



**Diseño CAD y arquitectura de control, de la simulación de un brazo robótico de aplicación de pintura electrostática para la empresa mundo lujos, capaz de ejecutar dos trayectorias de pintura por medio de los softwares SolidWorks y Simscape de MATLAB.**

**Andres Arturo Torres Guerra**

10431413473

**Sebastian Camilo Aponte Guarin**

10431411466

Universidad Antonio Nariño  
Programa Ingeniería Electromecánica  
Facultad de Ingeniería Mecánica, Electrónica y Biomédica  
Bogotá D.C., Colombia

2023

**Diseño CAD y arquitectura de control, de la simulación de un brazo robótico de aplicación de pintura electrostática para la empresa mundo lujos, capaz de ejecutar dos trayectorias de pintura por medio de los softwares SolidWorks y Simscape de MATLAB.**

**Andres Arturo Torres Guerra, Sebastian Camilo Aponte Guarin**

Proyecto de grado presentado como requisito parcial para optar al título de:

**Ingeniero Electromecánico**

Director:

Carlos Arturo García Gómez, MSc.Director

Codirector:

Christian Camilo Erazo Ordoñez, Ph.D.Codirector

Universidad Antonio Nariño

Programa Ingeniería Electromecánica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Bogotá D.C., Colombia

2023

*(Dedicatoria)*

Dedico este logro, en primer lugar, a Dios por brindarme perseverancia. También quiero agradecer a mi madre y a mi padre, quienes siempre soñaron con verme convertirme en un profesional. A lo largo de todo el proceso de aprendizaje, su apoyo incondicional ha sido fundamental.

“Los días más importantes en tu vida es el día en que naces y el día en que descubres por qué ”

Mark Twain

## Agradecimientos

Le agradezco a Dios por la fuerza y la persistencia que me brindó para culminar con esta meta. También quiero expresar mi gratitud a mi familia por creer en mí durante todo el proceso. Agradezco a los profesores por sus enseñanzas y por contribuir a forjar la persona y el ingeniero en el que me he convertido.

## Lista de figuras

Figura 1 Aplicación de pintura al interior de la carrocería de un automóvil. [1] .....	- 5 -
Figura 2. Eficiencia y calidad con brazo robot de aplicación de pintura. [1] .....	- 5 -
Figura 3. Estadística de instalación anual de robots industriales en el mundo. [2] .....	- 5 -
Figura 4. Componentes básicos de un sistema robot. ....	- 10 -
Figura 5. sistema robot cartesiano y espacio de trabajo. [4] .....	- 12 -
Figura 6. manipulador Gantry con espacio de trabajo. [4] .....	- 12 -
Figura 7. Manipulador robot cilíndrico y su espacio de trabajo. [4].....	- 13 -
Figura 8. manipulador esférico y su espacio de trabajo. [4] .....	- 13 -
Figura 9. manipulador SCARA y su espacio de trabajo. [4] .....	- 14 -
Figura 10. manipulador antropomórfico y su espacio de trabajo. [4].....	- 14 -
Figura 11. estructuras robot paralelo. [4].....	- 15 -
Figura 12. Interconexión entre los componentes de un sistema robot. [5] .....	- 15 -
Figura 13. Posición de un marco de referencia no inercial. [4] .....	- 19 -
Figura 14. Representación de un vector posición en dos marcos de referencia. [4].....	- 20 -
Figura 15. Vectores de posición de dos marcos de referencia diferentes. [4] .....	- 21 -
Figura 16. Descripción de la posición y orientación de un efector final. [4].....	- 22 -
Figura 17. Parámetros cinemáticos de Denavit-Hartenberg. [6].....	- 23 -
Figura 18. Perfil de velocidad para la ecuación de posición del efector final. [4].....	- 30 -
Figura 19. Perfil de velocidad del efector final. [4].....	- 30 -
Figura 20. Perfil de aceleración del efector final. [4] .....	- 30 -
Figura 21. La ley de control PD con precompensación se puede describir como se muestra en la ecuación (3.42). [11] .....	- 36 -
Figura 22. Control PD con precompensación. [5] .....	- 36 -
Figura 23. Control de doble integrador. [11] .....	- 37 -
Figura 24. planteamiento del desarrollo metodológico.....	- 40 -
Figura 25. espacios de trabajo para el proceso de la aplicación de pintura electrostática. ....	- 41 -
figura 26. Enlace 1 (Base) [fuente propia].....	- 42 -
figura 27. Enlace 2 [fuente propia] .....	- 43 -
figura 28. Enlace 3 [fuente propia] .....	- 43 -
figura 29. Enlace 4 [fuente propia] .....	- 43 -
figura 30. enlace 5 [fuente propia].....	- 44 -
figura 31. Ensamble final del brazo robot. [fuente propia].....	- 44 -
Figura 32. disposición inicial para iniciar la trayectoria. [fuente propia].....	- 45 -
Figura 33. definición de L5 usando vista superior. [fuente propia].....	- 45 -
Figura 34. vector resultante para cálculo de L2 y L3. [fuente propia].....	- 45 -
Figura 35. disposición del robot dentro de la cabina .....	- 46 -
Figura 36. (Diagrama de bloques del sistema dinámico brazo robot antropomorfo en Simulink). [fuente propia].....	- 47 -
Figura 37. sistemas de referencia desincronizados. [fuente propia] .....	- 48 -
Figura 38. marcos de referencia sincronizados con parámetros D-H. [fuente propia] .....	- 48 -

Figura 39. parámetros D-H robot antropomorfo 5 DOF..... - 49 -

Figura 40. función D-H para modelo cinemático del robot [fuente propia] ..... - 49 -

Figura 41. posición y orientación inicial del robot [fuente propia] ..... - 49 -

Figura 42. matriz de transformación homogénea con función D-H en MATLAB [fuente propia] - 50 -

Figura 43. Cinemática directa del robot antropomorfo [fuente propia]..... - 50 -

Figura 44. Triangulo de la posición inicial de enlace 1 y 5 [fuente propia] ..... - 51 -

Figura 45. Posición inicial para cinemática inversa [fuente propia]..... - 52 -

Figura 46. cinemática inversa en MATLAB y función de error de posición ..... - 53 -

Figura 47. operación del Jacobiano por medio de derivadas parciales [fuente propia] ..... - 54 -

Figura 48. Posicionamiento del brazo para realización de la primera trayectoria de aplicación de pintura. .... - 55 -

Figura 49. Posiciones, velocidades y aceleraciones de cada articulación, para la ubicación a pinar del brazo..... - 56 -

Figura 50. planeación 1 para la articulación 1 [fuente propia] ..... - 56 -

Figura 51. planeación 1 para la articulación 2 [fuente propia] ..... - 57 -

Figura 52. planeación 1 para la articulación 3 [fuente propia] ..... - 57 -

Figura 53. planeación 1 para la articulación 4 [fuente propia] ..... - 57 -

Figura 54. planeación 1 para la articulación 5 [fuente propia] ..... - 57 -

Figura 55 Posición inicial, intermedia y final, segunda trayectoria [fuente propia]..... - 58 -

Figura 56 Graficas de posición, velocidad y aceleración de cada articulación para la figura 55. [fuente propia]..... - 58 -

