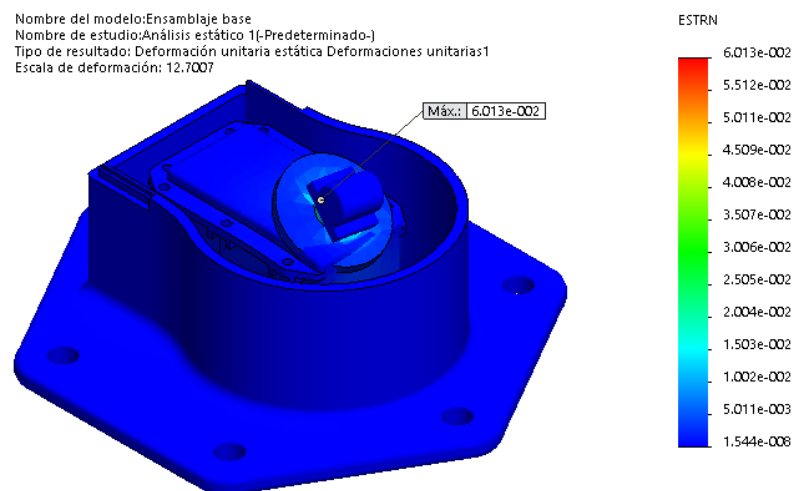


Figura 47. Deformación unitaria de la base.

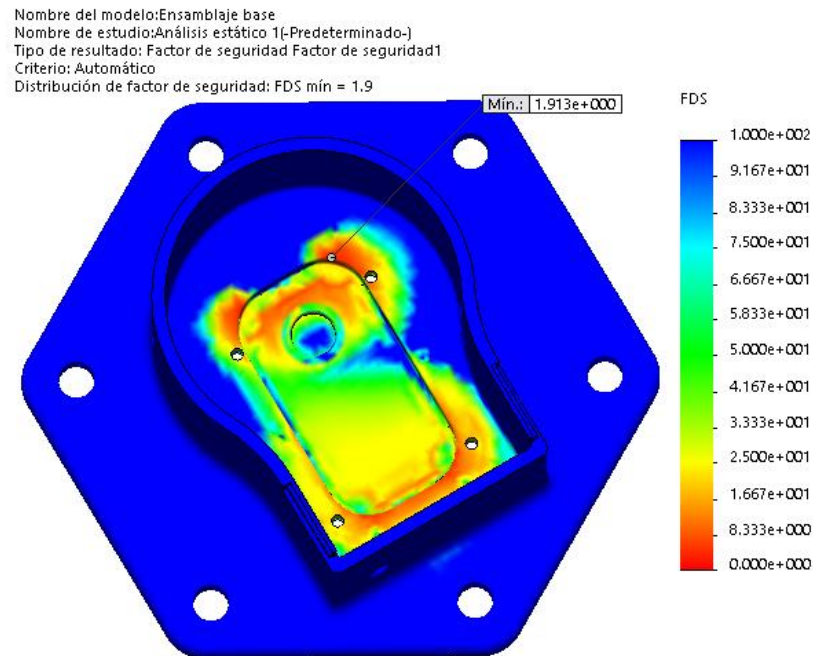


Figura 48. Factor de seguridad en la base

Eslabón 1: El termoplástico ABS es el material con el que se construyó el eslabón uno, está sujeto en la parte inferior por un pernos al motor anterior, soporta una carga de 2.4 kg aplicada sobre el eje del motor correspondiente al peso de los eslabones anteriores y un torque de 6 Nm en la brida del motor así como de 6N.m producidos por el motor sobre la base del eslabón como se ilustra en la Figura 49, con estas propiedades, sujeciones y cargas se realiza un estudio estático del eslabón dos, los resultados del análisis se muestran en las Figuras 50, 51, 52 y 53.

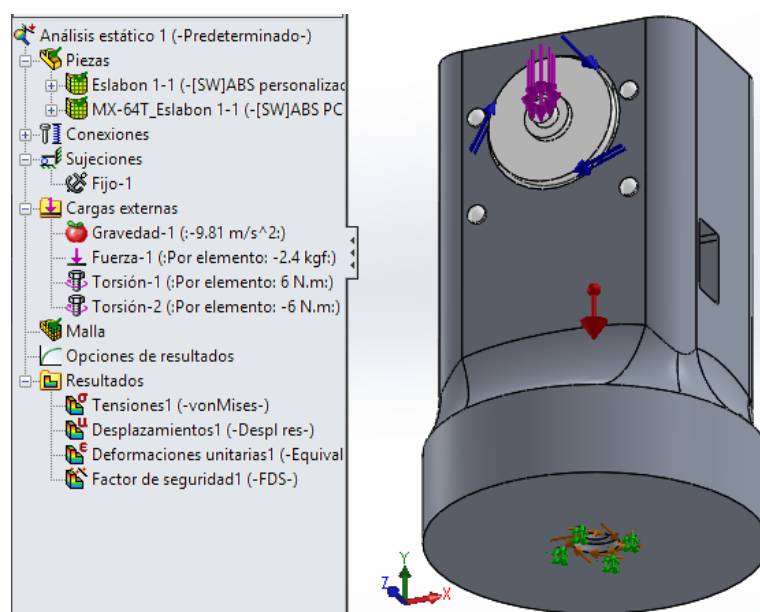


Figura 49. Cargas aplicadas al eslabón 1 para el análisis de esfuerzos

Nombre del modelo:Ensamblaje eslabon 1
 Nombre de estudio:Análisis estático 1(-Predeterminado-)
 Tipo de resultado:Análisis estático tensión nodal Tensiones1
 Escala de deformación: 25.2346

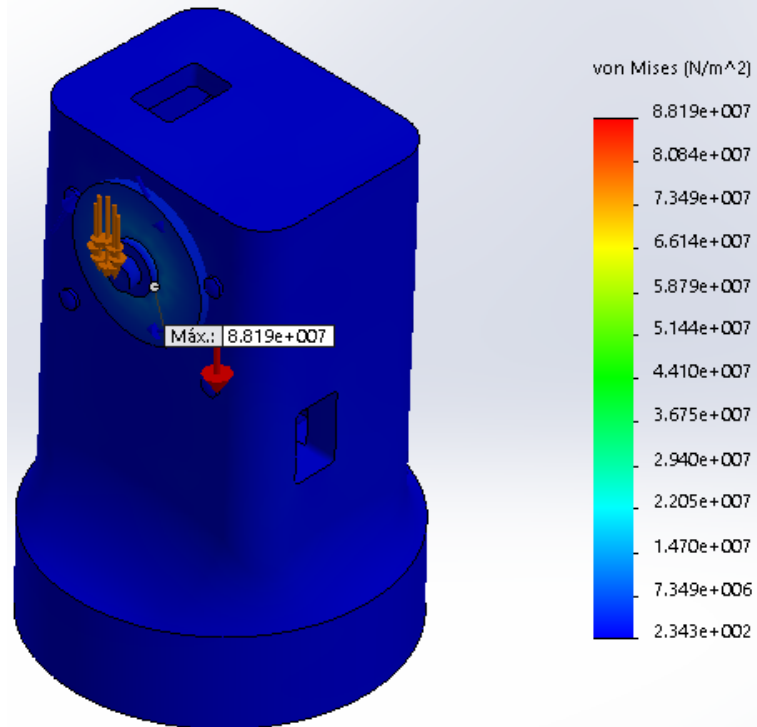


Figura 50. Tensiones von Mises del eslabón 1.

Nombre del modelo:Ensamblaje eslabon 1
 Nombre de estudio:Análisis estático 1(-Predeterminado-)
 Tipo de resultado:Desplazamiento estático Desplazamientos1
 Escala de deformación: 25.2346

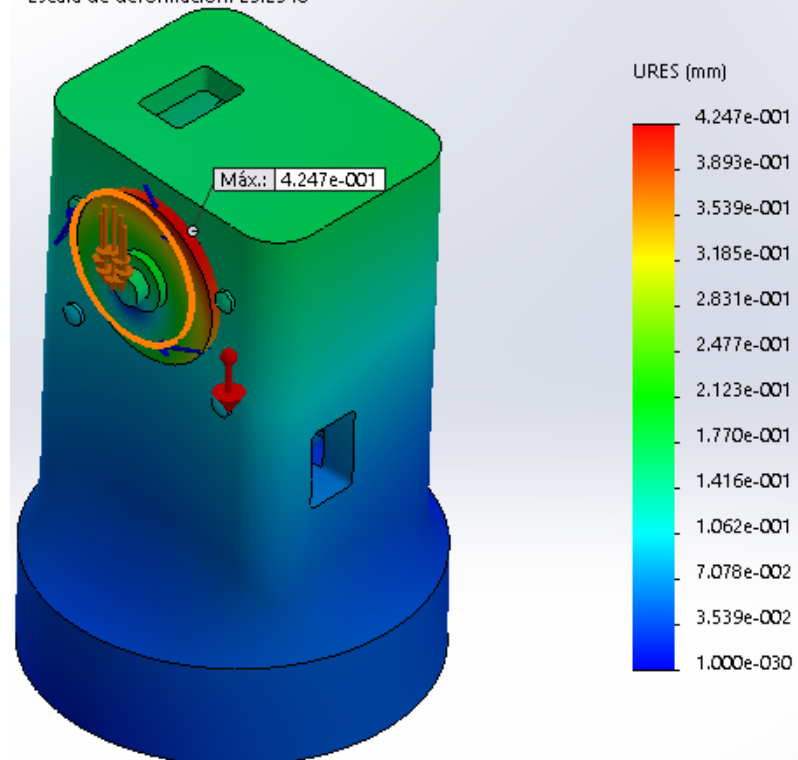


Figura 51. Desplazamiento del eslabón 1.

Nombre del modelo:Ensamblaje eslabon 1
 Nombre de estudio:Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Deformación unitaria estática Deformaciones unitarias1
 Escala de deformación: 25.2346

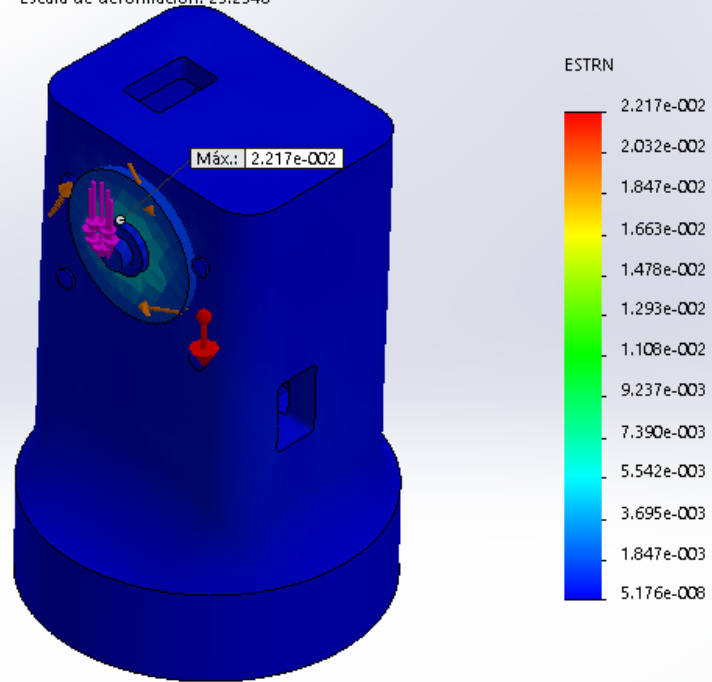


Figura 52. Deformación unitaria del eslabón 1.

Nombre del modelo:Ensamblaje eslabon 1
 Nombre de estudio:Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Factor de seguridad Factor de seguridad1
 Criterio: Automático
 Distribución de factor de seguridad: FDS mín = 2.6

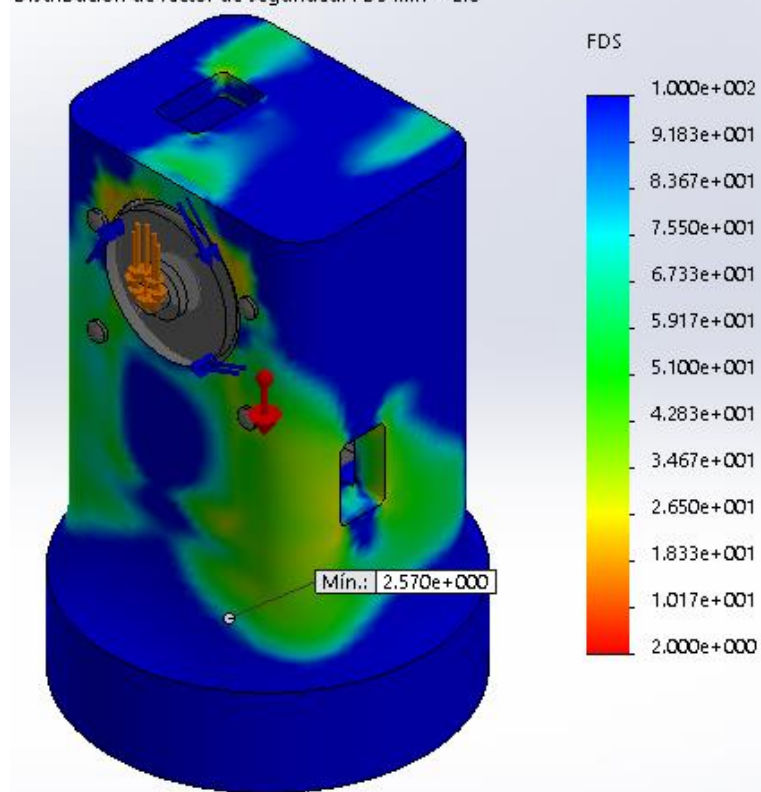


Figura 53. Factor de seguridad del eslabón 1.

Eslabón 2: El siguiente análisis se realizó considerando que el eslabón fue fabricado en ABS y cuenta con un refuerzo hecho en acero inoxidable recocido; se sujeta en un extremo del mismo mediante tornillos como se muestra en la Figura 54.

La carga de 1.647 Kgf sobre el eje del motor correspondiente a los eslabones que siguen en la cadena cinemática, esta carga también produce un torque de 3.4 Nm que está representada en la brida del motor como se representa en la Figura 55.

El motor anterior que se encuentra en el eslabón dos y el que se sujeta al eslabón tres son del tipo MX-64T, cada uno produce un torque máximo de 6 Nm; mediante un software CAD se realiza un análisis estático y los resultados se los visualiza en las Figuras 56, 57, 58 y 59.

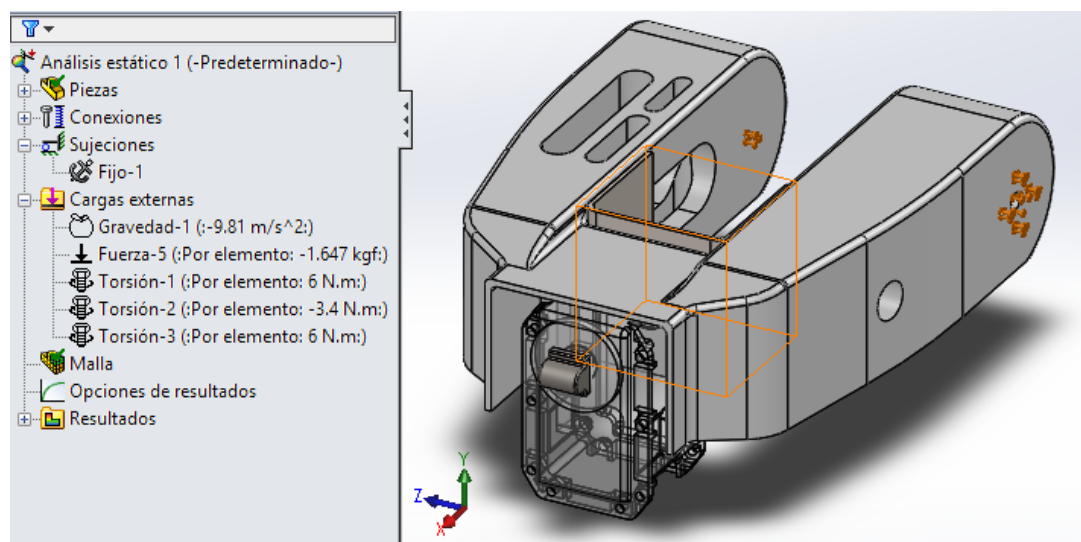


Figura 54. Sujeciones aplicadas al eslabón 2 para el análisis de esfuerzos.

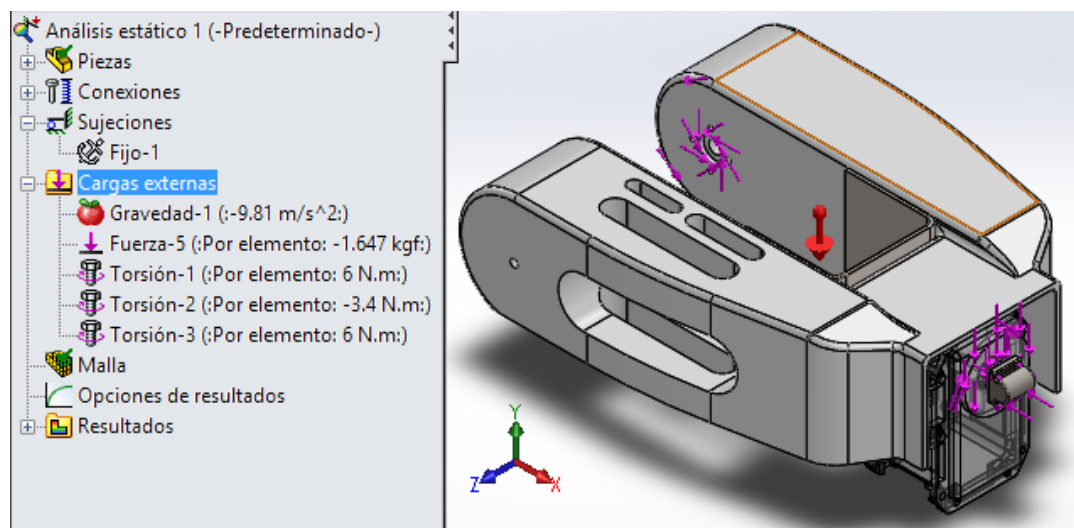


Figura 55. Cargas aplicadas al eslabón 2 para análisis de esfuerzos.

Nombre del modelo: E Eslabon 2 final
 Nombre de estudio: Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Análisis estático tensión nodal Tensiones1
 Escala de deformación: 9,78598

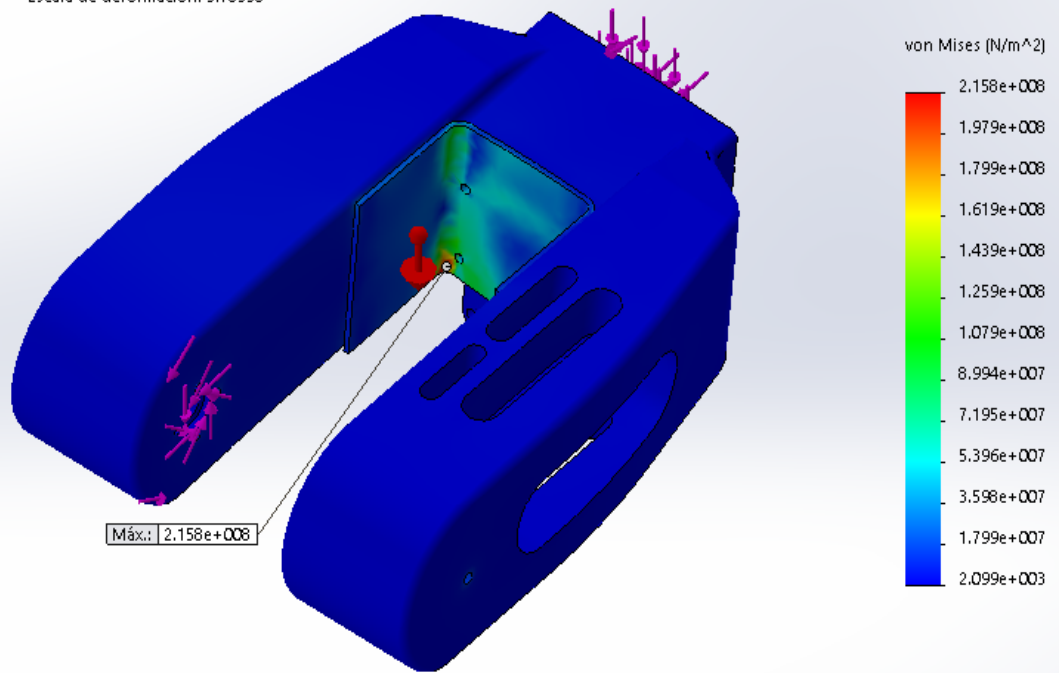


Figura 56. Tensiones von Mises en el eslabón 2.

Nombre del modelo: E Eslabon 2 final
 Nombre de estudio: Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Desplazamiento estático Desplazamientos1
 Escala de deformación: 9,78598

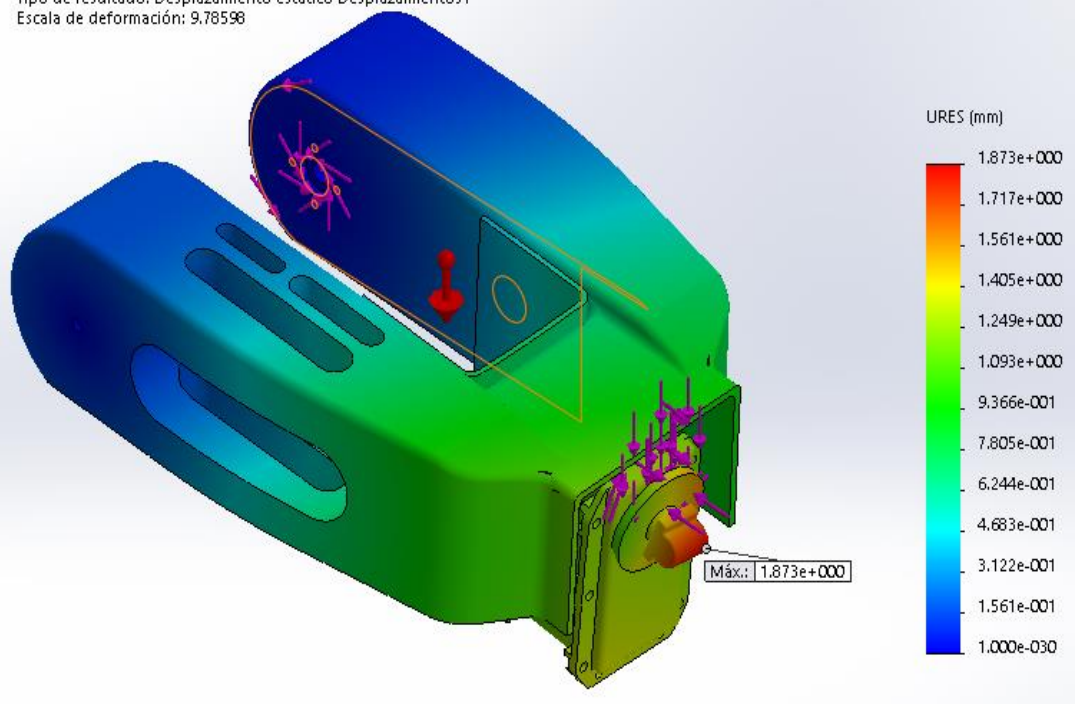


Figura 57. Desplazamiento en el eslabón 2.

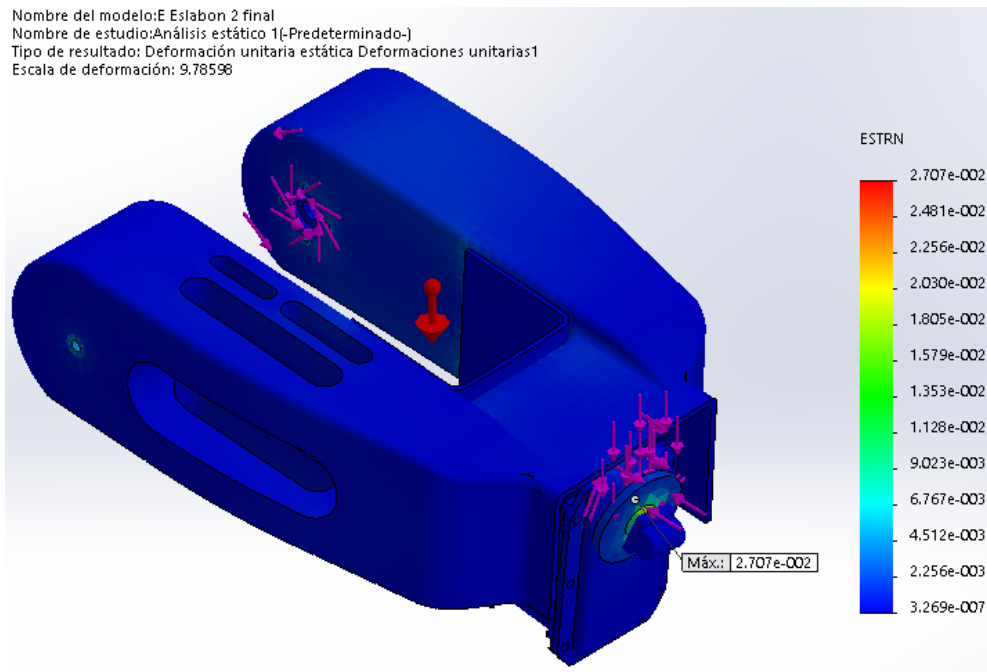


Figura 58. Deformación unitaria en el eslabón 2.

