



POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-30-No.508-2019

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

DESARROLLO DE UN TOOLBOX PARA RESOLVER
TRAYECTORIAS DE ROBOTS MANIPULADORES
MEDIANTE ENTRENAMIENTO BASADO EN
ALGORITMOS GENÉTICOS.

AUTOR:

ALEXIS PATRICIO SANCHEZ FRIAS

DIRECTOR:

MANUEL RAFAEL JAYA DUCHE

QUITO - ECUADOR
2021

Autor/a:



Alexis Patricio Sánchez Frías

Ingeniero en Mecatrónica

Candidato a Magister en Electrónica y Automatización con Mención en Informática Industrial por la Universidad Politécnica Salesiana – Sede Quito.

asanchezf4@est.ups.edu.ec

Dirigido por:



Manuel Rafael Jaya Duche

Ingeniero en electrónica y telecomunicaciones

Magister en redes de información y conectividad

mjaya@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2021 Universidad Politécnica Salesiana.

QUITO – ECUADOR – SUDAMÉRICA

SÁNCHEZ FRÍAS ALEXIS PATRICIO

DESARROLLO DE UN TOOLBOX PARA RESOLVER TRAYECTORIAS DE ROBOTS MANIPULADORES MEDIANTE ENTRENAMIENTO BASADO EN ALGORITMOS GENÉTICOS.

Índice general

Índice de Figuras	6
Índice de Tablas	9
Abstract	10
1. Introducción	12
1.1. Descripción general del problema	14
1.2. Objetivos	14
1.2.1. Objetivo general	14
1.2.2. Objetivos específicos	14
1.3. Contribuciones	15
1.4. Organización del manuscrito	16
2. Algoritmos Genéticos en la cinemática inversa de robots manipuladores	17
2.1. Estado del Arte	18
2.2. Definiciones Previas	18
2.2.1. Morfología de un robot industrial	19
2.2.2. Partes y tipos de brazos roboticos industriales	19
2.2.3. Grados de libertad	21
2.2.4. Cinemática en Robots Manipuladores	21
2.2.5. Cinemática Directa	21
2.2.6. Cinemática Inversa	22
2.2.7. Sistemas Inteligentes	22
2.2.8. Algoritmos Genéticos	23
3. Diseño HMI del toolbox para robots manipuladores	26
3.1. Diseño de Software	27
3.1.1. Guía Gedis	27

3.1.2.	Pantalla de inicio	28
3.1.3.	Pantalla Seleccionar Robot	29
3.1.4.	Diseñar Robot	30
3.1.5.	Pantalla de Programación y Simulación	31
3.1.6.	Pausa	35
3.1.7.	Pantalla Mover	35
3.1.8.	Pantalla Mover recta	35
3.1.9.	Pantalla Trayectoria	35
4.	Algoritmos genéticos para solución de trayectorias	40
4.1.	Análisis Cinemático.	41
4.2.	Solución de la cinemática inversa mediante un algoritmo genético	43
4.2.1.	Restricciones	44
4.2.2.	Población inicial	45
4.2.3.	Selección	46
4.2.4.	Mutación	46
4.2.5.	Cruzamiento	46
4.2.6.	Función de aptitud	46
4.2.7.	Tamaño de la población y generaciones	49
4.2.8.	Generación de trayectorias mediante interpolación mediante polinomios	49
4.2.9.	Trayectorias rectas	49
4.2.10.	Trayectorias indefinidas	50
5.	Pruebas Y Resultados	51
5.1.	Prueba de la funcionalidad de la opción teach para cinemática directa en el robot FANUC LRMATE 200iC	52
5.2.	Resultados de la funcionalidad de la opción teach para cinemática directa en el robot Fanuc LRMATE-200iC.	54
5.3.	Resultados del algoritmo	58
5.3.1.	Exactitud de posición y orientación	58
5.3.2.	Repetibilidad de posición y orientación	59
5.4.	Prueba de la solución de cinemática inversa y trayectoria en dos planos diferentes mediante algoritmos genéticos en el robot KUKA-KR16-2	60
5.5.	Resultados solución de cinemática inversa y trayectoria irregular en dos planos diferentes mediante algoritmos genéticos en el robot KUKA KR 16 2	61

<i>ÍNDICE GENERAL</i>	5
5.6. Pruebas funcionales del toolbox con robot un redundante de 7 grados de libertad	65
5.7. Resultados del toolbox con robot un redundante de 7 grados de libertad	65
5.8. Pruebas Funcionales del toolbox para la solución de trayectorias irregulares en el robot Mitsubishi RV2AJ.	68
5.8.1. Comunicación entre el robot Mitsubishi RV2AJ y el toolbox	69
5.9. Prueba trayectorias con el robot Mitsubishi RV2AJ	70
5.10. Discusión	72
5.10.1. Ventajas y desventajas del método implementado en el toolbox con los resultados obtenidos en las pruebas funcionales	72
6. Conclusiones y Recomendaciones	74
6.1. Conclusiones	75
6.2. Recomendaciones	76

Índice de Figuras

2.1. Constitución física de un robot industrial [Autores, 2015] . . .	19
2.2. Configuraciones de robots industriales, (A)Cartesiano, (B)Esférico, (C)Angular,(D)Scara [Aranda, 2014]	20
2.3. Cinemática inversa y directa[Barrientos et al., 2007]	21
2.4. Cinemática directa mediante Denavit-Hartenberg [Barrientos et al., 2007]	22
2.5. Diagrama de flujo algoritmo genético [Cruz, 2011]	25
3.1. Diagrama de flujo Pantalla Inicio (Fuente propia)	28
3.2. Pantalla de Inicio (Fuente propia)	29
3.3. Diagrama de Seleccionar Robot (Fuente propia)	30
3.4. Pantalla Seleccionar Robot (Fuente propia)	31
3.5. Pantalla Diseñar Robot (Fuente propia)	32
3.6. Diagrama de Diseñar Robot (Fuente propia)	33
3.7. Pantalla Programación y Simulación (Fuente propia)	34
3.8. Diagrama de flujo- botón home (Fuente propia)	34
3.9. Pantalla – Pausa (Fuente propia)	35
3.10. Diagrama de flujo - mover (Fuente propia)	36
3.11. Pantalla -Mover (Fuente propia)	37
3.12. Pantalla – mover recta (Fuente propia)	37
3.13. Pantalla – Trayectoria (Fuente propia)	38
3.14. Trayectoria interactiva (Fuente propia)	38
3.15. Diagrama del algoritmo para trayectoria interactiva (Fuente propia)	39
4.1. Articulaciones Robot Mitsubishi RV-2AJ [Slamet and dan Elektronik, 2013]	41
4.2. Dimensiones del Robot Mitsubishi RV-2AJ [Slamet and dan Elektronik, 2013]	42

4.3. Ingreso de Datos del Robot Mitsubishi RV2AJ (Fuente propia)	43
4.4. Gráfico en 3 dimensiones del Robot Mitsubishi generado con herramienta de Peter Corke (Fuente propia)	44
4.5. Ingreso de parámetros de Denavit Hartenberg (Fuente propia)	45
4.6. matriz de transformación Tf [Barrientos et al., 2007]	47
4.7. Matriz homogénea para cada eslabón mediante los parámetros de Denavit Hatenberg [Barrientos et al., 2007]	47
5.1. Pantalla de programación mediante la opción de teach de 3 puntos a velocidades diferentes en el robot Fanuc LRMate 200iC.	53
5.2. Simulación de movimiento 3d del robot FANUC LRTMATE 200iC (Fuente propia)	53
5.3. Resultados de posición de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	55
5.4. Resultados de velocidad de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	55
5.5. Resultados de aceleración de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	56
5.6. Resultados de posición de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	56
5.7. Resultados de velocidad de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	57
5.8. Resultados de aceleración de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)	57
5.9. Trayectoria hexagonales en el espacio de trabajo del robot KUKA-KR16-2 (Fuente propia)	60
5.10. Primera trayectoria hexagonal en el espacio de trabajo del robot KUKA-KR16-2 vista superior del plano xy (Fuente propia)	61
5.11. Segunda trayectoria hexagonal en el espacio de trabajo del robot KUKA-KR16-2 vista superior del plano xz (Fuente propia)	61
5.12. Resultados de posición de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)	62
5.13. Resultados de velocidad de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)	63
5.14. Resultados de aceleración de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)	64
5.15. Pantalla de programación para Robot Mitsubishi 7gdl (Fuente propia)	65
5.16. Cuadrado en el espacio de trabajo del Robot Mitsubishi 7gdl (Fuente propia)	66

5.17. Vista superior de la trayectoria de prueba para Robot Mitsubishi 7gdl (Fuente propia)	66
5.18. Resultados de posición de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)	67
5.19. Resultados de velocidad de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)	68
5.20. Resultados de aceleración de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)	69
5.21. Menu de comunicación Serial (Fuente propia)	69
5.22. Letra A manuscrita dibujada en el espacio de trabajo (Fuente propia)	70
5.23. Robot Mitsubishi RV-2AJ dibujando la letra manuscrita (Fuente propia)	71
5.24. Robot Mitsubishi RV-2AJ dibujando trayectoria de soldadura (Fuente propia)	71

Índice de Tablas

2.1. Expresiones utilizadas en genética y su equivalente en algoritmos genéticos [Cruz, 2011]	23
4.1. Parámetros de Denavit Hartenberg del robot Mitsubishi RV2AJ [Slamet and dan Elektronik, 2013]	42
4.2. Límites del Robot Mitsubishi RV2AJ [Slamet and dan Elektronik, 2013]	43
5.1. Exactitud del algoritmo genético en trayectoria robot KUKA-KR16-2	62
5.2. Repetibilidad del algoritmo genético en trayectoria robot KUKA-KR16-2	63
5.3. Exactitud del algoritmo genético en trayectoria robot redundante Mitsubishi 7gdl	67
5.4. Repetibilidad del algoritmo genético en trayectoria robot redundante Mitsubishi 7gdl	68

Resumen

El presente proyecto tiene como objetivo analizar la utilización de un método de inteligencia computacional como son los algoritmos genéticos para resolver la cinemática inversa y planificar la trayectoria de los robots manipuladores y basados en ello desarrollar un toolbox para poder visualizar e interactuar con varios tipos y configuraciones de robots. El toolbox fue desarrollado en MATLAB software que tiene una gran capacidad de manejar matrices y además tiene herramientas y funciones incluidas para inteligencia artificial, el espacio GUIDE también permite diseñar el HMI el cual se ha realizado basados en la guía Gedis, la cual brinda parámetros para hacer una interfaz que sea adecuada, el toolbox desarrollado permite resolver la cinemática inversa mediante algoritmos genéticos y obtener las trayectorias que se desean en un robot manipulador y que además visualizar el movimiento de los eslabones para escoger el robot que mejor se adapte a una aplicación e inclusive diseñar un robot cambiando parámetros cinemáticos como grados de libertad, longitudes de eslabones y así encontrar un diseño que cumpla con el requerimiento de movimiento en el espacio de trabajo dado, el toolbox es una herramienta académica que permite observar el comportamiento del robot y simular su movimiento, es bastante útil para la enseñanza de la robótica industrial mostrando en una interfaz gráfica los principales parámetros que se tienen en cuenta en el aspecto cinemático además tiene una base de datos que permite simular los principales robots industriales en el mercado y una interfaz que permite conectar un robot mediante comunicación serial. Finalmente se han realizado una serie de pruebas, analizando la precisión y repetibilidad del algoritmo en diferentes modelos de robots manipuladores industriales incluyendo además pruebas que validan la utilidad del método de algoritmos genéticos en robots redundantes.

Palabras claves: robótica, algoritmos genéticos, cinemática inversa, trayectorias, toolbox, desarrollo, HMI, Matlab.

Abstract

This project has the objective of analyzing the use of a computational intelligence method such as genetic algorithms to solve the inverse kinematics and plan the trajectory of manipulator robots and based on it to develop a toolbox to visualize and interact with various types and configurations of robots. The toolbox was developed in MATLAB software that has a great capacity to handle matrices and also has tools and functions included for artificial intelligence, the GUIDE space also allows to design the HMI which has been done based on the Gedis guide which gives us parameters to make an interface that is suitable, The toolbox developed allows to solve the inverse kinematics through genetic algorithms and obtain the desired trajectories in a manipulator robot and also to visualize the movement of the links to choose the robot that best suits an application and even design a robot by changing kinematic parameters such as degrees of freedom, The toolbox is an academic tool that allows to observe the behavior of the robot and simulate its movement, it is quite useful for teaching industrial robotics showing in a graphical interface the main parameters that are taken into account in the kinematic aspect also has a database that allows to simulate the main industrial robots in the market and an interface that allows to connect a robot through serial communication. Finally, a series of tests have been carried out, analyzing the accuracy and repeatability of the algorithm in different models of industrial manipulator robots, including tests that validate the usefulness of the genetic algorithms method in redundant robots.

Keywords: robotics, genetic algorithms, inverse kinematics, trajectories, toolbox, development, HMI, Matlab.

Capítulo 1

Introducción

En la actualidad los robots manipuladores son herramientas de alta importancia en varios campos de aplicación, con el crecimiento industrial es muy común ver brazos robóticos ensambladores, soldadores, pintores en las plantas o fábricas ya que pueden trabajar en ambientes peligrosos en condiciones de temperaturas fuera de un rango aceptable ayudando a no poner en riesgo a las personas, son instrumentos que se utilizan también fuera de la industria existen brazos quirúrgicos, de exploración, y además se han utilizado en lugares extremos, bajo el océano o el espacio, sus diferentes diseños y configuraciones que cambian de acuerdo a la aplicación también han ido evolucionando, uno de los principales problemas que se presenta al momento de diseñar o elegir un brazo robótico es poder resolver la cinemática inversa y el problema se complica cuando los grados de libertad aumentan, es por ello que los nuevos diseños ya no buscan una solución cerrada matemáticamente sino mas bien algún método computacional que resuelva el problema de manera precisa.

Los robots manipuladores son herramientas de alta importancia en varios campos de aplicación, con el crecimiento industrial es muy común observar brazos robóticos ensambladores o soldadores en las plantas o fábricas ya que pueden trabajar en ambientes peligrosos en condiciones de temperaturas que podrían poner en riesgo a las personas, son instrumentos que se utilizan también fuera de la industria manufacturera, se puede mencionar brazos quirúrgicos, de exploración, desactivación de bombas y además se han utilizado robots manipuladores en lugares de climas extremos, bajo el océano o el espacio, sus diferentes diseños y configuraciones que cambian de acuerdo a la aplicación también han ido evolucionando. Los robots manipuladores son diseñados cada vez con más grados de libertad y esto da como resultados

complejidad en el análisis de su cinemática para lo cual no existe una solución matemática cerrada que describa el movimiento del efector final es decir la solución de la cinemática inversa. Para analizar esta problemática es necesario mencionar sus causas. La principal causa es que mediante la metodología tradicional dada por las matrices de Denavit Hartenberg se obtienen sistemas no lineales con una mayor número de incógnitas, sin embargo los algoritmos evolutivos de optimización han demostrado ser aplicables a este tipo de problemática no lineal y multiobjetivo, por lo cual el proyecto utilizará un algoritmo evolutivo como base de un toolbox didáctico que permita observar que dicha solución es efectiva para planificar la trayectoria de los robots manipuladores.

1.1. Descripción general del problema

La trayectoria que puede realizar el efector final de un robot manipulador es lo más importante para el diseño o selección del robot ya que determina la precisión y exactitud del mismo para realizar la aplicación que se requiera programar, La solución de la cinemática inversa de los robots manipuladores industriales es un problema de relevancia en los procesos de producción industrial por ejemplo en la soldadura o la pintura, como se menciona en [Pan et al., 2010] la optimización de la programación y el seguimiento de la trayectoria, pueden tardar hasta ocho meses antes de que la línea de producción se inicie, es por esto que resolver dicha trayectoria mediante un algoritmo que optimice los resultados es esencial porque ahorra muchos recursos y evita forzar a pruebas directamente al robot, además es necesario visualizar una simulación en tres dimensiones que permita apreciar el movimiento del robot y su comportamiento en su respectivo espacio de trabajo, de manera que la solución cinemática sea apreciable en forma de movimiento y no solamente sea un conjunto de matrices y ecuaciones matemáticas, sin mencionar lo costoso que es acceder y mantener un laboratorio de robótica además de la situación actual que se vive con la pandemia de Covid 19 la cual limita el acceso a los laboratorios de forma presencial, resultaría de gran utilidad tener al alcance un entorno virtual para el aprendizaje de robótica industrial.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un toolbox basado en algoritmos genéticos que permita resolver la cinemática inversa y trayectorias de robots manipuladores mediante la integración de las funciones de optimización y ambientes de simulación de Matlab para analizar la utilidad del entrenamiento empleando técnicas de inteligencia artificial como método de programación en la robótica.