Figura 57. planeación 2 para la articulación 1 [fuente propia] ..... - 59 -

Figura 58. planeación 2 para la articulación 2 [fuente propia] ..... - 59 -

Figura 59. planeación 2 para la articulación 3 [fuente propia] ..... - 59 -

Figura 60. planeación 2 para la articulación 4 [fuente propia] ..... - 59 -

Figura 61. planeación 2 para la articulación 5 [fuente propia] ..... - 60 -

Figura 62. Posición inicial y final de tercera trayectoria [fuente propia] ..... - 60 -

Figura 63. Graficas de posición, velocidad y aceleración de cada articulación en la tercera trayectoria. [fuente propia]..... - 61 -

Figura 64. planeación 3 para la articulación 1 [fuente propia] ..... - 61 -

Figura 65. planeación 3 para la articulación 2 [fuente propia] ..... - 61 -

Figura 66. planeación 3 para la articulación 3 [fuente propia] ..... - 62 -

Figura 67. planeación 3 para la articulación 4 [fuente propia] ..... - 62 -

Figura 68. planeación 3 para la articulación 5 [fuente propia] ..... - 62 -

Figura 69. cuarta planeación, segunda aplicación de pintura. .... - 63 -

Figura 70. planeación 4 articulación 1 [fuente propia] ..... - 63 -

Figura 71. planeación 4 para la articulación 2 [fuente propia] ..... - 63 -

Figura 72. planeación 4 para la articulación 3 [fuente propia] ..... - 63 -

Figura 73. planeación 4 para la articulación 4 [fuente propia] ..... - 64 -

Figura 74. planeación 4 para la articulación 5 [fuente propia] ..... - 64 -

Figura 75. Posiciones, velocidades y aceleraciones de las articulaciones 1,2 y 3 [fuente propia] . - 65 -

Figura 76. Posiciones, velocidades y aceleraciones de las articulaciones 4 y 5 [fuente propia]- 65 -

Figura 77. masas de enlaces y constante de gravedad [fuente propia] .....- 66 -

Figura 78. distancias a centros de masas de cada enlace [fuente propia] .....- 66 -

Figura 79. tensores de inercia [fuente propia] .....- 66 -

figura 80. operación de la energía cinética del manipulador con la ecuación (3.35) [fuente propia] .....- 66 -

Figura 81. operación de la energía potencial del manipulador con la ecuación (3.36) [fuente propia] .....- 66 -

Figura 82. ecuación lagrangiana [fuente propia] .....- 67 -

Figura 83. derivada del lagrangeano respecto a posiciones angulares [fuente propia].....- 67 -

Figura 84. derivada del lagrangeano respecto a velocidades angulares [fuente propia].....- 67 -

Figura 85. derivadas respecto al tiempo [fuente propia].....- 67 -

Figura 86. ecuación de movimiento para cada articulación [fuente propia].....- 67 -

Figura 87. Diagrama de control PD con precompensación desacoplado y unificado [fuente propia] .....- 68 -

Figura 88. subsistema del brazo robot con entradas y salidas. [fuente propia] .....- 69 -

Figura 89. modelo robot exportado de solidworks .....- 70 -

Figura 90. función de torques para la acción de precompensación [fuente propia] .....- 70 -

Figura 91. Error posición deseada en negro y posición real controlada de azul articulación 1 [fuente propia].....- 71 -

Figura 92. Error de posición angular en función del tiempo, articulación 1. [fuente propia]..- 71 -

Figura 93. error de posición en la articulación 2. [fuente propia].....- 72 -

Figura 94 señal de error de posición angular vs tiempo, corregida con nuevas ganancias  $K_p=10000$  y  $K_v=200$ . [fuente propia].....- 72 -

Figura 95. señal de error de posición en la articulación 3. [fuente propia].....- 73 -

Figura 96. señal de error de posición para la articulación 3 con nuevas ganancias  $K_p$  y  $K_v$ . [fuente propia].....- 73 -

Figura 97. Error de posición para la articulación 4. [fuente propia].....- 73 -

Figura 98. Grafica de error de posición vs tiempo. Sintonización para la articulación 4. [fuente propia] .....- 74 -

Figura 99. grafica de error de posición en la articulación 5. [fuente propia].....- 74 -

1 Contenido

**Resumen** ..... - 1 -

**Abstrac**..... - 2 -

**Palabras clave:**..... - 3 -

**Keywords:** ..... - 3 -

**Introducción** ..... - 4 -

    1.1 Planteamiento del problema ..... - 7 -

    1.2 Justificación ..... - 8 -

**2 Objetivos**..... - 9 -

    2.1.1 Objetivo general..... - 9 -

    2.1.2 Objetivos específicos..... - 9 -

**3 Marco teórico**..... - 10 -

    3.1 Preliminares ante datos teóricos ..... - 10 -

        3.1.1 sistema robot cartesiano ..... - 11 -

        3.1.2 manipulador Gantry ..... - 12 -

        3.1.3 Manipulador cilíndrico o estructura robot cilíndrica ..... - 12 -

        3.1.4 Manipulador esférico ..... - 13 -

        3.1.5 manipulador robot SCARA ..... - 13 -

        3.1.6 manipulador antropomorfo ..... - 14 -

        3.1.7 manipulador paralelo..... - 14 -

    3.2 Diseño CAD..... - 17 -

    3.3 Modelamiento cinemático de brazos antropomorfos..... - 18 -

        3.3.1 Posición de un cuerpo rígido..... - 18 -

        3.3.2 Matrices de rotación ..... - 19 -

        3.3.3 Vectores de posición de dos marcos de referencia ..... - 20 -

        3.3.4 Composición de matrices de rotación ..... - 21 -

        3.3.5 Matriz de transformación homogénea ..... - 21 -

        3.3.6 Cinemática directa ..... - 22 -

        3.3.7 Convención Denavit-Hartenberg ..... - 23 -

        3.3.8 Cinemática inversa ..... - 24 -

        3.3.9 Cinemática diferencial (Jacobiano) ..... - 25 -

    3.4 Planeación de trayectorias..... - 28 -

        3.4.1 planificación de trayectoria punto a punto ..... - 28 -



3.5	Dinámica .....	- 31 -
3.5.1	Método de Lagrange .....	- 32 -
3.5.2	Operación de la energía cinética.....	- 32 -
3.5.3	Operación de la energía potencial .....	- 33 -
3.5.4	Ecuación de movimiento.....	- 33 -
3.6	Control de articulaciones en sistemas robots.....	- 35 -
3.6.1	control PD con precompensación o alimentación anticipada .....	- 35 -
<b>4</b>	<b>Desarrollo metodológico.....</b>	<b>- 39 -</b>
4.1	Definición de los parámetros de diseño .....	- 41 -
4.2	Diseño del brazo antropomorfo.....	- 42 -
4.3	Exportación de diseño mecánico a Simulink de MATLAB .....	- 47 -
4.4	Modelamiento cinemático directo e inverso.....	- 49 -
4.5	Cinemática directa .....	- 50 -
4.6	Cinemática inversa .....	- 51 -
4.7	Cinemática diferencial (Jacobiano) .....	- 54 -
4.8	Planeación de trayectorias.....	- 55 -
4.9	Modelamiento dinámico de robot antropomorfo.....	- 66 -
4.10	Acción de control PD con alimentación anticipada .....	- 68 -
<b>5</b>	<b>Análisis de resultados .....</b>	<b>- 75 -</b>
<b>6</b>	<b>Conclusiones .....</b>	<b>- 76 -</b>
<b>7</b>	<b>Referencias .....</b>	<b>- 77 -</b>

## **Resumen**

El presente proyecto es un esfuerzo para modelar y simular un brazo robot de 5DOF (grados de libertad) en un entorno de aplicación de pintura industrial en la empresa mundo lujos LTDA, con el objetivo de proporcionar una idea de cómo se comportaría un diseño real de un brazo robot diseñado desde cero. Esto puede ser útil para evaluar la viabilidad y eficacia de un brazo robot en aplicaciones industriales de pintura antes de construir un prototipo físico.