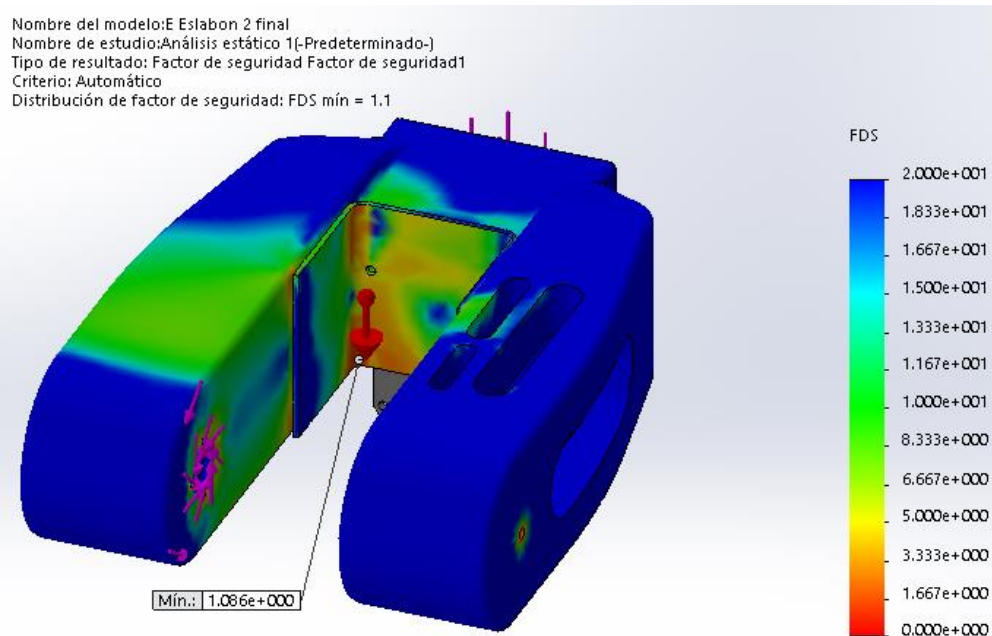


Figura 59. Factor de seguridad del eslabón 2.

Eslabón 3: Se realiza el ensamble con el motor dynamixel MX-28T, las propiedades físicas del eslabón corresponden a las del termoplástico ABS y está sujeto en su base mediante pernos.

Sobre el eje del motor se apoyan los siguientes eslabones, motores y actuador final que tienen un peso total 1.462 kg, las cargas y sujeciones se observan en la Figura 60.

Además el torque máximo que puede ejercer el motor MX-28T es de 2.5 Nm; bajo estas consideraciones se realiza el análisis estático del eslabón, los resultados se observan en las Figuras: 61, 62, 63 y 64.

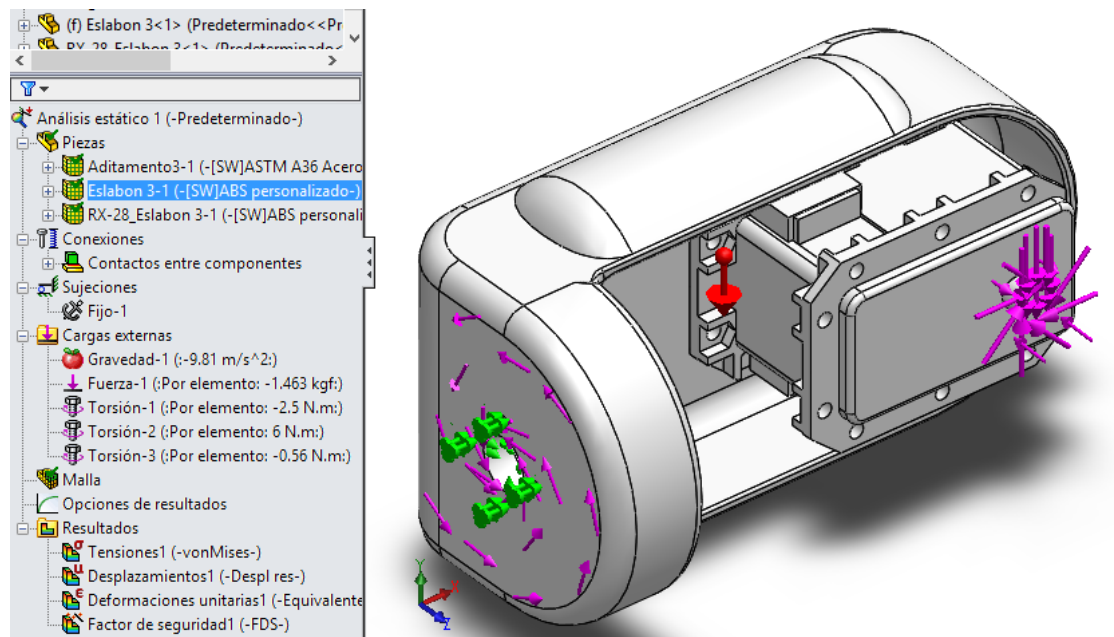


Figura 60. Cargas y Sujeciones aplicadas a Eslabón 3.

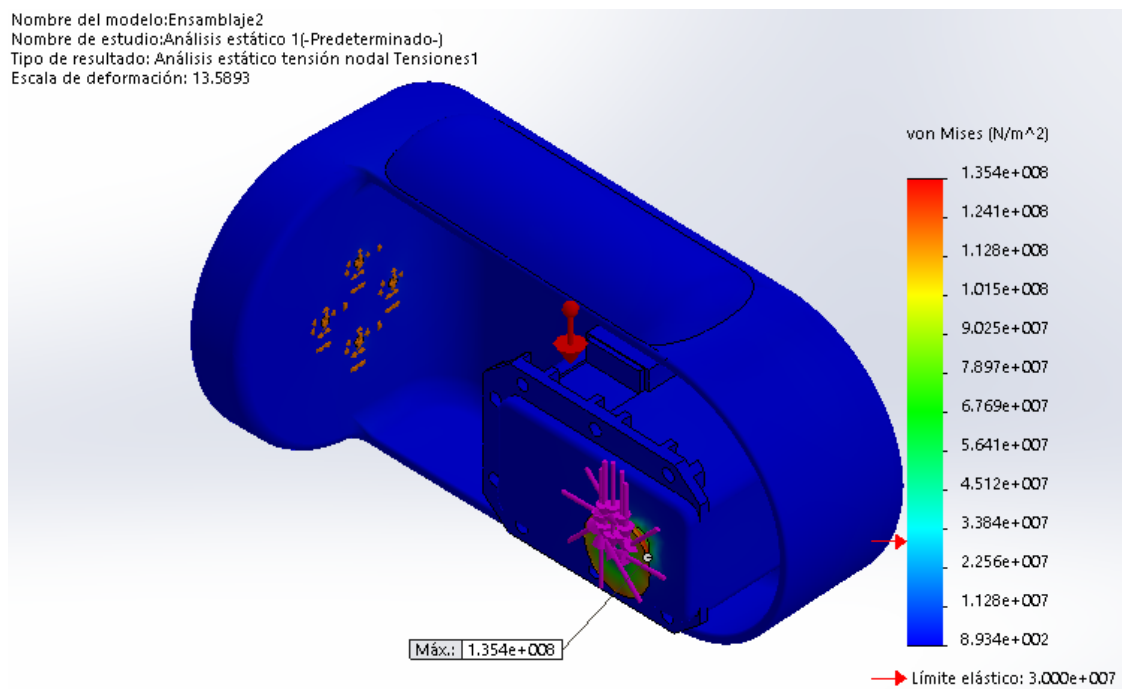


Figura 61. Tensiones von Mises del eslabón 3.

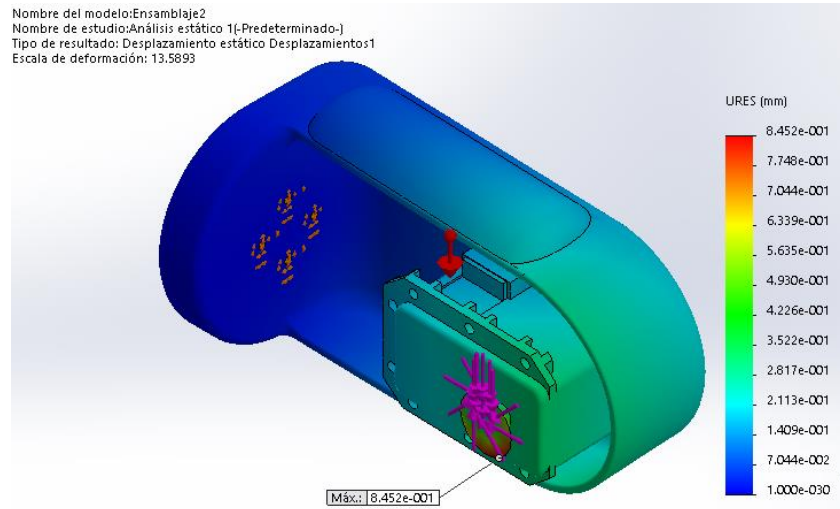


Figura 62. Desplazamiento del eslabón 3.

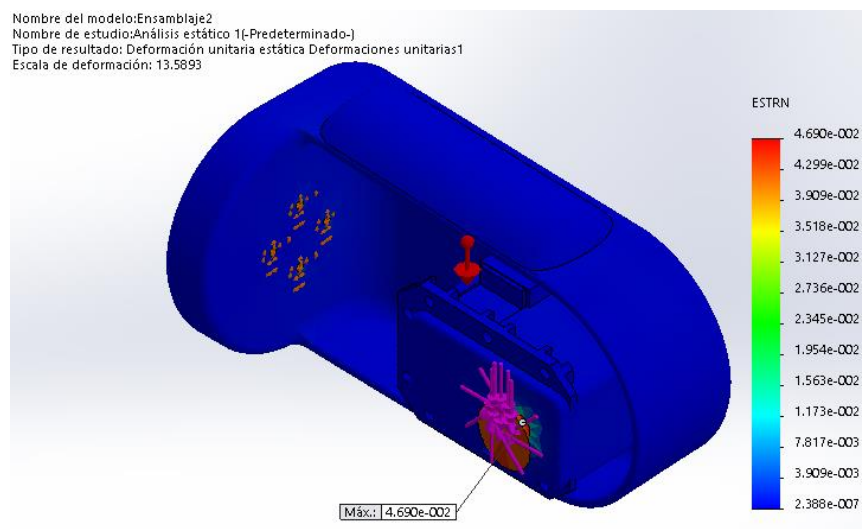


Figura 63. Deformación unitaria del eslabón 3.

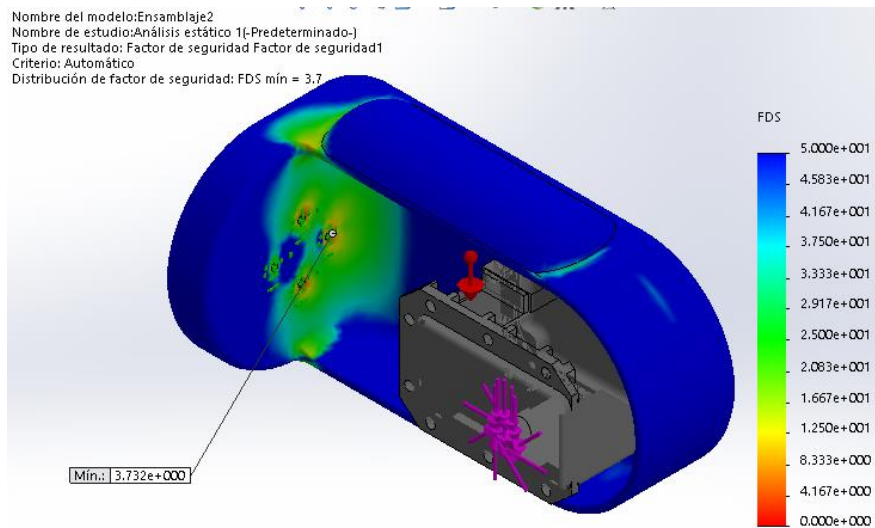


Figura 64. Factor de seguridad del eslabón 3.

Eslabón 4: En el software CAD realizamos un ensamblaje con el motor dynamixel AX-12A, luego seleccionamos el material ABS con sus respectivas propiedades.

Aplicamos la carga en el eje del motor donde soporta la carga y el valor es el peso de los eslabones posteriores más la carga aplicada ya considerada en el efector final, esto nos da una carga de 1.355 Kg, además consideramos un momento en el eje del motor con el valor del torque del motor que es de 1.6Nm, consideramos también el torque que ejerce el motor anterior MX-28T es de 2.5 Nm y el torque que se produce por consecuencia del peso de los elementos siguientes y la carga aplicada que resulta de 2.464N.m como se observa en la Figura 65.

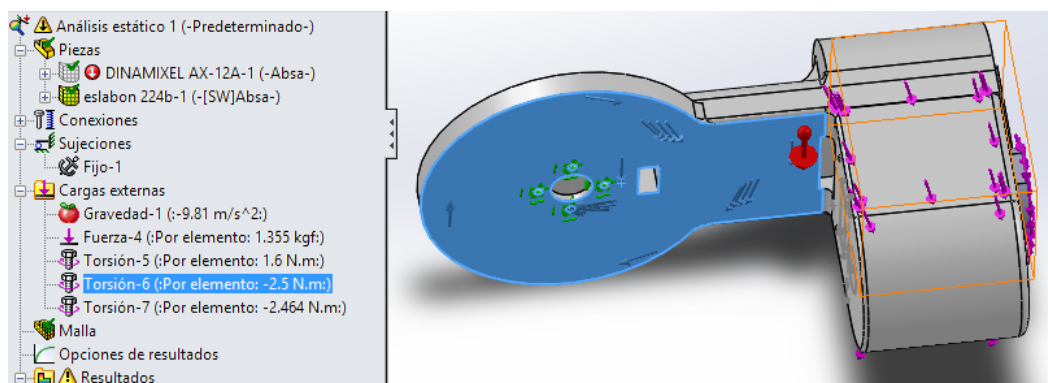


Figura 65. Cargas aplicadas al eslabón 4 para análisis de esfuerzos.

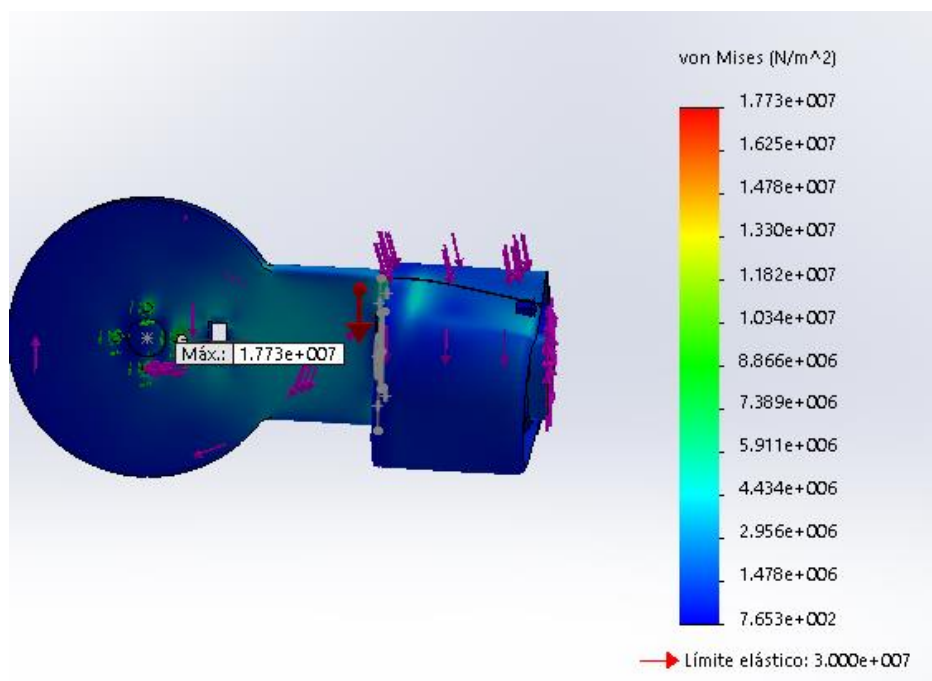


Figura 66. Tensiones von Mises del eslabón 4.

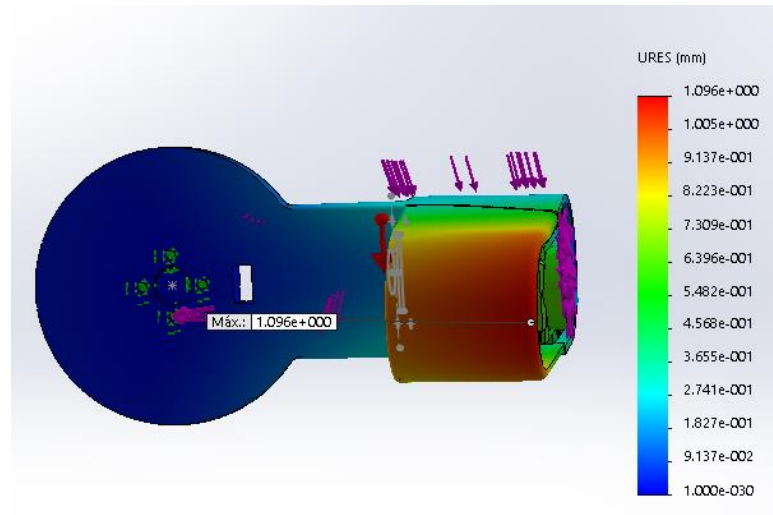


Figura 67. Desplazamientos del eslabón 4.

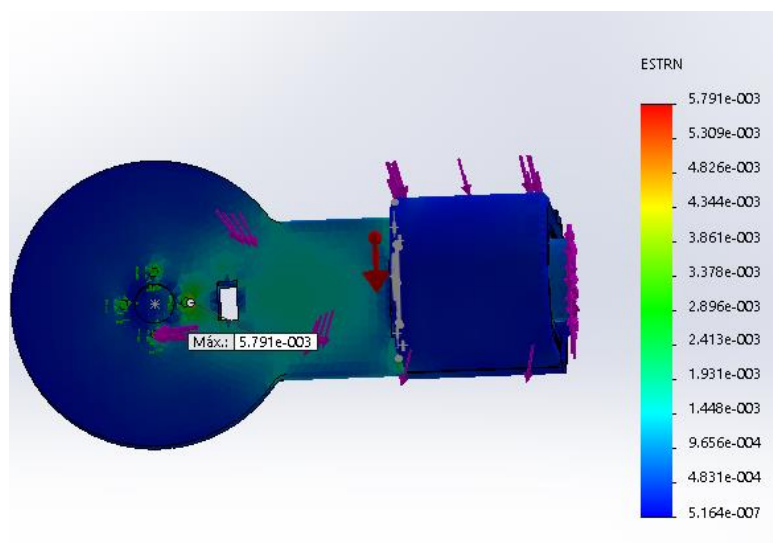


Figura 68. Deformación unitaria del eslabón 4.

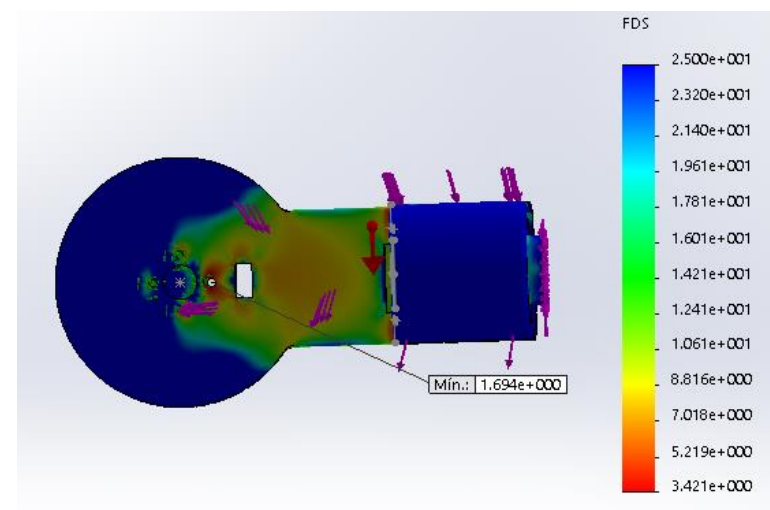


Figura 69. Factor de seguridad del eslabón 4.

Eslabón 5: En el software CAD realizamos un ensamblaje con el motor dynamixel AX-12A, aplicamos el material ABS con sus respectivas propiedades.

Realizamos las sujeciones correspondientes y aplicamos las cargas en el extremo así como el peso del ensamblaje como se observa en la Figura 70, aplicamos la carga en el eje del motor donde soporta la carga y el valor es el peso de los eslabones posteriores más la carga aplicada ya considerada en el efector final, esto nos da una carga de 1.236 Kg, además aplicamos un momento en el eje del motor con el valor del torque del motor que es 1.6Nm, consideramos también el torque que ejerce el motor anterior AX-12A es de 1.6 Nm y el torque que se produce por consecuencia del peso de los elementos siguientes y la carga aplicada que resulta de 1.71N.m.

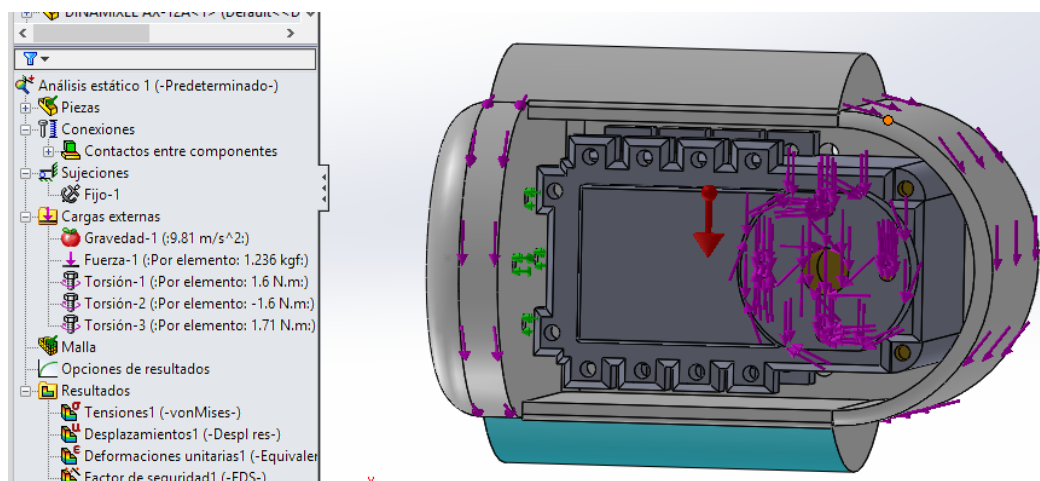


Figura 70. Cargas aplicadas al eslabón 5 análisis de esfuerzos.

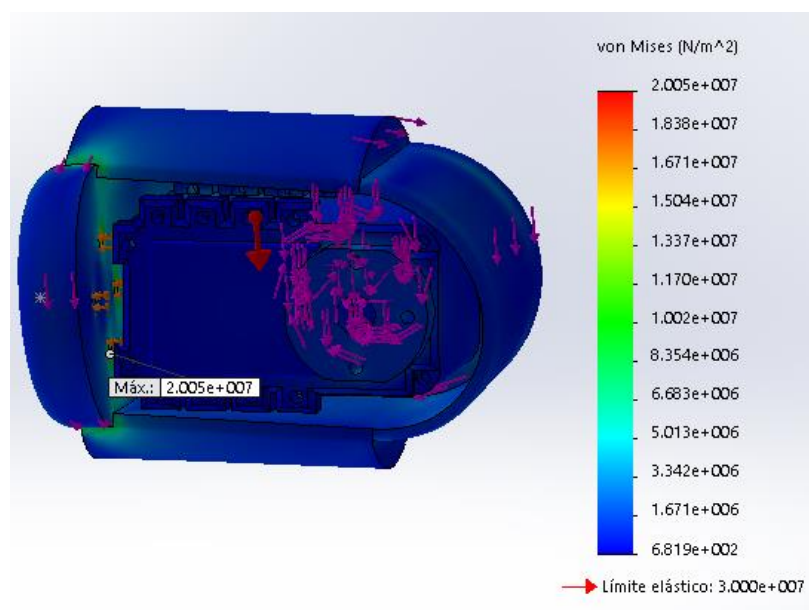


Figura 71. Tensiones von Mises del eslabón 5.

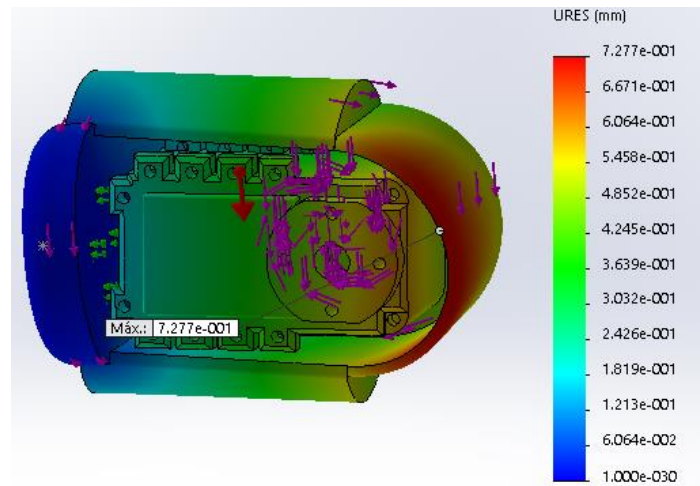


Figura 72. Desplazamiento del eslabón 5.

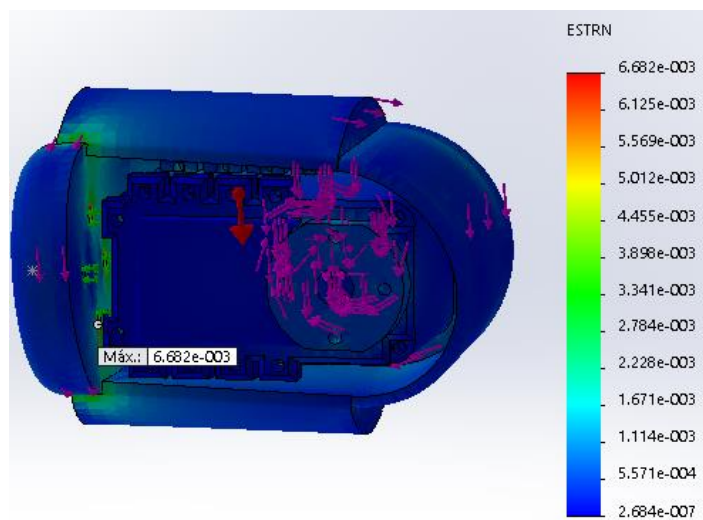


Figura 73. Deformación unitaria del eslabón 5.