1.2.2. Objetivos específicos

- Analizar el estado del arte de los algoritmos genéticos aplicados a resolver la cinemática inversa de robots manipuladores mediante una revisión de trabajos previos sobre dicha aplicación para que el algoritmo que se diseñe en el toolbox tenga buenos resultados.

- Diseñar una interfaz (HMI) mediante la herramienta GUI de Matlab que permita visualizar el movimiento del robot y el progreso del aprendizaje del algoritmo con gráficas y animaciones en tiempo real para que el software tenga una correcta y didáctica interacción con el usuario.
- Analizar el desempeño de los algoritmos genéticos utilizados en el toolbox para la resolución de trayectorias en varios tipos de robots manipuladores.
- Realizar pruebas funcionales del entrenamiento de una serie de trayectorias en un robot manipulador utilizando el toolbox desarrollado.
- Obtener las ventajas y desventajas del método mediante un análisis de los resultados obtenidos en las pruebas funcionales.

1.3. Contribuciones

El presente proyecto pretende plantear un método capaz de resolver la cinemática inversa de cualquier tipo de robot manipulador sin importar la complejidad de su configuración ya que mientras más grados de libertad tiene un robot se vuelve más compleja la solución de la cinemática inversa. La manera de resolver para cada modelo de robot es diferente y depende de sus ecuaciones resultantes al analizar sus matrices de posición y orientación, los métodos tradicionales se centran en encontrar la solución de un robot en específico, con el método que se desarrolla en esta investigación es posible hallar la solución de cualquier robot de manera generalizada, además es útil en robots redundantes con sistemas tan complejos que no se puede obtener una solución cerrada mediante los clásicos métodos matriciales. Simular la trayectoria de los robots manipuladores mediante un software que optimice los resultados es importante ya que se ahorra muchos recursos y evita forzar a pruebas directamente al robot. Igual de importante resulta poder visualizar una simulación en tres dimensiones que permita apreciar el movimiento del robot y su comportamiento en su respectivo espacio de trabajo, esto es importante porque da un enfoque didáctico e intuitivo en el caso de clases virtuales. El toolbox facilita la programación ya que el entorno que se pretende diseñar se asemeja más a una herramienta CAD que a un lenguaje de programación, y permite introducir trayectorias más complejas permitiendo aplicaciones que potenciarán la utilidad del robot que se esté analizando.

1.4. Organización del manuscrito

El presente proyecto se compone de seis capítulos. El primer capítulo es la introducción al proyecto. El segundo capítulo analiza el estado del arte con respecto a los algoritmos genéticos aplicados en la solución de la cinemática inversa de robots manipuladores. El tercer capítulo trata del diseño de la estructura del toolbox y los elementos que componen el desarrollo del mismo, como la declaración de variable, funciones y objetos, interfaces, flujo, HMI, etc. El cuarto capítulo trata sobre planteamiento de la solución de la cinemática inversa mediante algoritmos genéticos en los robots manipuladores dentro del toolbox y como se aborda los elementos de este método para la planificación de trayectorias. El quinto capítulo comprende la validación del desarrollo con una serie de pruebas establecidas de acuerdo a normativa vigente para robótica y además se realizarán pruebas utilizando el hardware robot manipulador Mitsubishi RV2AJ que se tiene disponible en los laboratorios de la Universidad Politécnica Salesiana de modo que demuestre la utilidad del proyecto para el ámbito académico. El sexto y último capítulo comprende las conclusiones y recomendaciones a partir de los resultados obtenidos.

Capítulo 2

Algoritmos Genéticos en la cinemática inversa de robots manipuladores

El capítulo 2 abarcara en primer lugar el estado del arte con respecto a los algoritmos genéticos aplicados en la solución de la cinemática inversa de robots manipuladores. Luego se definirán los términos y conceptos presentes en el proyecto y finalmente se formulará el problema para describir las bases del proyecto adecuadamente.

2.1. Estado del Arte

A través del tiempo se han diseñado un sin fin de tipos de robots con diversas configuraciones, sin embargo algunas configuraciones se han mantenido por su utilidad dependiendo de la aplicación, una de las configuraciones más comunes son la configuración cartesiana, la cilíndrica, la polar o esférica, la angular, la SCARA y robots redundantes.

La inteligencia artificial tuvo su introducción como una metodología viable para resolver problemas matemáticos complejos con el aumento de la capacidad de procesamiento de las computadoras, y como se menciona en [Teigens et al.] se desarrollaron descripciones matemáticas precisas para paradigmas de inteligencia computacional como redes neuronales y algoritmos evolutivos, estos métodos de inteligencia artificial y entre ellos los algoritmos genéticos demostraron ser de gran utilidad en la industria tecnológica y entre los problemas matemáticos complejos que se empezaron a investigar a parte de buscadores, minería de datos, también se aplicó al campo de la robótica industrial.

En el año 2003 [Karlra and Prakash, 2003] usa una combinación de redes neuronales y algoritmos genéticos para desarrollar un algoritmo que resuelva la cinemática inversa de un robot manipulador en esta investigación la red neuronal utilizada es multicapa con un algoritmo genético como algoritmo de aprendizaje. Los algoritmos genéticos son algoritmos bioinspirados, cuyo origen se explica en [Lahoz-Beltrá, 2004], la posibilidad de simulación de la evolución mediante una computadora fue por primera vez con el avance de los primeros ordenadores por Jonh Von Newman y Ulam en los años cincuenta sin embargo no sería hasta la década del sesenta donde se realizan experimentos con las primeras poblaciones por parte de Fraser(1969). Fogel, Owens y Walsh (1966) realizan experimentos con poblaciones que presentan mutación variando características de la población aleatoriamente, el término algoritmo genético fue acuñado por Bagley en 1967.

2.2. Definiciones Previas

A continuación se describen y explica los fundamentos, definiciones y herramientas importantes que se va a utilizar para el abordaje y análisis del presente proyecto de investigación.

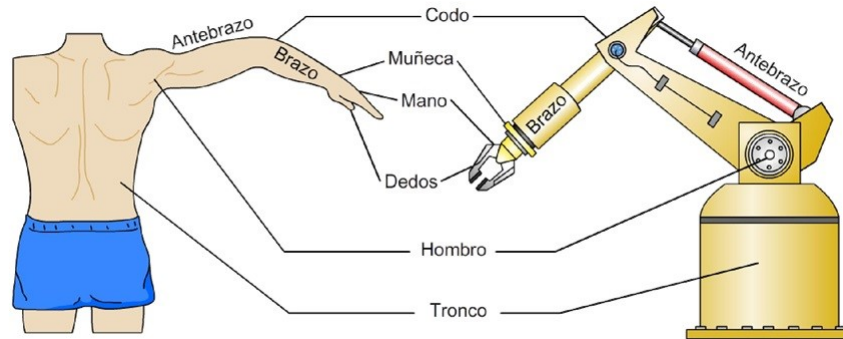


Figura 2.1: Constitución física de un robot industrial [Autores, 2015]

2.2.1. 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 2.1.

2.2.2. Partes y tipos de brazos robóticos industriales

Como menciona [Barrientos et al., 2007] Un robot está constituido por los siguientes elementos :

- 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).
- Elementos terminales: Pinzas, herramientas, dispositivos de sujeción, etc.
- Sistema sensorial: Existen dos tipos de sensores, sensores internos y sensores externos. Los sensores internos registran 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.

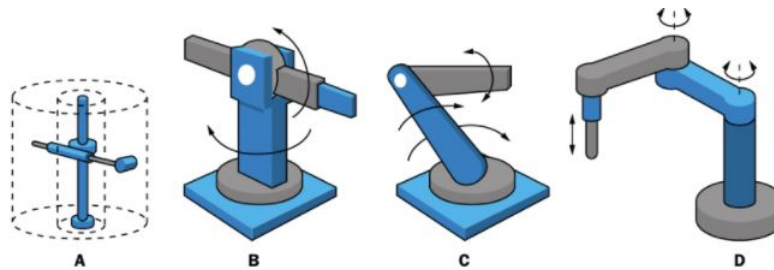


Figura 2.2: Configuraciones de robots industriales, (A)Cartesiano, (B)Esférico, (C)Angular,(D)Scara [Aranda, 2014]

- Sistema de control: Gobierna el funcionamiento del robot, puede contar o no de un teach pendant y un ordenador.

Se puede tener un gran número de configuraciones de robots industriales al combinar los tipos de articulaciones de las cuales hay algunas básicas que son empleadas ampliamente en la industria entre estos tenemos el robot cartesiano (Cintura: Prismático, Hombro: Prismático, Codo: Prismático) también conocido como prismático tiene tres articulaciones prismáticas o lineales, el robot cilíndrico (Cintura: Rotacional torsión, Hombro: Lineal, Codo: lineal) tiene una articulación rotacional de torsión y dos articulaciones lineales, el robot esférico (Cintura: Rotacional torsión, Hombro: Rotacional, Codo: Lineal) o polar tiene una articulación rotacional de torsión, una rotacional y una lineal o prismática, el robot antropomórfico (Cintura: Rotacional torsión, Hombro: Rotacional, Codo: Rotacional), articulado vertical o angular, tiene una articulación rotacional de torsión y dos articulaciones rotacionales, el robot SCARA (Cintura: Rotacional torsión, Hombro: Rotacional, Codo: Lineal) o articulado horizontal tiene 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 2.2. 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.

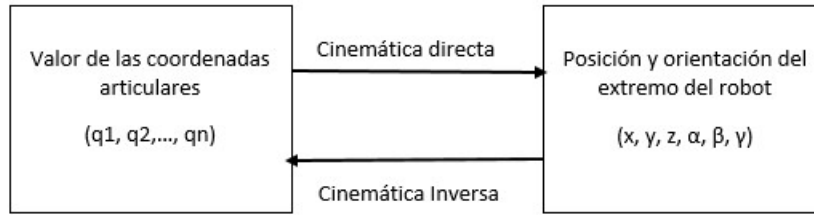


Figura 2.3: Cinemática inversa y directa [Barrientos et al., 2007]

2.2.3. Grados de libertad

Grados de libertad, son el número de parámetros independientes que determinan la posición del efector final, por lo general el número de eslabones a veces coincide con el número de grados de libertad de la cadena cinemática.

2.2.4. Cinemática en Robots Manipuladores

[Barrientos et al., 2007] Define a la cinemática como el estudio del movimiento del robot con respecto a un sistema de referencia, en el cual se enfoca principalmente en las relaciones que hay 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 2.3).

2.2.5. Cinemática Directa

Consiste en encontrar la posición y orientación del efector final teniendo como datos las coordenadas articulares. Para esta investigación se utilizará como base para el análisis los parámetros de Denavit-Hatenberg para obtener el modelo cinemático directo del robot manipulador. En el año de 1955 Denavit y Hatenberg [Hayat et al., 2013] propusieron un algoritmo que permite resolver el problema cinemático directo mediante un método matricial y sistemático, encontrando las relaciones entre la posición del efector final y las coordenadas articulares.

[Barrientos et al., 2007] menciona que a partir de cuatro matrices de transformación básicas que son: Rotación alrededor del eje z un ángulo theta, translación a lo largo de z una distancia d, translación a lo largo de x una distancia a y rotación alrededor de x un ángulo alpha. A continuación se

$$\begin{aligned}
{}^{i-1}A_i &= T(z, \theta_i)T(0,0, d_i)T(a_i, 0,0)T(x, \alpha_i) \\
{}^{i-1}A_i &= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{i-1}A_i &= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
T &= {}^0A_1, {}^1A_2, \dots, {}^{n-1}A_n
\end{aligned}$$

Figura 2.4: Cinemática directa mediante Denavit-Hartenberg [Barrientos et al., 2007]

multiplica las matrices de rotación y translación obteniendo la representación de cada eslabón, luego se obtiene la matriz de transformación de cada eslabón y se multiplican entre si ordenadamente para obtener la matriz T la cual define la orientación y posición del robot en función de las coordenadas articulares es decir la solución a la cinemática directa como se observa en las ecuaciones de la Figura 2.4.

2.2.6. Cinemática Inversa

Según [Craig, 2006], siempre que se especifica una posición y orientación del efector final, debe calcularse la cinemática inversa para poder encontrar los ángulos que deben tener las articulaciones para alcanzar dicha posición. En [Jiménez] describe a la cinemática inversa como el estudio de las posiciones geométricas y del movimiento de las articulaciones y eslabones sin tomar en cuenta las causas generadoras.

2.2.7. Sistemas Inteligentes

Según [Zilouchian and Jamshidi, 2001] define a los sistemas inteligentes como la base de lo que se denomina “soft computing” en la cual la precisión y la certeza tienen un costo. Por lo tanto, es realista considerar la integración del cálculo, el razonamiento y la toma de decisiones para alcanzar un equilibrio entre precisión e incertidumbre.

Tabla 2.1: Expresiones utilizadas en genética y su equivalente en algoritmos genéticos [Cruz, 2011]

Evolución biológica	Algoritmos Genéticos
Gen	Posición de cadena
Función de aptitud o aptitud	Valor de la función objetivo
Alelo	Valor en una posición determinada
Genotipo	Código de cadena
Fenotipo	Punto sin codificar
Cromosoma	Cadena

Para conseguir estos objetivos los principales elementos son la lógica difusa, la computación de redes neuronales, los algoritmos genéticos y el razonamiento probabilístico, el presente proyecto está enfocado en el método de algoritmos genéticos.

2.2.8. Algoritmos Genéticos

Según [Cruz, 2011] 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.

Un algoritmo genético se compone 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 examinar cada una de estas operaciones es necesario codificar los individuos y convertirlos en una cadena binaria a este proceso se lo denomina codificación, en la Tabla 2.1 se observa las equivalencias mas importantes entre terminos de la evolución natural y términos utilizados en algoritmos genéticos computacionales.

Un genotipo 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 tiene información única de cada individuo. Fenotipo son 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.

El proceso de selección se trata de 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).

Un elemento muy importante es la mutación que tiene como objetivo cambios aleatorios en los valores de una determinada posición de la cadena de bits o su equivalente término genético alelo en ciertos individuos, esta mutación puede ser favorable o no, pero su importancia radica en introducir ciertas características de las que carecen sus padres.

El cruzamiento se da entre dos progenitores previamente seleccionados con características aptas se fusionan produciendo así hijos o descendientes que heredan las mejores características de sus padres.

Finalmente, el remplazo 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.

Para el desarrollo de este toolbox se ha utilizado las funciones GA predefinidas en Matlab que poseen todas las herramientas de mutación, selección y limitación requerida la cual se puede ir modificando de acuerdo a nuestro requerimiento de solución, el algoritmo genético de Matlab utilizado tiene la lógica que se observa en el diagrama de flujo que se observa en la Figura 2.5.