Se desarrolló el proyecto a partir de una visita a la empresa donde se evaluó y dimensionó el área de la cabina y las posiciones posibles para el brazo robot, luego, utilizando SolidWorks, se diseñó un brazo robot antropomorfo basado en las mediciones y requisitos recopilados. Para determinar la cinemática del robot se implementó la metodología Denavit-Hartenberg para obtener la cinemática directa del brazo, lo que proporcionó las posiciones del efector final en coordenadas rectangulares con respecto a la base. Se utilizó la metodología de punto a punto con splines cúbicos y cinemática inversa para calcular las posiciones angulares de las articulaciones a partir de las posiciones inicial y final del efector final en coordenadas rectangulares, para llevar a cabo la planeación de dos trayectorias de aplicación de pintura. Con lo que finalmente en MATLAB por medio de la biblioteca Simscape se realizó una simulación de un control PD con acción de pre-compensación para el seguimiento de las trayectorias, en donde la acción de pre-compensación sugiere que se están tomando medidas para anticipar y corregir los errores antes de que ocurran por medio del modelo dinámico del brazo robot.

## **Abstrac**

The present project is an effort to model and simulate a 5DOF (degrees of freedom) robot arm in an industrial paint application environment at Mundo Lujos LTDA, with the objective of providing an idea of how a real design of a robot arm designed from scratch would behave. This can be useful for evaluating the feasibility and effectiveness of a robot arm in industrial painting applications prior to building a physical prototype.

The project was developed from a visit to the company where the area of the cabin and the possible positions for the robot arm were evaluated and dimensioned, then, using SolidWorks, an anthropomorphic robot arm was designed based on the measurements and requirements collected. To determine the kinematics of the robot, the Denavit-Hartenberg methodology was implemented to obtain the direct kinematics of the arm, which provided the end-effector positions in rectangular coordinates with respect to the base. The point-to-point methodology with cubic splines and inverse kinematics was used to calculate the angular positions of the joints from the initial and final positions of the end effector in rectangular coordinates, to carry out the planning of two paint application trajectories. Finally, a simulation of a PD control with pre-compensation action for trajectory tracking was performed in MATLAB using the Simscape library, where the "pre-compensation action" suggests that measures are being taken to anticipate and correct errors before they occur by means of the dynamic model of the robot arm.

**Palabras clave:** DOF (degrees of freedom), parámetros D-H, cinemática inversa, Jacobiano, Lagrangeano, planeación, trayectoria, robot, morfología, control PD, modelo dinámico.

**Keywords:** DOF (degrees of freedom), D-H parameters, inverse kinematics, Jacobian, Lagrangean, planning, trajectory, robot, morphology, PID control.

## Introducción

El desarrollo de simulaciones ha sido crucial tanto en el ámbito industrial como académico, especialmente en el campo de la robótica. El software de simulación permite visualizar presentaciones en tiempo real, ofreciendo ventajas significativas al proporcionar un sistema virtual que se asemeja de cerca a la realidad. En el contexto industrial, se destaca la importancia de la programación de robots para simulación offline, donde se realizan modelamientos mecánicos para llevar a cabo simulaciones. Esto permite analizar aspectos como ajustes finos, colisiones, prevención de posibles accidentes, detección de coordenadas fuera del espacio de trabajo, y la obtención de información valiosa, como tiempos de ciclo. Además, las simulaciones facilitan la extracción de medidas fundamentales del robot, como masas de los enlaces e inercias, brindando una comprensión detallada del rendimiento del sistema. [1] [2]

A nivel mundial, los softwares utilizados en control y simulación de estaciones robóticas son MATLAB, ROS (Robot Operating System), Gazebo, RoboDK, Cosimir/Robotica, Webots, KUKA.Sim, Fanuc Roboguide, Miranda – tdrobotica, Almacam Weld, CoppeliaSim, Free CAD. Que son entornos virtuales para la simulación de celdas robóticas offline. [3]

La implementación de brazos robóticos ha venido en auge desde hace más de dos décadas, debido a la necesidad de obtener mayor eficiencia en los procesos industriales de manufactura, con el fin de agilizar y darle mayor precisión a los procesos de producción que requieran movimiento de piezas pesadas, exposición a gases tóxicos, actividades repetitivas, en general actividades que pongan en peligro la integridad de las personas. Los países que lideran la implementación de brazos en sus procesos productivos son China, Alemania, Japón, Corea, en menor cantidad en EE.UU. Esto implica que las compañías se pongan a la vanguardia ante la automatización de procesos industriales como industria automotriz, metalmecánica, plásticos, productos químicos, comida, entre otros, algunos ejemplos se pueden evidenciar en las figuras 1 y 2 para procesos de aplicación de pintura automatizados con brazos robot. [1] [2]

Los principales fabricantes de robots en el mundo son ABB, KUKA, FANUC, YASKAWA, STAUBLI, WITTMANN, GUEDEL, YUANDA, KAWASAKI, ESTUN ROBOTICS, UNIVERSAL ROBOTS, AISONG, PANASONIC, DAIHEN, COMAU, OMRON, NACHI, entre otros. [3] [4]



Figura 1 Aplicación de pintura al interior de la carrocería de un automóvil. [1]



Figura 2. Disposición de brazos robot de aplicación de pintura. [1]

Según la International Federation of Robotics la instalación de robot en el mundo viene con un aumento por año del 11% desde el 2016 al 2021 como se muestra en la estadística realizada en la figura 3. [3]

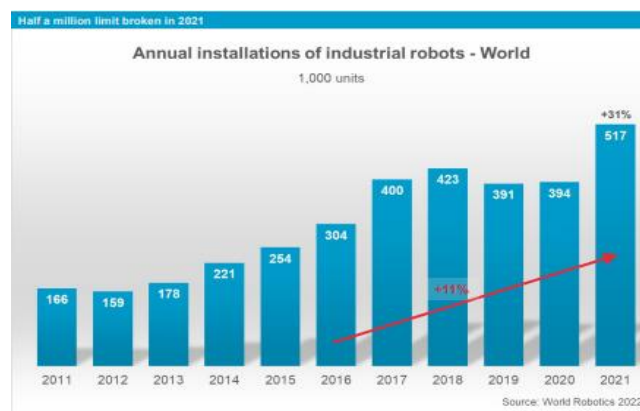


Figura 3. Estadística de instalación anual de robots industriales en el mundo. [2]

Las universidades, al formar semilleros y ofrecer programas en disciplinas relacionadas con la robótica, han contribuido significativamente al conocimiento y al talento en este campo. La investigación y las tesis de grado proporcionan una base sólida para la comprensión de los conceptos y la aplicación de la robótica en diferentes contextos.