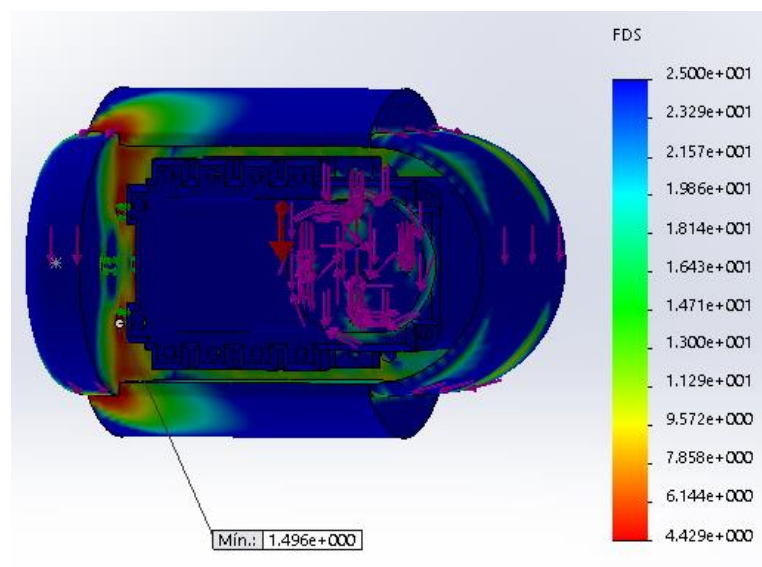


Figura 74. Factor de seguridad del eslabón 5.

Eslabón 6: En el software CAD realizamos un ensamblaje con el motor dynamixel AX-12A, aplicamos el material ABS con sus respectivas propiedades.

Realizamos las sujeciones correspondientes y aplicamos las cargas en el extremo así como el peso del ensamblaje como se observa en la Figura 75, aplicamos la carga en el eje del motor donde soporta la carga y el valor es el peso de los eslabones posteriores más la carga aplicada ya considerada en el efector final, esto nos da una carga de 1.1 Kg, además aplicamos un momento en el eje del motor con el valor del torque de dicho motor, consideramos también el torque que ejerce el motor anterior AX-12A es de 1.6 Nm y el torque que se produce por consecuencia del peso de los elementos siguientes y la carga aplicada que resulta de 0.78815N.m.

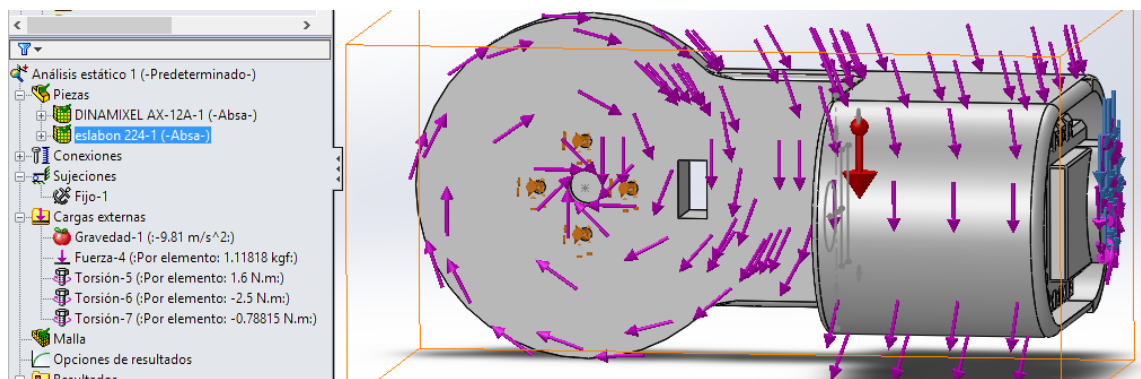


Figura 75. Cargas aplicadas al eslabón 6 para análisis de esfuerzos.

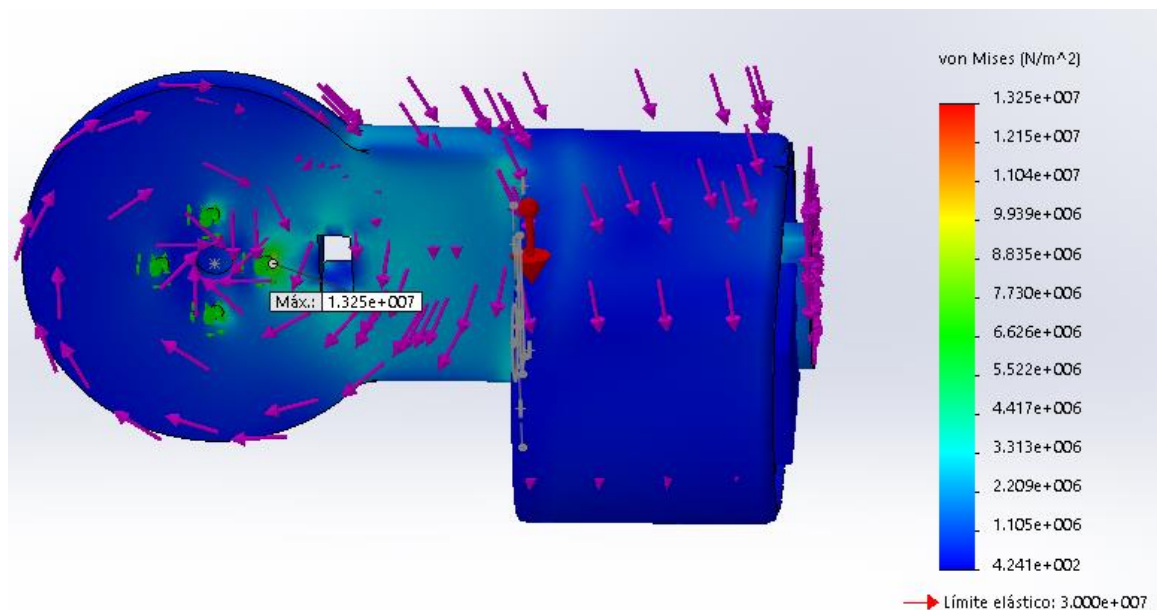


Figura 76. Tensiones von Mises del eslabón 6.

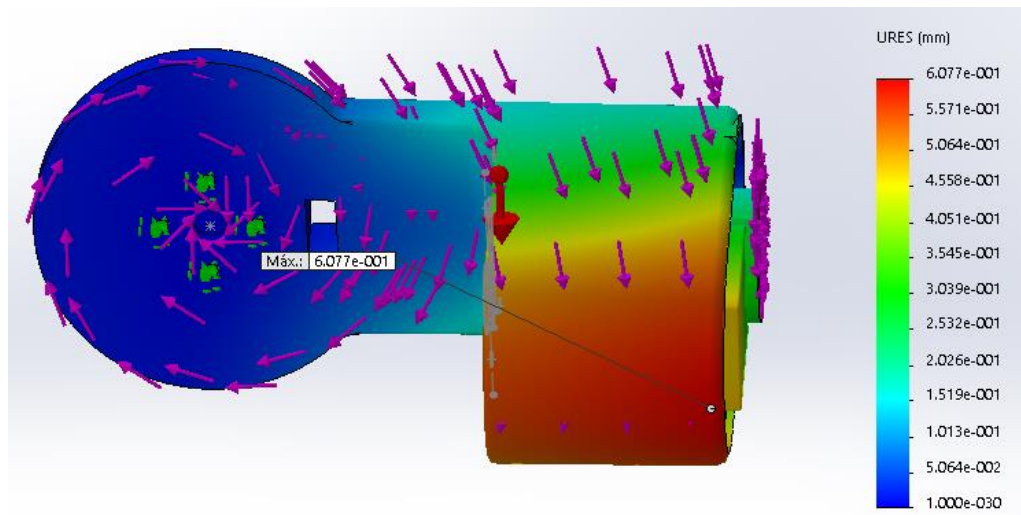


Figura 77. Desplazamientos del eslabón 6.

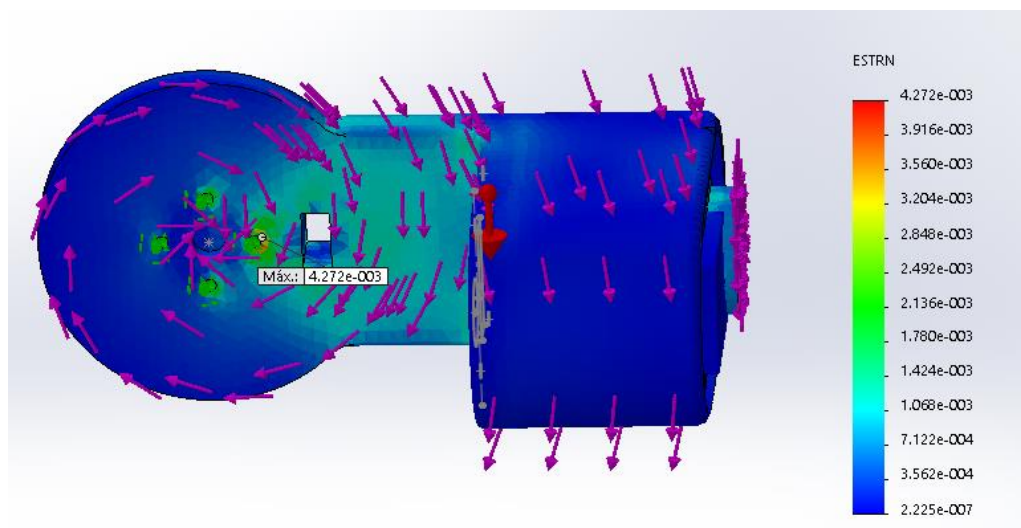


Figura 78. Deformación unitaria del eslabón 6.

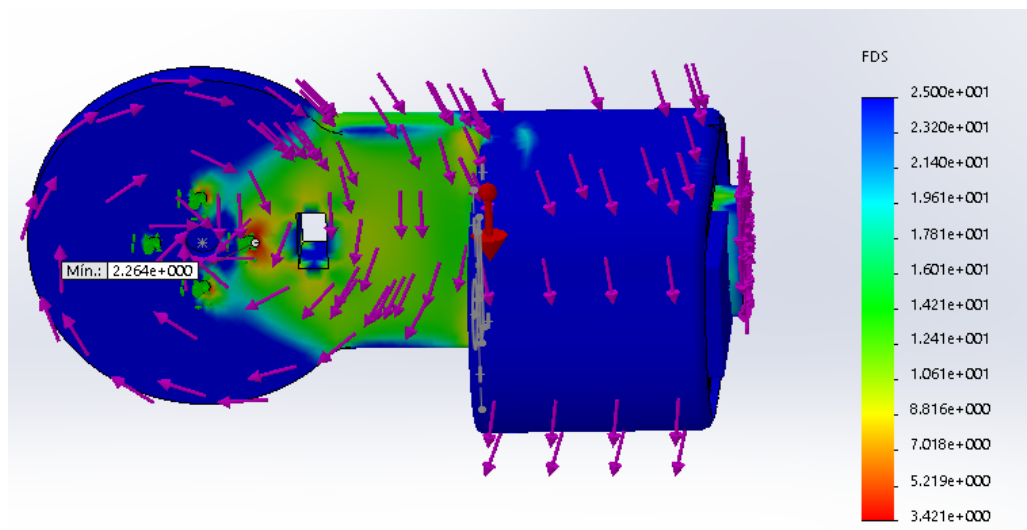


Figura 79. Factor de seguridad del eslabón 6.

Soporte del motor 8 y del efector final: En el software CAD realizamos un ensamblaje con el motor dynamixel AX-12A, aplicamos el material ABS con sus respectivas propiedades.

Realizamos las sujeciones correspondientes y aplicamos las cargas en el extremo así como el peso del ensamblaje como se observa en la Figura 80, aplicamos la carga en el eje del motor donde soporta la carga y el valor es el peso de los eslabones posteriores más la carga aplicada ya considerada en el efector final, esto nos da una carga de 1.043 Kg, además aplicamos un momento en el eje del motor con el valor del torque de dicho motor, consideramos también el torque que ejerce el motor anterior AX-12A es de 1.6 Nm y el torque que se produce por consecuencia del peso de los elementos siguientes y la carga aplicada que resulta de 0.024N.m.

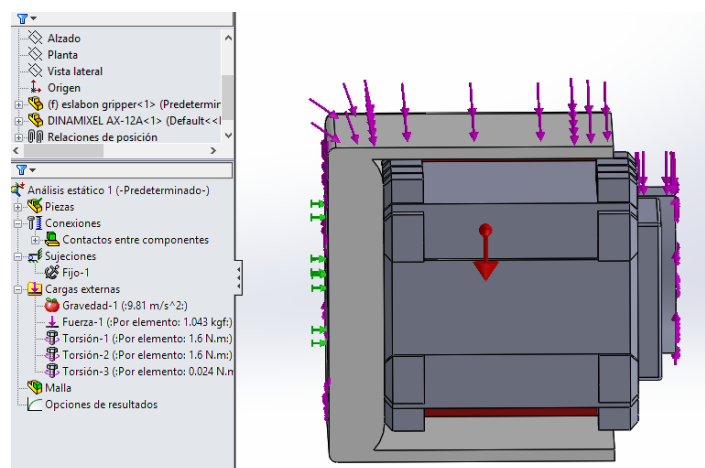


Figura 80. Cargas aplicadas a soporte del efector final para análisis de esfuerzos.

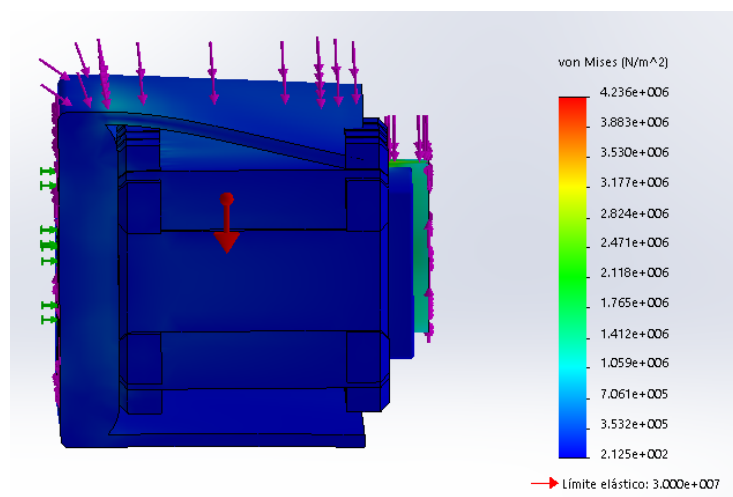


Figura 81. Tensiones von Mises del soporte para el efector final.

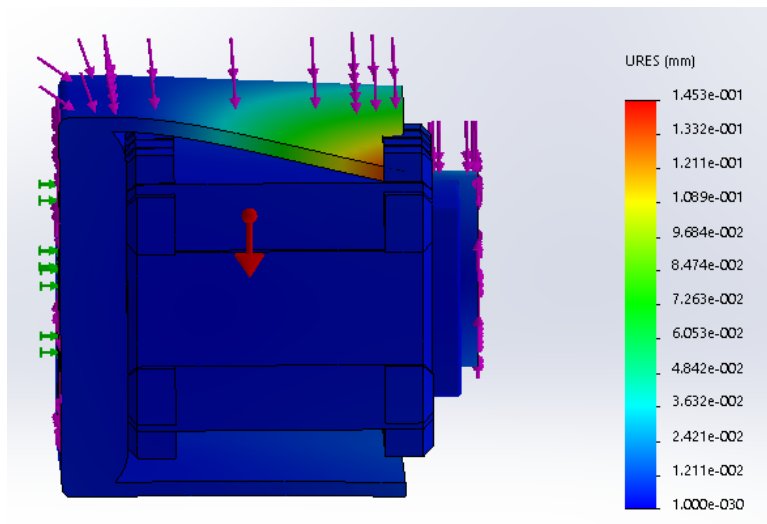


Figura 82. Desplazamientos del soporte para el efector final.

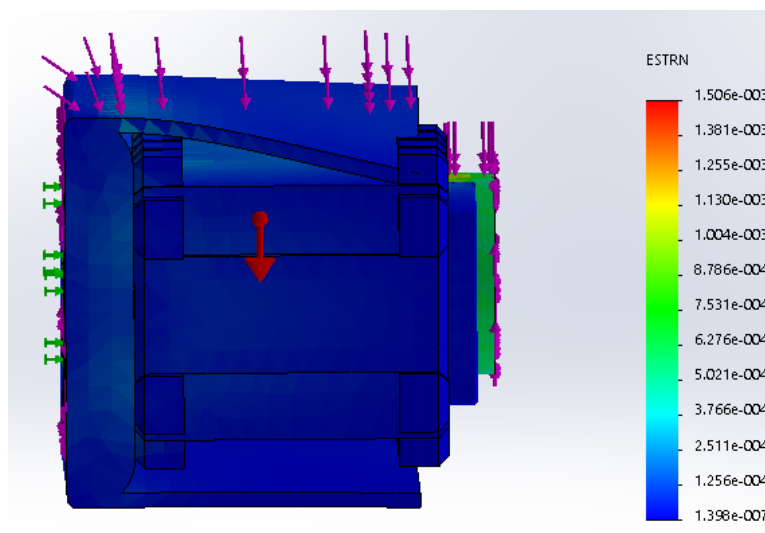


Figura 83. Deformación unitaria del soporte para el efector final.

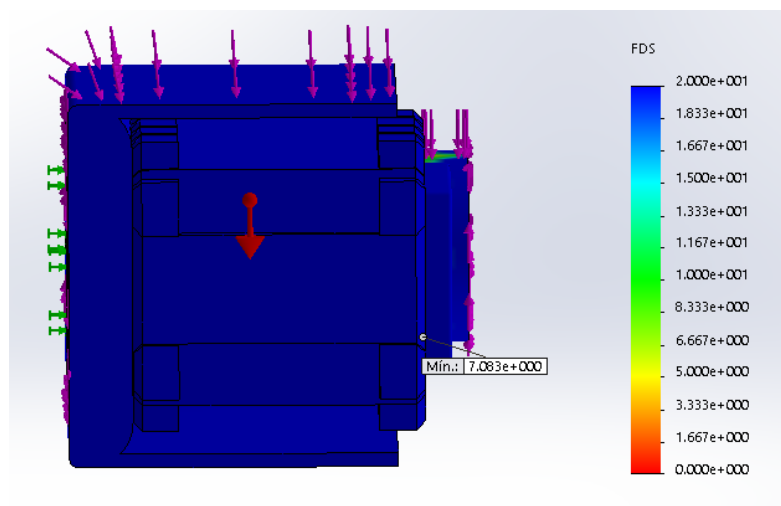


Figura 84. Factor de seguridad del soporte para el efector final.

3.1.5 Evaluación

En la tabla 8 podemos observar los resultados del análisis estático en cada pieza del robot, podemos observar que el factor de seguridad es mayor a uno a pesar de que cargamos el sistema con un 1Kg en el efector final, lo que nos indica que los elementos no fallaran mecánicamente.

Tabla 8.

Resultados del análisis estático de los eslabones

<i>Piezas</i>	<i>Esfuerzo-Von Mises (N/m²)</i>	<i>Desplazamiento (mm)</i>	<i>Deformación unitarias</i>	<i>Factor de seguridad</i>
Base	1.926x10 ⁸	1.220	6.010x10 ⁻¹	1.900
Eslabón 1	8.820x10 ⁷	4.247x10 ⁻¹	2.220x10 ⁻²	2.570
Eslabón 2	2.160x10 ⁸	1.873x10 ⁻¹	2.210x10 ⁻²	1.100
Eslabón 3	1.354x10 ⁸	8.452x10 ⁻¹	4.690x10 ⁻²	3.732
Eslabón 4	1.770x10 ⁷	10.960x10 ⁻¹	5.791x10 ⁻³	1.694
Eslabón 5	6.773x10 ⁶	2.510x10 ⁻¹	6.682 ⁻³	1.496
Eslabón 6	2.005x10 ⁷	7.270x10 ⁻¹	4.272x10 ⁻³	2.264
Soporte motor 8	4.230x10 ⁶	1.453 x10 ⁻¹	1.506x10 ⁻³	7.083

3.1.6 Implementación del diseño mecánico.

Los motores dynamixel, el gripper y algunos tornillos fueron adquiridos en trossenrobotics.com, la fuente de alimentación, los tornillos faltantes y la placa para la base fueron conseguidos de distintos locales de Quito y Sangolquí.

Una vez adquiridos los materiales se fabricaron los eslabones en la máquina de prototipado rápido Makerbot Replicator 2x indicada en la figura 85, perteneciente al área de impresión del laboratorio de Procesos de Manufactura de la ESPE; con los parámetros de impresión que se indican en la Tabla 9. El acabado superficial con el espesor de capa de 0.2 mm se ve estéticamente bien, la densidad mayor al 90 % se emplea para piezas funcionales.



Figura 85. Máquina de prototipado rápido Makerbot Replicator x2.

Tabla 9.

Parámetros para el prototipado de los eslabones.

<i>Parámetro</i>	<i>Especificación</i>
Material	ABS
Espesor de capa	< 0.25 mm
Densidad de la pieza	> 90 %

En la figura 86 se puede observar una vista superior de la fabricación del segundo eslabón, los orificios que se notan pertenecen al diseño de la pieza pero la máquina crea superficie de respaldo para que no se caigan las siguientes capas de ABS impreso y no se dañe el eslabón.

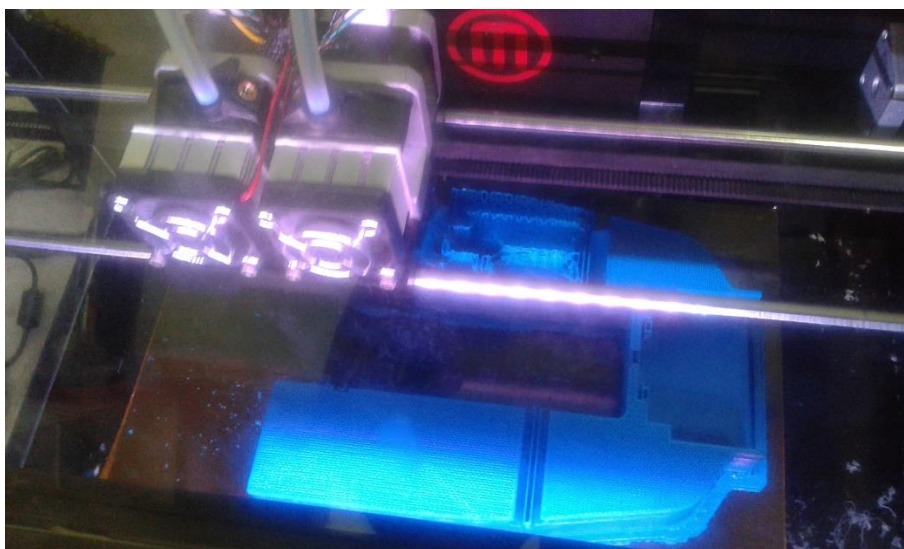


Figura 86. Vista superior del prototipado del eslabón 2.

En la figura 87 se muestra una vista lateral de la impresión del eslabón 2, el prototipado se encuentra sobre el 50%, el orificio que se muestra a la derecha permite aligerar el peso de la pieza.

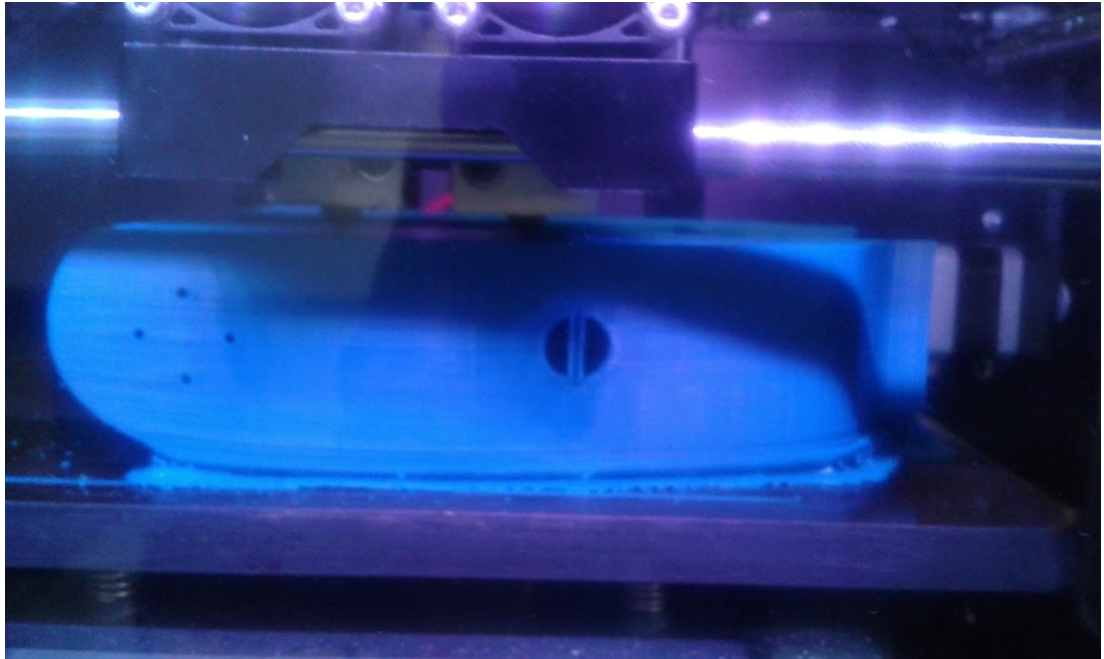


Figura 87. Vista lateral del prototipado del eslabón 2.

Después de realizar el prototipado de todos los eslabones, se les coloca una pequeña capa de pegamento para darles un poco de brillo y mejorar la unión de la superficie de las capas externas del ABS en las piezas. El resultado se muestra en la figura 88.



Figura 88. Eslabones y motores del robot.

Los eslabones del final hacia el inicio se ensamblan mediante los tornillos y tuercas, teniendo cuidado de no estrangular los cables ni dejarlos atascados entre el motor y la pieza. En la figura 89 está el gripper ensamblado con el séptimo motor AX-12A.



Figura 89. Gripper ensamblado al séptimo motor.

Al séptimo motor AX-12A se le sujeta al eslabón correspondiente de manera que con una holgura para que se conecten los cables del anterior motor al motor sujeto. Como se muestra en la figura 90 la conexión entre los motores es en serie reduciendo el cableado necesario.



Figura 90. Séptimo eslabón ensamblado al gripper.

Al séptimo eslabón se acopla la brida del sexto motor AX-12A mediante cuatro tornillos y éste al sexto eslabón, teniendo cuidado no dañar los cables de conexión entre los motores, de la misma forma se ensambla el quinto motor y eslabón como se indica en la figura 91.



Figura 91. Quinto, sexto y séptimo eslabón ensamblado.