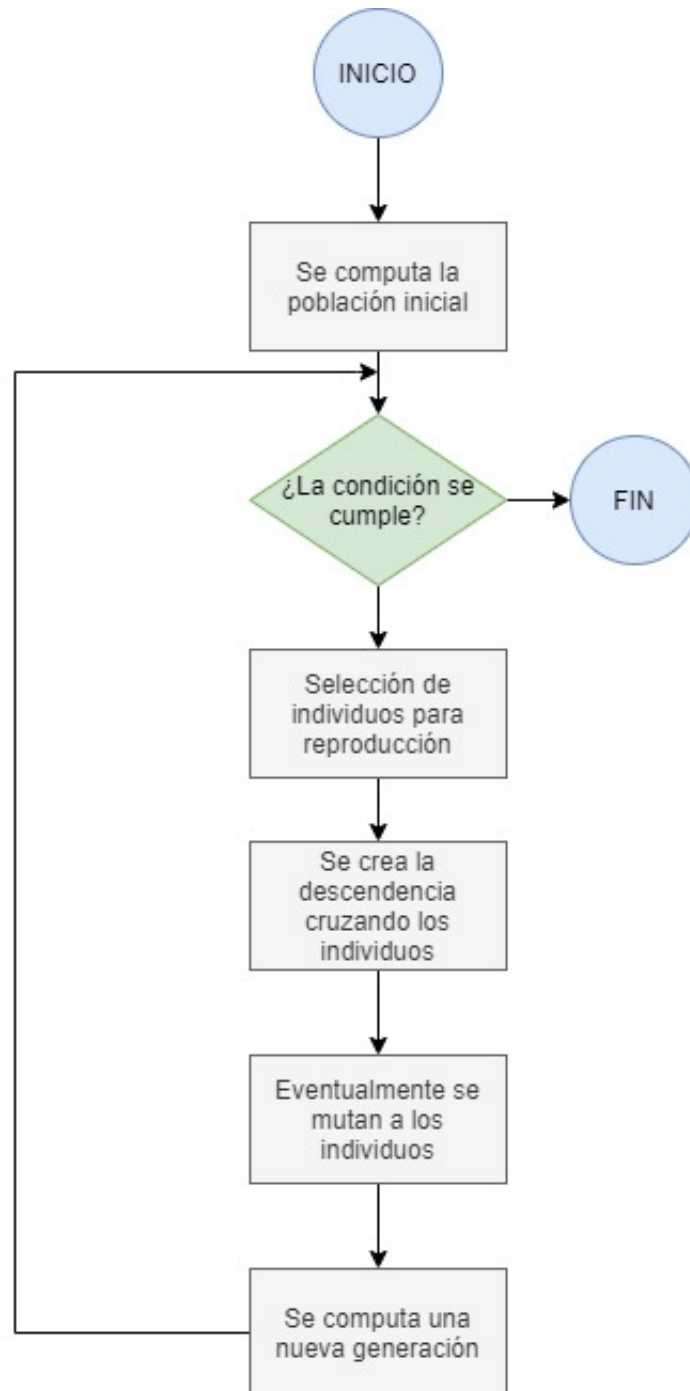


Figura 2.5: Diagrama de flujo algoritmo genético [Cruz, 2011]

Capítulo 3

Diseño HMI del toolbox para robots manipuladores

El Toolbox fue desarrollado en Matlab que es un software que posee una gran cantidad de herramientas sobretodo es fuerte con la operación de matrices que como se definió anteriormente el análisis de la posición y rotación viene dada por matrices. El toolbox diseñado consiste de una pantalla principal de inicio con datos informativos y una opción para seleccionar entre un robot predefinido dentro de una base de datos o ingresar manualmente los parámetros de Denavit Hartenberg llenando una matriz interactiva, a continuación se abrirá una pantalla en donde el robot definido o diseñado podrá ser visualizado, programado y simulado mediante cinemática directa o inversa. Las trayectorias programadas podrán ser guardadas y modificadas a conveniencia del usuario. En este capítulo se explicará cada uno de los pasos y elementos utilizados para el desarrollo del toolbox

3.1. Diseño de Software

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 está la guía Gedis que nace de varios estándares para el diseño de HMI.[[Pecherková et al., 2008](#)]

3.1.1. 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, 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.

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

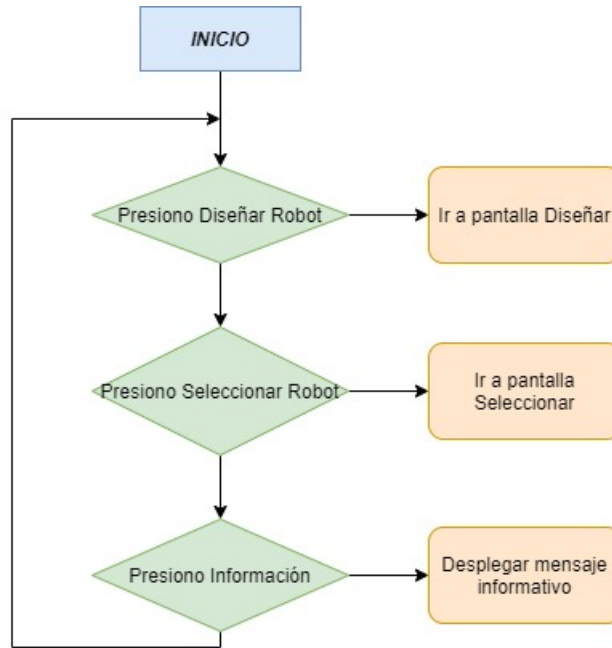


Figura 3.1: Diagrama de flujo Pantalla Inicio (Fuente propia)

- Alarmas: Evitar el exceso de alarmas innecesarias y mostrar el motivo del error o alarma.

3.1.2. Pantalla de inicio

Esta pantalla es la que se muestra al iniciar el programa consta de una interfaz ligera como se muestra en la Figura 3.2, Con datos informativos y tres botones que permiten acceder a las funcionalidades del software, la lógica se puede observar en el diagrama de flujo que se observa en la Figura 3.1.

El botón “Información” muestra una descripción del programa y lo que se puede realizar mediante el mismo, el botón “Seleccionar Robot” accede a la pantalla que permite seleccionar entre uno de los robots predefinidos. El último botón es “Diseñar Robot” y está destinado a que el usuario diseñe las características cinemáticas de un robot manipulador.



Figura 3.2: Pantalla de Inicio (Fuente propia)

3.1.3. Pantalla Seleccionar Robot

Esta pantalla permite seleccionar entre uno de los robots ya definidos previamente dentro del programa para los cuales se ha elegido los modelos de robots manipuladores más comunes en las universidades y en la industria, el diagrama de flujo de lógica para seleccionar el robot se observa en la Figura 3.3 mientras que el diseño de la pantalla se observa en la Figura 3.4.

La pantalla consiste de un popup menú que se despliega con las opciones de los diferentes robots disponibles y dos botones, uno que permite regresar a la pantalla de inicio y otro que accede a la pantalla de programación. La base de datos que conforma los robots del toolbox desarrollado por Arturo Gil en la Universidad Miguel Hernandez de España, el cual es un toolbox muy completo que posee funciones bien diseñadas para aplicaciones roboticas de cinemática directa, cinemática inversa, visión artificial, etc. En este caso es bastante útil porque posee ya una base de Datos de robots manipuladores de una gran cantidad de marcas y modelos donde ya están definidos los parámetros de Denavit Hartenberg que son la entrada que requiere el toolbox a desarrollar, pero además posee en varios de los modelos una serie de archivos cad stl que se enlazan con MATLAB para observar simulaciones muy reales de los

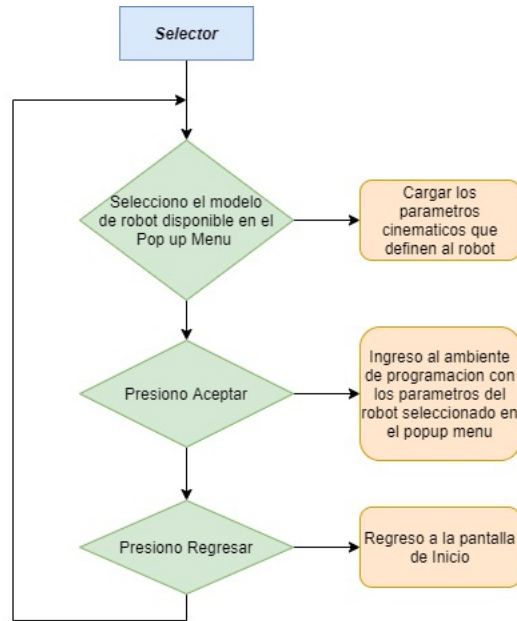


Figura 3.3: Diagrama de Seleccionar Robot (Fuente propia)

robots disponibles y con la posibilidad de agregar mas robots. Las marcas y modelos disponibles que se han agregado al toolbox en la pantalla de Selección de Robots las cuales fueron obtenidas de la librería ARTE. El algoritmo del botón Importar permite utilizar los datos de la librería ARTE [Castelló et al., 2018], al iniciar la librería y convocar a un robot por su marca y modelo, el toolbox ARTE crea un objeto del robot con varios datos de los cuales se extrae los parámetros de Denavit Hartenberg, el número de grados de libertad, los límites de movimiento y la representación gráfica de los modelos.

3.1.4. Diseñar Robot

Esta pantalla permitirá ingresar los parámetros cinemáticos del robot que se simulará mediante la matriz de Denavit Hartenberg, consta de un Text Edit para para escribir el número de grados de libertad del robot, lo cual permite definir el tamaño de la matriz, el siguiente elemento importante es la matriz que puede ser llenada con los parámetros de Denavit Hartenberg y la matriz de límites que define los máximos y mínimos permitidos en el



Figura 3.4: Pantalla Seleccionar Robot (Fuente propia)

movimiento de las articulaciones del robot. Finalmente en el menú de la parte inferior se encuentran los botones que permiten regresar a la página de inicio o ingresar a la pantalla de programación el diseño de la pantalla se observa en la Figura 3.5 y la lógica de esta sección se encuentra detallada en el diagrama de la Figura 3.6.

3.1.5. Pantalla de Programación y Simulación

Esta pantalla permitirá simular, programar y poner en línea el robot activando el puerto serial que controla los motores del robot RV2AJ moverlo mediante cinemática inversa o directa con los menús en el centro de la pantalla, el diseño se puede observar en la Figura 3.7. Además, posee un botón teach que permitirá guardar una posición deseada con una velocidad a escoger un botón de home para que el robot vuelva a su posición inicial, el menú de cinemática inversa para generar trayectorias donde se programan los algoritmos genéticos.

Esta pantalla es la principal y la que posee más botones interactivos, las acciones principales de los botones son las siguientes:

- Home: Permite regresar al robot simulado a la posición home, la lógica de programación del botón se puede observar en la Figura 3.8.

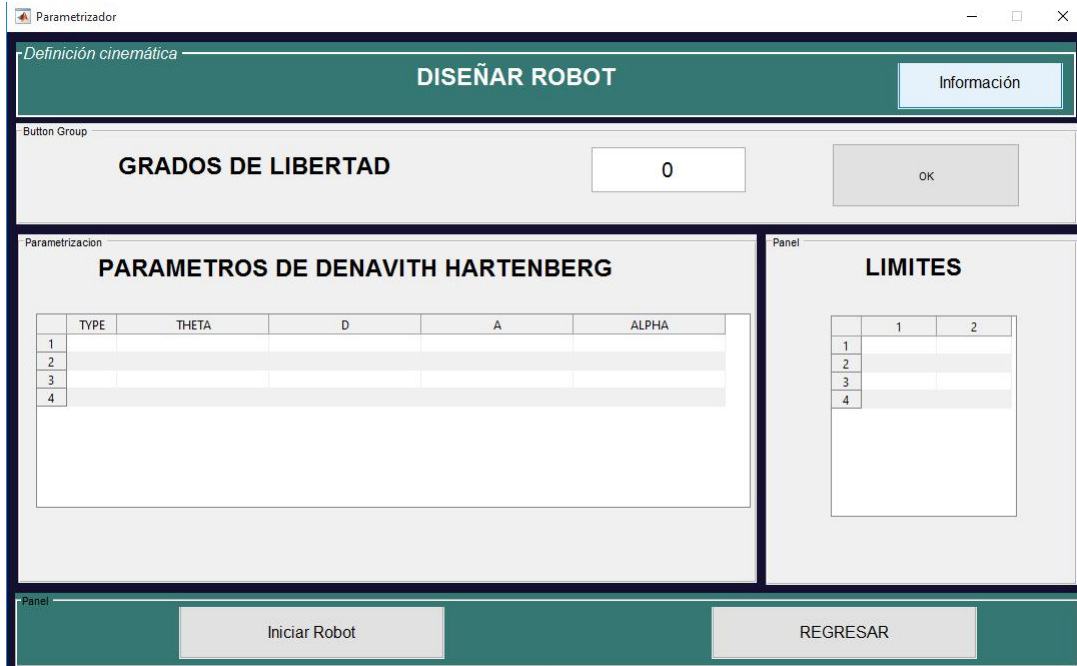


Figura 3.5: Pantalla Diseñar Robot (Fuente propia)

- Guardar: Permite guardar los puntos y trayectorias generadas con el botón teach.
- Abrir: Permite abrir un programa previamente guardado
- Borrar: Borra la última posición guardada
- Borrar Todo: Borra todas las posiciones guardada
- 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.

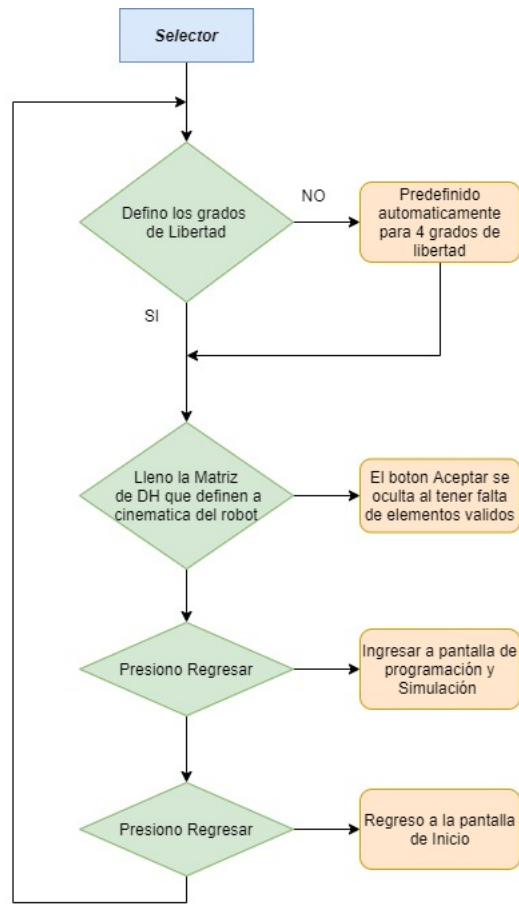


Figura 3.6: Diagrama de Diseñar Robot (Fuente propia)

- Menú Principal: Regresa a la pantalla de inicio.
- Simular: Permite simular la trayectoria del robot mediante un gráfico 3d, así como visualizar la posición.
- Teach: Este botón permite guardar la posición actual del robot, para luego simularla o reproducirla con el robot físico, Se pueden guardar las líneas que sean necesarias
- Guardar: Permite guardar los puntos y trayectorias generadas con el botón teach.