La presencia de brazos robóticos en grandes multinacionales como General Motors, Bavaria, Bimbo y Corona destaca la relevancia y la aceptación de esta tecnología en la industria colombiana. La adaptación de brazos robóticos en los procesos productivos no solo mejora la eficiencia, sino que también impulsa la innovación en la fabricación y la automatización.

Empresas pioneras en Colombia con implementación de robots como Eurobots y Robotika S.A.S. demuestra cómo el conocimiento generado en el ámbito académico se traduce en soluciones prácticas para la industria. Estas empresas desempeñan un papel crucial al ofrecer soluciones de ingeniería y al difundir el conocimiento sobre el uso y la aplicación real de la robótica industrial en Colombia.[3]

### **1.1 Planteamiento del problema**

La empresa Mundo Lujos Ltda. es una empresa que empezó realizando la fabricación de piezas pequeñas como, grapas, fundiciones en aluminio, diseño de moldes para las fundiciones. El proceso de pintura en la empresa se inició con aplicación de pintura líquida con la que se encontraron problemas de calidad, desperdicios de pintura y costos de aplicación mucho más elevados que otros procesos. Con lo que al transcurrir del tiempo y el crecimiento de producción tomaron la decisión de mejorar el proceso pasando de la pintura líquida a la pintura en polvo aplicada electrostáticamente. La cual tenía menores costos de aplicación y presentaba una mejor calidad en las piezas eso sin que el operario tuviese gran pericia al momento de la aplicación La pintura electrostática en la empresa presenta problemas de polución que se traduce en pérdida de pintura y suciedad de los alrededores por el proceso, la solución a nivel industrial más conocida es la de automatizar estos procesos con robots teniendo entornos de trabajo más herméticos reduciendo las pérdidas de pintura por contaminación cruzada y la polución que se genera alrededor de otros procesos.

A la empresa mundo lujos se propuso hacer un diseño CAD en conjunto con el modelamiento y simulación del robot en Matlab con un control en Simulink esto para tener una idea del robot que podrían implementar según el espacio de trabajo para llevar a cabo las tareas que se necesitarían programar.



## **1.2 Justificación**

Este proyecto presenta una simulación de un sistema mecánico y cinemático para aplicar pintura electrostática a un tubo utilizando un modelo CAD para con la simulación obtener datos importantes de las trayectorias.

Nuestro proyecto quiere darle a la empresa Mundo lujos, la visión de cómo se puede ver una estación automatizada de un robot aplicando las trayectorias programadas para aplicar pintura electrostática a un tubo, obteniendo los datos de velocidades, grados de libertad y desplazamiento de cada enlace, tiempos de respuesta y torques necesarios para llevar a cabo las tareas programadas.

Con lo que ya para un futuro tendrían la idea de que robot les podría servir para una posible implementación y que características de velocidades y torques aproximados necesitarían para llevar a cabo una implementación satisfactoria.

## 2 Objetivos

### *2.1.1 Objetivo general*

- ✓ Diseñar y simular en 3D el sistema de control de un brazo robótico para la aplicación de pintura electrostática, que será capaz de llevar a cabo dos trayectorias de pintura.

### *2.1.2 Objetivos específicos*

- ✓ Realizar el diseño CAD del brazo robótico, con el fin de obtener el modelamiento de la cinemática del robot teniendo en cuenta las limitaciones y condiciones de trabajo.
- ✓ Migrar el modelamiento de la cinemática del brazo robótico a la herramienta Simscape de MATLAB, para realizar la planeación de las trayectorias.
- ✓ Desarrollar la simulación 3D y el algoritmo de control de planeación de las dos trayectorias en el software MATLAB.
- ✓ Evaluar el tiempo de respuesta del sistema en función de la acción de control.

### 3 Marco teórico

#### 3.1 Preliminares ante datos teóricos

La fabricación de los robots básicamente se basó en el desarrollo de máquinas que pudieran sustituir la actividad realizada por una persona, así de esta forma se diseñaron distintas formas, estructuras y morfologías que fueron adoptadas por ingenieros de automatización, mecatrónicos, mecánicos y electromecánicos entre otros, para llevar a cabo estas tareas. Sin embargo, toda morfología desarrollada tecnológicamente está sujeta a unos componentes básicos que todo robot debería tener como mínimo, tales componentes se muestran en la siguiente figura 4. [4]

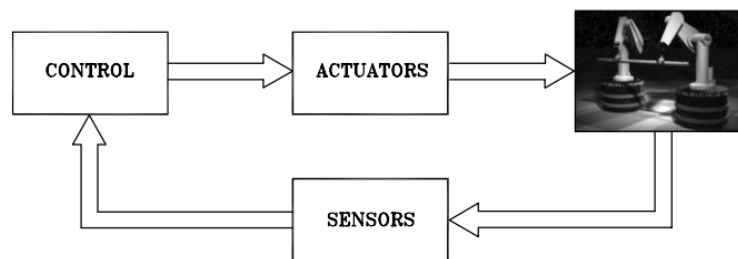


Figura 4. Componentes básicos de un sistema robot.

Los sensores, estos dispositivos miden magnitudes físicas como posición, velocidad, fuerza, o cualquier otro parámetro relevante para el control del brazo robot, proporcionan información al sistema sobre el entorno y el estado del robot, permitiendo ajustes y correcciones en tiempo real. Los actuadores, Son los componentes que ejecutan las acciones físicas requeridas por el sistema, en el caso del brazo robot, los actuadores serían responsables de mover las articulaciones y herramientas para realizar las tareas asignadas. El controlador, Actúa como el "cerebro" del sistema, utilizando algoritmos parametrizados y la retroalimentación de los sensores para tomar decisiones, basado en esa información, emite órdenes a los actuadores para lograr y mantener el estado deseado del brazo robot.

Las estructuras mecánicas de los sistemas robots se basan en la unión de cuerpos rígidos, una serie de (enlaces) mediante nodos llamados (articulaciones), los cuales pueden conformar una cadena cinemática abierta o una cadena cinemática cerrada, se dice que es una cadena abierta cuándo existe una sola posible unión de los enlaces, es decir, que cada articulación bien sea prismática o de revolución, otorga a la estructura un único grado de libertad (DOF) y una cadena cinemática cerrada en robótica se refiere a un mecanismo o sistema de enlace donde la conexión

entre sus elementos forma un bucle cerrado. En otras palabras, la salida de un eslabón vuelve a conectarse al eslabón de entrada, creando una estructura de bucle.

Desde un punto de vista topológico, las articulaciones que son las uniones entre los enlaces pueden suplir de movimientos traslacionales, si se trata de una articulación prismática o movimientos rotacionales si son de revolución. Teniendo en cuenta esto, el número de grados de libertad (DOF) debe ser proporcionalmente distribuido por toda la cadena cinemática, de tal manera que vaya cumpliendo con la ejecución de una tarea en específica, con el fin de obtener un número óptimo de articulaciones.

Si lo que se desea es ubicar un objeto en un espacio 3D se requieren como mínimo 6 grados de libertad, 3 para posicionar el elemento y 3 para orientarlo con respecto a un marco de referencia. Si hay más grados de libertad que variables de tarea, se considera que la estructura robot es redundante. La presencia de más grados de libertad puede brindar ventajas y desafíos en la funcionalidad y el control de un sistema robótico.

La clasificación se basa en cómo estos robots se mueven y cuántos grados de libertad tienen. Cada tipo se adapta mejor a diferentes tareas y entornos de trabajo específicos. La elección del robot más adecuado depende de las necesidades de la tarea a realizar, la precisión requerida y las limitaciones del entorno de trabajo.