Mediante los tornillos adquiridos se acoplan los motores MX-28T al cuarto eslabón y el motor MX-64T al tercer eslabón, un tornillo en el centro de la brida de los motores permite una sujeción mayor entre los eslabones y el motor. En la figura 92 se pueden observar los eslabones y motores ensamblados desde el tercer eslabón hasta el gripper.

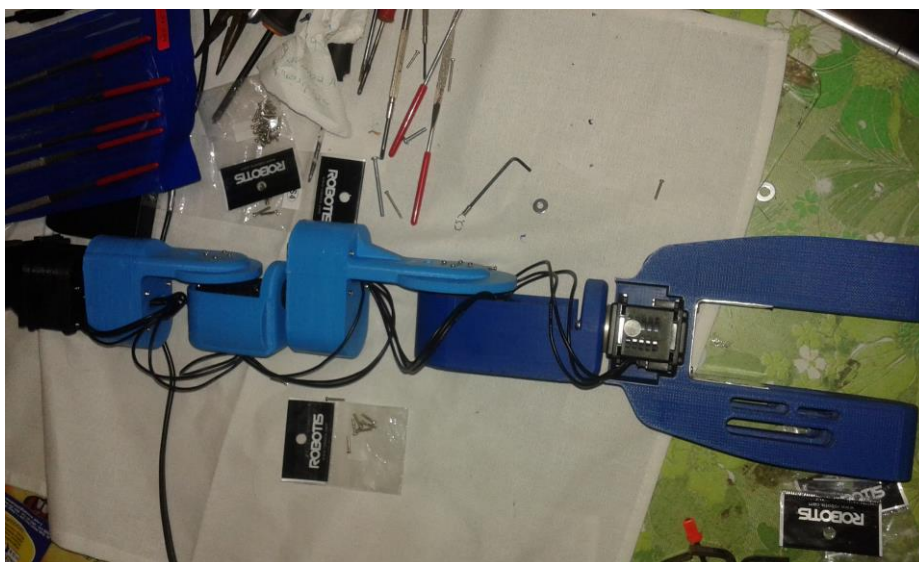


Figura 92. Robot ensamblado desde el tercer eslabón.

La base o eslabón 0 se ensambla con el eslabón 1 por facilidad y se colocan los motores MX-64T; en el gráfico 93 se observa los cables que llegarán al tercer motor y tres cables que salen de la base para conectar los motores al controlador USB2Dynamixel y a la fuente de poder.



Figura 93. Base y eslabón 1 ensamblados.

Una vez ensamblado el robot en dos partes se unen el segundo motor con el segundo eslabón, armándose todo el robot para probar su funcionamiento antes de anclarlo al soporte de la base. Al soporte de la base se le realizan los orificios para que se pueda anclar la base del robot y se sujeta el robot a la misma como se ilustra en la figura 94.



Figura 94. Robot ensamblado.

3.2 Implementación Electrónica

3.2.1 Fuente de alimentación

En Ecuador el voltaje más común de encontrar es de 120 V AC a una frecuencia de 50 Hz, en la Tabla 10 se indica una corriente de 19.7 A necesaria para cuando todos los motores funcionen a la vez a su máxima capacidad y a 12 V DC.

Tabla 10.

Corriente necesaria por los motores dynamixel del robot.

<i>Cantidad</i>	<i>Modelo</i>	<i>Corriente máxima (A)</i>	<i>Corriente total (A)</i>
3	MX-64T	4.1	12.3
1	MX-28T	1.4	1.4
4	AX-12 ^a	1.5	6.0
		Corriente total (A)	19.7

La fuente de poder marca Altek modelo ATX-750 entrega +12 V DC, posee una salida de +12 V con una corriente máxima de 38 A, indicada en la tabla 11. El robot ocupa 19.7 A; existen 18 A disponibles para una aplicación adicional así como las salidas de +5, +3.3 y -12 V DC.

Tabla 11.

Características de la fuente de poder ALTEK ATX-750

	<i>Voltaje (V)</i>	<i>Corriente (A)</i>
Entrada voltaje AC	220 50 Hz	3
Salidas voltaje DC	+5	30
	+12	38
	+3.3	27
	-12	0.5

3.2.2 Controlador usb2Dynamixel

USB2Dynamixel mostrado en la Figura 95 es un controlador empleado para gobernar los motores Dynamixel desde un ordenador con un puerto USB o el puerto Serial.



Figura 95. Controlador USB2Dynamixel.

Fuente: (Robotis, 2010)

Los motores se conectan forma serial entre si al igual que con el controlador como se muestra en la Figura 96, se observa una línea de poder que se emplea para energizar los motores porque emplean +12 V DC y el controlador USB2Dynamixel funciona con los +5 V DC que entrega el puerto USB o serial de la computadora.

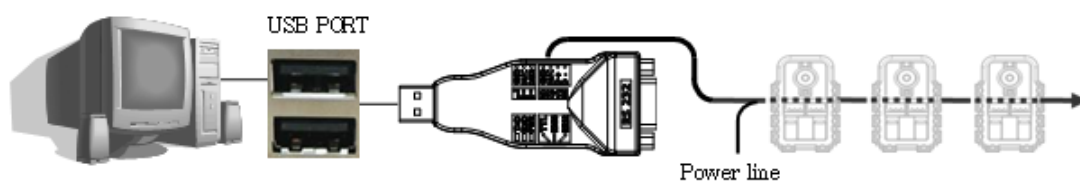


Figura 96. Diagrama de conexión PC/USB2Dynamixel/motores dynamixel

Fuente: (Robotis, 2010)

La línea de poder se muestra en la Figura 97, el PIN#1 correspondiente al negativo o tierra de la conexión DC se conecta en el negativo del controlador y en el negativo de la fuente. El PIN#2 es el positivo que energiza a los motores, se conecta únicamente al positivo de 12 V DC de la fuente de poder.

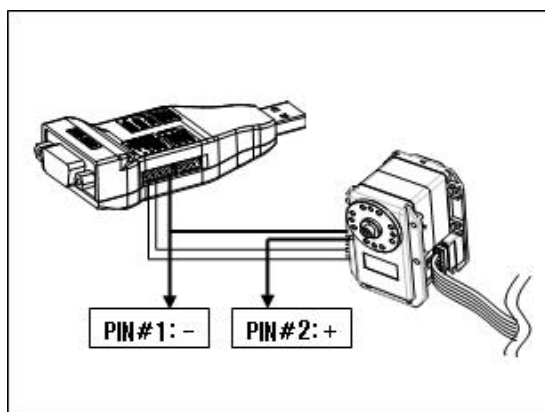
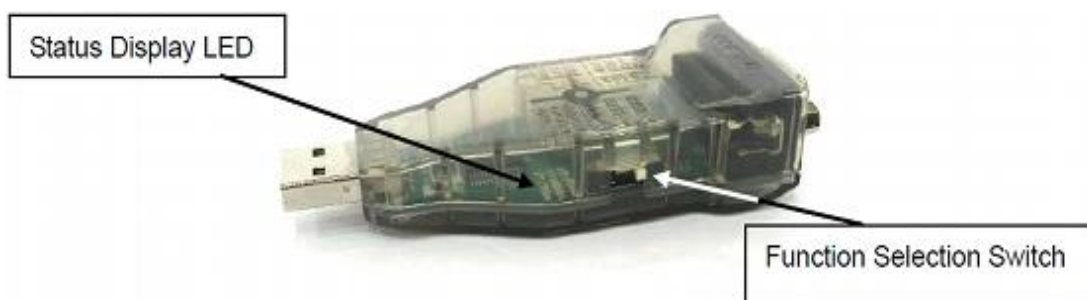


Figura 97. Diagrama de conexión de la fuente de poder a los motores.

Fuente: (Robotis, 2010)

Partes del controlador USB2Dynamixel

El controlador consta de las partes indicadas en la Figura 98 y en la Figura 99, en la tabla 12 se indican sus funciones.



Fuente: (Robotis, 2010)

Figura 98. Status y selector de función del controlador USB2Dynamixel



Figura 99. Conectores del controlador USB3Dynamixel

Fuente: (Robotis, 2010)

Tabla 12.

Partes del controlador USB2Dynamixel.

<i>Parte</i>	<i>Descripción</i>
Status Display LED	Visualiza fuente de alimentación, TXD (escritura de datos) y RXD (lectura de datos) de estado.
Function Slection Switch	Selecciona el método de comunicación de TTL, RS-485 y RS-232.
3P Connector	Conecta Dynamixels de Serie AX a través de la comunicación TTL.
4P Connector	Conecta Dynamixels de DX, serie RX través de la comunicación RS-485.
Serial Connector	Cambia de puerto USB a través del puerto serie de comunicación RS-232.
USB Connector	Conecta controlador con el ordenador.

Fuente: (Robotis, 2010)

Características del software

El instalador para este dispositivo se lo puede descargar de la página de www.robotis.com, se puede configurar la velocidad de transmisión, el tiempo de respuesta, entre otros parámetros. El software es muy versátil para emplearse en varios entornos de programación bajo el sistema operativo Windows como se muestran en la Tabla 13. Matlab es el entorno de programación en el cual se desarrolla el algoritmo genético, se requiere la versión Dinamixel Communication 1.0 para el presente trabajo.

Tabla 13.

Características del software del controlador USB2Dynamixel.

<i>Parámetro</i>	<i>Especificaciones</i>
Sistema operativo	Windows XP / Vista / 7 (32 / 64 bit)
Versiones	Dynamixel Communication 1.0 Dynamixel Communication 2.0
Entornos de programación	Dynamixel Communication 1.0 Visual C++ Visual Basic Visual C# MATLAB LabView Dynamixel Communication 2.0 Visual C++ Visual C#
<i>Parámetro</i>	<i>Especificaciones</i>
Sistema operativo	Linux 2.6.21 recomendado
Versión	Dynamixel 1.0
Entornos de programación	Linux

Fuente: (Robotis, 2010)

3.3 Sistema de control

3.3.1 Control de Posición Motores Dynamixel

Los motores dynamixel que utilizamos para el presente proyecto tienen integrado un controlador PID el cual permite que cada motor tenga un sistema de control en lazo cerrado y así alcanza una posición determinada.

Acción proporcional P: produce una acción de control en forma de ganancia proporcional al error.

Acción integral I: produce una acción de control proporcional a la integral del error con respecto al tiempo

Acción derivativa D: produce una acción de control proporcional a la rapidez con que se produce el cambio del error.

Las ganancias del controlador PID de cada motor vienen dadas de fábrica y de acuerdo al modelo del motor

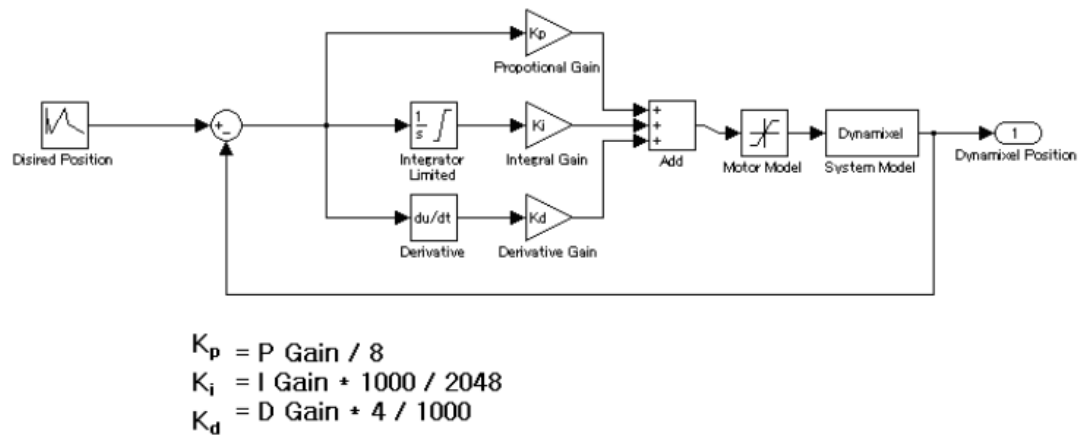


Figura 100. Controlador PID interno de motores dynamixel.

Fuente: (Robotis, 2010)

3.4 Diseño del HMI

El diseño del HMI tiene que ser intuitivo y claro para el operador o usuario, el software que se diseñó además de monitorear el robot, permitirá la correcta programación e interacción con el mismo.

Para desarrollar pantallas de manera adecuada existen varias normativas y guías entre las más utilizadas para sistemas SCADA y procesos industriales esta la guía Gedis que nace de varios estándares para el diseño de HMI.

3.4.1 Guía Gedis

La guía Gedis es una herramienta ergonómica para el desarrollo de HMI, que se basa en normas presentadas por diversos organismos mundiales como por ejemplo:

NASA-STD-3000, El estándar SAS (Safety Automation System) es un estándar de la industria petrolífera de Noruega, estándar NUREG 0700 desarrollado por la asociación nuclear de Estados Unidos, etc.

La guía Gedis plantea estándares para los siguientes aspectos como se menciona en (Ponsa & Granollers).

- Arquitectura : La arquitectura que se recomienda es de forma jerárquica
- Distribución de las pantallas: la distribución nos orienta en aspectos como ubicación del título, ubicación del menú, ubicación de alarmas, ubicación de tendencias y tablas.

En (Ponsa & Granollers, pág. 5), menciona las siguientes directrices se debe considerar que el centro de la pantalla es importante, la información debe estar en la parte inferior izquierda, la mejor posición de los gráficos es a la izquierda de la pantalla y considerar que el movimiento de los ojos va de arriba hacia abajo y de izquierda a derecha.

- Navegación: este ítem permite que la pantalla sea fácil de usar, aquí entran Menús, barra de botones, Sliders, etc.

Las directrices de navegación más importantes son el área de contacto para pulsar o navegar debe ser lo suficientemente amplio, botones que permitan retroceder o avanzar pantallas y opciones de salir o cerrar las mismas.

- Color: No utilizar más de 4 colores para usuarios inexpertos, no utilizar contrastes incompatibles, tratar de no usar intermitencia de colores, para el fondo se recomienda colores neutros como (gris, arena, azul).
- Texto: El tamaño del texto debe ser adecuado y no difícil de ver, el texto debe contrastar con el fondo de la pantalla, evitar aglutinamientos, no utilizar más de tres fuentes.
- Comandos e ingreso de datos: los comandos deben ser visibles, deben estar debidamente etiquetados, el área del botón o similar debe ser lo suficientemente grande, para el ingreso de datos debe existir un botón de aceptar o confirmar.
- Alarmas: evitar el exceso de alarmas innecesarias y mostrar el motivo del error o alarma.

Siguiendo todas estas directrices hemos diseñado las pantallas de la siguiente manera:

3.4.2 Pantalla de inicio

Esta pantalla está diseñada para poder navegar a través de las diferentes sub-áreas del programa que son: Simulación, modo manual, Programación, la lógica de programación de la pantalla se muestra en la Figura 101.

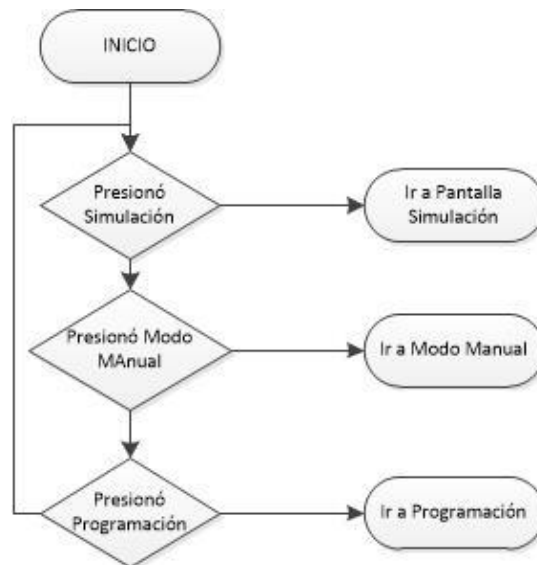


Figura 101. Diagrama de flujo Pantalla Inicio.

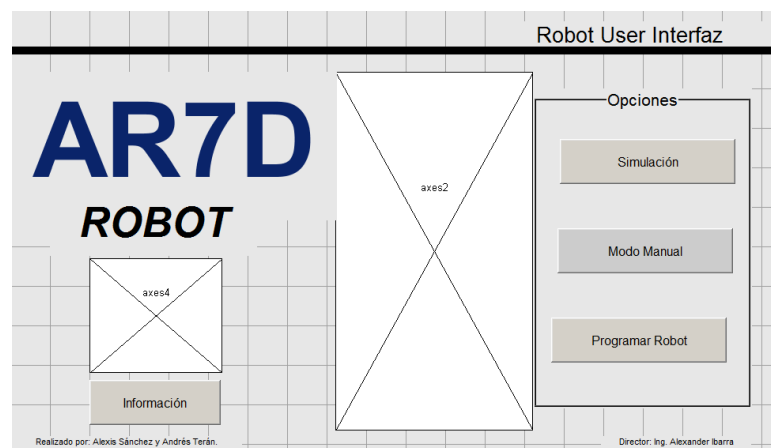


Figura 102. Pantalla de Inicio.

3.4.3 Pantalla de Simulación

Esta pantalla nos permite ingresar como datos de entrada la posición y orientación que se desea alcanzar seguido por la velocidad, la opción generar trayectoria mediante un algoritmo genético resolverá la cinemática inversa para luego generar una trayectoria que se podrá simular y visualizar tanto en un diagrama esquemático de barras unidas, como en un modelo 3D del robot, el cual nos ayudara a verificar el correcto movimiento y a identificar colisiones.

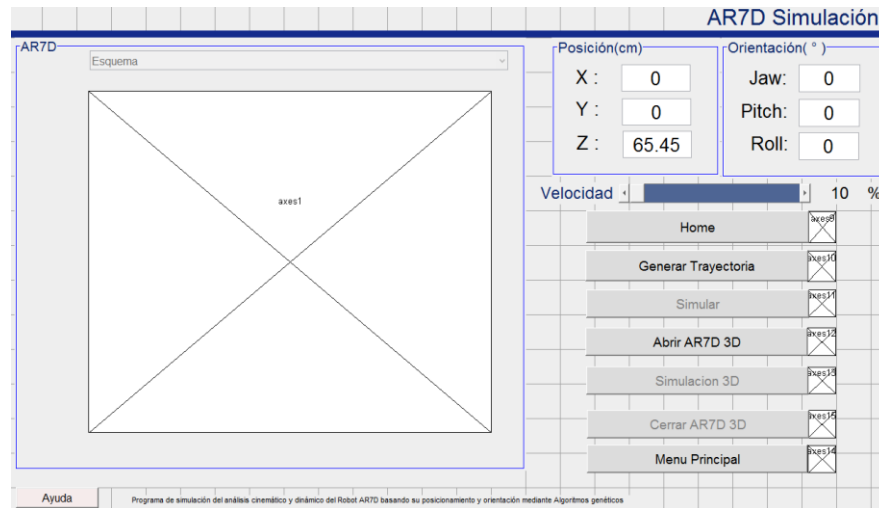


Figura 103. Pantalla Simulación.

Home: permite regresar al robot simulado a la posición home, la lógica de programación del botón se puede observar en la Figura 104.

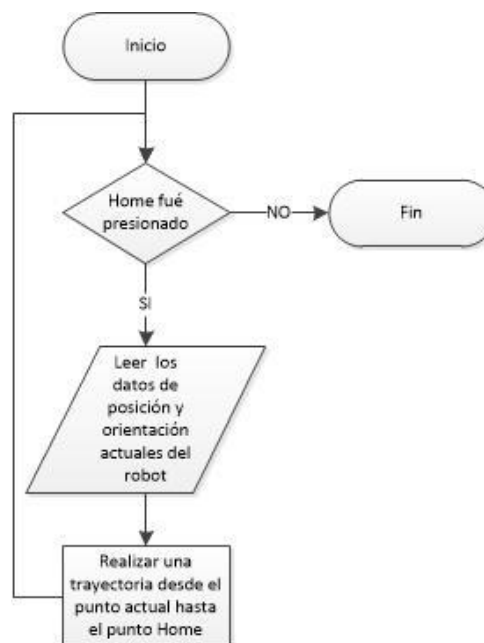


Figura 104. Diagrama de flujo- botón home.

Generar trayectoria: Resuelve la cinemática inversa mediante un algoritmo genético y luego genera una trayectoria, la lógica de programación del botón se puede observar en la Figura 105.

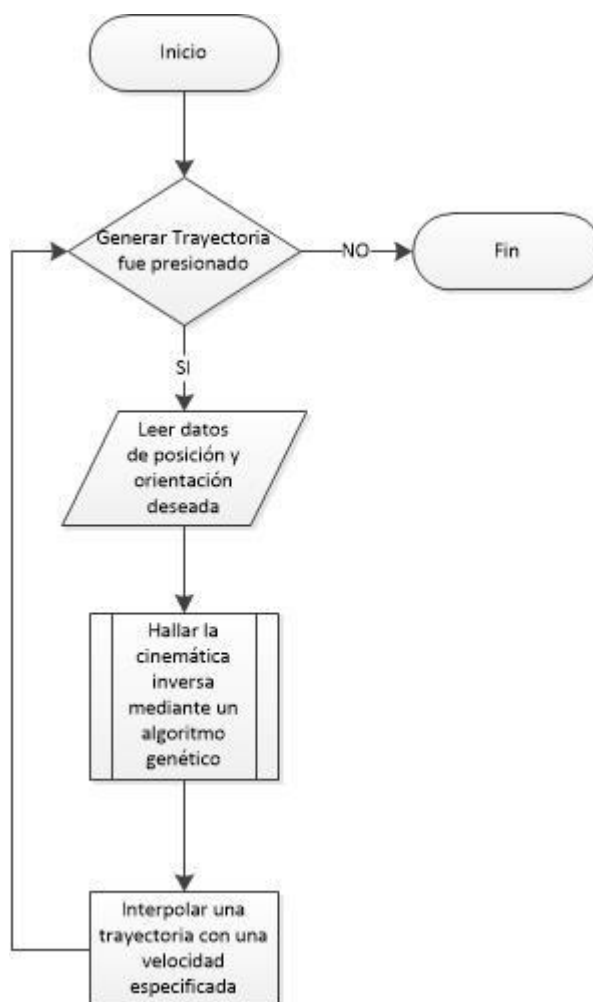


Figura 105. Diagrama de flujo- botón Generar Trayectoria.

Simular: Permite simular la trayectoria del robot mediante un diagrama de barras, así como visualizar la posición, la velocidad, la aceleración y el Torque de cada articulación con respecto al tiempo si seleccionamos el menú en la parte superior de la gráfica, la lógica de programación del botón se puede observar en la Figura 106.

Abrir 3D: Abre el diagrama de bloques de Simulink que representan el robot y así visualizar un modelo 3D.

Simulación 3D: Permite simular la trayectoria generada en el modelo 3D.

Cerrar 3D: Cierra el diagrama de bloques de Simulink.

Menú principal: Regresa a la pantalla de Inicio.

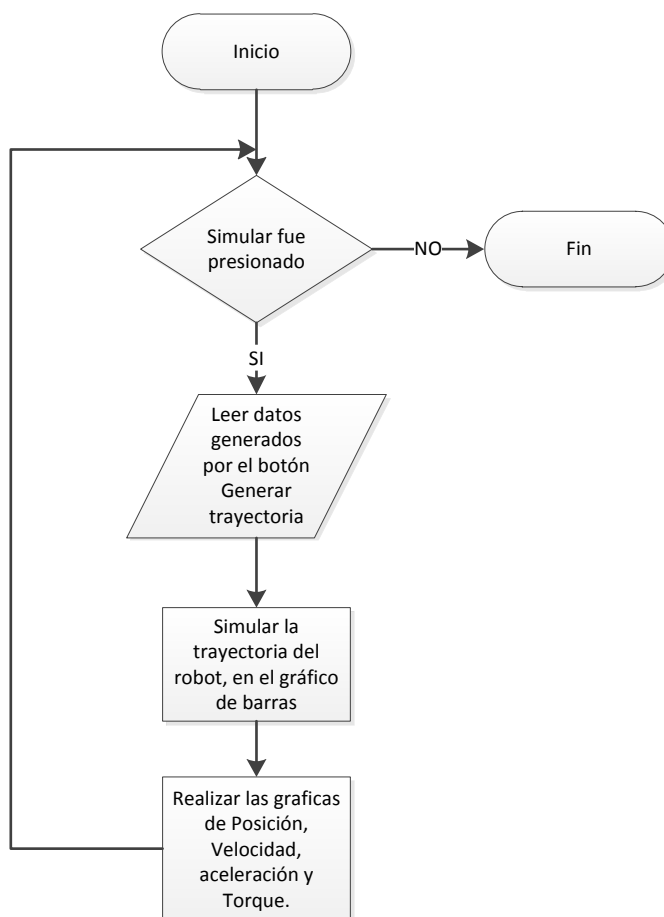


Figura 106. Diagrama de flujo- botón Simular.

3.4.4 Pantalla de Modo Manual

Esta pantalla nos permitirá simular, poner en línea el robot activando el puerto serial que controla los motores Dynamixel y moverlo de modo manual con los Sliders en el centro de la pantalla, además posee un botón teach que nos permitirá guardar una posición deseada con una velocidad a escoger y así podrá ser programado según se vayan guardando los puntos requeridos, que a su vez se visualizan en la parte derecha, también se ha programado un botón de home para que el robot vuelva a su posición inicial, las acciones son las siguientes:

- **Teach:** Este botón permite guardar la posición actual del robot, para luego simularla o reproducirla con el robot físico, Se pueden guardar un total de 20 posiciones, la lógica de programación del botón se puede observar en la Figura 107.
- **Guardar:** permite guardar los puntos y trayectorias generadas con el botón teach.