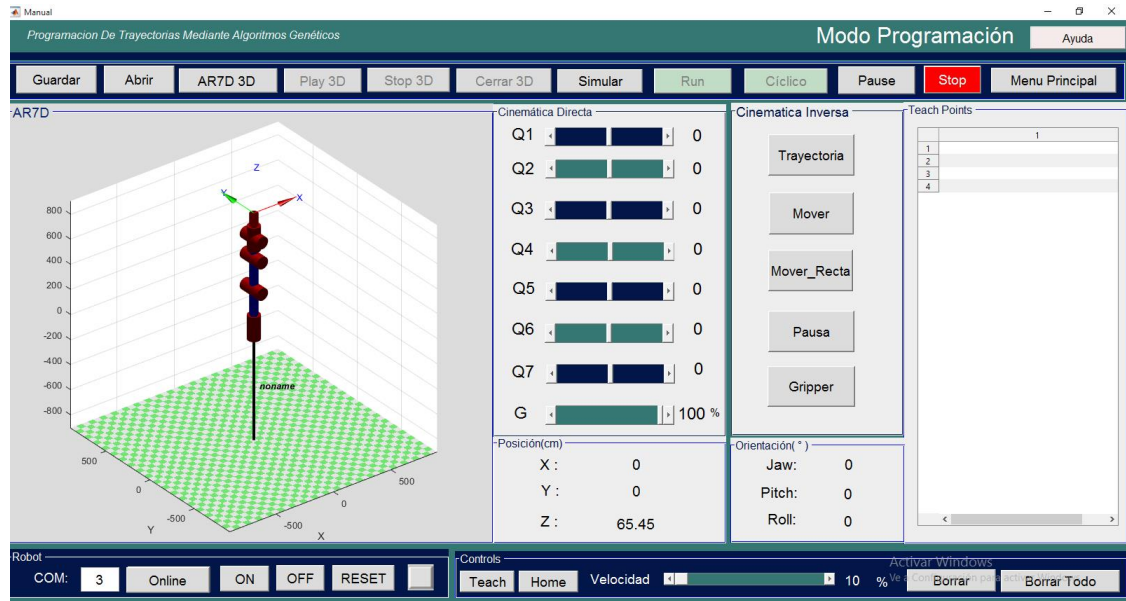


Figura 3.7: Pantalla Programación y Simulación (Fuente propia)

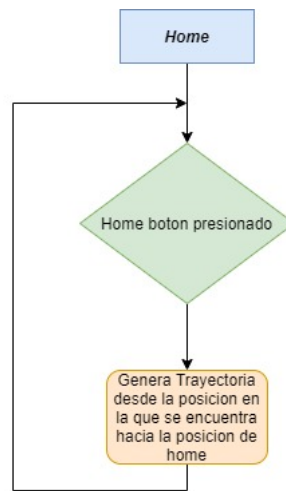


Figura 3.8: Diagrama de flujo- botón home (Fuente propia)

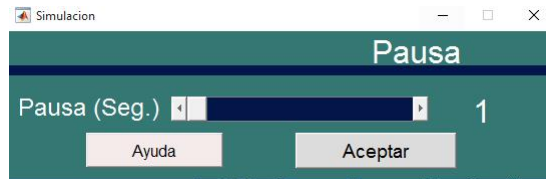


Figura 3.9: Pantalla – Pausa (Fuente propia)

3.1.6. Pausa

El comando Pausa permite programar una pausa de acuerdo a la tarea que se desee que el robot realice (Figura 3.9).

3.1.7. Pantalla Mover

Al presionar sobre el botón mover en la pantalla de Programación, se abrirá la sub-pantalla Mover (Figura 3.11), en esta pantalla se puede ingresar los datos de orientación en grados y posición, seleccionar una velocidad, y con el botón calcular se resolverá la cinemática inversa mediante un algoritmo genético, el diagrama de flujo en la Figura 3.10 muestra la lógica de programación.

3.1.8. Pantalla Mover recta

Al presionar mover recta en la pantalla de programación se despliega la sub- pantalla mover recta, en la que se escribe una posición deseada y al presionar calcular se interpolará y resolverá la cinemática inversa mediante algoritmos genéticos procurando mantener la orientación 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.

3.1.9. Pantalla Trayectoria

Al presionar trayectoria se abre una pantalla que permite crear o importar una trayectoria con los vectores x , y , z , la trayectoria puede ser cualquiera dentro del ambiente de trabajo, esta es la función que tiene una ventaja muy alta ya que puede generar trayectorias complejas o inclusive con adquisición de datos lo que permite un trabajo futuro para potenciar el toolbox Figura 3.12.

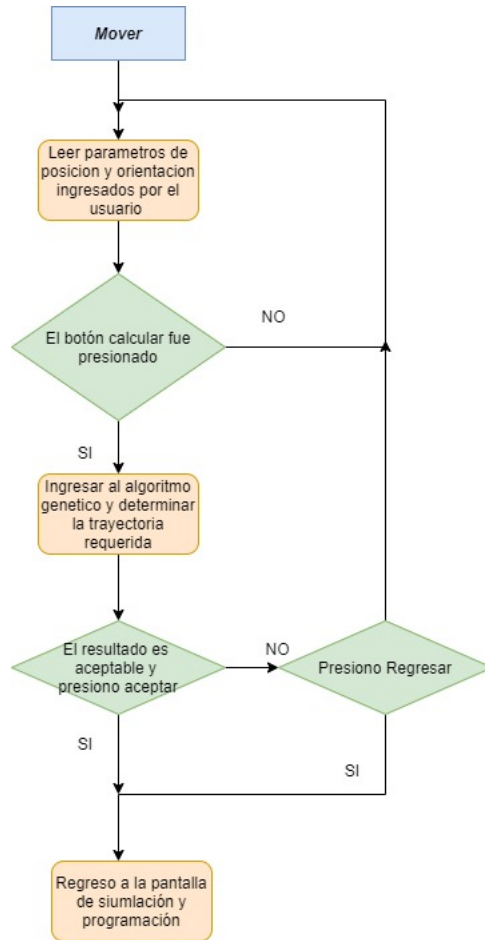


Figura 3.10: Diagrama de flujo - mover (Fuente propia)

Dentro de la pantalla trayectoria se tiene la opción de trayectoria interactiva Figura 3.13 la cual permite trazar una serie de puntos interpolados con el mouse en el entorno de trabajo, esta opción es bastante útil y abre un campo importante para una futura investigación ya que la acción del mouse puede ser reemplazada o potenciada con una sensor de acelerómetro o con visión artificial. En la Figura 3.14 se muestra la interfaz de esta pantalla, para lograr que los puntos adquiridos se den manera intuitiva como cualquier programa de dibujo CAD, para lo cual se han utilizado dos funciones de MATLAB:



Figura 3.11: Pantalla -Mover (Fuente propia)

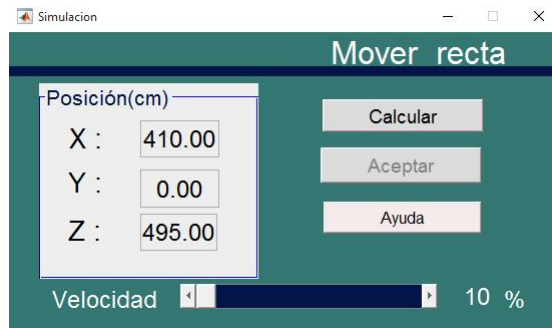


Figura 3.12: Pantalla – mover recta (Fuente propia)

$x,y,boton = ginput(1)$: El cual permite adquirir la posición del mouse en la gráfica en los vectores x,y y el estatus de los botones de mouse en el vector botón

- $Get(gca, 'CurrentPoint')$: El cual permite complementar a $ginput$ para que funcione en 3 dimensiones ya que retorna la posición del mouse en cada plano dentro de una imagen 3D y además se ha utilizado una variable de offset para desplazarse en profundidad dependiendo de cada plano, en la Figura 3.15 se puede observar el diagrama del algoritmo del graficador.

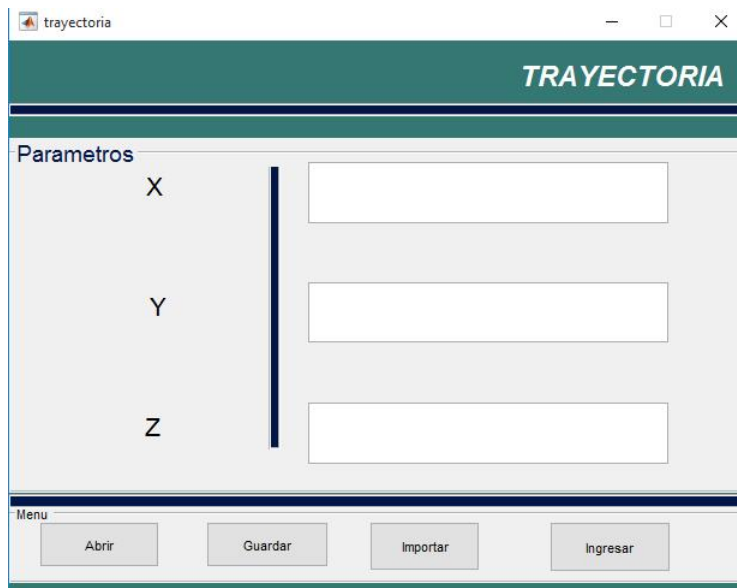


Figura 3.13: Pantalla – Trayectoria (Fuente propia)

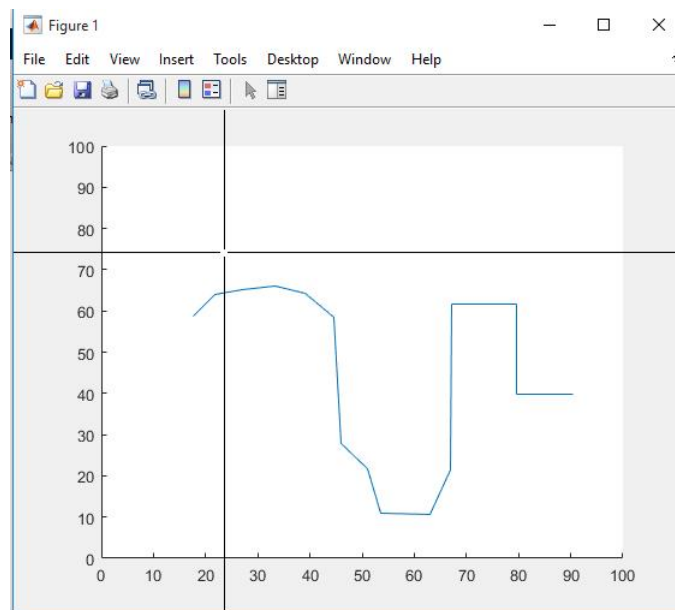


Figura 3.14: Trayectoria interactiva (Fuente propia)

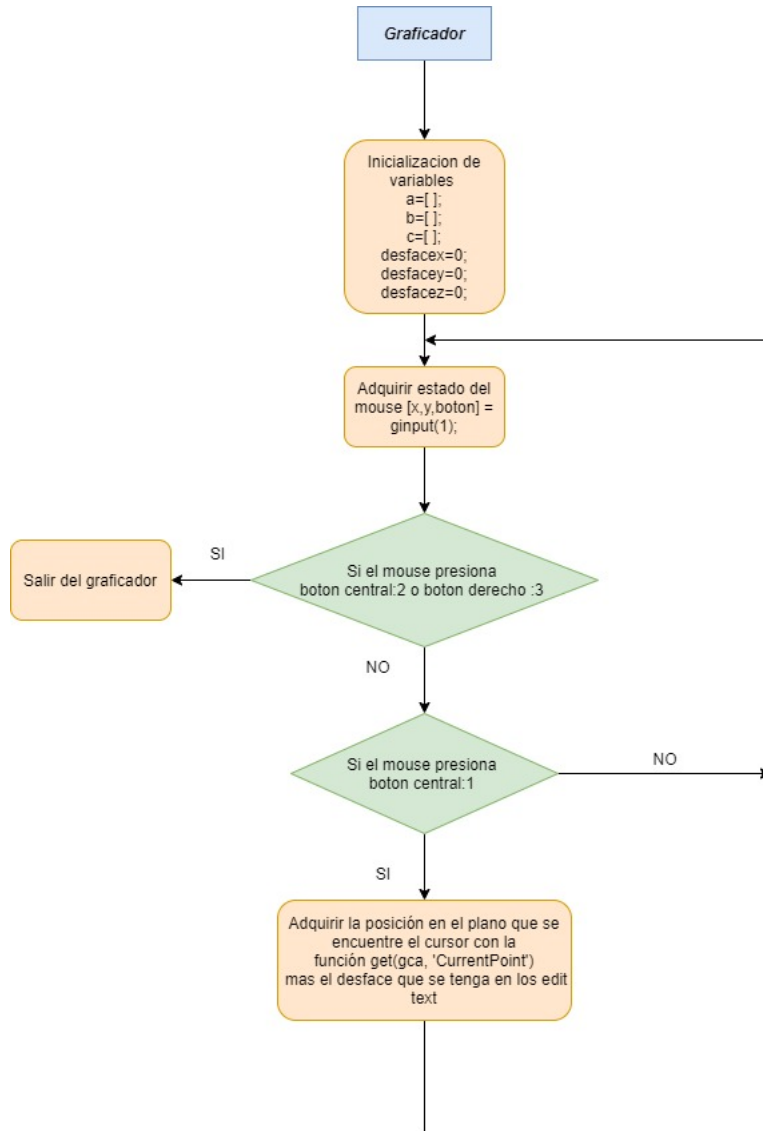


Figura 3.15: Diagrama del algoritmo para trayectoria interactiva (Fuente propia)

Capítulo 4

Algoritmos genéticos para solución de trayectorias

En este capítulo se analiza en primer lugar como se definen los parámetros iniciales de un robot mediante un ejemplo con el robot Mitsubishi Rv2Aj, a continuación se encuentra la utilización de gatool de matlab y como se seleccionó los parámetros de las funciones del mismo para obtener los mejores resultados en la solución del problema.

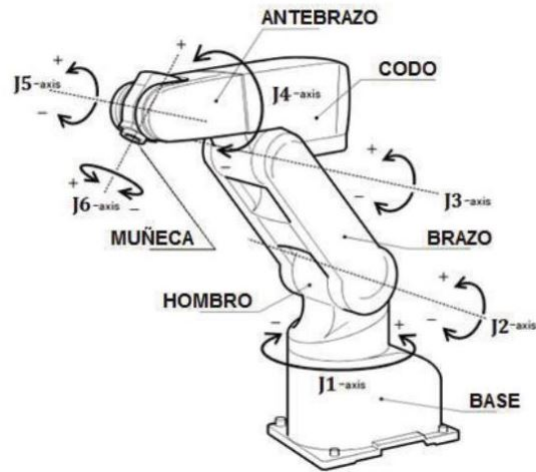


Figura 4.1: Articulaciones Robot Mitsubishi RV-2AJ [Slamet and dan Elektronik, 2013]

4.1. Análisis Cinemático.

Para empezar con el análisis cinemático es necesario obtener los parámetros de Denavit Hartenberg que representan al robot ya que esta matriz contiene los parámetros de entrada del Toolbox, la matriz es diferente dependiendo de las dimensiones del robot, números de grados de libertad, configuraciones de los eslabones y límites del robot este análisis da como resultados una matriz que es ingresada en el Toolbox. Para demostrar el ejemplo de funcionamiento se utilizó el robot Mitsubishi RV2AJ (Figura 4.1), que tiene la configuración cinemática que se observa en la Figura 4.2 y cuyos parámetros D-H vienen dados por la Tabla 4.1.