### ***3.1.1 sistema robot cartesiano***

Está compuesto por tres articulaciones prismáticas mutuamente ortogonales entre ellas, una estructura mecánica básica de un robot cartesiano se puede ver en la figura 5. Ejemplos comunes incluyen sistemas de posicionamiento en líneas de ensamblaje, máquinas de corte o grabado y sistemas de manipulación que requieren movimiento preciso en varias direcciones.

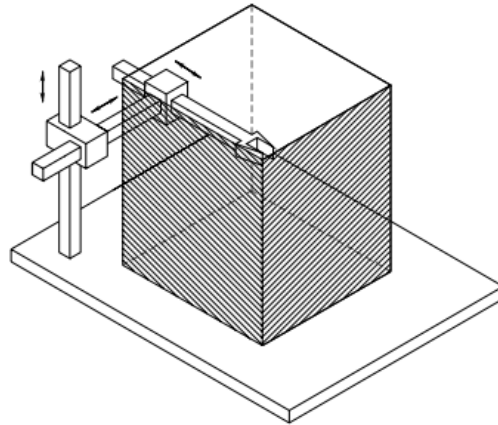


Figura 5. sistema robot cartesiano y espacio de trabajo. [4]

### 3.1.2 manipulador Gantry

Es básicamente un robot cartesiano compuesto de cuatro articulaciones prismáticas montadas sobre un pórtico y son altamente implementados en la industria para el transporte de objetos pesados y ensamble de piezas.

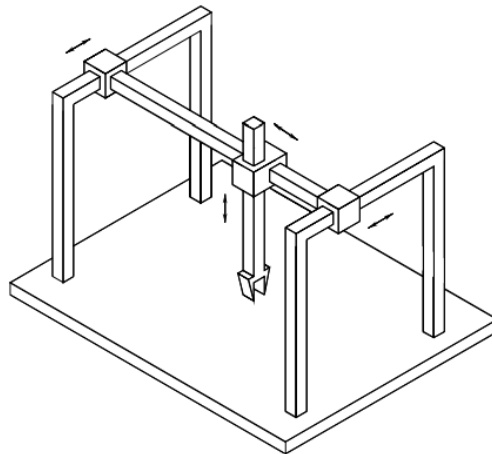


Figura 6. manipulador Gantry con espacio de trabajo. [4]

### 3.1.3 Manipulador cilíndrico o estructura robot cilíndrica.

En este caso esta estructura porta dos articulaciones prismáticas y una articulación de revolución. La porción de espacio para este manipulador es de una tubería cilíndrica.

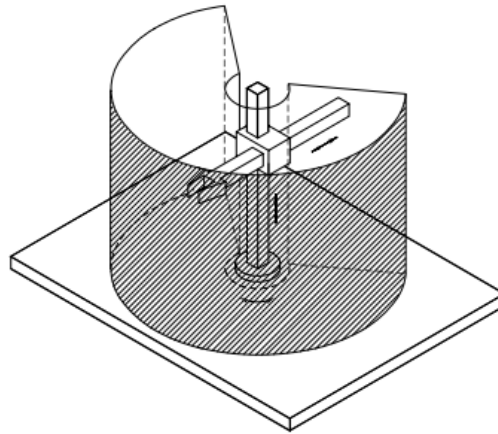


Figura 7. Manipulador robot cilíndrico y su espacio de trabajo. [4]

### 3.1.4 Manipulador esférico.

Este manipulador robot está compuesto por 3 articulaciones en donde todas son de revolución. Teniendo un espacio de trabajo alcanzable dentro de una esfera. Este tipo de manipulador con tres articulaciones de revolución es comúnmente utilizado en entornos industriales para tareas de montaje, inspección y manipulación que requieren movimientos precisos y controlados dentro de un espacio de trabajo esférico.

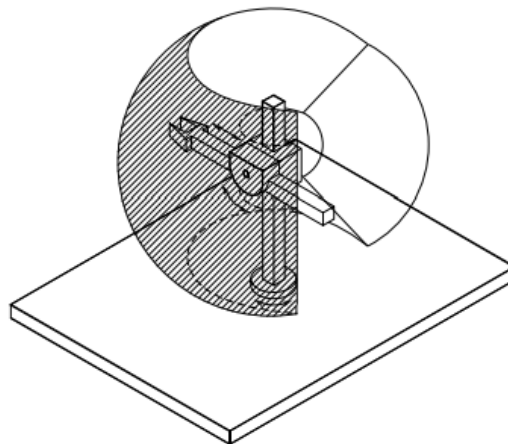


Figura 8. manipulador esférico y su espacio de trabajo. [4]

### 3.1.5 manipulador robot SCARA

De la integración de sus siglas stand for selective compliance assembly robot arm (Brazo Robot de Montaje de Cumplimiento Selectivo), típicamente implementado en tareas de izaje de cargas para realizar movimientos verticales y horizontales, puede ser compuesto por dos articulaciones de revolución y una prismática.

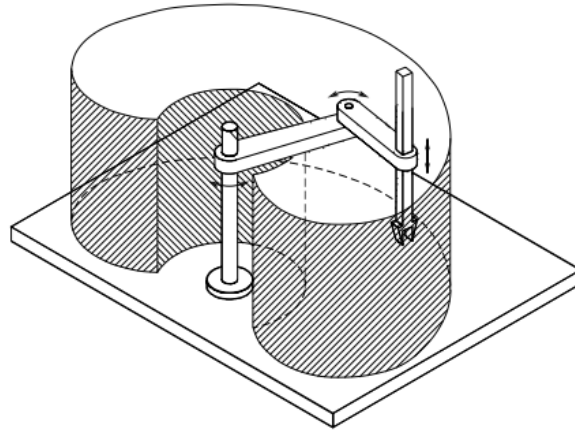


Figura 9. manipulador SCARA y su espacio de trabajo. [4]

### 3.1.6 manipulador antropomorfo

La estructura antropomórfica se compone de todas sus articulaciones de revolución, se asemeja mucho a la morfología de un brazo humano, de ahí su nombre. En muchas veces es considerado un robot redundante, pero con ventajas en la flexibilidad para llevar a cabo trayectorias de mayor complejidad.

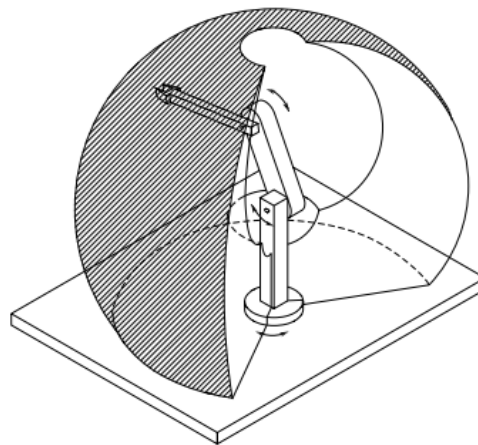


Figura 10. manipulador antropomórfico y su espacio de trabajo. [4]

### 3.1.7 manipulador paralelo

Conformados por cadenas cinemáticas cerradas, tienen una compleja y rígida estructura, implementados en tareas de alto esfuerzo y manipulación de cargas que exijan alta fuerza. La estructura rígida y cerrada de estas cadenas cinemáticas, aunque puede limitar su movilidad en comparación con otros tipos de robots, les confiere la capacidad de soportar cargas pesadas sin

sacrificar la estabilidad o la precisión. Esto los hace valiosos en entornos donde se requiere fuerza y resistencia para realizar tareas específicas.