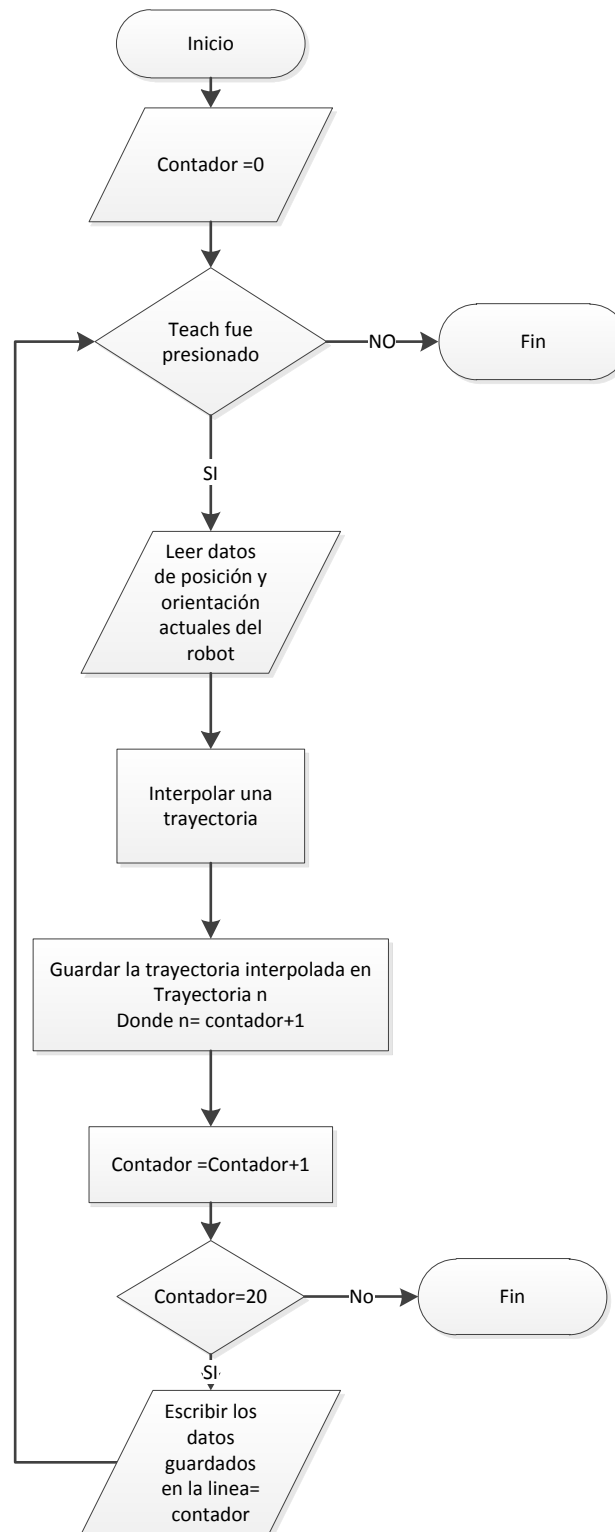


Figura 107. Diagrama de flujo- botón Teach.

- **Abrir:** permite abrir un programa previamente guardado
- **Borrar:** Borra la última posición guardada
- **Borrar Todo:** Borra todas las posiciones guardada

- **AR7D 3D:** permite abrir la pantalla en Simulink, que nos permitirá visualizar el robot a través de la herramienta SimMechanics de Matlab.
- **Play 3D:** permite correr el diagrama de bloque de Simulink y mover el modelo en 3D del robot con los Slider del centro de la pantalla
- **Run:** Si el robot está en línea, lo moverá igual que en la simulación.
- **Cíclico:** permite que el robot se mueva de manera cíclica repitiendo el movimiento programado
- **Pause:** Realiza una pausa del movimiento del robot, Si se presiona otra vez continuará el movimiento.
- **Stop:** Detiene el movimiento del robot.
- **Menú Principal:** Regresa a la pantalla de inicio.

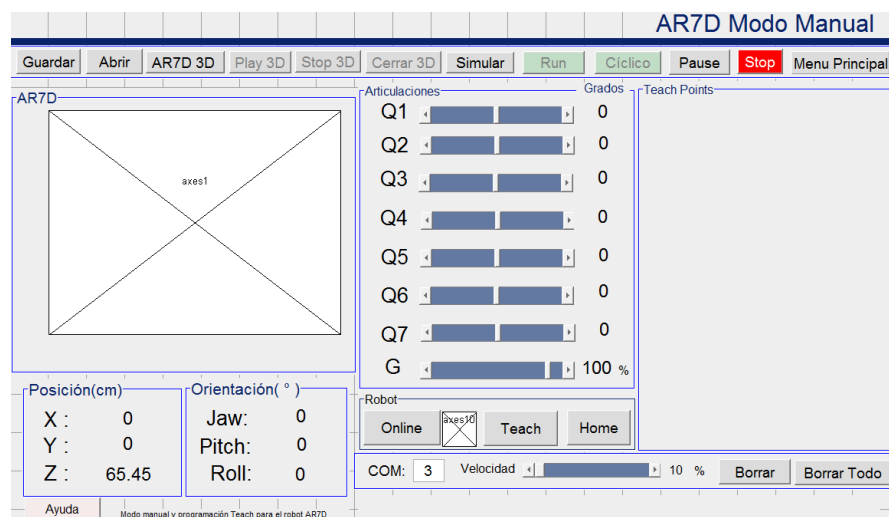


Figura 108. Pantalla Modo Manual.

3.4.5 Pantalla de Programación

Esta pantalla está destinada para programar el robot mediante varios comandos básicos que se utilizan en la programación de robots por instrucciones estos son:

- Home
- Mover
- Mover recta
- Pausa
- Gripper

Al presionar sobre cada uno de estos comandos se programara una instrucción, de acuerdo a la tarea que queramos que el robot realice.

Además cuenta con una barra de herramientas en la parte superior similar a la pantalla de Modo Manual.

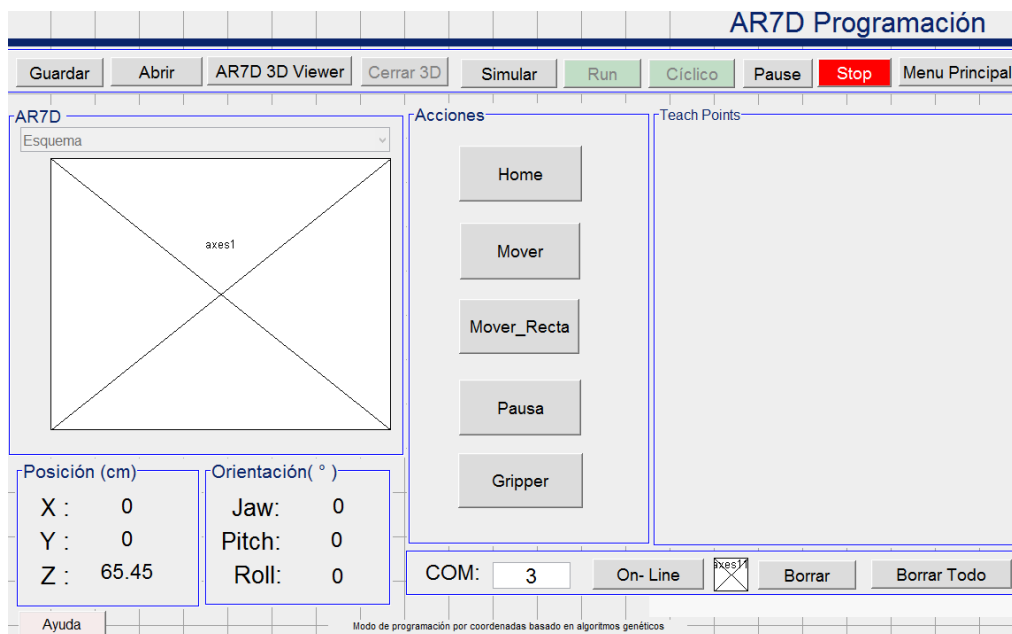


Figura 109. Pantalla -Programación.

3.4.5.1 Pantalla Mover

Al presionar sobre el botón mover en la pantalla de Programación, se abrirá la sub-pantalla Mover, en esta pantalla podremos ingresar los datos de orientación en grados y posición en centímetros, seleccionar una velocidad, y con el botón calcular se resolverá la cinemática inversa mediante un algoritmo genético, también se puede seleccionar un modo el cual realiza un cambio en los parámetros del algoritmo genético y así encontrar otra solución en caso de que la primera no sea satisfactoria, el diagrama de flujo en la Figura 111 nos muestra la lógica de programación.

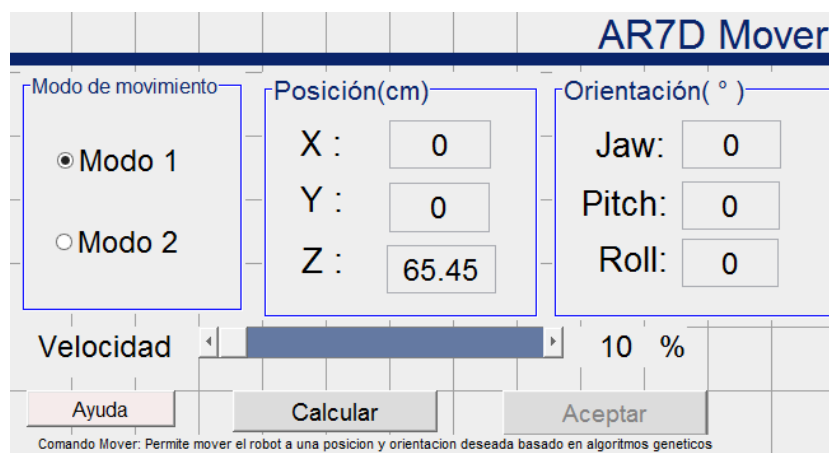


Figura 110. Pantalla -Programación.

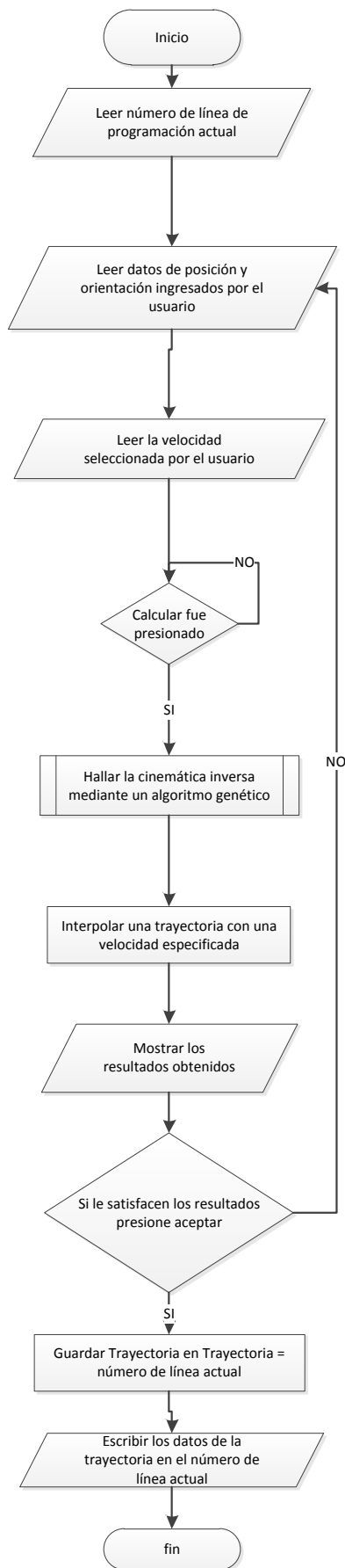


Figura 111. Diagrama de flujo - mover

3.4.5.2 Pantalla Mover recta

Al presionar mover recta en la pantalla de programación, nos llevara a la sub-pantalla mover recta, en la que podemos escribir una posición y una velocidad deseada y al presionar calcular se interpolará y resolverá la cinemática inversa mediante algoritmos genéticos procurando mantener la orientación y siguiendo una trayectoria recta, este comando es muy común en los robots industriales, la lógica de programación de la pantalla mover recta es la misma que la de la pantalla mover.

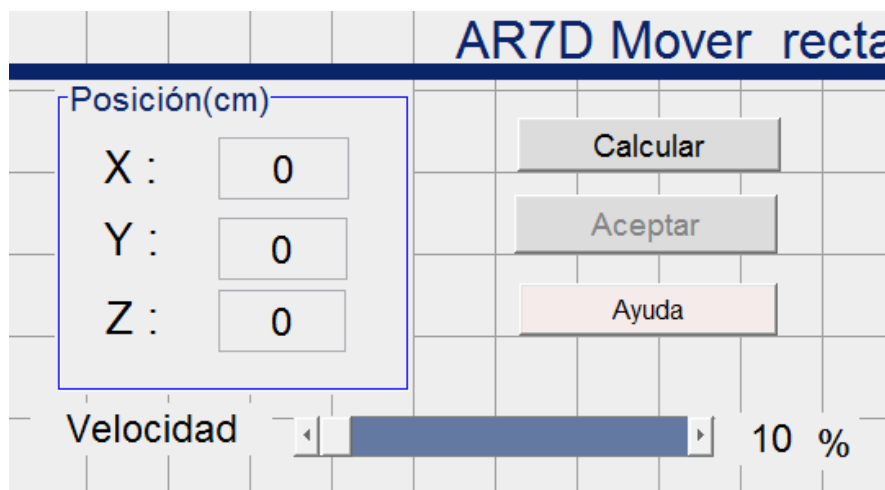


Figura 112. Pantalla – mover recta.

3.4.5.3 Pantalla Pausa

El comando Pausa nos permite programar una pausa de acuerdo a la tarea que se desee que el robot realice

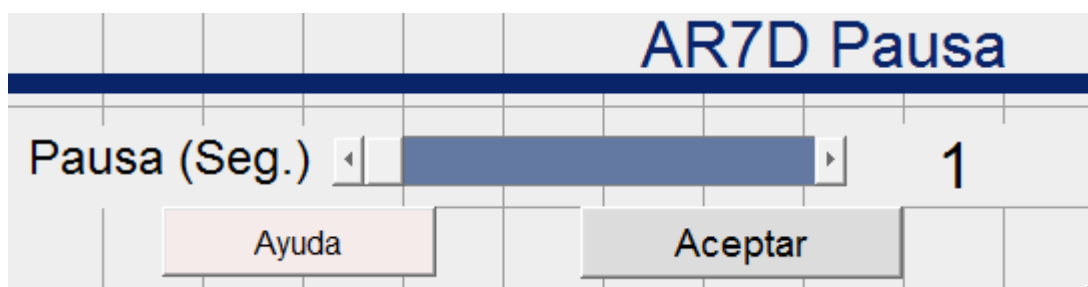


Figura 113. Pantalla – Pausa.

3.4.5.4 Pantalla Gripper

El comando gripper permite controlar la apertura del efector final, en el caso de nuestro robot es una pinza o gripper.

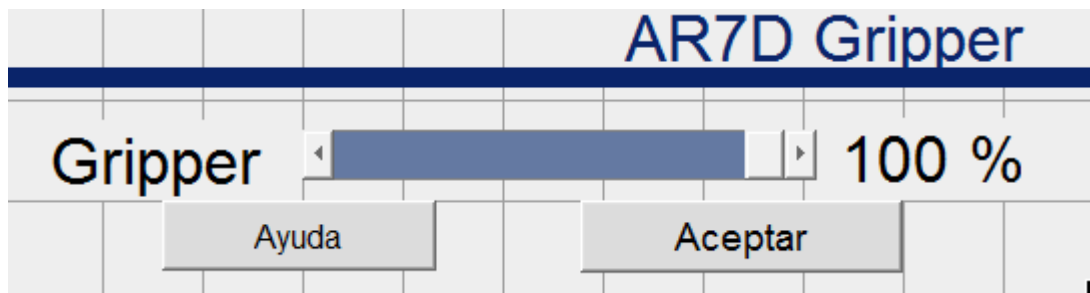


Figura 114. Pantalla – Gripper.

3.4.6 SimMechanics

El robot posee siete grados de libertad y por ello muchas posibilidades de movimiento para alcanzar un punto pese a que el algoritmo genético se diseña para que realice el mínimo cambio articular el robot pueden presentar trayectoria con colisiones que no se observan en el diagrama esquemático de barras, así que se ha utilizado un complemento de Simulink de Matlab denominado SimMechanics, el cual utilizando un ensamblaje en Solidworks permitirá una interacción tridimensional con el modelo en 3D.

En primer lugar debemos ir al sitio web de Matlab y descargar el complemento SimMechanics Link, el cual nos permite unir un modelo CAD a Simulink.

Una vez descargado e instalado el complemento se procede a importar el ensamblaje de Solidworks, en herramientas, SimMechanics Link, Export, SimMechanics First Generation, como se aprecia en la Figura 115.

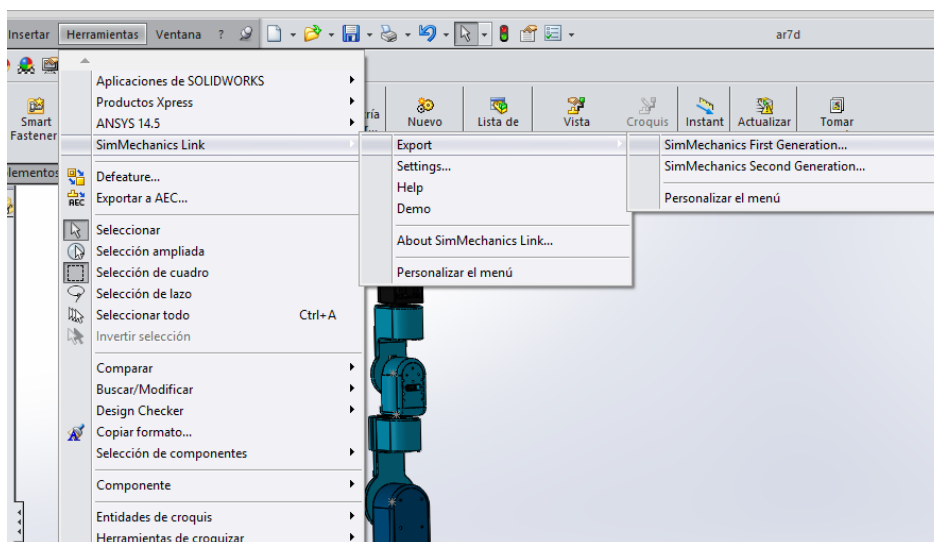


Figura 115. Exportación de un ensamblaje CAD a SimMechanics.

En el área de trabajo de Matlab escribimos el siguiente comando en función del nombre con el que hayamos guardado el ensamblaje: `mech_import('Nombre del ensamblaje.xml')`, y automáticamente se crearan los bloques en Simulink que representan el ensamblaje.

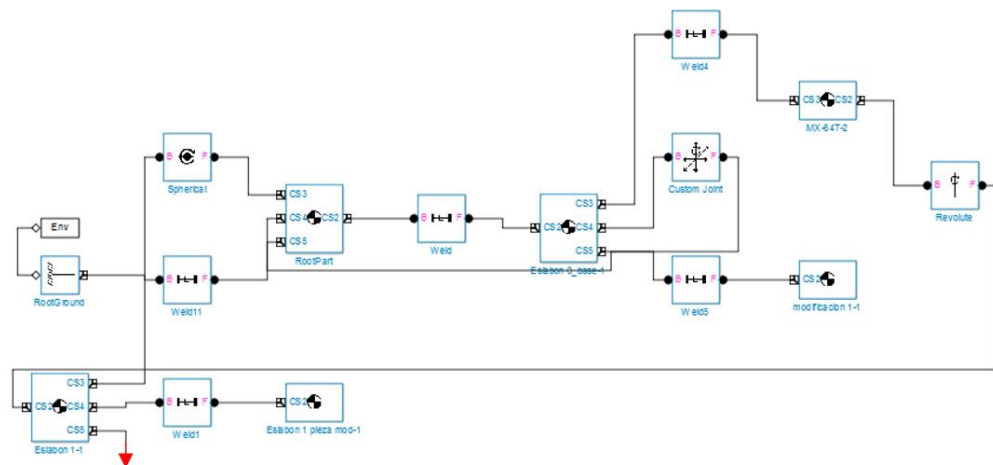


Figura 116. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 1.

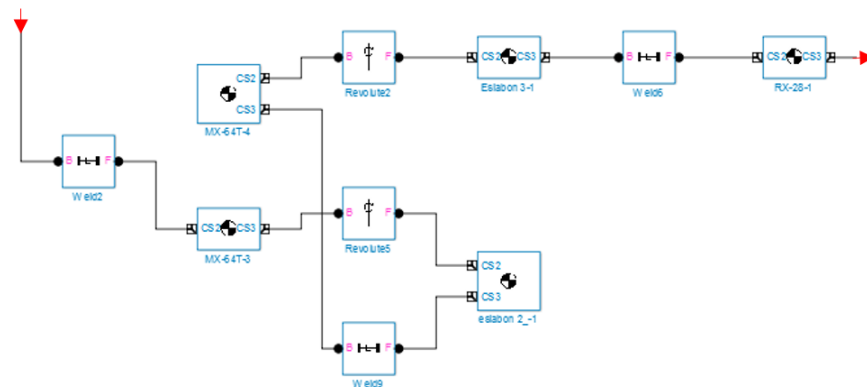


Figura 117. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 2.



Figura 118. Diagrama de bloques en Simulink del ensamblaje importado desde un ensamblaje CAD parte 3.

A continuación se modifican los bloques para que pueda interactuar con nuestro programa, para esto en cada bloque Revolute, en los parámetros del bloque, subimos a 1 en la opción: number of sensor/ actuator ports.

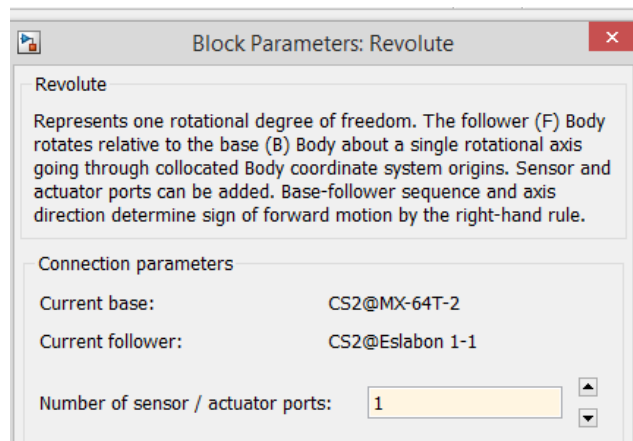


Figura 119. Parámetros del bloque Revolute.

Finalmente en cada uno de las articulaciones del robot le insertamos un bloque Joint Actuator, con los parámetros de posición angular, velocidad angular y aceleración angular como se observa en la Figura 120.

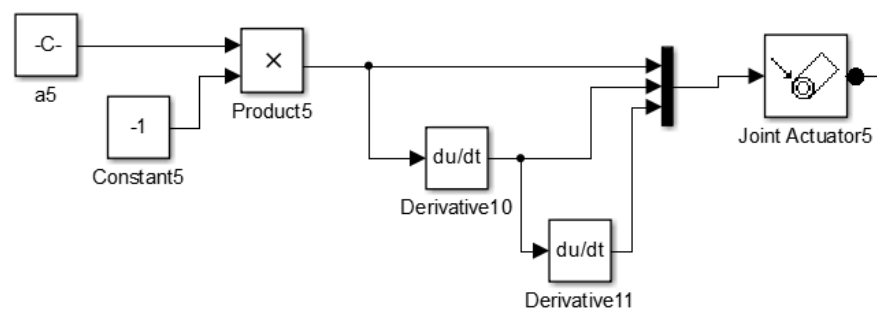


Figura 120. Bloque Joint Actuator con constantes de posición, velocidad y aceleración.

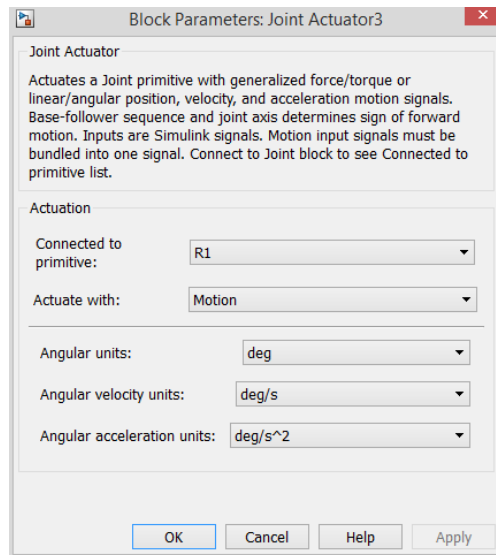


Figura 121. Parámetros del bloque Joint Actuator.

El resultado será un diagrama de bloques en Simulink que representa al robot el cual puede ser controlado mediante bloques de constantes que representen el desplazamiento angular de cada eslabón en la Figura 122 se puede observar el modelo en 3D importado.

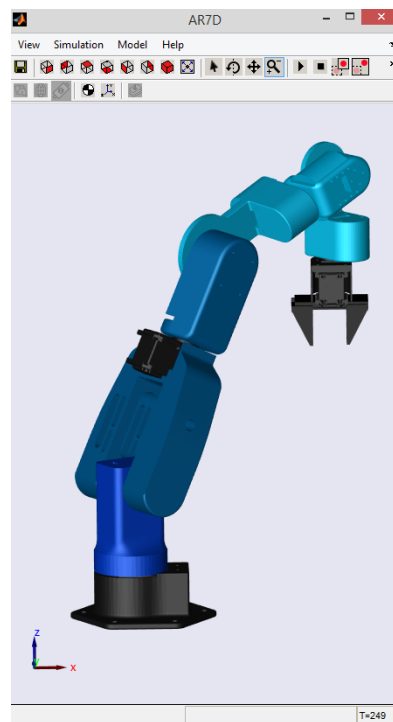


Figura 122. Ensamblaje CAD importado a Matlab.

3.4.7 Programación de Motores Dynamixel

Los motores dynamixel cuentan con una librería que permite su fácil interacción con Matlab, Dynamixel SDK's, una vez agregada al path de Matlab, se puede programar cada uno de los motores dynamixel, cabe recalcar que dentro del computador debe existir un compilador en lenguaje C, por ejemplo: Visual Studio, Turbo C++, Visual Basic, etc, debido a que las librerías están basadas en lenguaje C.