Es necesario conocer los límites de movimiento de los motores de Robot ya que permiten restringir la búsqueda del algoritmo dentro de este rango, los límites del Robot Mitsubishi RV2AJ vienen dados en su manual por la Tabla 4.2

Los parámetros se ingresan mediante la pantalla de parametrización Figura 4.3, la cual crea la matriz de DH y a continuación se procede a crear el robot como un objeto utilizando la herramienta desarrollada por Pete Corke Robotic Toolbox para Matlab la cual genera una serie de herramientas para graficar y operar matrices orientadas a la robótica, el diagrama de flujo

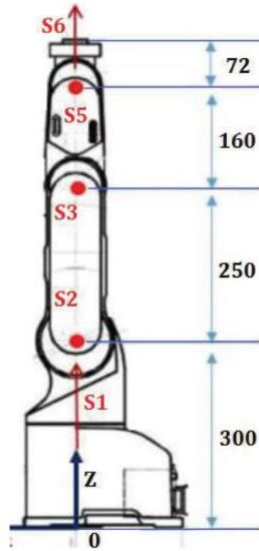


Figura 4.2: Dimensiones del Robot Mitsubishi RV-2AJ [Slamet and dan Elektronik, 2013]

Tabla 4.1: Parámetros de Denavit Hartenberg del robot Mitsubishi RV2AJ [Slamet and dan Elektronik, 2013]

Eslabón	a	alpha	d	theta
1	0	90°	300	Q1
2	250	0°	0	Q2
3	160	0°	0	Q3
4	0	-90°	0	Q4
5	0	0°	72	Q5

explica cómo se genera el robot como objeto interactivo, para lo cual se utiliza los comandos:

- Link (Contiene la información de DH de cada eslabón)
- Serial Link (Une los parámetros de cada eslabón y crea el Robot como objeto)
- Plot (Gráfica el objeto creado)

Tabla 4.2: Límites del Robot Mitsubishi RV2AJ [Slamet and dan Elektronik, 2013]

Articulación	Límites de movimiento (grados)
J1	+ -150
J2	180 (-60+120)
J3	230 (-110+120)
J5	+ -90
J6	+ -200

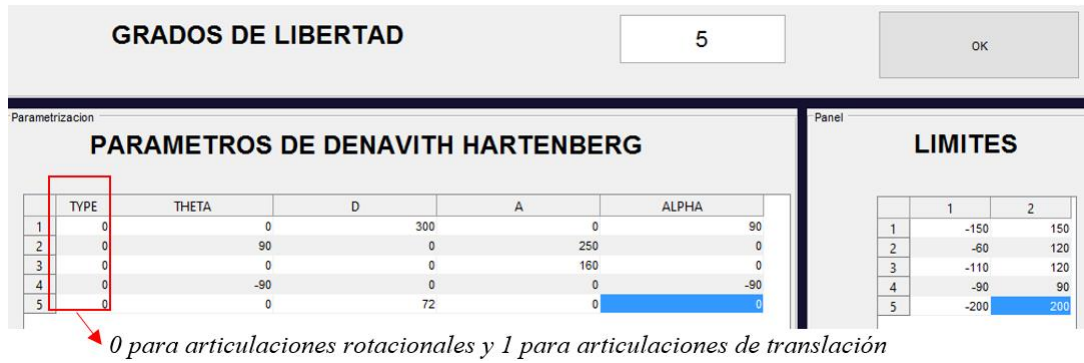


Figura 4.3: Ingreso de Datos del Robot Mitsuishi RV2AJ (Fuente propia)

- FKine (Resuelve la cinemática directa con los valores iniciales ini en este caso con todos los ángulos theta igual a cero)

Al dar aceptar se obtiene un diagrama en tres dimensiones con las dimensiones del robot como se observa en la Figura 4.4, la lógica del ingreso de parámetros se observa en el diagrama de la Figura 4.5.

4.2. Solución de la cinemática inversa mediante un algoritmo genético

Se utilizó el toolbox de Matlab gatool el cual 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);`

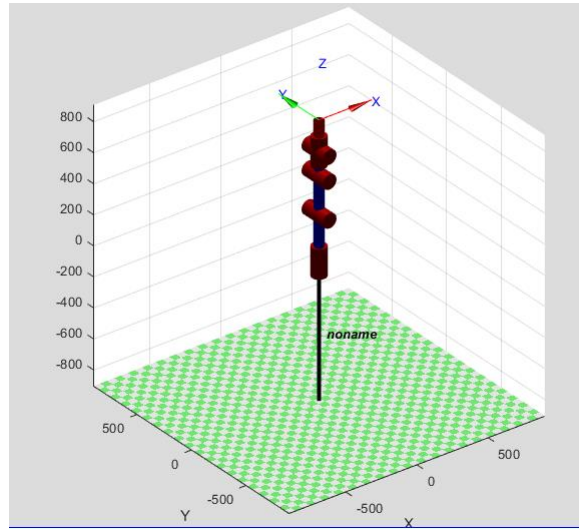


Figura 4.4: Gráfico en 3 dimensiones del Robot Mitsubishi generado con herramienta de Peter Corke (Fuente propia)

El algoritmo genético de Matlab trabaja de la siguiente manera: Recibe parámetros iniciales como número de variables, restricciones y límites de las variables, los límites serían 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 último 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

4.2.1. 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 permite restringir ángulos que no pueden ser alcanzados físicamente, o ángulos que pueden causar una colisión, las restricciones para los robots vienen definidas

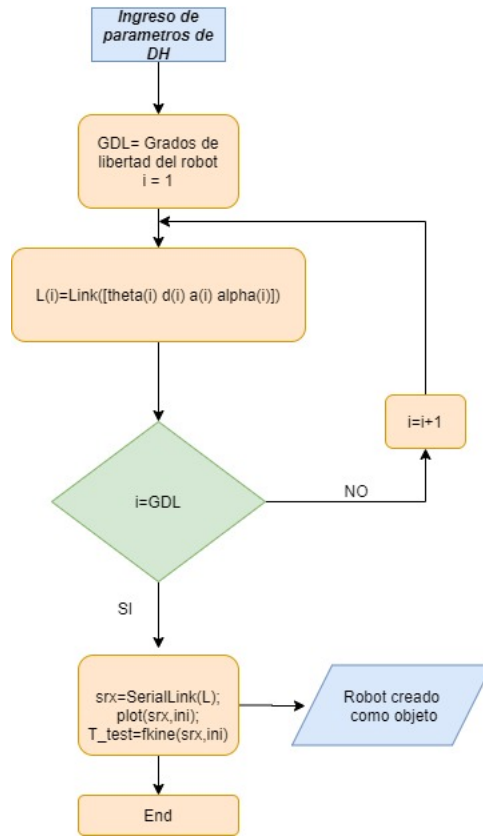


Figura 4.5: Ingreso de parámetros de Denavit Hartenberg (Fuente propia)

en una tabla al cargar los adtos de entrada del programa.

4.2.2. 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)`.

4.2.3. Selección

Para la función de selección se utilizó la opción de gatool 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. Esta es la función de selección predeterminada en el gatool de Matlab.

4.2.4. Mutación

Para la función de mutación que se utiliza la opción de 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.

4.2.5. 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]

4.2.6. Función de aptitud

La Función de aptitud o función objetivo es el modelo en el cual se va a evaluar 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 es multiobjetivo con tres que se requieren 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,

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

Figura 4.6: matriz de transformación Tf [Barrientos et al., 2007]

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

Figura 4.7: Matriz homogénea para cada eslabón mediante los parámetros de Denavit Hatenberg [Barrientos et al., 2007]

estos son posición en x, y, z, orientación en los ángulos de Euler pitch, yaw, roll.

- Pitch: Gira el sistema con respecto al eje Y
- Yaw: Gira el sistema con respecto al eje Z
- Roll: Gira el sistema con respecto al eje X

La matriz de transformación Tf (Figura 4.6), representa los parámetros se desea alcanzar y será constante dentro del algoritmo genético, a continuación hay que hallar la matriz de transformación que variará hasta acercarse a la matriz deseada Tf. La matriz homogénea para cada eslabón (Figura 4.7) es desarrollada mediante los parámetros de Denavit Hatenberg.

Multiplicando las matrices de cada eslabón se obtiene la matriz de transformación homogénea del sistema representado por la ecuación 4.1:

$$T = {}^0 A_1 * {}^1 A_2 * \dots * {}^{n-1} A_n \quad (4.1)$$

Por último se diseña la función de aptitud en términos de las matrices Tf y T, para lo cual consideramos:

Error de posición (epos): Es la diferencia entre los términos de posición de las matrices de transformación Tf y T como se observa en la ecuación 4.2.

$$epos = [Tf(1,4) - T(1,4)]^2 + [Tf(2,4) - T(2,4)]^2 + [Tf(3,4) - T(3,4)]^2 \quad (4.2)$$

Error de orientación (eor) Es la diferencia entre los términos de orientación de las matrices de transformación Tf y T como se observa en la ecuación 4.3.

$$\begin{aligned} eor = & [Tf(1,1) - T(1,1)]^2 + [Tf(1,2) - T(1,2)]^2 + [Tf(1,3) - T(1,3)]^2 \\ & \dots + [Tf(2,1) - T(2,1)]^2 + [Tf(2,2) - T(2,2)]^2 + [Tf(2,3) - T(2,3)]^2 \\ & \dots + [Tf(3,1) - T(3,1)]^2 + [Tf(3,2) - T(3,2)]^2 + [Tf(3,3) - T(3,3)]^2 \end{aligned} \quad (4.3)$$

Mínimo cambio articular (mca). El mínimo cambio articular definido en la ecuación 4.4, está en función de la posición anterior y la posición actual, al minimizar esta función se garantiza que el robot tenga el mínimo cambio articular posible con respecto a la posición previa .

$$mca = \sum_{i=1}^n (q0(i) - q(i))^2 \quad (4.4)$$

Donde:

- q0= ángulo de la posición anterior
- q= ángulo de la posición actual
- i= número de articulación

Para la función de aptitud se usa la exponencial inversa de manera que los valores resultantes 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 ecuación 4.5. 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.

$$faptitud = e^{-\frac{1}{\sqrt{epos + C * eor + mca}}} \quad (4.5)$$

4.2.7. Tamaño de la población y generaciones

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. La cinemática inversa tiene que explorar un espacio amplio debido 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.

Para el toolbox desarrollado se tomó una población de 100 individuos por generación, con un total de 400 generaciones. El número de generaciones significa las veces que una población vuelve a pasar por los procesos de selección, mutación, etc. El máximo número de generaciones es también un criterio de parada, por la dificultad de resolver la cinemática inversa se ha considerado al menos 400 generaciones.

4.2.8. Generación de trayectorias mediante interpolación mediante polinomios

La interpolación mediante polinomios permite obtener una trayectoria en la cual tenemos varios puntos de restricción, en este caso se conoce la posición inicial y gracias a la cinemática inversa conocemos la posición final, además es utilizando la interpolación se asegura 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.

4.2.9. Trayectorias rectas

La trayectoria recta es muy común en los robots industriales ya que hay muchas aplicaciones que requieren de movimientos rectos, es por ello que en el toolbox se tiene la opción de generar trayectorias rectas en 3 dimensiones siempre y cuando estén dentro del espacio de trabajo del robot, para realizar

la trayectoria se utiliza la función de Matlab `linspace(x1,x2,n)` genera n puntos. El espaciado entre los puntos es $(x2-x1)/(n-1)$. `linspace` es similar al operador de dos puntos, “:”, pero proporciona control directo sobre el número de puntos y siempre incluye los extremos. “lin” en el nombre “linspace” se refiere a generar valores espaciados linealmente, a diferencia de su función hermana `logspace`, que genera valores espaciados logarítmicamente

4.2.10. Trayectorias indefinidas

Si bien la solución de la cinemática inversa mediante algoritmos genéticos es sumamente útil para robots redundantes de más de 6 grados de libertad, en robots redundantes como no redundantes la ventaja es que permite resolver una trayectoria irregular que no está definida por arcos exactos o líneas rectas sino por una serie de puntos que representan posiciones en el espacio de trabajo, esto permite realizar tareas más complejas sin necesidad de programar con los lenguajes regulares con Melfa. El algoritmo para resolver una trayectoria se ha definido de manera diferente, se trata de resolver la cinemática inversa en cada punto cuyos vectores corresponden a la posición en x en y y en z y además se requiere mantener la orientación que el robot tiene al iniciar la trayectoria, el algoritmo genético implementado se diferencia principalmente en que los límites superior e inferior permisibles en cada articulación se han limitado a un máximo de 20° sin sobrepasar el límite permitido, esto es necesario para que el algoritmo no busque ángulos muy separados entre un punto y otro de manera que presente un movimiento suave a lo largo de la trayectoria, al ser por lo general puntos cercanos no necesitan un universo amplio ya que el punto siguiente está próximo, así se optimiza el movimiento y se logra una trayectoria suave.

Capítulo 5

Pruebas Y Resultados

El toolbox es sometido a una serie de pruebas de sus funcionalidades, las cuales son el modo de programación teach mediante cinemática directa, prueba de el algoritmo genético para resolver trayectoria graficadas en el espacio de trabajo del robot, la utilidad del algoritmo para robots redundantes y las pruebas de la comunicación entre el toolbox y el robot Mitsubishi Rv2aj, así como las trayectorias que se probaron en el mismo.

5.1. Prueba de la funcionalidad de la opción teach para cinemática directa en el robot FANUC LRMATE 200iC

La prueba consiste en importar correctamente los parámetros del robot Fanuc LRMATE -200iC, a continuación observar su gráfica interactiva en tres dimensiones la cual se puede manipular mediante los sliders de la pantalla de programación (Figura 5.1), que moverá el robot en cinemática directa y mediante el botón teach se guarda 3 posiciones deseadas con tres diferentes velocidad determinadas por el slider de velocidad. El robot FANUC LRTMATE 200iC tiene seis grados de libertad rotacionales, los ángulos de las articulaciones son las siguientes:

- P0: 0 0 0 0 0 0
- P1: 0.7156 0 0 0.7156 0.6283 0
- P2: -0.8552 -0.6109 -0.7330 0.7156 1.4312 -1.2043
- P3: 0.8029 -1.2566 -0.7330 0.7156 1.4312 -1.2043

La primera prueba se realiza con las siguientes velocidades:

- Velocidad de P0 a P1 : 40
- Velocidad de P1a P2 : 40
- Velocidad de P2 a P3 : 80

La segunda prueba se realiza con las siguientes velocidades:

- Velocidad de P0 a P1 : 10
- Velocidad de P1a P2 : 80
- Velocidad de P2 a P3 : 20

A continuación se simula el movimiento del robot (Figura 5.2) a la velocidad deseada y se cambia la velocidad de cada una de las trayectorias a través de la tabla de velocidades de la pantalla de programación, para que realice el mismo movimiento pero a diferente velocidad.

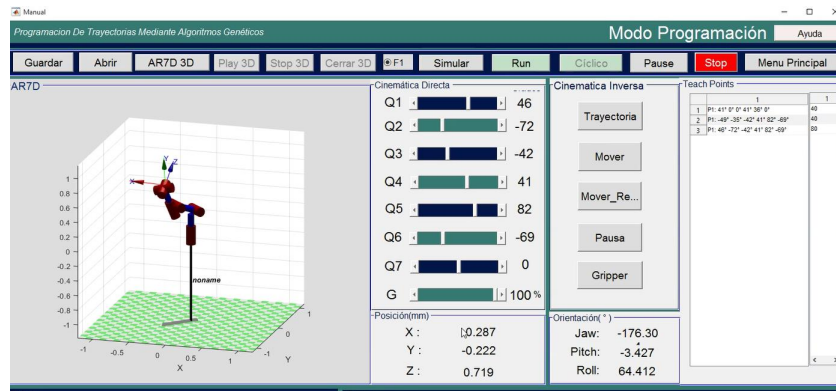


Figura 5.1: Pantalla de programación mediante la opción de teach de 3 puntos a velocidades diferentes en el robot Fanuc LRMate 200iC.

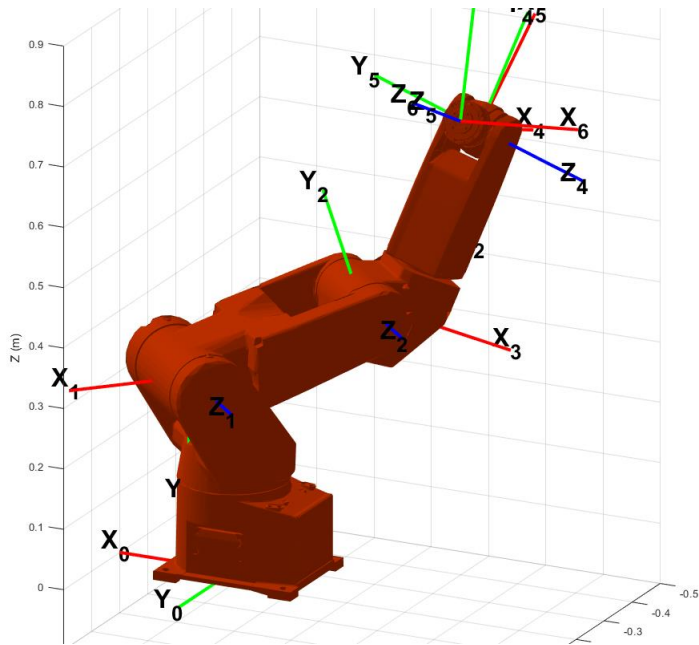


Figura 5.2: Simulación de movimiento 3d del robot FANUC LRMTATE 200iC (Fuente propia)

5.2. Resultados de la funcionalidad de la opción teach para cinemática directa en el robot Fanuc LRMATE-200iC.

Para visualizar los resultados de la trayectoria es necesario analizar la posición, velocidad y aceleración de todo el recorrido del robot, en el caso de esta prueba tenemos como resultado la matriz de cada punto interpolado dentro de la trayectoria dependiendo de la velocidad a la que fue inicializado el movimiento, la derivada de la posición da como resultado la velocidad y la derivada de la velocidad da como resultado la aceleración, en Matlab para derivar usamos los comandos:

- Velocidad: $V = \text{diff}(\text{vertcat}(c, \text{pos}));$
- Aceleración: $Ac = \text{diff}(\text{vertcat}(c, v));$

En la Figura 5.3 se observa la posición con respecto al tiempo en cada una de las articulaciones del robot, la interpolación implementada permite cambios suaves, en la Figura 5.4 se observa las velocidades seleccionadas y en la Figura 5.5 se observa la aceleración.