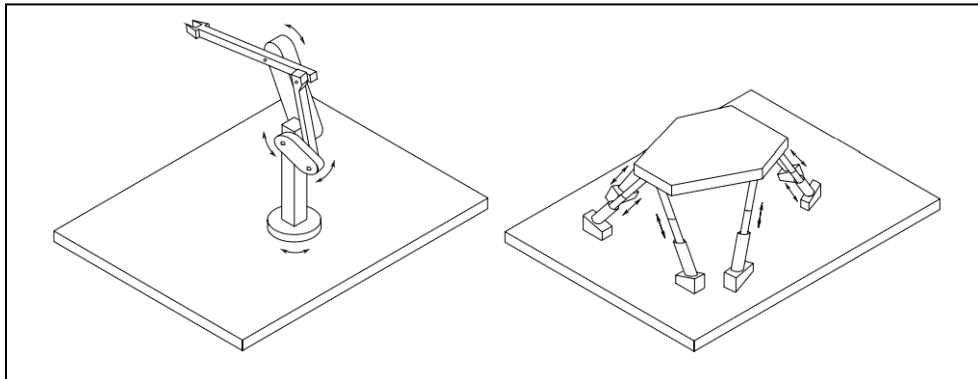


Figura 11. estructuras robot paralelo. [4]

Una vez definida la morfología del robot se debe tener claro los elementos mínimos que debe de contener un sistema de un manipulador robot, así, se tendrá una fuente de energía eléctrica y mecánica (accionadores), un procesador para comunicar y controlar el sistema, sensores de posición angular, algoritmo de planeación de tareas, herramienta de efector final, la figura 12, demuestra la interacción de cada una de las etapas entre sí del sistema del manipulador robot.

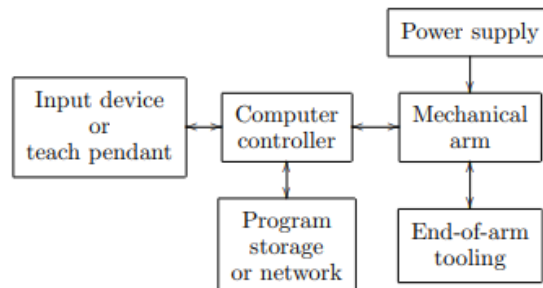


Figura 12. Interconexión entre los componentes de un sistema robot. [5]

La fuente de energía eléctrica o mecánica que provee la energía necesaria para el movimiento. Los accionadores pueden ser motores eléctricos, hidráulicos o neumáticos que generan la potencia para desplazar las articulaciones y ejecutar las tareas, procesador para la comunicación y el control, el procesador es el cerebro del sistema. Se encarga de procesar la información de los sensores, ejecutar algoritmos de control y coordinar el movimiento del robot, los sensores, son esenciales para determinar la posición y orientación de las articulaciones.



Proporcionan retroalimentación al sistema de control para garantizar que el robot se encuentre en la posición correcta, la planeación de trayectorias, es el algoritmo que determina cómo se deben realizar las tareas. Basado en las entradas del usuario o del entorno, planifica la secuencia de movimientos del robot para completar la tarea de manera eficiente, el efector final o herramienta, es la parte del robot que interactúa directamente con el entorno, como una pinza, un brazo, una herramienta de soldadura, etc. Es responsable de ejecutar la tarea designada.

Estos aspectos son todos necesarios para la simulación e incluso la fabricación y puesta en marcha de una estación automatizada con un brazo robot industrial.

### 3.2 Diseño CAD

El diseño asistido por computadora CAD es un programa que permite realizar diseños mecánicos a partir de la concepción de una idea, SolidWorks es una solución de diseño en 3D ya que esta integra un gran número de herramientas que facilitan el modelamiento de diferentes piezas, crear ensambles y gestionar planos para la posterior construcción de estas. Además de esto se tiene un entorno intuitivo en el cual tenemos más de 45 aplicaciones complementarias las cuales nos ayudan a facilitar el desarrollo de proyectos.

En SolidWorks la creación los sólidos en 3D se realizan por medio de planos o croquis a lo que se les aplican operaciones de extrusión, revolución, redondeo, corte etc. Las cuales se van a ir visualizando por orden en un árbol de operaciones.

Las operaciones de croquizado se basan en las medidas generadas para crear dicho croquis el que generalmente se convierte en un sólido a través de extrusiones, barridos, rotaciones o recubrimientos.

- ✓ La extrusión base es una operación que permite darle una altura a una geometría cerrada conformando un sólido base para iniciar la conformación de una pieza
- ✓ La revolución de base es una operación que se realiza por medio de una geometría cerrada que gira alrededor de un eje de referencia esto para conformar un sólido como bujes o ejes
- ✓ La herramienta de extrusión y corte es utiliza para poner o quitar material de un sólido en la que podemos especificar la profundidad y la dirección de acuerdo al croquis diseñado.
- ✓ Las operaciones de chaflán, redondeo y vaciado son las operaciones que modifican directamente el sólido sin depender de un croquis.

La herramienta toolbox nos permite importar componentes mecánicos estandarizados como lo son las tuercas, pernos, rodamientos, engranajes y anillos seeger etc. Las dimensiones de estos componentes están en cumplimiento en normas como ISO, DIN, ANSI.

El módulo de ensamble es una herramienta para relacionar las piezas del diseño y permite la exportación de piezas de las bibliotecas y piezas creadas, para con estas conformar las relaciones de posición entre ellas con operaciones básicas y avanzadas como son las relaciones concéntricas, coincidencia, tangencial, simétrica y centro de perfil, las restricciones de movimiento o grados de libertad máximo se tomaron teniendo en cuenta la estructura y sus posibles colisiones.

### **3.3 Modelamiento cinemático de brazos antropomorfos**

La realización del modelamiento de un brazo robótico de  $n$  grados de libertad se basa en la configuración de las articulaciones, es decir, como se clasifican, si son articulaciones prismáticas o de revolución. Se necesita definir la cinemática de análisis del mismo, teniendo en cuenta el efector final como punto de análisis, con el fin de modelar la posición, la orientación y la localización del efector final de un brazo robot.

Este modelado cinemático es esencial para planificar y controlar los movimientos del brazo robot, permitiendo que se realicen tareas específicas de manera precisa y eficiente. Además, es la base para desarrollar algoritmos de control y planificación de trayectorias que guían al robot en la ejecución de las tareas deseadas.

Para los brazos robot se cuenta con herramientas para modelar matemáticamente el sistema, como la cinemática directa, que se centra en determinar la posición y orientación del efector final a partir de las variables de las articulaciones mediante los parámetros D-H y la cinemática inversa. Si lo que se desea es dar solución al problema de la cinemática directa por medio de la definición de los parámetros de Denavit-Hartenberg se hace uso de la transformación homogénea, los cuales son unos parámetros especiales que definirán el robot. Así, de este modo se expresan los mínimos argumentos y conocimientos que deben tenerse en cuenta para la realización de un modelo cinemático en robótica.