- Lo primero que tenemos que tener en cuenta es la inicialización de la librería que se realiza con el comando: `loadlibrary('dynamixel', 'dynamixel.h');`
- Para inicializar la comunicación serial se utiliza el comando: `calllib('dynamixel','dxl_initialize',DEFAULT_PORTNUM, DEFAULT_BAUDNUM);`
- En el cual hay que especificar el Puerto serial al que está conectado el controlador y el número de baudios a los que se realiza la comunicación en nuestro caso es de Mbps. Para cerrar la comunicación el comando es: `calllib('dynamixel','dxl_terminate');`
- Por ultimo para mover los motores a una posición determinada tenemos que enviar un paquete con toda la información requerida para que la información llegue a todos los motores, cada motor tiene un número de identificación previamente dispuesto.
- La línea de comandos que se debe seguir para transmitir un paquete de instrucciones se puede apreciar en la Figura 123.

```
% Make syncwrite packet
calllib('dynamixel','dxl_set_txpacket_id',BROADCAST_ID);
calllib('dynamixel','dxl_set_txpacket_instruction',INST_SYNC_WRITE);
calllib('dynamixel','dxl_set_txpacket_parameter',0,P_GOAL_POSITION);
calllib('dynamixel','dxl_set_txpacket_parameter',1,2);
for i = 0:1:NUM_ACTUATOR-1
    calllib('dynamixel','dxl_set_txpacket_parameter',2+3*i,id(i+1));
    GoalPos = ((sin(theta+phase(i+1)) + 1) * AmpPos);
    Position(i+1) = int32(GoalPos);
    low = calllib('dynamixel','dxl_get_lowbyte',GoalPos);
    calllib('dynamixel','dxl_set_txpacket_parameter',2+3*i+1,low);
    high = calllib('dynamixel','dxl_get_highbyte',GoalPos);
    calllib('dynamixel','dxl_set_txpacket_parameter',2+3*i+2,high);
end
calllib('dynamixel','dxl_set_txpacket_length',(2+1)*NUM_ACTUATOR+4);
calllib('dynamixel','dxl_tx_packet');
```

Figura 123. Paquete de transmisión de datos de Motores Dynamixel en Matlab.

Fuente: (Trossen Robotis, 2015)

- La instrucción 'dxl_set_txpacket_id', se encuentra en función del número de identificación del servomotor, para incluir a todos los motores se utiliza el número 254.
- La instrucción 'dxl_set_txpacket_instruction', indica la instrucción que se llevara a cabo, para mover todos los motores a la vez se aplica el número 131.
- La instrucción 'dxl_set_txpacket_length', indica el tamaño del paquete y depende del número de actuadores conectados
- La instrucción 'dxl_set_txpacket_parameter', está en función de la acción que se realiza, y la posición dentro del paquete, siendo la posición 0 y 1 reservadas para indicar que la acción es mover el servomotor, los demás parámetros se utilizan para indicar el número de identificación de cada motor seguido de la posición a la que se desea mover el mismo, los números van de 0 a 1023 en los motores AX-12, y de 0 a 4095 en los modelos MX-28 y MX-64.
- Finalmente la instrucción 'dxl_txrx_packet', transmite el paquete y así se logra que todos los motores se muevan a la vez a una posición determinada.

CAPÍTULO 4

ANÁLISIS CINEMÁTICO Y DINÁMICO BASADO EN ALGORITMOS GENÉTICOS.

4.1 Análisis Cinemático.

4.1.1 Parámetros de Denavit Hatenberg

Para empezar con el análisis cinemático necesitamos obtener los parámetros de Denavit Hatenberg que representan al robot como se observa en la Figura 124 y obteniendo como resultado la tabla 14.

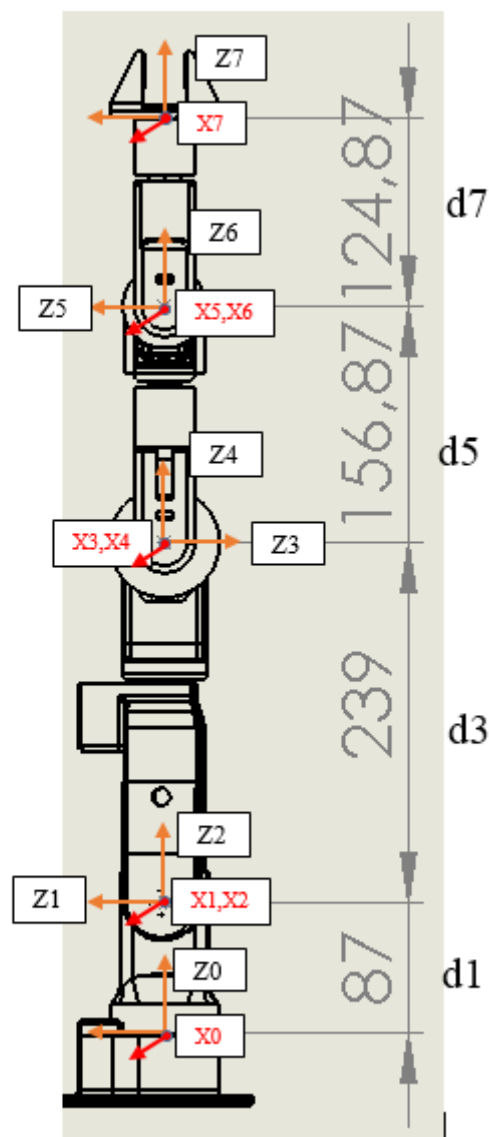


Figura 124. Diagrama que representa los parámetros de Denavit Hatenberg.

Tabla 14.

Parámetros de Denavit Hatenberg del robot AR7D

	a	α	D	θ
Eslabón 1	0	90°	8.700	Q1
Eslabón 2	0	-90°	0.000	Q2
Eslabón 3	0	-90°	23.900	Q3
Eslabón 4	0	90°	0.000	Q4
Eslabón 5	0	90°	15.687	Q5
Eslabón 6	0	-90°	0.000	Q6
Eslabón 7	0	0°	12.487	Q7

4.1.2 Solución de la cinemática inversa mediante un algoritmo genético

Se utilizó el toolbox de Matlab gatool el cual nos permite optimizar y variar los parámetros de algoritmo genético hasta que el resultado sea favorable, el comando en Matlab para obtener la minimización de una función es el siguiente:

```
ga(@función, nvars,[],[],[],[],lb,ub,[],[],options);
```

El algoritmo genético de Matlab trabaja de la siguiente manera:

Recibe parámetros iniciales como numero de variables, restricciones y límites de las variables, los limites serian el máximo y mínimo ángulo que puede alcanzar el servomotor o el ángulo máximo que no produzca una colisión, esto permitirá que las variables estén siempre dentro de este rango y no existan ángulos inalcanzables para el robot

Crea una población inicial que puede ser controlada o aleatoria, evalúa los individuos y selecciona los mejores.

Crea una secuencia de nuevas generaciones tomando en cuenta parámetros como la aptitud, elitismo, generando padres que se cruzan o mutando padres solteros, por ultimo reemplazando la población por la nueva población o hijos.

Se detiene cuando llega a un criterio de parada que puede ser el máximo número de generaciones, haber alcanzado un error aceptable, o que ya no varíen los individuos a lo largo de varias generaciones.

4.1.2.1 Población inicial

La población inicial en este proyecto se genera de manera aleatoria, sin embargo en la opción de generar una trayectoria recta la población inicial son los ángulos en los que se encuentra actualmente el robot para tratar de obtener un mínimo cambio articular y aprovechar la cercanía del punto anterior al punto que se ha interpolado a continuación.

La población inicial se puede controlar mediante el siguiente comando del toolbox gatool de Matlab: `gaoptimiset(options,'Initial Population',IP)`.

4.1.2.2 Restricciones

Una de las ventajas de los algoritmos genéticos es que se puede restringir el espacio de exploración.

En la solución de la cinemática inversa nos permite restringir ángulos que no pueden ser alcanzados físicamente, o ángulos que pueden causar una colisión, las restricciones para el robot se pueden observar en la Tabla 15.

Tabla 15.

Restricciones de las articulaciones del robot.

<i>Articulación (Numerado desde la base hacia el efector final)</i>	<i>Límite Inferior (grados)</i>	<i>Límite Superior (grados)</i>
1	-180	180
2	-80	80
3	-180	180
4	-100	100
5	-150	150
6	-100	100
7	-150	150

4.1.2.3 Selección

Utilizamos la función de selección estocástica uniforme, esta función asigna a cada padre una sección de recta de longitud proporcional a su aptitud, el algoritmo se mueve a lo largo de la recta con pasos iguales y los individuos apuntados serán los que quedan seleccionados como se observa en la Figura 125. Esta es la función de selección predeterminada en el gatool de Matlab.

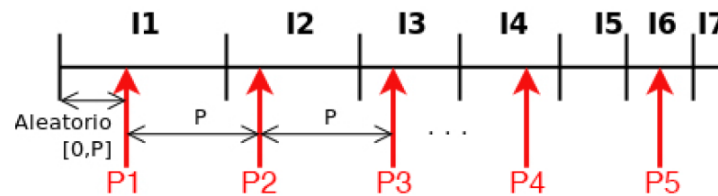


Figura 125. Selección uniforme estocástica.

Fuente: (Valles Pérez, s.f.)

4.1.2.4 Mutación

La Función de mutación que utilizamos es mutación gaussiana que cambia en los padres un gen seleccionado por un número aleatorio tomado de una distribución gaussiana o campana de gauss con una media de cero y desviación variable que disminuye según avanza el número de generaciones. A través de esto se obtienen individuos con otros valores, que pueden o no mejorar su aptitud, con lo cual se explora un mayor campo de probabilidades.

4.1.2.5 Cruzamiento

La función de cruzamiento utilizada se denomina Dispersos, crea una cadena binaria aleatoria, los elementos de esta cadena que son de valor 1 toman los valores del primer padre mientras que los elementos de la cadena de valor 0 toman los valores del segundo padre.

Ejemplo:

- Primer padre: [a b c d e f]
- Segundo padre: [u v w x y z]
- Cadena binaria aleatoria: [1 0 0 1 1 0]
- Hijo resultante: [a v w d e z]

4.1.2.6 Función de aptitud

La Función de aptitud o función objetivo es muy importante en un algoritmo genético, siendo los individuos más óptimos los que maximicen o minimicen la función de acuerdo a como este diseñado el algoritmo, en Matlab el gatool trata de minimizar la función.

El diseño de la función de aptitud en este caso parte de tres objetivos a alcanzar, la posición del robot, la orientación del robot y el mínimo cambio articular.

El primer paso consiste en poner la matriz de transformación en función de los parámetros de entrada, estos son posición en x, y, z, orientación en los ángulos de Euler pitch, yaw, roll.

Pitch (θ): gira el sistema con respecto al eje Y

Yaw (ψ): gira el sistema con respecto al eje Z

Roll (ϕ): gira el sistema con respecto al eje X

$$Tf = \begin{bmatrix} C\phi C\theta & C\phi S\theta S\psi - S\phi C\psi & C\phi S\theta C\psi + S\phi S\psi & x \\ S\phi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\phi S\theta C\psi - C\phi S\psi & y \\ -S\theta & C\theta C\psi & C\theta S\psi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

La matriz de transformación Tf, representa los parámetros que queremos alcanzar y será constante dentro del algoritmo genético, ahora debemos hallar la matriz de transformación que variará hasta acercarse a la matriz deseada Tf.

Desarrollamos la matriz homogénea para cada eslabón mediante los parámetros de Denavit Hartenberg.

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Con los parámetros de Denavit Hartenberg antes encontrados las matrices para cada eslabón son:

$${}^0A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & 8.7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} C\theta_2 & 0 & -S\theta_2 & 0 \\ S\theta_2 & 0 & C\theta_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} C\theta_3 & 0 & -S\theta_3 & 0 \\ S\theta_3 & 0 & C\theta_3 & 0 \\ 0 & -1 & 0 & 23.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4 = \begin{bmatrix} C\theta_4 & 0 & S\theta_4 & 0 \\ S\theta_4 & 0 & -C\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4A_5 = \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & 15.687 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5A_6 = \begin{bmatrix} C\theta_6 & 0 & -S\theta_6 & 0 \\ S\theta_6 & 0 & C\theta_6 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^6A_7 = \begin{bmatrix} C\theta_7 & -S\theta_7 & 0 & 0 \\ S\theta_7 & C\theta_7 & 0 & 0 \\ 0 & 0 & 1 & 12.487 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplicando las matrices de cada eslabón se obtiene la matriz de transformación homogénea del sistema:

$$T = {}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * {}^4A_5 * {}^5A_6 * {}^6A_7 \quad (7)$$

Por ultimo diseñamos la función de aptitud en términos de las matrices Tf y T, para lo cual consideramos:

1. Error de posición (epos):

Es la diferencia entre los términos de posición de las matrices de transformación Tf y T.

$$\begin{aligned} epos = & [Tf(1,4) - T(1,4)]^2 + [Tf(2,4) - T(2,4)]^2 \\ & + [Tf(3,4) - T(3,4)]^2 \end{aligned} \quad (8)$$

2. Error de orientación (eor)

Es la diferencia entre los términos de orientación de las matrices de transformación Tf y T.

$$\begin{aligned} eor = & [Tf(1,1) - T(1,1)]^2 + [Tf(1,2) - T(1,2)]^2 + [Tf(1,3) - T(1,3)]^2 + \dots \\ & [Tf(2,1) - T(2,1)]^2 + [Tf(2,2) - T(2,2)]^2 + [Tf(2,3) - T(2,3)]^2 + \dots \\ & [Tf(3,1) - T(3,1)]^2 + [Tf(3,2) - T(3,2)]^2 + [Tf(3,3) - T(3,3)]^2 \end{aligned} \quad (9)$$

En las cuales Tf (k, j) y T (k, j) representan los elementos de la fila k, columna j de cada una de las matrices Tf y T respectivamente.

3. Mínimo cambio articular (mca).

El mínimo cambio articular está en función de la posición anterior y la posición actual, al minimizar esta función garantizamos que el robot tenga el mínimo cambio articular posible con respecto a la posición previa.

$$mca = \sum_{i=1}^7 (q_0(i) - q(i))^2 \quad (10)$$

Donde:

q_0 = ángulo de la posición anterior

q = ángulo de la posición actual

i = número de articulación

Para la función de aptitud utilizamos la función exponencial de manera que los valores resultantes de la función de aptitud no crezcan hasta el infinito, el máximo que puede alcanzar es uno en el caso de que el radical de como resultado infinito. Finalmente la función de aptitud o función objetivo queda definida por:

$$f_{aptitud} = e^{-\frac{1}{\sqrt{e_{pos} + C * eor + mca}}} \quad (11)$$

Donde C es una constante que escala al error de orientación debido a que la submatriz de orientación produce valores muy pequeños y pueden generar errores en orientación.

Por experimentación obtuvimos que el valor de $C=200$, porque escala de manera adecuada a los términos de orientación, como se observa en la Tabla 16

Se trata de alcanzar la posición: $X=15$, $Y=15$, $Z=15$, y orientación: $Yaw=90^\circ$, $Pitch=60^\circ$, $Roll=30^\circ$.

Tabla 16.

Variación de la constante C para sintonizar el error de orientación.

<i>Generaciones:</i>		<i>1000</i>				
Tamaño de la población		120				
<i>C</i>	<i>X (cm)</i>	<i>Y (cm)</i>	<i>Z (cm)</i>	<i>Yaw (°)</i>	<i>Pitch (°)</i>	<i>Roll (°)</i>
1	15.12	15.01	15.10	-151.61	31.85	-139.67
50	15.25	15.25	15.21	86.31	61.07	32.90
100	15.20	15.23	15.18	88.09	60.49	31.60
150	15.27	15.30	15.21	88.74	60.45	31.06
200	15.29	15.29	15.22	89.20	60.34	30.61
250	15.32	15.28	15.25	89.31	60.27	30.54

En el experimento anterior podemos observar que al aumentar la constante C mejora la orientación, y aumenta el error de posición aunque no significativamente, los mejores resultado se obtuvieron con $C=150$, $C=200$ y $C=250$, por lo cual hemos decidido tomar el valor medio que es $C=200$, teniendo buenos resultados en el desempeño del algoritmo genético.

4.1.2.7 Tamaño de la población

El tamaño de la población corresponde al número de individuos por cada generación, para escoger el adecuado tamaño de la población se tiene que analizar el espacio que debe ser explorado, para un espacio pequeño bastará con una población de pocos individuos.

Nuestro problema tiene que explorar un espacio amplio debido a sus siete variables y a todas las posibles combinaciones de ángulos en las articulaciones que puedan existir, a mayor población mayor posibilidad de encontrar los mejores individuos, sin embargo esto genera costo computacional y tiempo de procesamiento.

Hemos encontrado un buen tamaño de población en un rango de 100 a 200 individuos por generación como se puede observar en la Tabla 17.

Se trata de alcanzar la posición: $X=20$, $Y=20$, $Z=20$, y orientación: $Yaw=30^\circ$, $Pitch=30^\circ$, $Roll=30^\circ$, variando el tamaño de la población.

Tabla 17.

Variación del tamaño de la población.

Generaciones	1000						
<i>Tamaño de la población</i>	<i>X(cm)</i>	<i>Y(cm)</i>	<i>Z(cm)</i>	<i>Yaw(°)</i>	<i>Pitch(°)</i>	<i>Roll(°)</i>	<i>Tiempo(s)</i>
10	20.14	20.22	20.08	29.79	29.98	30.37	7.66
50	20.17	20.16	20.11	30.08	29.85	29.98	10.67
100	20.19	20.16	20.13	29.98	29.90	30.02	15.12
120	20.18	20.14	20.13	29.90	30.06	30.09	16.40
200	20.19	20.16	20.13	30.00	29.95	30.03	23.15
500	20.19	20.17	20.13	29.94	29.92	30.07	48.11

El valor que tomamos fue de 120 individuos por generación ya que presenta una buena exploración del espacio se puede observar una mejoría sobretodo en orientación al aumentar el tamaño de la población.

El tiempo de computación a 120 individuos es de 16 segundos, un tiempo que consideramos adecuado, en el toolbox gatool de Matlab el comando para especificar el tamaño de la población es: `gaoptimset (options,'PopulationSize', 120);`

4.1.2.8 Generaciones

El número de generaciones significa las veces que una población vuelve a regenerarse. El máximo número de generaciones es también un criterio de parada, por la dificultad de resolver la cinemática inversa de un robot de siete grados de libertad hay que considerar un alto número de generaciones.

La Tabla 18 muestra la variación del algoritmo al incrementar las generaciones.

Se trata de alcanzar la posición: X=20, Y=15, Z=10, y orientación: Yaw=45°, Pitch=45°, Roll=45°, variando el número de generaciones.

Tabla 18.

Variación del número de generaciones.

<i>Tamaño de la población</i>		<i>120</i>					
Generaciones	X(cm)	Y(cm)	Z(cm)	Yaw(°)	Pitch(°)	Roll(°)	Tiempo(s)
10	28.44	14.84	18.95	12.06	66.81	75.28	0.80
50	20.01	10.63	11.51	49.76	57.33	36.88	1.30
100	20.51	15.41	10.08	44.64	44.48	45.21	2.10
500	20.27	15.19	10.10	44.58	45.11	45.33	8.01
1000	20.28	15.10	10.08	45.05	44.83	44.86	16.47
2000	19.98	15.01	9.99	45.04	45.03	44.94	35.00

Se observa que mientras mayor sea el número de generaciones, el algoritmo mejora su exactitud. Concluimos que 1000 generaciones es un límite adecuado para resolver la cinemática inversa en un tiempo relativamente corto.

4.2 Generación de trayectorias.

Una vez obtenida la cinemática inversa hay que interpolar los puntos y hallar una trayectoria que nos dará datos de posición, velocidad y aceleración en función del tiempo.

4.2.1.1 Interpolación mediante polinomios

La interpolación mediante polinomios nos permite obtener una trayectoria en la cual tenemos varios puntos de restricción, en este caso conocemos la posición inicial y gracias a la cinemática inversa conocemos la posición final, además debemos asegurarnos de que la velocidad y la aceleración sean continuas.

El complemento Robotics toolbox de Matlab tiene una función que genera una trayectoria suave en términos de los ángulos de las articulaciones en la posición inicial y final, la función en Matlab es: $[Q \ QD \ QDD] = \text{jtraj}(Q0, Q1, T)$;

Donde $Q0$ es un vector con los ángulos de las articulaciones en la posición inicial, $Q1$ es un vector con los ángulos de las articulaciones en la posición final, T es un vector que representa el tiempo. Los datos que se obtienen son tres matrices que representan la posición Q , velocidad QD y aceleración QDD en cada instante de tiempo T .

El comando jtraj utiliza un polinomio de grado siete, que permite que se cumplan las condiciones de velocidad inicial y final igual a 0, y aceleración inicial y final igual a 0, estos parámetros pueden ser modificados si se requiere.

4.2.1.2 Seguimiento de una Trayectoria recta

El seguir una trayectoria recta es una herramienta muy útil en un robot manipulador, pero es más complicado lograr que se cumpla este objetivo en un robot que tiene solamente articulaciones rotacionales y más aún si se trata de siete grados de libertad.

Para lo cual vamos a aprovechar una ventaja al modificar los parámetros del algoritmo genético.

En primer lugar dividimos la trayectoria recta deseada en tres segmentos iguales obteniendo una posición inicial conocida (P1) y tres posiciones desconocidas (P2, P3 P4) que tienen en común la misma orientación.



Figura 126. Trayectoria recta dividida en cuatro puntos.

Realizamos la cinemática inversa mediante el algoritmo genético en cada posición desconocida, pero modificando ciertos parámetros del algoritmo.

La modificación más importante es la población inicial. En el algoritmo genético que utilizamos previamente la población inicial es aleatoria, en este caso la población inicial tiene que ser la posición anterior, sabemos que P1 ya alcanzó la posición y orientación deseada y conocemos los valores de la cinemática inversa, entonces a partir de este punto, el algoritmo tardará menos en converger ya que la orientación fue alcanzada y la posición es relativamente cercana al punto P2, gracias a esto se utiliza menos generaciones y menos tiempo para hallar la solución.

Otra ventaja es la función de mínimo cambio articular que ayuda a que la posición del robot en P1 sea similar a la posición en P2.

Para encontrar la posición P3, se toma como posición inicial la posición P2 obtenida anteriormente.

Este proceso se repite hasta obtener las tres posiciones desconocidas y como la orientación se mantiene, el movimiento que realiza el robot se aproxima a una trayectoria recta.

4.3 Análisis dinámico.

4.3.1.1 Parámetros dinámicos del robot

Una vez obtenida la cinemática inversa e interpolada una trayectoria se obtiene valores de posición angular en cada eslabón con respecto al tiempo, al derivar la posición obtenemos la velocidad angular y la aceleración angular de cada eslabón en cada instante de tiempo.

Para analizar la dinámica del robot además de conocer la cinemática es necesario encontrar también la distribución de masa del robot y los momentos de inercia en cada eslabón.

Se tomó los datos de inercia, coordenadas del centro de gravedad y masas de cada uno de los eslabones que conforman el robot ayudados del Software CAD.

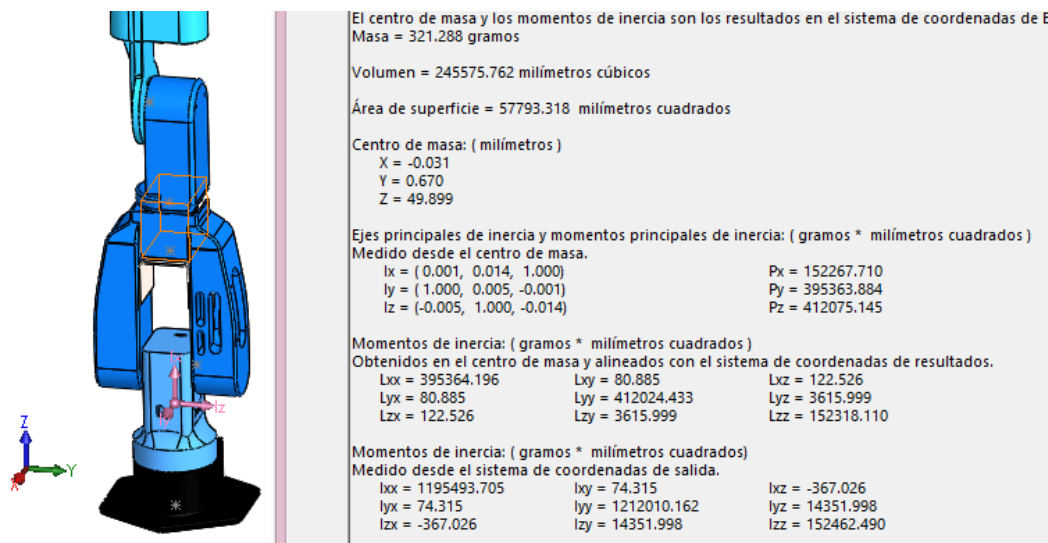


Figura 127. Masa, centros de gravedad y momentos de inercia del eslabón 1.

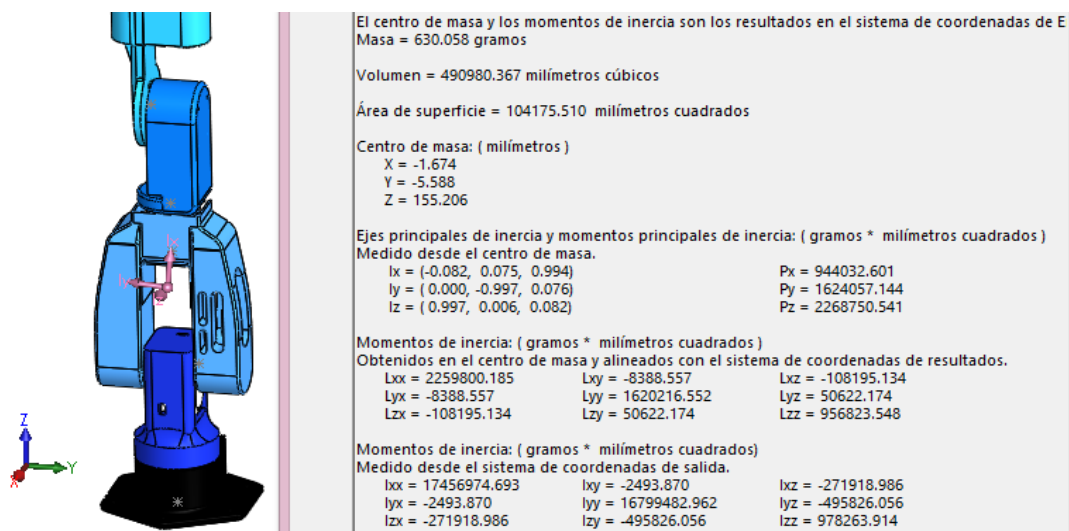


Figura 128. Masa, centros de gravedad y momentos de inercia del eslabón 2.