De la misma manera la prueba dos que tiene diferentes velocidades se puede observar el cambio en el movimiento y en las trayectorias obtenidas tanto en posición (Figura 5.6), velocidad (Figura 5.7) y aceleración (Figura 5.8).

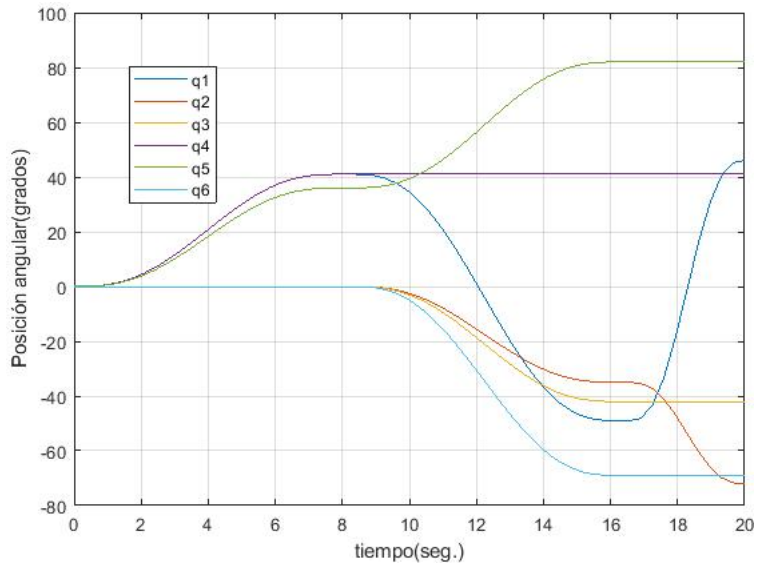


Figura 5.3: Resultados de posición de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

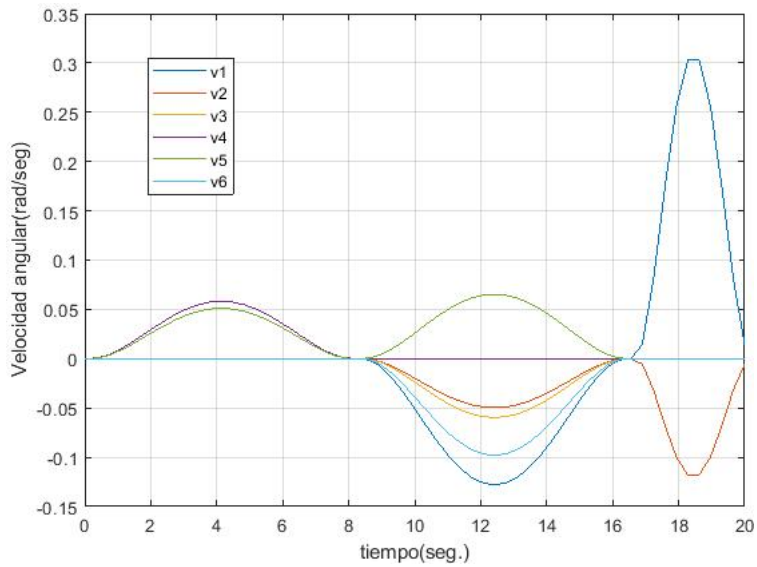


Figura 5.4: Resultados de velocidad de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

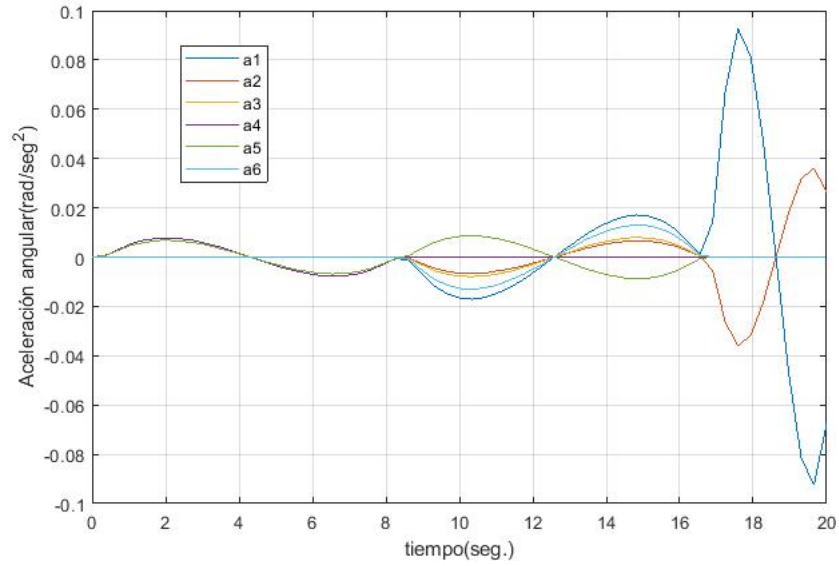


Figura 5.5: Resultados de aceleración de prueba 1 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

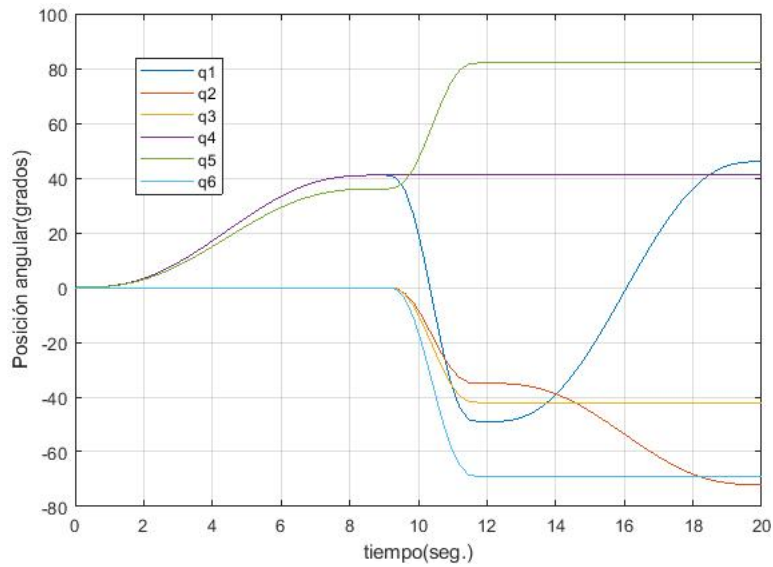


Figura 5.6: Resultados de posición de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

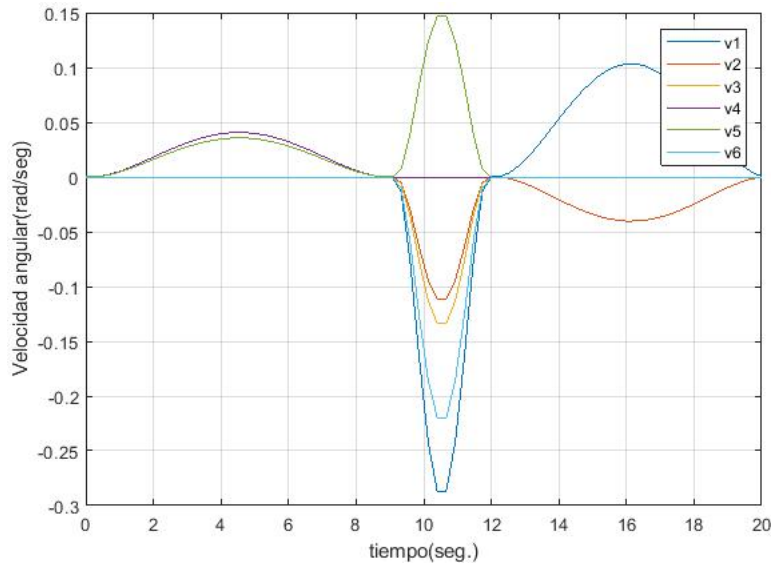


Figura 5.7: Resultados de velocidad de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

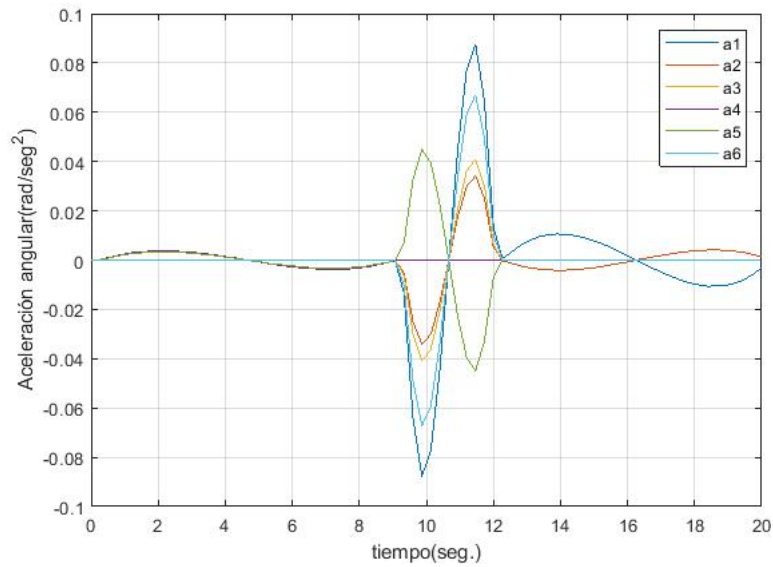


Figura 5.8: Resultados de aceleración de prueba 2 cinemática directa con el robot Fanuc LRMATE-200iC (Fuente propia)

5.3. Resultados del algoritmo

Para determinar la precisión del algoritmo para resolver trayectorias se ha seguido procedimientos de las normas utilizadas en robots industriales [ISO ISO 9283:1998]

5.3.1. Exactitud de posición y orientación

Según la norma ISO 9283:1998 la exactitud de posición se define en la ecuación 5.1 donde: x', y', 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.

$$Exactituddeposicin = \sqrt{(x' - x_c)^2 + (y' - y_c)^2 + (z' - z_c)^2} \quad (5.1)$$

donde:

$$x' = \frac{1}{n} \sum_{i=1}^n x_i$$

$$y' = \frac{1}{n} \sum_{i=1}^n y_i$$

$$z' = \frac{1}{n} \sum_{i=1}^n z_i$$

Para la exactitud de orientación definida por las ecuaciones 5.2, 5.3 y 5.4, los elementos a', b', 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.

$$Exactituddeorientacina = (a' - a_c) \quad (5.2)$$

$$Exactituddeb = (b' - b_c) \quad (5.3)$$

$$Exactituddeorientacinc = (c' - c_c) \quad (5.4)$$

donde:

$$a' = \frac{1}{n} \sum_{i=1}^n a_i$$

$$b' = \frac{1}{n} \sum_{i=1}^n b_i$$

$$c' = \frac{1}{n} \sum_{i=1}^n c_i$$

5.3.2. Repetibilidad de posición y orientación

De igual manera la norma indica como calcular la repetibilidad de posición con la ecuación 5.5.

$$RP = \sqrt{(x' - xc)^2 + (y' - yc)^2 + (z' - zc)^2} \quad (5.5)$$

donde:

$$l' = \frac{1}{n} \sum_{j=1}^n l_j$$

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

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

Mientras que la repetibilidad de orientación esta dada por las ecuaciones 5.6, 5.7 y 5.8.

$$RP = \pm 3 \sqrt{\frac{\sum_{j=1}^n (a_j - a')^2}{n - 1}} \quad (5.6)$$

$$RP = \pm 3 \sqrt{\frac{\sum_{j=1}^n (b_j - b')^2}{n - 1}} \quad (5.7)$$

$$RP = \pm 3 \sqrt{\frac{\sum_{j=1}^n (c_j - c')^2}{n - 1}} \quad (5.8)$$

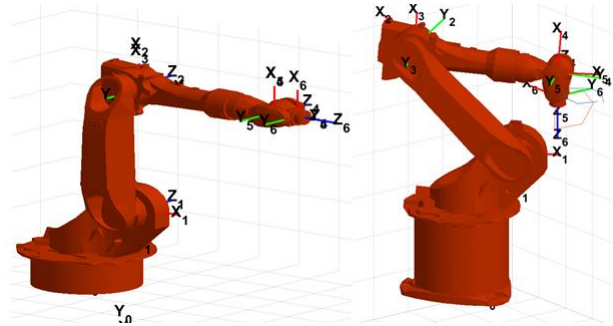


Figura 5.9: Trayectoria hexagonales en el espacio de trabajo del robot KUKA-KR16-2 (Fuente propia)

5.4. Prueba de la solución de cinemática inversa y trayectoria en dos planos diferentes mediante algoritmos genéticos en el robot KUKA-KR16-2

La prueba está diseñada para probar las funcionalidades principales del toolbox, en primer lugar se debe posicionar y orientar el robot desde home hasta el punto deseado, a continuación mediante la herramienta de dibujar trayectoria se dibujará una trayectoria de un hexágono irregular en el espacio en el plano xy , después se dibujará con la misma herramienta otro hexágono irregular en el plano xz , el objetivo de la prueba es que algoritmo genético resuelva la cinemática inversa de la trayectoria dibujada en cada uno de los planos y que permita seguir las trayectorias al robot de manera precisa manteniendo la orientación y con una velocidad deseada en cada trayectoria.

En la Figura 5.9 se observa los hexágonos en el espacio de trabajo del robot, en la Figura 5.10 y Figura 5.11 se observa desde una vista superior a cada uno de los planos los hexágonos dibujados con la herramienta para introducir una trayectoria.

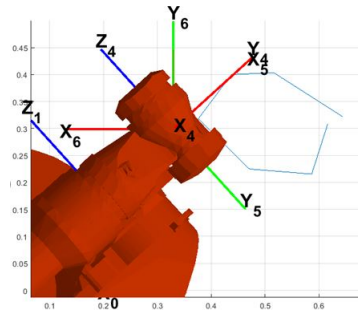


Figura 5.10: Primera trayectoria hexagonal en el espacio de trabajo del robot KUKA-KR16-2 vista superior del plano xy (Fuente propia)

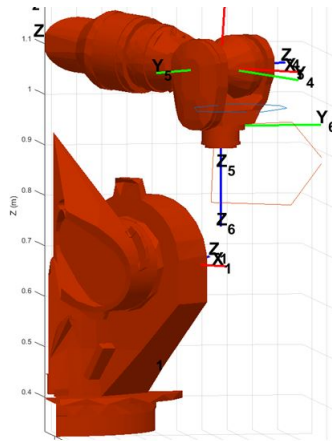


Figura 5.11: Segunda trayectoria hexagonal en el espacio de trabajo del robot KUKA-KR16-2 vista superior del plano xz (Fuente propia)

5.5. Resultados solución de cinemática inversa y trayectoria irregular en dos planos diferentes mediante algoritmos genéticos en el robot KUKA KR 16 2

La trayectoria fue seguida exitosamente manteniendo la orientación con un margen de error de 20mm, de igual manera como se hizo en las pruebas de cinemática directa se puede observar los resultados de las trayectorias en

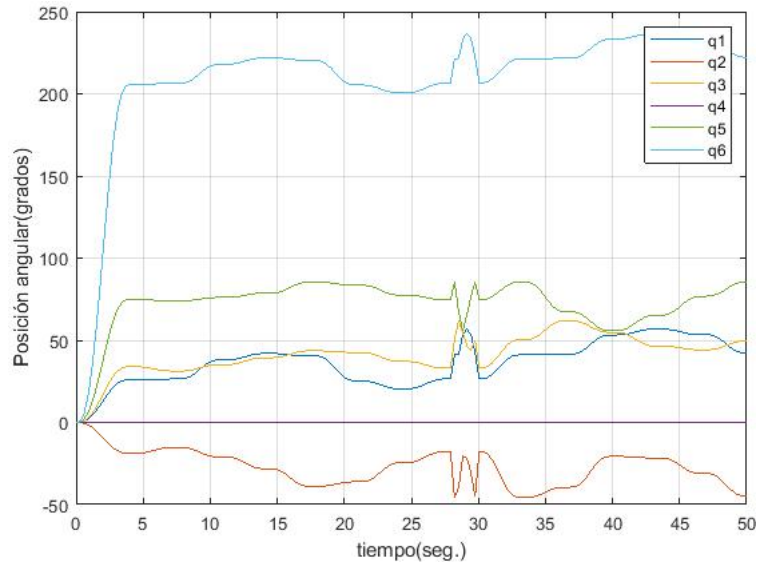


Figura 5.12: Resultados de posición de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)

Tabla 5.1: Exactitud del algoritmo genético en trayectoria robot KUKA-KR16-2

Exactitud	Valor
Posición	14mm
Yaw	1.8 ^o
Pitch	1.5 ^o
Roll	1.45 ^o

posición (Figura 5.12), velocidad (Figura 5.13) y aceleración (Figura 5.14). En las tablas 5.1 y 5.2 se observan los resultados de exactitud y repetibilidad calculados como se indicó en la sección 5.3 .