Por otro lado, la cinemática inversa es el proceso que resuelve el problema opuesto, dada una posición y orientación deseada para el efector final, se calcula las configuraciones de las articulaciones necesarias para alcanzar esa posición. [4]

#### ***3.3.1 Posición de un cuerpo rígido***

Para definir la posición de un marco de referencia no inercial, como el marco  $O'$ , en un sistema robótico, se relaciona cada componente (posición y orientación) con respecto a un marco inercial situado en el origen del brazo del robot. La posición vista desde un marco de referencia  $O'$  que se ilustra en la figura 13 es:

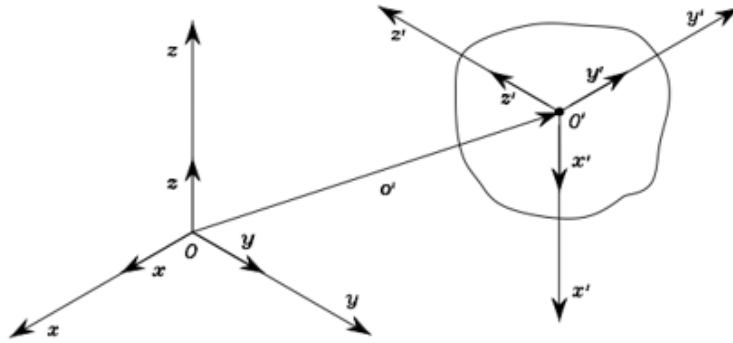


Figura 13. Posición de un marco de referencia no inercial. [4]

La representación vectorial de la posición del marco de referencia no inercial con respecto al origen en componentes rectangulares es:

$$O' = \begin{bmatrix} O'_x \\ O'_y \\ O'_z \end{bmatrix} \quad 3.1$$

De la misma forma se relaciona cada componente con el marco de referencia inercial situado en el origen O, con el fin de mostrar su dirección y orientación en la ecuación 3.2 a 3.4:

$$X' = X'_x \mathbf{X} + X'_y \mathbf{Y} + X'_z \mathbf{Z} \quad 3.2$$

$$Y' = Y'_x \mathbf{X} + Y'_y \mathbf{Y} + Y'_z \mathbf{Z} \quad 3.3$$

$$Z' = Z'_x \mathbf{X} + Z'_y \mathbf{Y} + Z'_z \mathbf{Z} \quad 3.4$$

### 3.3.2 Matrices de rotación

La representación de la orientación de un cuerpo rígido está dada por la ecuación (3.5) a (3.7), la forma matricial que denota la rotación mediante una matriz de (3x3), se expresa como los senos y cosenos de los ángulos de rotación de un vector de posición con respecto a un eje coordenado. Donde cada componente tiene un vector característico dependiendo la referencia de rotación. [4]

Estas matrices de rotación son fundamentales en la cinemática y la descripción espacial de cuerpos rígidos, permitiendo representar y entender cómo se mueven y se orientan en el espacio tridimensional, ofreciendo una base matemática para la representación de la orientación y la rotación en sistemas físicos.

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 3.5$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad 3.6$$

$$\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad 3.7$$

De la misma forma en como se demuestra la posición de un cuerpo rígido en (3.1), se puede determinar por medio de trigonometría, la representación de un vector en el punto P visto desde dos marcos de referencia que comparten el origen, con el fin de relacionar las posiciones entre los marcos y una componente rotacional aplicada a uno de ellos, el cual no es inercial, representa la transformación de un marco de referencia a otro se puede obtener a partir de evaluar la figura 14:

### 3.3.3 Vectores de posición de dos marcos de referencia

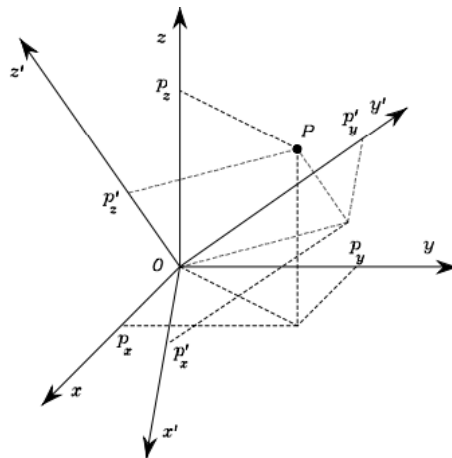


Figura 14. Representación de un vector posición en dos marcos de referencia. [4]

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad 3.8$$

$$P' = \begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} \quad 3.9$$

Cada una de las filas simboliza la magnitud de cada componente en relación al punto P, lo cual se representa en una matriz (3x1), de igual forma para el punto P'. La transformación de un sistema a otro se obtiene aplicando la matriz de rotación al sistema P' y trasponiendo la matriz de rotación para despejar la posición vista desde P'.

$$P = R * P' \tag{3.10}$$

$$P' = R^T * P \tag{3.11}$$

### 3.3.4 Composición de matrices de rotación

Para realizar una composición de matrices capaz de representar la transformación a un marco de referencia general, desde un punto actual en un cuerpo rígido, se realiza la propagación o multiplicación de cada matriz de rotación de los n-simos grados de libertad, es decir, por ejemplo, como se ve la posición desde cero a otros dos marcos de referencia 1 y 2 como se muestra en las siguientes ecuaciones (3.12), (3.13) y (3.14).

$$P_1^0 = R_1^0 * P^1 \tag{3.12}$$

$$P_2^1 = R_2^1 * P^2 \tag{3.13}$$

$$P_2^0 = R_2^0 * P^2 \tag{3.14}$$

### 3.3.5 Matriz de transformación homogénea

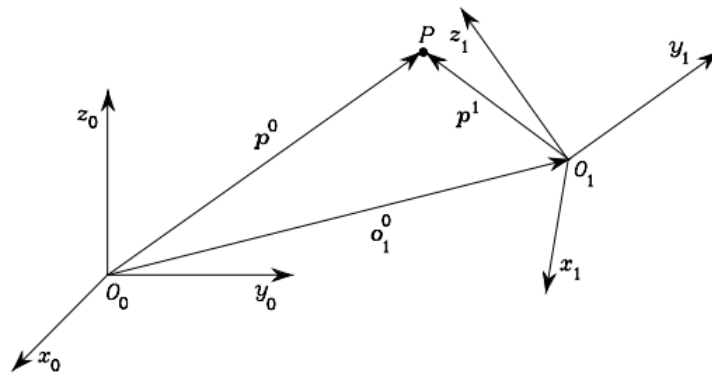


Figura 15. Vectores de posición de dos marcos de referencia diferentes. [4]

La matriz de transformación homogénea (también conocida como matriz de transformación homogénea o matriz de transformación 4x4) es una herramienta fundamental en robótica y gráficos computacionales. Se utiliza para representar y calcular las transformaciones tridimensionales, como traslaciones, rotaciones y escalas en un espacio tridimensional como se muestra en la ecuación 3.15.

$$A_1^0 = \begin{bmatrix} R_1^0 & O_1^0 \\ 0^T & 1 \end{bmatrix} \quad 3.15$$

Una aplicación en robótica para la matriz de transformación homogénea se muestra en la ecuación 3.16 la cual representa la suma vectorial para definir cómo se transforman las coordenadas entre dos marcos de referencia, considerando la componente rotacional que se aplica al marco no inercial. La rotación se expresa típicamente a través de un producto vectorial o cruz entre la matriz de rotación y el vector que representa la posición en el marco no inercial, la cual se deduce a partir de la figura 15.

$$P^0 = O_1^0 + R_1^0 \times P^1 \quad 3.16$$

### 3.3.6 Cinemática directa

Un manipulador o brazo robótico está compuesto por una serie de cuerpos rígidos (enlaces), conectados por medio de articulaciones o pares cinemáticos. Las articulaciones pueden ser prismáticas o de revolución. Uno de los extremos de la cadena cinemática debe ser el efector final (pinza o herramienta) y el otro extremo debe estar acoplado a la base como se muestra en la figura 16.