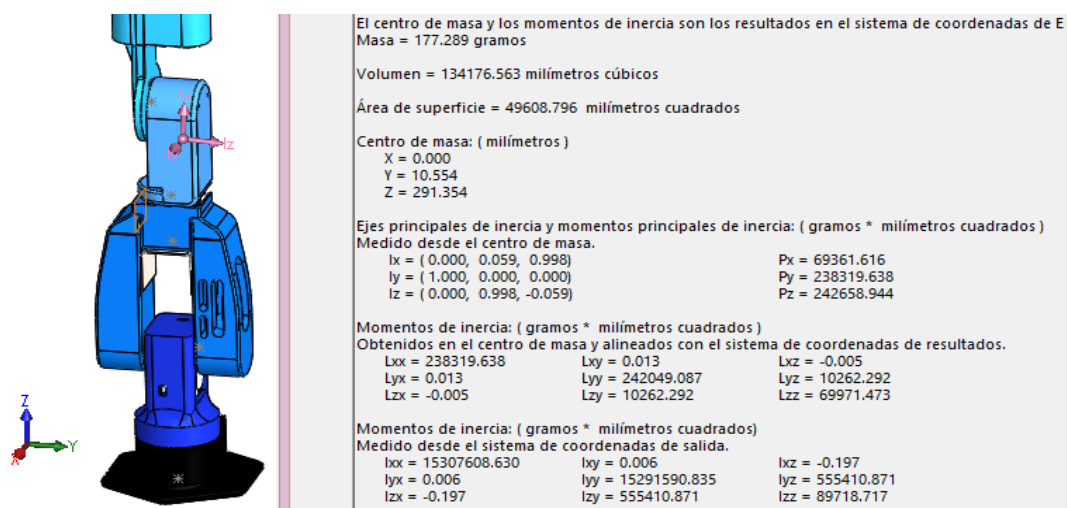


Figura 129. Masa, centros de gravedad y momentos de inercia del eslabón 3.

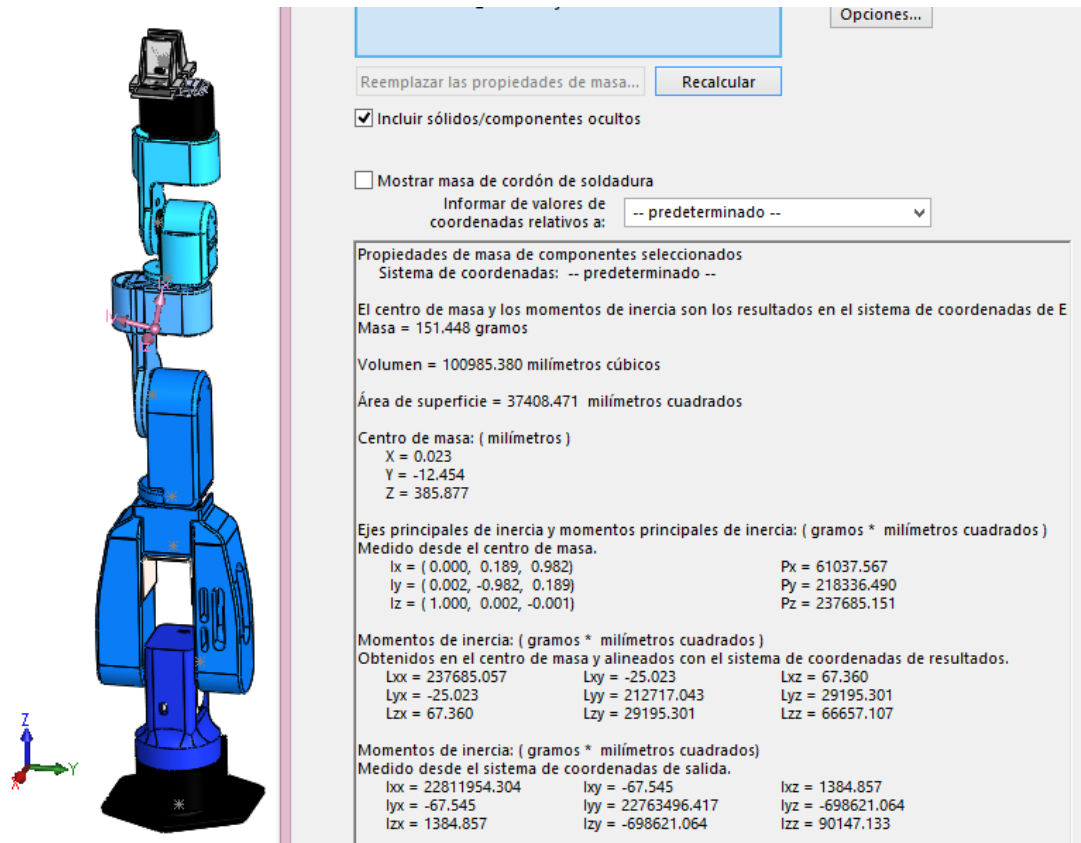


Figura 130. Masa, centros de gravedad y momentos de inercia del eslabón 4.

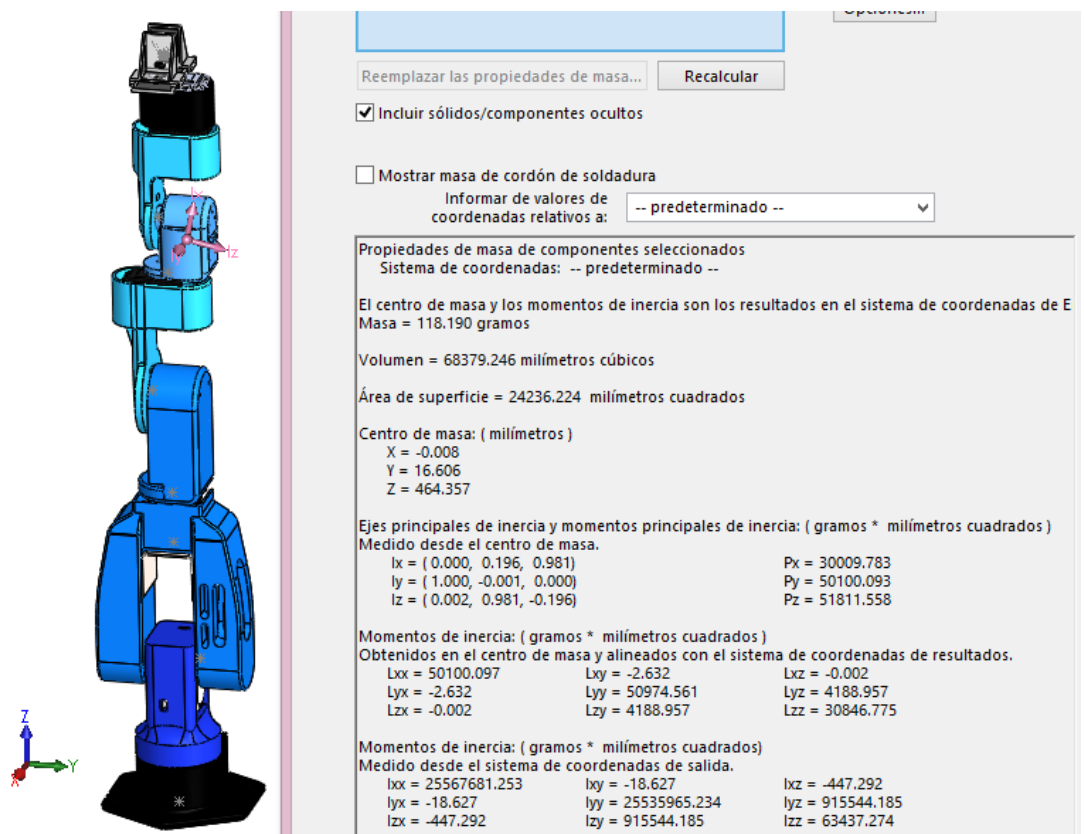


Figura 131. Masa, centros de gravedad y momentos de inercia del eslabón 5.

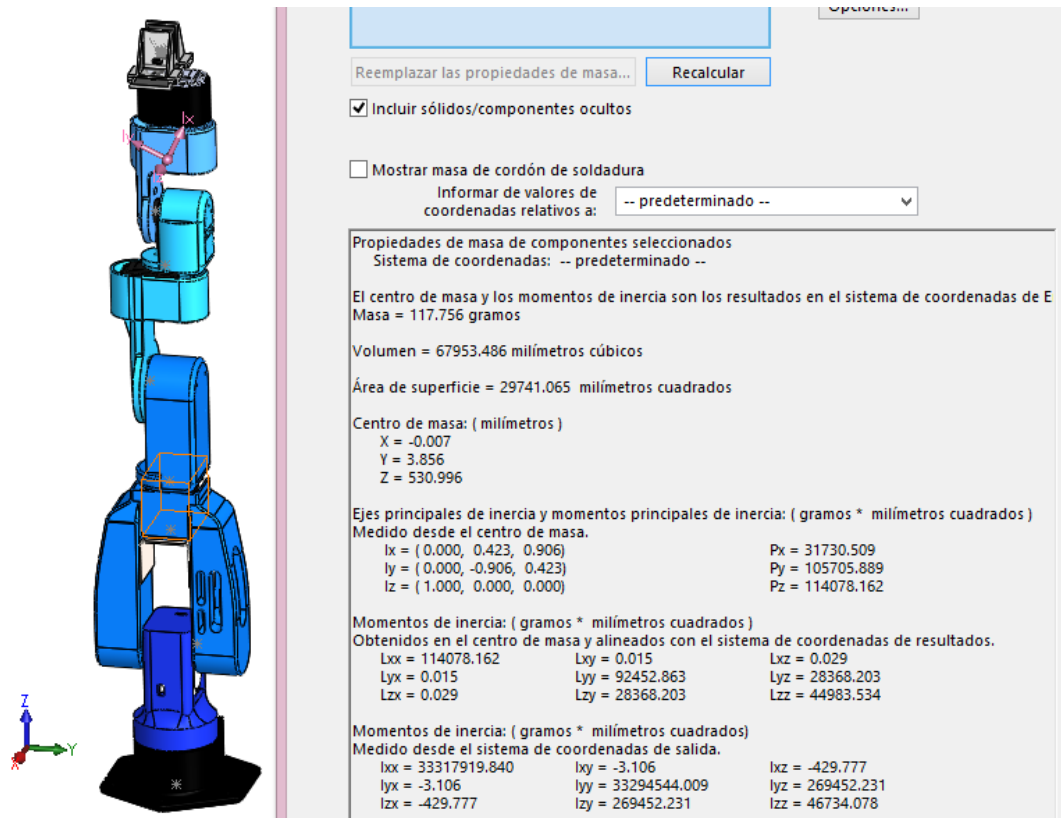


Figura 132. Masa, centros de gravedad y momentos de inercia del eslabón 6.

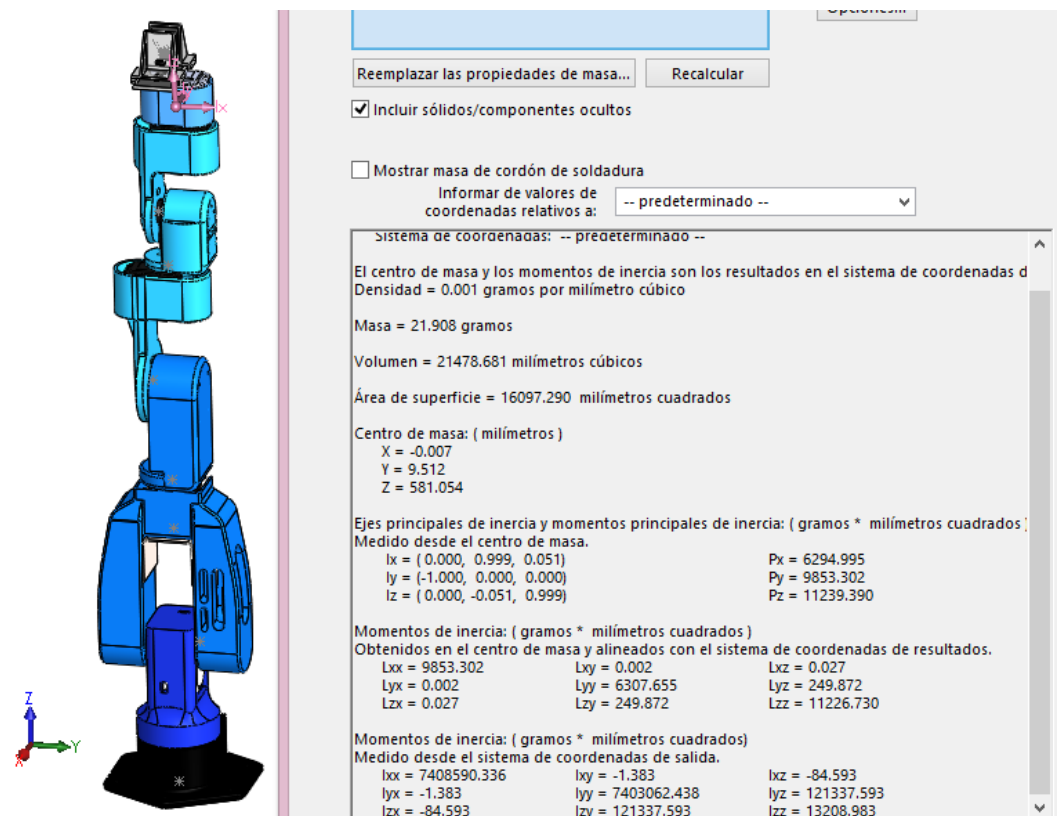


Figura 133. Masa, centros de gravedad y momentos de inercia del soporte del efector final.

4.3.1.2 Método Recursivo de Newton-Euler

El complemento de Matlab Robotics toolbox de Pete Corke posee una función muy buena como herramienta para el análisis dinámico de una trayectoria.

La función es en Matlab es: $\tau = \text{rne}(\text{ROBOT}, q, \dot{q}, \ddot{q}, g)$; y resuelve la dinámica inversa mediante el método recursivo de Newton-Euler que esta descrito por las siguientes ecuaciones:

La ley de Newton Euler para el equilibrio de fuerzas está definida por la ecuación (1), y para el equilibrio de momentos en la ecuación (2)

Se obtienen las matrices de rotación ${}^{i-1}R_i$

Y sus inversas ${}^iR_{i-1} = ({}^{i-1}R_i)^{-1} = ({}^{i-1}R_i)^T$

$${}^{i-1}R_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i \\ 0 & S\alpha_i & C\alpha_i \end{bmatrix} \quad (12)$$

Condiciones iniciales:

Para la base $\{S_0\}$:

$${}^0\omega_0: \text{Velocidad angular} = [0,0,0]^T$$

$${}^0\dot{\omega}_0: \text{Aceleración angular} = [0,0,0]^T$$

$${}^0v_0: \text{Velocidad lineal} = [0,0,0]^T$$

$${}^0\dot{v}_0: \text{Aceleración lineal} = [g_x, g_y, g_z]^T$$

En el extremo del robot:

$${}^{n+1}f_{n+1}: \text{Fuerza ejercida externamente}$$

$${}^{n+1}n_{n+1}: \text{Par ejercido externamente}$$

$$z_0 = [0,0,1]^T$$

En cada sistema:

$${}^i p_i: \text{Coordenadas del origen del sistema } \{S_i\} \text{ respecto a } \{S_{i-1}\} = [a_i, d_i S_i, d_i C_i]$$

$${}^i s_i: \text{Coordenadas del centro de masas del eslabón } i \text{ respecto del sistema } \{S_i\}$$

${}^i I_i$: Matriz de inercia del eslabón i respecto de su centro de masas en $\{S_i\}$

Para los eslabones $i = 1 \dots n$, excluyendo la base:

Velocidad angular del sistema $\{S_i\}$

$${}^i \omega_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} \omega_{i-1} + z_0 \dot{q}_i) & \text{si el eslabón } i \text{ es de rotación} \\ {}^i R_{i-1} {}^{i-1} \omega_{i-1} & \text{si el eslabón } i \text{ es de traslación} \end{cases} \quad (13)$$

Aceleración angular del sistema $\{S_i\}$

$${}^i \dot{\omega}_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} \dot{\omega}_{i-1} + z_0 \ddot{q}_i) + {}^{i-1} \omega_{i-1} \times z_0 \dot{q}_i & \text{si el eslabón } i \text{ es de rot.} \\ {}^i R_{i-1} {}^{i-1} \dot{\omega}_{i-1} & \text{si el eslabón } i \text{ es de tras.} \end{cases} \quad (14)$$

Aceleración lineal del sistema i

$${}^i \dot{v}_i = \begin{cases} {}^i \dot{\omega}_i \times {}^i p_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i p_i) + {}^i R_{i-1} {}^{i-1} \dot{v}_{i-1} & \text{rotación} \\ {}^i R_{i-1} (z_0 \ddot{q}_i + {}^{i-1} \dot{v}_{i-1}) + {}^i \dot{\omega}_i \times {}^i p_i + \\ 2 {}^i \omega_i \times {}^i R_{i-1} z_0 \dot{q}_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i p_i) & \text{traslación.} \end{cases} \quad (15)$$

Aceleración lineal del centro de gravedad del eslabón i :

$${}^i a_i = {}^i \dot{\omega}_i \times {}^i S_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i S_i) + {}^i \dot{v}_i \quad (16)$$

Para todos los eslabones: ($i=n \dots 1$)

Fuerza ejercida sobre el eslabón i :

$${}^i a_i = {}^i \dot{\omega}_i \times {}^i S_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i S_i) + {}^i \dot{v}_i \quad (17)$$

Par ejercido sobre el eslabón i :

$${}^i n_i = {}^i R_{i+1} [{}^{i+1} n_i + ({}^{i+1} R_i {}^i p_i) \times {}^{i+1} f_{i+1}] + ({}^i p_i + {}^i s_i) \times m_i {}^i a_i + {}^i I_i {}^i \dot{\omega}_i + {}^i \omega_i \times ({}^i I_i {}^i \omega_i) \quad (18)$$

Fuerza o par aplicado a la articulación i

$$\tau_i = \begin{cases} {}^i n_i^T {}^i R_{i-1} z_0 & \text{si el eslabón } i \text{ es de rotación} \\ {}^i f_i^T {}^i R_{i-1} z_0 & \text{si el eslabón } i \text{ es de traslación} \end{cases} \quad (19)$$

Los parámetros de entrada son el objeto ROBOT que contiene toda la información requerida para la dinámica descrita en la Tabla 19 y en la Tabla 20, donde m es la masa, r_x , r_y , r_z son las coordenadas del centro de gravedad de cada eslabón, y los demás parámetros representan los momentos de inercia del eslabón con respecto al centro de gravedad.

Tabla 19.

Masas y coordenadas del centro de gravedad de los eslabones del robot AR7D.

<i>Eslabón</i>	<i>m (g)</i>	<i>Rx (mm)</i>	<i>Ry (mm)</i>	<i>Rz (mm)</i>
1	321.270	-0.031	0.670	49.900
2	630.060	-1674.000	-5.588	155.206
3	177.290	0.000	10.554	291.354
4	151.450	0,023	-12.454	385.877
5	118.190	-0,008	16.600	464.360
6	117.760	-0,070	3.856	530.990
7	21.900	-0,070	9.510	581.050

Tabla 20.

Momentos de inercia de los eslabones del robot AR7D

<i>Eslabón</i>	<i>Ixx (gr.mm²)</i>	<i>Iyy (gr.mm²)</i>	<i>Izz (gr.mm²)</i>	<i>Ixy (gr.mm²)</i>	<i>Iyz (gr.mm²)</i>	<i>Ixz (gr.mm²)</i>
1	395364.190	3615.990	152318.110	80.885	3615.990	122.526
2	2259800.100	1620216.500	956823.54	-8388.557	50622.174	-108195.130
3	238319.640	242049.090	69971.470	0.013	10262.290	-0.005
4	237685.060	212717.040	66657.100	-25.020	29195.300	67.360
5	50100.097	50974.561	30846.775	-2.632	4188.960	-0.002
6	114078.200	92452.860	44983.534	0.015	28368.200	0.029
7	9853.300	7403062.400	11226.730	0.002	249.872	0.027

El siguiente parámetro de entrada es una matriz que representa la posición angular de cada eslabón, después una matriz que representa la velocidad angular de cada eslabón.

El tercer parámetro es una matriz que representa la aceleración angular de cada eslabón y finalmente se agrega un vector que representa la gravedad.

La función `rne` de Matlab nos devuelve como resultado una matriz que representa el torque que requiere cada eslabón del robot en cada instante de tiempo.

En el siguiente ejemplo se puede observar tanto el análisis cinemático como el análisis dinámico de una trayectoria requerida.

4.4 Ejemplo del análisis cinemático y dinámico.

Se trata de alcanzar la posición: $X=20$, $Y=10$, $Z=20$, y orientación: $Yaw=30^\circ$, $Pitch=60^\circ$, $Roll=45^\circ$.

Se aplicó el algoritmo genético para resolver a cinemática inversa obteniendo los resultados que se muestra en la Tabla 21 y en la Tabla 22, la posición final del robot se puede observar en las Figura 134 y en la Figura 135.

Tabla 21.

Posición y orientación alcanzada por el algoritmo genético.

Generaciones:	1000				
Tamaño de la población	120				
X (cm)	Y (cm)	Z (cm)	Yaw ($^\circ$)	$Pitch$ ($^\circ$)	$Roll$ ($^\circ$)
20.2	10.1	20.2	29.4	60.3	45.6

Tabla 22.

Cinemática inversa resuelta por algoritmo genético.

Q1	Q2	Q3	Q4	Q5	Q6	Q7
42°	-28°	20°	128°	70°	113°	50°

A continuación se interpolo una trayectoria entre el punto inicial home $X=0$, $Y=0$, $Z=65.45$, y orientación: $Yaw=0^\circ$, $Pitch=0^\circ$, $Roll=0^\circ$ y el punto obtenido mediante el algoritmo genético. La Figura 136 muestra la posición angular de cada eslabón respecto al tiempo, numerados desde la base hasta el efector final.

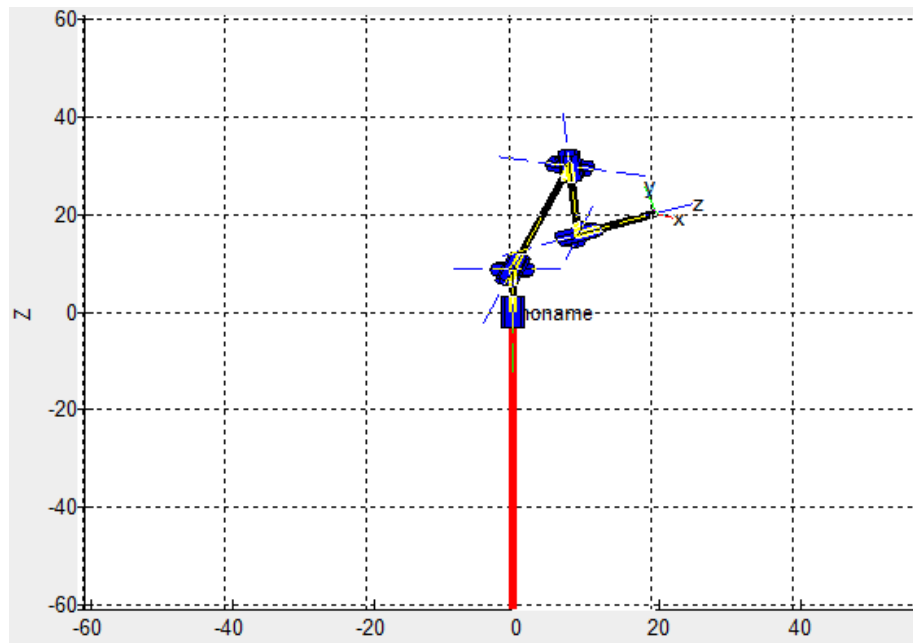


Figura 134. Posición final del robot AR7D para el ejemplo en el plano X-Z.



Figura 135. Posición final del robot AR7D para el ejemplo en vista 3D.

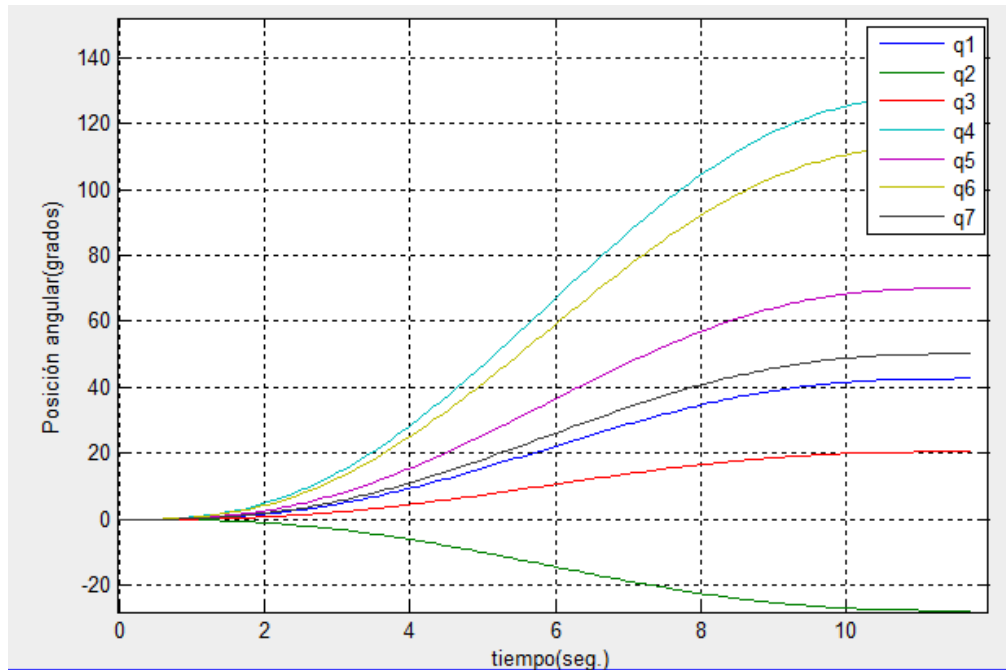


Figura 136. Posición angular.

Una vez obtenidos los resultados de la cinemática inversa e interpolada una trayectoria hallamos la velocidad angular (Figura 137), aceleración angular (Figura 138) y calculamos la dinámica inversa para hallar el torque requerido por cada uno de los eslabones del robot (Figura 139).