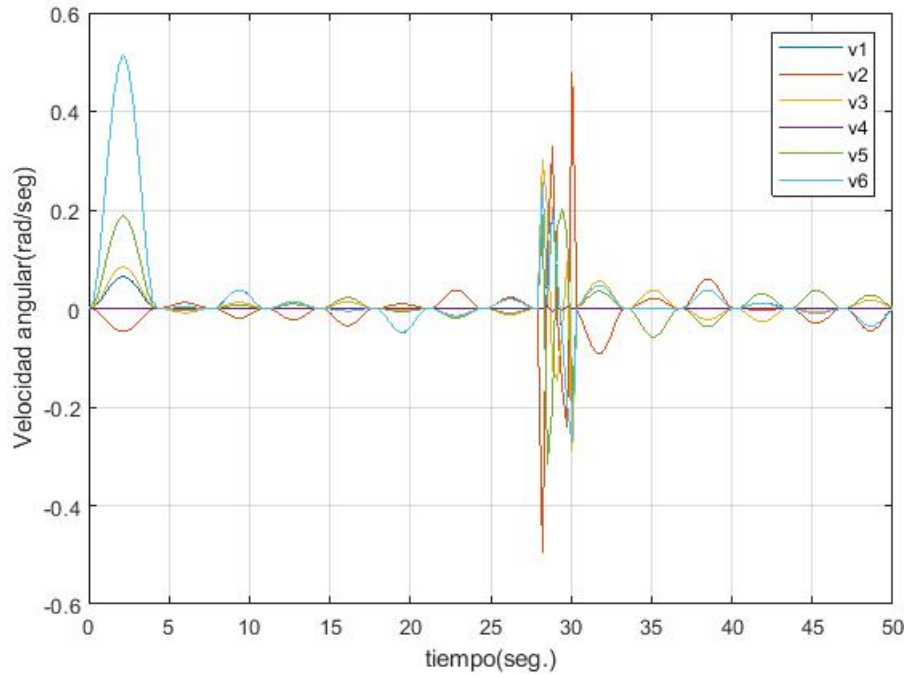


Figura 5.13: Resultados de velocidad de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)

Tabla 5.2: Repetibilidad del algoritmo genético en trayectoria robot KUKA-KR16-2

Repetibilidad	Valor
Posición	8.2mm
Yaw	1.4 ^o
Pitch	1.7 ^o
Roll	1.9 ^o

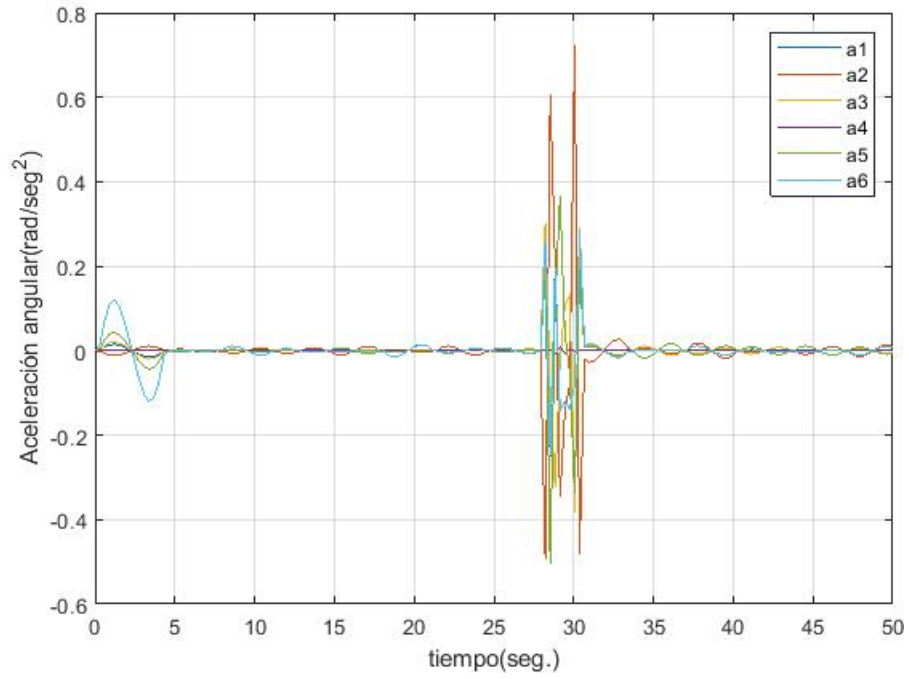


Figura 5.14: Resultados de aceleración de prueba cinemática inversa con el robot KUKA-KR16-2 (Fuente propia)

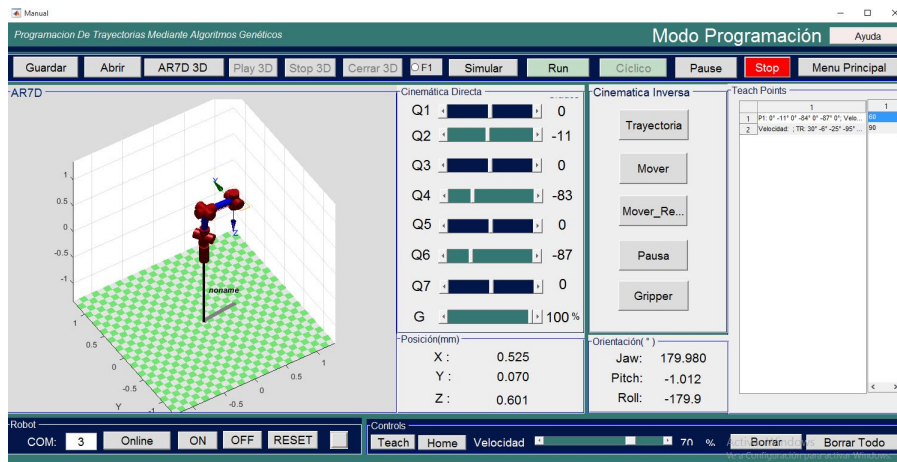


Figura 5.15: Pantalla de programación para Robot Mitsubishi 7gdl (Fuente propia)

5.6. Pruebas funcionales del toolbox con robot un redundante de 7 grados de libertad

La utilidad de usar algoritmos genéticos para resolver la cinemática inversa es bastante notoria en robots redundantes en los cuales no existe una solución cerrada de la cinemática inversa ya que la matriz de transformación da como máximo seis ecuaciones y seis incógnitas utilizando operaciones trigonométricas, sin embargo el método puede ser efectivo para resolver la cinemática inversa de robots redundantes como se analiza en esta prueba.

La prueba en este robot consiste en usar el graficador de trayectoria para dibujar un cuadrado y luego resolver esta trayectoria para el robot Mitsubishi 7gdl y simular el movimiento, la pantalla de programación de esta prueba se observa en la Figura 5.15.

5.7. Resultados del toolbox con robot un redundante de 7 grados de libertad

La trayectoria se ha seguido correctamente como se observa en la Figura 5.16 desde un plano isométrico y en la Figura 5.17 desde un plano superior. Los resultados son similares a los obtenidos para robots no redundantes lo que indica que el algoritmo converge de la misma manera para robots

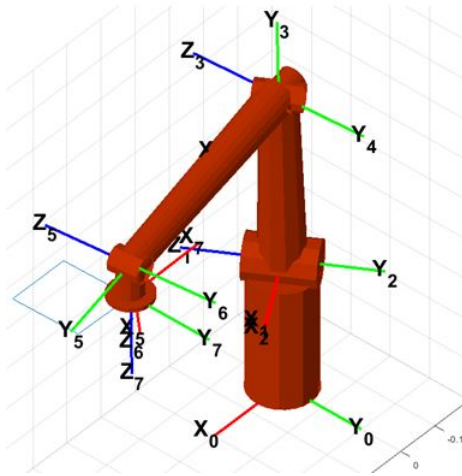


Figura 5.16: Cuadrado en el espacio de trabajo del Robot Mitsubishi 7gdl (Fuente propia)

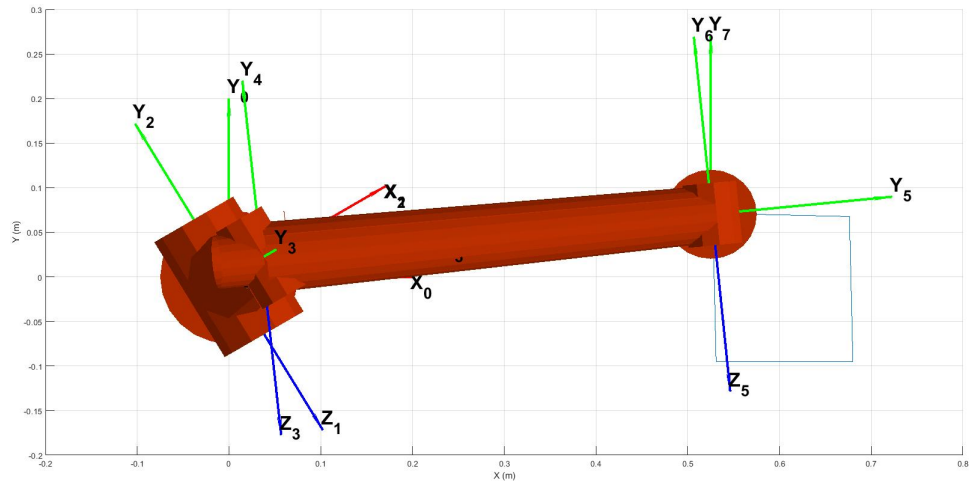


Figura 5.17: Vista superior de la trayectoria de prueba para Robot Mitsubishi 7gdl (Fuente propia)

redundantes, lo que demuestra que el método es general independientemente de los grados de libertad del robot a analizar, los resultados de posición

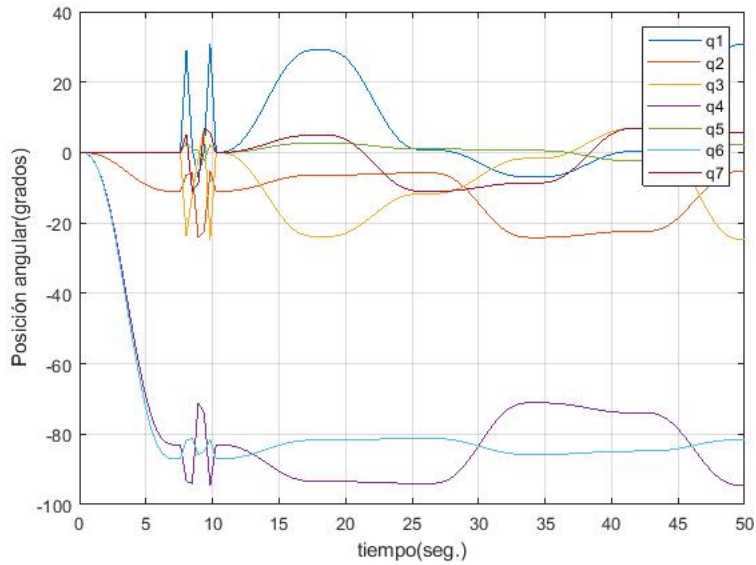


Figura 5.18: Resultados de posición de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)

Tabla 5.3: Exactitud del algoritmo genético en trayectoria robot redundante Mitsubishi 7gdl

Exactitud	Valor
Posición	20mm
Yaw	2 ^o
Pitch	2,2 ^o
Roll	2.3 ^o

(Figura 5.18) , velocidad (Figura 5.19) y aceleración (Figura 5.20) mantienen trayectorias sin discontinuidad y son transferibles a un robot real, de igual manera se ha calculado la precisión (Tabla 5.3) y repetibilidad (Tabla 5.4) utilizando las normas descritas en la sección 5.3.

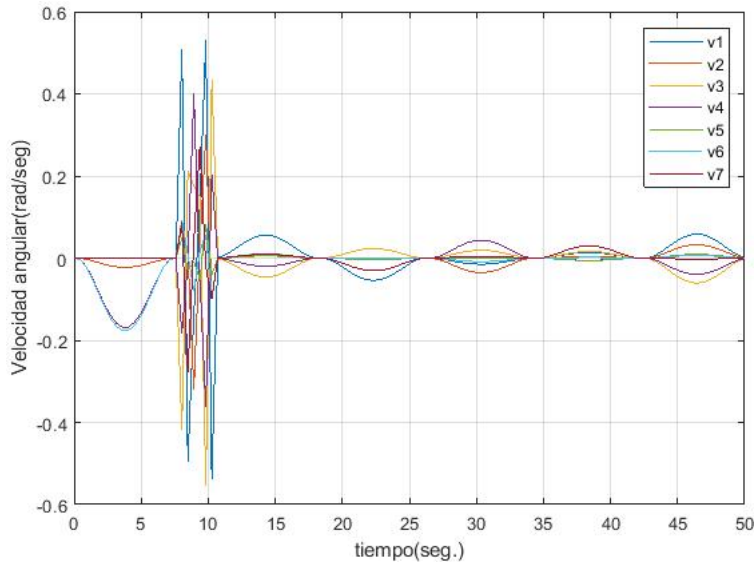


Figura 5.19: Resultados de velocidad de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)

Tabla 5.4: Repetibilidad del algoritmo genético en trayectoria robot redundante Mitsubishi 7gdl

Repetibilidad	Valor
Posición	9.6mm
Yaw	1.1 ^o
Pitch	1.9 ^o
Roll	1.4 ^o

5.8. Pruebas Funcionales del toolbox para la solución de trayectorias irregulares en el robot Mitsubishi RV2AJ.

El robot Mitsubishi RV2AJ está disponible en los laboratorios de la Universidad politécnica Salesiana y con el cual se ha podido realizar múltiples pruebas funcionales que permite vincular el toolbox con el robot en simulación y en operación físicamente.