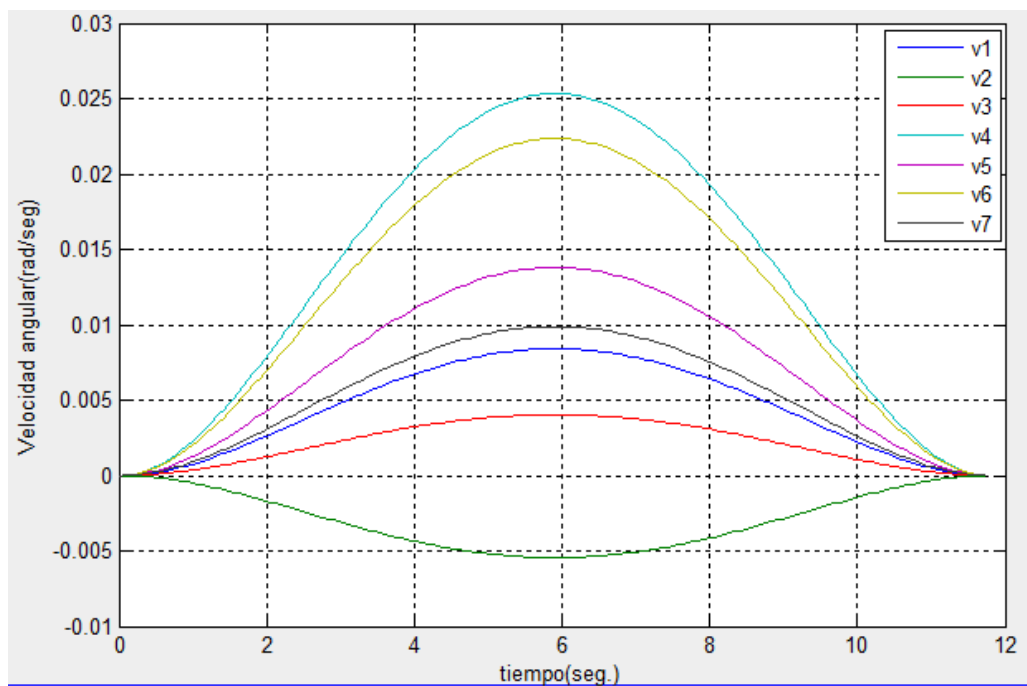


Figura 137. Velocidad angular.

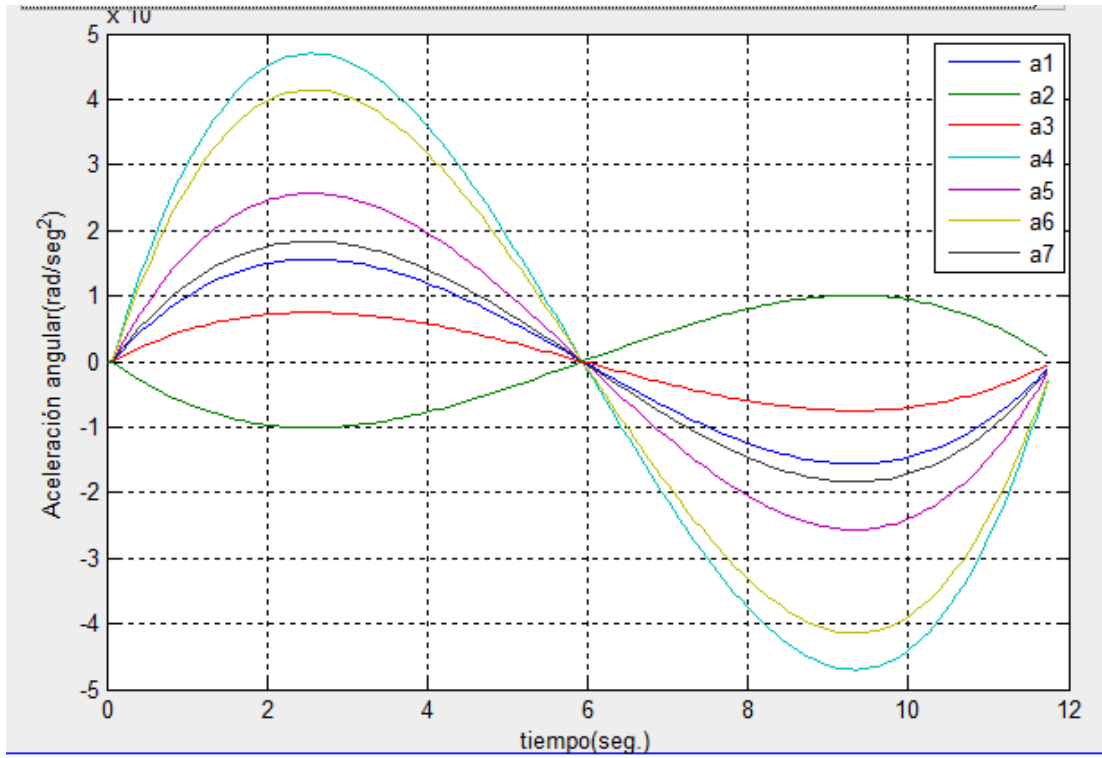


Figura 138. Aceleración angular.

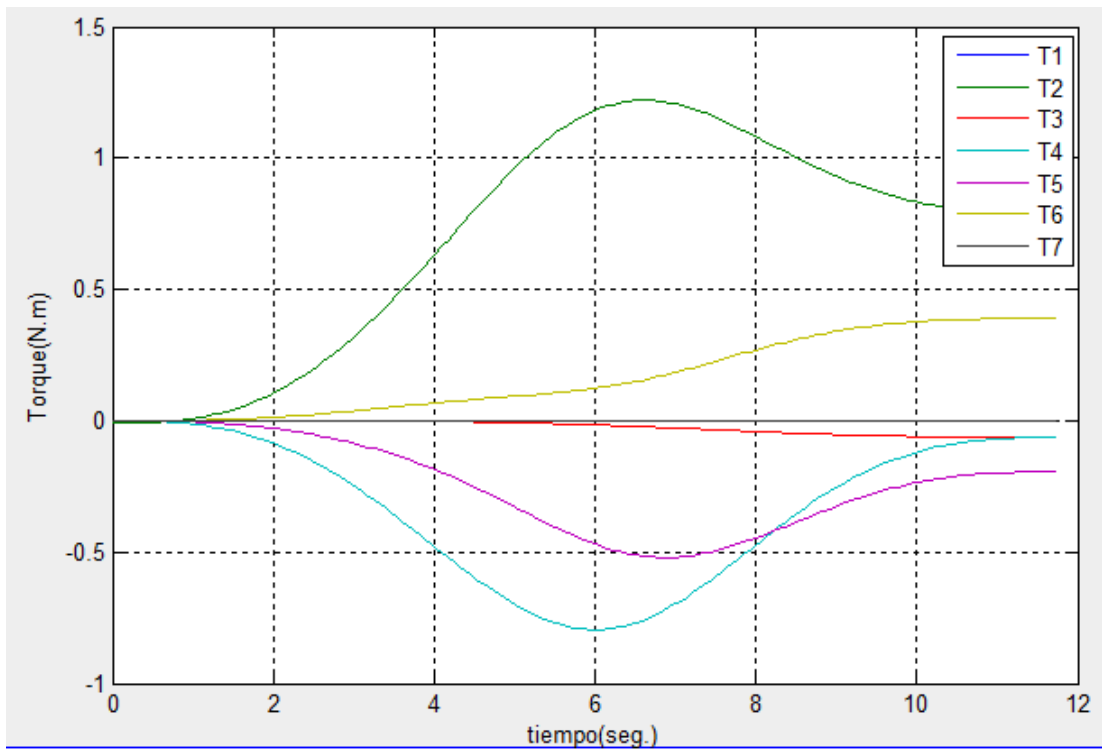


Figura 139. Torque requerido.

CAPÍTULO 5

PRUEBAS Y RESULTADOS.

5.1 Pruebas.

5.1.1 Pruebas del algoritmo genético

Para determinar el error del algoritmo hemos tratado de alcanzar el mismo punto 10 veces, con un número de población de 120 y 1000 generaciones se intenta alcanzar la posición: $X=-10$, $Y=15$, $Z=6$, y orientación: $Yaw=-75^\circ$, $Pitch=30^\circ$, $Roll=120^\circ$, los resultados obtenidos se muestran en la tabla 19:

Tabla 19.

Pruebas de exactitud del algoritmo genético.

<i>X</i>	<i>y</i>	<i>Z</i>	<i>yaw</i>	<i>pitch</i>	<i>roll</i>	<i>aptitud</i>
-9.96	14.99	6.01	-75.27	30.09	120.38	0.6977
-10.03	15.03	5.95	-75.19	29.92	120.52	0.7971
-10.01	14.97	5.96	-75.14	30.05	120.19	0.6967
-9.98	15.06	5.93	-75.31	29.97	120.38	0.8015
-10.01	15.01	5.97	-75.10	30.04	120.15	0.6967
-10.02	14.98	5.96	-75.19	30.02	120.33	0.6998
-9.98	15.08	5.93	-75.29	29.96	120.43	0.8014
-9.74	14.59	5.93	-75.36	29.95	120.60	0.8270
-10.00	14.94	6.03	-74.66	29.74	119.97	0.7741
-9.98	15.15	5.93	-75.32	30.14	120.47	0.8034

5.1.2 Pruebas del robot

Para determinar el rendimiento de un robot es necesario obtener parámetros que definen el funcionamiento y limitaciones del robot.

5.1.2.1 Precisión del robot

La precisión del robot depende de la resolución espacial, la exactitud y la repetibilidad.

Resolución espacial: es el incremento más pequeño de movimiento que puede realizar el robot, este incremento depende de los actuadores en la tabla 20 se puede observar la resolución de cada actuador en cada articulación.

Tabla 20.

Resolución de las articulaciones

<i>Articulación (numeradas desde la base hasta el efector final)</i>	<i>Resolución</i>
1	0.088 °
2	0.088 °
3	0.088 °
4	0.088 °
5	0.290 °
6	0.290 °
7	0.290 °

Exactitud: es la capacidad que tiene el robot para alcanzar el punto programado

Repetibilidad: es la capacidad que tiene el robot para alcanzar el punto programado las veces que se requiera.

5.1.2.2 Pruebas de repetibilidad y exactitud.

La norma ISO 9283 es una de las principales normas a seguir para determinar las características de los robots industriales y proporciona ejemplos de pruebas y parámetros para determinar la exactitud y repetibilidad, así como también indica cómo interpretar los resultados.

Sin embargo la mayoría de las pruebas recomendadas mencionan instrumentación especializada y capaz de medir errores con resoluciones menores que 0.05mm.



Figura 140. Robot AR7D realizando las pruebas de exactitud y repetibilidad.

Siguiendo los mismos principios que nos dicta la norma hay que tomar datos tanto de posición y orientación en una pose previamente determinada al azar durante 30 ciclos repetidos, los instrumentos utilizados fueron calibrador pie de rey de 0.05mm de resolución, graduador de 1° de resolución.

La prueba consiste en alcanzar el punto X= 7, Y=15, Z=11, con una orientación de yaw=180, pitch=0, roll=180, obteniendo los siguientes resultados.

Tabla 20.

Datos de posición y orientación del robot AR7D..

<i>N°</i>	<i>X(cm)</i>	<i>Y(cm)</i>	<i>Z(cm)</i>	<i>Yaw(°)</i>	<i>Pitch(°)</i>	<i>Roll(°)</i>
1	7.40	14.50	10.40	182	2	182
2	7.55	14.40	10.50	183	2	182
3	7.44	14.45	10.45	182	2	182
4	7.40	14.50	10.50	183	2	182
5	7.60	14.40	10.50	183	2	182
6	7.55	14.50	10.50	183	3	183
7	7.30	14.50	10.50	183	3	183
8	7.40	14.50	10.45	182	3	183
9	7.30	14.40	10.50	182	3	183
10	7.60	14.40	10.60	183	3	183
11	7.50	14.40	10.50	182	3	183
12	7.60	14.40	10.40	183	2	182
13	7.50	14.45	10.60	183	2	182
14	7.60	14.50	10.60	182	2	182
15	7.60	14.50	10.50	183	2	182
16	7.60	14.40	10.50	183	2	182
17	7.55	14.45	10.50	182	2	182
18	7.50	14.40	10.50	182	3	183
19	7.40	14.50	10.50	183	3	183
20	7.50	14.50	10.45	183	3	183
21	7.40	14.50	10.40	184	3	183
22	7.50	14.40	10.50	183	2	182
23	7.40	14.40	10.50	182	2	182
24	7.45	14.40	10.50	183	2	182
25	7.55	14.45	10.50	182	1	181
26	7.40	14.50	10.45	182	1	181
27	7.50	14.45	10.45	183	1	181
28	7.50	14.40	10.50	183	3	183
29	7.40	14.40	10.50	183	3	183
30	7.40	14.50	10.50	182	2	182

5.2 Resultados.

5.2.1 Resultado del algoritmo.

Los resultados del error promedio del algoritmo genético se puede observar en la Tabla 22, donde podemos encontrar que el error es muy pequeño cumpliendo con el objetivo de posicionar y orientar la cadena cinemática del robot AR7D.

Tabla 21.

Pruebas de exactitud del algoritmo genético.

<i>error x</i> (cm)	<i>error y</i> (cm)	<i>error z</i> (cm)	<i>error yaw</i> (°)	<i>error pitch</i> (°)	<i>error roll</i> (°)
-0.04	0.01	-0.01	0.27	-0.09	-0.38
0.03	-0.03	0.05	0.19	0.08	-0.52
0.01	0.03	0.04	0.14	-0.05	-0.19
-0.02	-0.06	0.07	0.31	0.03	-0.38
0.01	-0.01	0.03	0.10	-0.04	-0.15
0.02	0.02	0.04	0.19	-0.02	-0.33
-0.02	-0.08	0.07	0.29	0.04	-0.43
-0.26	0.41	0.07	0.36	0.05	-0.60
0.00	0.06	-0.03	-0.34	0.26	0.03
-0.02	-0.15	0.07	0.32	-0.14	-0.47

Tabla 22.

Error promedio del algoritmo genético.

<i>error promedio x</i>	0.430 mm
<i>error promedio y</i>	0.860 mm
<i>error promedio z</i>	0.480 mm
<i>error promedio yaw</i>	0.251 °
<i>error promedio pitch</i>	0.080 °
<i>error promedio roll</i>	0.348 °

5.2.2 Exactitud de Posición.

Según la norma ISO 9283:1998 la exactitud de posición es:

$$\text{Exactitud de posición} = \sqrt{(\bar{x} - xc)^2 + (\bar{y} - yc)^2 + (\bar{z} - zc)^2} \quad (20)$$

Donde:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j$$

$$\bar{y} = \frac{1}{n} \sum_{j=1}^n y_j$$

$$\bar{z} = \frac{1}{n} \sum_{j=1}^n z_j$$

$\bar{x}, \bar{y}, \bar{z}$, son la media de cada conjunto de datos obtenidos para x, y, z después de que el robot haya alcanzado esa posición n=30 veces, x_c, y_c, z_c son la posición que se requería alcanzar, y x_j, y_j, z_j las coordenadas alcanzadas en cada intento realizado.

5.2.3 Exactitud de orientación.

Según la norma ISO 9283:1998 la exactitud de posición es:

$$\text{Exactitud de orientación } a = (\bar{a} - ac) \quad (21)$$

$$\text{Exactitud de orientación } b = (\bar{b} - bc) \quad (22)$$

$$\text{Exactitud de orientación } c = (\bar{c} - cc) \quad (23)$$

Donde:

$$\bar{a} = \frac{1}{n} \sum_{j=1}^n a_j$$

$$\bar{b} = \frac{1}{n} \sum_{j=1}^n b_j$$

$$\bar{c} = \frac{1}{n} \sum_{j=1}^n c_j$$

$\bar{a}, \bar{b}, \bar{c}$, son la media de cada conjunto de datos obtenidos para yaw, pitch, roll después de que el robot haya alcanzado esa posición n=30 veces, x_c, y_c, z_c son la orientación que se requería alcanzar, y a_j, b_j, c_j los ángulos de orientación alcanzados en cada intento realizado.

Realizando las operaciones indicadas obtuvimos los resultados que se observan en la Tabla 23:

Tabla 23.

Exactitud del robot AR7D

<i>Exactitud de posición</i>	8.80 mm
<i>Exactitud yaw</i>	2.63 °
<i>Exactitud pitch</i>	2.30 °
<i>Exactitud roll</i>	2.30 °

5.2.4 Repetibilidad de posición.

$$RP = \bar{l} + 3S \quad (24)$$

$$\bar{l} = \frac{1}{n} \sum_{j=1}^n l_j \quad (25)$$

$$l_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \quad (26)$$

$$S = \sqrt{\frac{\sum_{j=1}^n (l_j - \bar{l})^2}{n - 1}} \quad (27)$$

5.2.5 Repetibilidad de orientación.

$$Rpa = \pm 3 \sqrt{\frac{\sum_{j=1}^n (a_j - \bar{a})^2}{n - 1}} \quad (28)$$

$$Rpb = \pm 3 \sqrt{\frac{\sum_{j=1}^n (b_j - \bar{b})^2}{n - 1}} \quad (29)$$

$$Rpc = \pm 3 \sqrt{\frac{\sum_{j=1}^n (c_j - \bar{c})^2}{n - 1}} \quad (30)$$

Realizando las operaciones indicadas obtuvimos los resultados que se observan en la Tabla 24:

Tabla 24.

Repetibilidad del robot AR7D

<i>Repetibilidad de posición</i>	+1.260 mm
<i>Repetibilidad yaw</i>	+ 0.305°
<i>Repetibilidad pitch</i>	+ 0.357 °
<i>Repetibilidad roll</i>	+0.357°

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES.

6.1 Conclusiones.

- Se diseñó y construyó un brazo robótico antropomórfico de siete grados de libertad, y a pesar de que resultaba matemáticamente complejo se pudo analizar cinemática y dinámicamente, ayudados de un algoritmo genético como herramienta.
- Los eslabones del robot fueron fabricados en termoplástico ABS mediante impresión 3D con una densidad del 90% en los laboratorios de Ciencias de la Energía y Mecánica de la Universidad de las Fuerzas Armadas ESPE.
- La resolución espacial del robot es de 0.26° en los eslabones que poseen motores Dynamixel AX12-A, mientras que en los motores MX64-T, MX28-T la resolución es de 0.088° .
- La carga útil del robot es de 200 gramos, que es suficiente para manipular piezas livianas ya que es un robot demostrativo.
- Se probó el robot mediante una serie de procedimientos diseñados para calcular su rendimiento según la norma ISO: 9283:1998, obteniendo una exactitud de 8.8 mm y una repetibilidad de ± 1.26 mm.
- Se confirmó que un algoritmo genético es una herramienta válida para resolver un problema multivariable y complejo como resolver la cinemática inversa de un robot de siete grados de libertad.
- Los mejores resultados del algoritmo genético fueron obtenidos con una población de 120 individuos, durante 1000 generaciones.
- El análisis dinámico nos muestra que el eslabón 2 es siempre el que más torque requiere para mover el robot. Sin embargo las trayectorias se

mantienen por debajo del torque que genera el motor dynamixel MX64-T de 5 Nm, para la carga útil de hasta 200gr.

- La exactitud promedio del algoritmo genético fue de 0.86mm en posición y 0.251° en el ángulo yaw, 0.1 en el ángulo, 0.35 en el ángulo roll

6.2 Recomendaciones.

- Al momento de imponer los parámetros de un algoritmo genético es recomendable, una gran cantidad de pruebas y así poder obtener un adecuado desempeño para cumplir el objetivo.
- Cuando no converge un problema multiobjetivo mediante algoritmos genéticos es recomendable analizar uno por uno los objetivos y sintonizar cada una de las funciones mediante constantes proporcionales que mejoren los resultados.
- Evitar que el robot se quede sin energía repentinamente, debido a que funciona con servomotores de corriente directa que pierden su Torque de trabajo cuando se apaga el sistema.
- Los Motores Dynamixel poseen un indicador led, que se prende cuando el Torque requerido supera al Torque proporcionado y el motor se apagará automáticamente; es recomendable quitar la energía de alimentación durante al menos 30 minutos.

6.3 Trabajos Futuros.

- Los controladores de los motores Dynamixel no han sido modificados ya que no eran el objetivo de esta investigación, Se recomienda analizar la posibilidad de realizar un estudio de los controladores para mejorar el rendimiento del robot AR7D.
- Al encontrarnos en la etapa de pruebas del robot, investigamos la instrumentación para medir exactitud y repetibilidad en un robot, encontrándonos con instrumentos con un alto precio, sería entonces un

proyecto futuro realizar este tipo de instrumentación, para que esté al alcance de los estudiantes y poder realizar mediciones con más resolución y por ende más precisas de estos parámetros.

- En el presente proyecto se demostró la utilidad de los algoritmos genéticos en un problema de ingeniería aplicado a la robótica, los algoritmos genéticos son capaces de optimizar muchas aplicaciones de ingeniería, En el campo de sistemas de control se puede realizar investigaciones utilizando algoritmos genéticos para control adaptable, auto-sintonización de controladores, etc.
- Al incrementar los grados de libertad aumenta la flexibilidad del Robot, Sería útil realizar una investigación para construir robots híbridos que sean móviles y manipuladores, para utilizarlos en lugares peligrosos, de difícil acceso y situaciones de rescate o quizá aplicaciones industriales.

REFERENCIAS

- Askeland, D. R., & Phulé, P. P. (2004). *Ciencia e ingeniería de los materiales* (Cuarta ed.). México D.F., México: Thomson.
- Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (1997). *Fundamentos de Robótica*. Madrid: McGraw-Hill.
- Becerra Pedraza, M. A. (2008). *Diseño y Simulación de un Robot Modular Autoconfigurable*. Mexico.
- Bravo Sancho, Á., & González Uzcátegui, C. (2012). *Evaluación de técnicas Evolutivas en el cálculo de la cinemática inversa de robots manipuladores*. Madrid.
- Budynas, R., & Nisbett, J. (2008). *Diseño en Ingeniería Mecánica de Shigley*. Mexico D.F., México: McGraw-Hill.
- Castañeda, S. (2009). *Diseño mecatrónico*. Recuperado el 7 de Agosto de 2015, de <http://slideplayer.es/slide/1130353/>
- Conti, F. (2005). *Motores paso a paso*. Madrid: ALSINA.
- Craig, J. (2006). *Robótica* (Tercera ed.). México: PEARSON EDUCACIÓN.
- Fanuc. (2015). *Aplicaciones de robots FANUC*. Recuperado el 19 de Agosto de 2015, de <http://www.fanuc.eu/es/es/aplicaciones>
- Giraldo, L. F., Delgado, E., & Castellanos, G. (2006). Cinemática inversa de un Brazo robot Utiizando Algoritmos Genéticos. *Avances en Sistemas e Informática*, 29-34.
- González, V. (2002). *Control y robótica*. Recuperado el 7 de Agosto de 2015, de http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/morfologia.htm
- Hernández Ordoñez, M., Ortiz Moctezuma, M. B., Calles Arriaga, C. A., & Rodríguez Portillo, J. C. (2015). *Robótica: Análisis, modelado, control e implementación*. Victoria: OmniaScience.
- Kelly, R., & Santibáñez, V. (2003). *Control de Movimiento de Robots Manipuladores*. Madrid: Pearson Educación.

- Mecalux. (2015). *Robot industrial de soldadura*. Recuperado el 19 de Agosto de 2015, de <http://www.logismarket.com.mx/fanuc-robotics/robot-industrial-de-soldadura/1637171848-1233048773-p.html>
- Motoman Robotics. (Julio de 2014). *Catálogo del robot SIA*. Recuperado el 20 de Agosto de 2015, de http://www.motoman.se/uploads/tx_catalogrobot/Flyer_Robot_SIA_Serie_E_05.13_09.pdf
- Motoman Robotics. (Julio de 2014). *Datasheet del robot SIA5D*. Recuperado el 20 de Agosto de 2015, de <http://www.motoman.com/datasheets/SIA5D.pdf>
- Moujahid, A., Inza, I., & Larrañaga, P. (11 de Noviembre de 2014). *Algoritmos genéticos*. Recuperado el 10 de 08 de 2015, de <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t2geneticos.pdf>
- Ollero, A. (2001). *Robótica Manipuladores y robót móviles*. Barcelona: Marcombo.
- Ponce Cruz, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.
- Ponsa, P., & Granollers, A. (s.f.). *Diseño de pantalla*. Universidad Politecnica de Catalunya, Diseño Industrial. Cataluña: Universidad Politecnica de Catalunya.
- Ramírez Gordillo, J. (2010). *Planeación de trayectorias en sistemas de manipulación robóticos múltiples*. Mexico.
- Rivas, H. (2010). *Robótica y automatización industrial*. Recuperado el 19 de Agosto de 2015, de <http://www.hurtadorivas.com/productos.php>
- Roboter GmbH. (2015). *Especificaciones robot KUKA LBR iiwa*. Recuperado el 20 de Agosto de 2015, de http://www.kuka-robotics.com/res/sps/e6c77545-9030-49b1-93f5-4d17c92173aa_Spez_LBR_iiwa_en.pdf
- Robotis. (2010). *Manual del controlador USB2Dynamixel*. Recuperado el 22 de Agosto de 2015, de http://support.robotis.com/en/product/auxdevice/interface/usb2dx1_manual.htm

- Robotis. (2010). *Manual motores dynamixel MX-64T / MX-64R*. Obtenido de http://support.robotis.com/en/product/dynamixel/mx_series/mx-64.htm
- Sánchez, O. (Marzo de 2008). *Cinemática de los manipuladores*. Recuperado el 20 de Agosto de 2015, de <http://es.slideshare.net/omarspp/cinemtica-de-los-manipuladores>
- Trossen Robotics. (2015). *Trossen Robotics Dynamixel Guide*. Recuperado el 31 de Agosto de 2015, de <http://learn.trossenrobotics.com/projects/159-trossen-robotics-dynamixel-guide.html>
- Trossen Robotis. (2015). *Dynamixel AX-12A Robot Actuator*. Recuperado el 27 de Agosto de 2015, de <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>
- Vaca Jiménez, S. (2012). *Diseño y simulación de un robot manipulador industrial tipo PUMA capaz de levantar pesos de hasta 20Kg*. Escuela Politécnica Nacional, Facultad de ingeniería Macánica, Quito.
- Valles Pérez, I. (s.f.). *Analysis of Genetic Algorithms applied on Extreme Learning Machine*. Recuperado el 20 de Agosto de 2015, de <http://www.uv.es/ivape3/memoria.pdf>
- Wikipedia. (2015). *Método científico*. Recuperado el 7 de Agosto de 2015, de https://es.wikipedia.org/wiki/M%C3%A9todo_cient%C3%ADfico