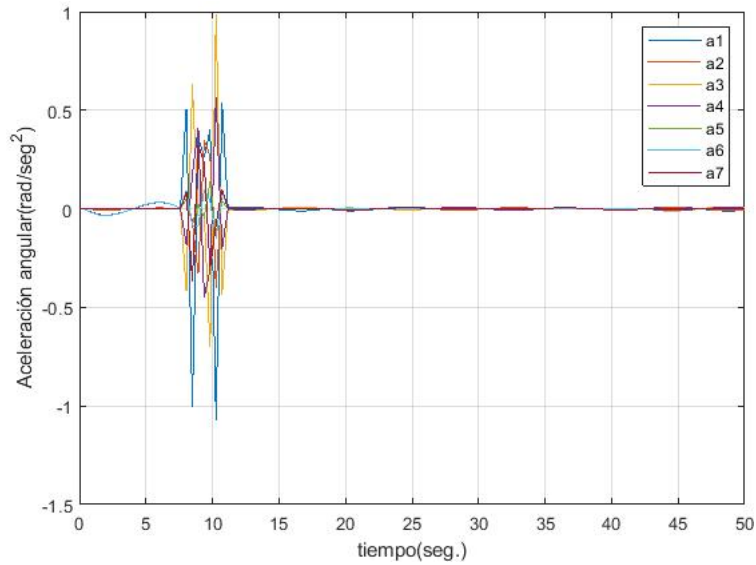


Figura 5.20: Resultados de aceleración de prueba cinemática inversa con el robot Mitsubishi 7gdl (Fuente propia)



Figura 5.21: Menú de comunicación Serial (Fuente propia)

5.8.1. Comunicación entre el robot Mitsubishi RV2AJ y el toolbox

Para la comunicación entre el robot y el toolbox se ha utilizado la comunicación serial entre el controlador del robot mediante un cable de rs232 a usb, después de instalar los controladores se seleccionara un com y se enviara al robot en línea para lo cual el toolbox tiene botones de comunicación serial en el menú inferior(Figura 5.21). Una vez que el robot está en línea se puede mover cada articulación mediante los sliders de cinemática directa y correr cualquier trayectoria resuelta mediante algoritmos genéticos

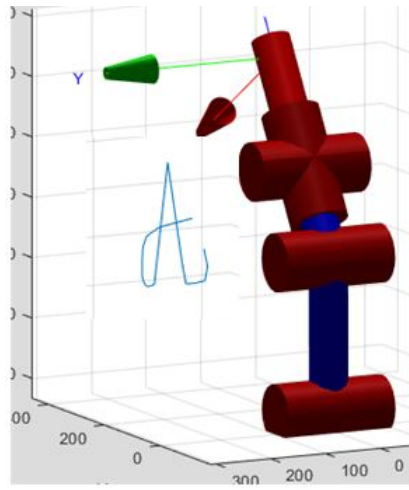


Figura 5.22: Letra A manuscrita dibujada en el espacio de trabajo (Fuente propia)

5.9. Prueba trayectorias con el robot Mitsubishi RV2AJ

Para probar la comunicación serial se ha utilizado el Robot Mitsubishi RV2AJ y se han realizado dos pruebas de escritura para observar el rastro de la trayectoria con mayor facilidad, para la escritura se ha utilizado la herramienta de trazado interactivo del toolbox desarrollado en la cual se puede dibujar una trayectoria en el espacio con el mouse, esto permite realizar cualquier dibujo y demostrar la gran ventaja que tiene el toolbox vs la programación normal que se utiliza para el robot, No es necesario aplicar ninguna línea de código para resolver la trayectoria ya que el toolbox se encarga de resolver la cinemática inversa e interpolar la trayectoria, las pruebas de escritura son:

- El trazado de una letra manuscrita A que se muestra en la Figura 5.22, la cual consiste en 84 puntos interpolados los cuales despues de ser importados al toolbox es resuelta por varias iteraciones del algoritmo genético y ha tardado 20 minutos en terminar de resolver la trayectoria, la cual se ha procedido simular para verificar los resultados obtenidos y a continuación se pone el línea al robot que cumple con la trayectoria satisfactoriamente como se observa en la Figura 5.23.

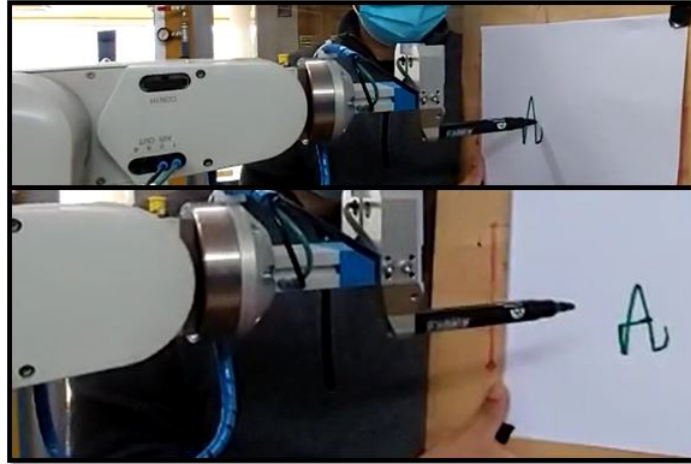


Figura 5.23: Robot Mitsubishi RV-2AJ dibujando la letra manuscrita (Fuente propia)

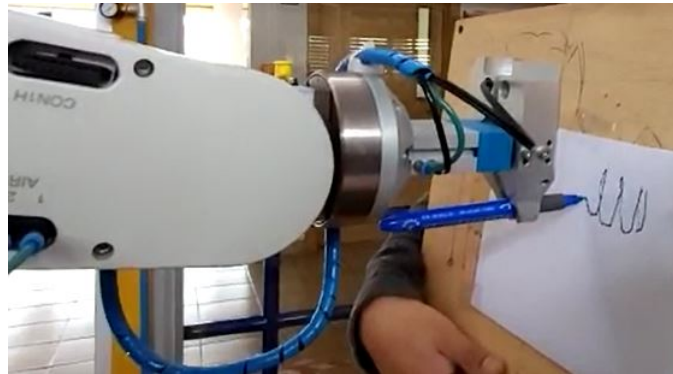


Figura 5.24: Robot Mitsubishi RV-2AJ dibujando trayectoria de soldadura (Fuente propia)

- Al igual que la prueba anterior mediante el mismo procedimiento se ha procedido a simular y aplicar una trayectoria irregular que asemeja a un movimiento de soldadura, como se observa en la Figura 5.24 el movimiento se ha implementado exitosamente.

5.10. Discusión

Al analizar los resultados en las Tablas 5.1, 5.2, 5.3 y 5.4 se pueden observar exactitudes aceptables de no mas de 2cm en posición y 5 grados en orientación despues de 30 pruebas como lo dicta la norma ISO 9283:199 revisada en la sección 5.3. Si bien una solución cerrada mediante los metodos trigonométricos daría en teoría mas exactitud, el toolbox permite que la solución de la cinemática inversa sea un método general para cualquier configuración de robot, por lo que el usuario evita el análisis trigonométrico complejo y se enfoca en la trayectoria en sí.

En la sección 5.7 el toolbox demostró ser efectivo para robots redundantes que no tienen una solución trigonométrica cerrada de mas de seis grados de libertad.

El toolbox fue desarrollado siguiendo la guía GEDIS como se observa en la sección 3.1.1 esta guía presenta una serie de parámetros enfocados al correcto diseño de la interfaz HMI, en el cual resalta la posibilidad de programar la trayectoria del robot de una manera diferente a los programas convencionales es decir cambiando las líneas de codigos con coordenadas por una interfaz que permite dibujar la trayectoria como si se tratara de un programa de CAD de esta manera en una sentencia se puede contener una trayectoria hexagonal como la que se observa en la sección 5.4, el software presenta también una correcta comunicación serial con el Robot Mitsubishi RV2AJ como se observa en la sección 5.8.

Como todo metodo computacional tiene un costo - beneficio, como se mencionó en la sección 4.2.7 a mas generaciones y mas poblaciones dentro del algoritmo genético se podría ganar en precisión sin embargo el tiempo de cómputo aumentaría.

A partir de los resultados analizados hemos obtenidos las siguientes ventajas y desventajas:

5.10.1. Ventajas y desventajas del método implementado en el toolbox con los resultados obtenidos en las pruebas funcionales

- La solución de la cinemática inversa mediante el algoritmo genético implementado en el desarrollo del toolbox funciona para cualquier configuración al contrario del análisis geométrico tradicional el cual obtiene funciones diferentes para cada modelo de robot.
- Es posible resolver trayectorias en robots de mas de seis grados de

libertad lo que no es posible con análisis geométrico tradicional ya que tenemos un máximo de seis ecuaciones e incógnitas.

- El método permite ingresar mas objetivos además de posición y orientación, en este caso a la función objetivo se agrega el mínimo cambio articular lo que hizo posible obtener trayectorias continuas y suaves como se observa en los gráficos de los resultados de posición velocidad y aceleración de las diferentes pruebas.
- El tiempo de cómputo es igual para todo tipo de configuraciones de robots debido al numero de generaciones y los criterios de parada del algoritmo son iguales, para robots de pocos grados de libertad donde la solución geométrica de la cinemática inversa es sencilla el tiempo de cómputo del algoritmo genético puede ser considerado alto.

Capítulo 6

Conclusiones y Recomendaciones

Finalmente se encuentran las conclusiones del proyecto, en el cual se ha obtenido resultados positivos y acordes a los objetivos, demostrando la efectividad de técnicas de optimización como los algoritmos genéticos para resolver trayectorias de robots manipuladores.

6.1. Conclusiones

- El toolbox desarrollado basado en algoritmos genéticos permite resolver la cinemática inversa de robots manipuladores mediante la integración de las funciones de optimización y ambientes de simulación de Matlab fueron y así interpolar trayectorias continuas como se observa en las gráficas de posición , velocidad y aceleración de las pruebas funcionales.
- Se demostró que los algoritmos genéticos pueden resolver la cinemática inversa de robots manipuladores inclusive resolver sistemas complejos y con más grados de libertad sin importar si son sistemas robóticos redundantes como se verificó en las pruebas funcionales de un robot de 7 grados de libertad obteniendo la trayectoria deseada con una exactitud de $\pm 19\text{mm}$.
- El HMI diseñado con la herramienta GUI de Matlab se ha realizado con una arquitectura jerárquica de pantallas en las cuales se han utilizado no mas de cuatro 4 colores neutros entre azules y grises, títulos con letras fáciles de visualizar y los respectivos menús de navegación que permiten el desplazamiento entre pantallas siguiendo los parámetros dictados en la guía Gedis.
- Se demostró que el software desarrollado puede comunicarse y ser utilizado en el robot Mitsubishi RV2Aj del laboratorio de la Universidad Politécnica en el cual se implementó una prueba de escritura que tuvo un tiempo de cómputo de 20 minutos para resolver una trayectoria de 84 puntos que fue reproducida correctamente por el robot y plasmada en un papel con la forma de una letra manuscrita.
- Se probó el robot mediante una serie de procedimientos diseñados para calcular su rendimiento según la norma ISO: 9283:1998, obteniendo una exactitud media de 14 mm y una repetitividad de $\pm 8.2\text{mm}$.
- Se confirmó que un algoritmo genético es una herramienta válida para resolver un problema multi-variable y complejo como es la solución de la cinemática inversa a lo largo de una trayectoria irregular.
- Los mejores resultados del algoritmo genético fueron obtenidos con una población de 120 individuos, durante 1000 generaciones, sin embargo el tiempo en el que una trayectoria es calculada depende de la cantidad de puntos que se tiene que alcanzar, el tiempo medido es de 15 segundos por punto lo cual es excelente al considerar que realiza el proceso de

cálculo una sola vez y después los datos se pueden guardar o utilizar a conveniencia del usuario.

6.2. Recomendaciones

- Es necesario identificar adecuadamente las características cinemáticas del Robot que vienen dados por los parámetros de Denavit Hartenberg que se desea analizar o programar ya que de esto depende el éxito de la solución de trayectorias.
- Si no hay conformidad con una solución de la trayectoria es necesario sintonizar parámetros dentro del algoritmo ya que la realimentación del algoritmo le permitirá encontrar una mejor solución.
- Hay que tener en cuenta los límites de los movimientos del robot real ya que toda simulación puede ser flexible sin embargo al implementar las trayectorias en robots reales hay que considerar el espacio de trabajos, obstáculos y colisiones.
- En la etapa de adquisición de trayectorias, se define varios métodos de ingreso, desde funciones matemáticas, adquisición en un entorno digital e ingreso de trayectoria predefinidas, esto es muy útil ya que facilita integrar el toolbox con cualquier método de adquisición de datos entre los cuales resultan novedosos adquirir los datos de una trayectoria mediante visión artificial o sensorización ultrasónica o sensores de movimiento. Los algoritmos genéticos se pueden combinar con otros algoritmos bioinspirados u otros métodos de inteligencia artificial como redes neuronales que pueden mejorar el rendimiento del mismo.

Bibliografía

- D. Aranda. *ELECTRÓNICA - Plataformas Arduino y Raspberry Pi: Plataformas Arduino y Raspberry Pi*. Electrónica. USERS, 2014. ISBN 9789871949564. URL <https://books.google.com.ec/books?id=QDS5DQAAQBAJ>.
- V. Autores. Morfología de un robot. Accedido en 12-04-2021 a url<http://lrobotc.blogspot.com/2015/06/morfologia-de-un-robot.html>, 2015.
- A. Barrientos, A. Cruz, L. Peñín, and C. Balaguer. *Fundamentos de robótica*. McGraw-Hill, 2007. ISBN 9788448156367. URL <https://books.google.com.ec/books?id=ArEMPAAACAAJ>.
- L. Castelló, A. Aparicio, A. Vidal, L. García, and Ó. García. *Uso de MATLAB en robótica y visión por computador*. Universidad Miguel Hernández, 2018. ISBN 9788416024537. URL <https://books.google.com.ec/books?id=-oVxDwAAQBAJ>.
- J. Craig. *Robotica*. Area ingeniería. Pearson Educación de México, 2006. ISBN 9789702607724. URL https://books.google.com.ec/books?id=hRzOp_qdxG8C.
- P. Cruz. *Inteligencia Artificial con Aplicaciones a la Ingeniería*. Marcombo, 2011. ISBN 9788426717061. URL <https://books.google.com.ec/books?id=myyQpWAACAAJ>.
- A. Hayat, R. Chittawadigi, A. Udai, and S. Saha. Identification of denavit-hartenberg parameters of an industrial robot. pages 1–6, 07 2013. doi: 10.1145/2506095.2506121.
- ISO ISO 9283:1998. MANIPULATING INDUSTRIAL ROBOTS. PERFORMANCE CRITERIA AND RELATED TEST METHODS. Standard, International Organization for Standardization, Mar. 1998.

- E. Jiménez. *Fundamentos de robótica y mecatrónica con MATLAB y Simulink*. URL <https://books.google.com.ec/books?id=oo2fDwAAQBAJ>.
- P. Karlra and N. Prakash. A neuro-genetic algorithm approach for solving the inverse kinematics of robotic manipulators. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, volume 2, pages 1979–1984 vol.2, 2003. doi: 10.1109/ICSMC.2003.1244702.
- R. Lahoz-Beltrá. *Bioinformática: simulación, vida artificial e inteligencia artificial*. Diaz de Santos, 2004. ISBN 9788479786458. URL <https://books.google.com.ec/books?id=1Lxt1Eviy8cC>.
- Z. Pan, J. Polden, N. Larkin, S. Duin, and J. Norrish. Recent progress on programming methods for industrial robots. volume 28, pages 1 – 8, 07 2010. doi: 10.1016/j.rcim.2011.08.004.
- P. Pecherková, M. Flídr, and J. Duník. *Robotics, Automation and Control*. IntechOpen, 2008. ISBN 9789537619183. URL <https://books.google.com.ec/books?id=ZQWQDwAAQBAJ>.
- S. Slamet and U. T. H. O. M. F. K. E. dan Elektronik. *Programming and 3D Simulation of Mitsubishi RV-2AJ Industrial Robot Using Cosimir Professional Software*. Universiti Tun Hussein Onn Malaysia, 2013. URL <https://books.google.com.ec/books?id=x4fpjwEACAAJ>.
- V. Teigens, P. Skalfist, and D. Mikelsten. *Inteligencia artificial: la cuarta revolución industrial*. Cambridge Stanford Books. URL <https://books.google.com.ec/books?id=sR3NDwAAQBAJ>.
- A. Zilouchian and M. Jamshidi. *Intelligent Control Systems Using Soft Computing Methodologies*. CRC Press, 2001. ISBN 9781420058147. URL <https://books.google.com.ec/books?id=SMoOEAAAQBAJ>.