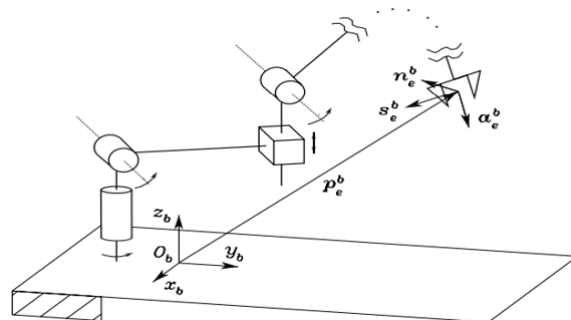


Figura 16. Descripción de la posición y orientación de un efector final. [4]

una cadena cinemática abierta se refiere a una secuencia de uniones y articulaciones que conectan ambos extremos de un robot sin cerrar en un bucle. La cinemática directa es el enfoque

que modela la ubicación espacial, orientación y posición del efecto final o extremo del robot con un marco de referencia particular en cada extremo de la cadena.

La posición y orientación del efecto final con respecto al marco de referencia  $O_b$  que se muestra en la figura 16, se obtiene de la matriz de transformación homogénea y se define como la función de cinemática directa como se muestra en la ecuación 3.17. Para determinar dicha ecuación es necesario realizar la multiplicación de las transformaciones homogéneas de cada articulación  $i+1$  en función del marco agregado a la articulación  $i$  tal y como se muestra en la ecuación 3.17.

[Robots y sistemas sensoriales. [6] [7] [8]

$$T_c^b(q) = \begin{bmatrix} n_c^b(q) & s_c^b(q) & a_c^b(q) & p_c^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.17$$

### 3.3.7 Convención Denavit-Hartenberg

La convención Denavit-Hartenberg permite obtener unos parámetros únicos que describen el robot y que lo definen desde sus grados de libertad hasta la orientación y posición del efector final, con ayuda de las transformaciones homogéneas de una cadena cinemática abierta. Esta convención describe lo que se conoce como la cinemática directa de un brazo robot articulado con solamente 4 variables, rotación en Z, translación en Z, rotación en X y translación en X. En la figura 17 se puede observar cómo se distribuyen la configuración general para obtener los parámetros DH. [6]

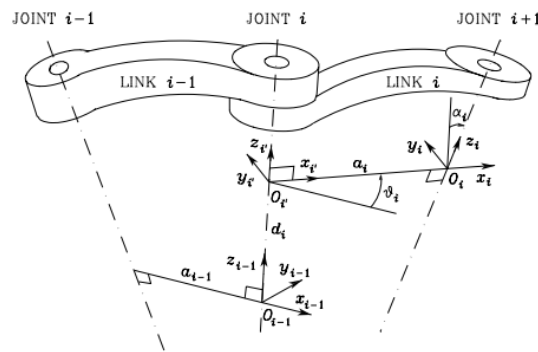


Figura 17. Parámetros cinemáticos de Denavit-Hartenberg. [6]

U <sub>i</sub>	El ángulo entre el eje X <sub>i-1</sub> y X <sub>i</sub> sobre el eje Z <sub>i-1</sub> debe tomarse positivo cuando la rotación se realiza en sentido a las manecillas del reloj.
d <sub>i</sub>	Coordenada de O <sub>i</sub> ' a lo largo de Z <sub>i-1</sub>



$a_i$	Distancia entre $O_i$ y $O_i'$
$\alpha_i$	Es el ángulo entre el eje $Z_{i-1}$ y $Z_i$ sobre el eje $X_i$ también debe ser positivo, pero cuando se toma la rotación en sentido contrario a las manecillas del reloj.

Tabla 1. Parámetros Denavit – Hartenberg. [6] [6]

Es necesario seguir una serie de reglas para definir la convención DH, dichas reglas se componen de la siguiente manera:

- ✓ Localizar y seleccionar los ejes de movimiento como los ejes Z, ejes de rotación para articulaciones de revoluta y ejes de desplazamiento para articulaciones prismáticas.
- ✓ Situar el marco de referencia O en la intersección del eje  $Z_i$  con la que forma la normal común al eje  $Z_{i-1}$  y  $Z_i$ . También localizar  $O'$  en la intersección de la normal común al eje  $Z_{i-1}$
- ✓ Elegir el eje  $X_i$  a lo largo de la línea normal común del eje  $Z_{i-1}$  y  $Z_i$  con dirección positiva saliendo de la articulación  $i$  a la articulación  $i + 1$
- ✓ Complete el eje  $Y_i$  con la regla de la mano derecha
- ✓ Si la articulación  $i$  es una articulación de revoluta la variable de junta es  $U_i$
- ✓ Si la articulación  $i$  es una articulación prismática la variable de junta es  $d_i$
- ✓ Las otras dos variables ( $a_i$ ,  $\alpha_i$ ) son siempre constantes y dependen de la geometría del sistema.

Luego la matriz que representa los parámetros DH se muestra en la ecuación 3.18 a continuación.

$$A_i^{i-1}(q_i) = A_i^{i-1} A_i^{i'} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} c_{\alpha_i} & s_{\vartheta_i} s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i} c_{\alpha_i} & -c_{\vartheta_i} s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.18$$

### 3.3.8 Cinemática inversa

La cinemática inversa se enfoca en determinar las configuraciones de las articulaciones del robot (es decir, los ángulos o longitudes) que logran una posición y orientación específica del efector final. Este proceso se basa en herramientas matemáticas como álgebra, trigonometría y cálculo vectorial para resolver las relaciones geométricas y matemáticas entre las articulaciones y los enlaces del robot.

La ley de cosenos es una herramienta vital en la cinemática inversa, ya que permite expresar ángulos en función de longitudes conocidas de los enlaces y la posición deseada del efector final,

al aplicar la ley de cosenos en un triángulo formado por los enlaces del robot, se puede resolver un ángulo como una función de las longitudes de los enlaces y las coordenadas del efector final.

La posición final del robot es la entrada en coordenadas (x, y, z), con la que se puede modelar como un vector resultante en el análisis de cinemática inversa. El análisis de la cinemática inversa es un desarrollo en base de la trigonometría intuitiva. El desarrollo es único para cada diseño.

La combinación de estas herramientas matemáticas permite formular ecuaciones que relacionan las coordenadas espaciales deseadas (x, y, z) con los ángulos requeridos en cada articulación para alcanzar esas posiciones. Este enfoque es fundamental para el diseño, la cinemática y la planificación de movimientos en brazos robóticos, permitiendo el control y la ejecución de tareas específicas. [11]

### 3.3.9 Cinemática diferencial (Jacobiano)

El Jacobiano geométrico es esencial para comprender cómo los cambios en las velocidades de las articulaciones afectan las velocidades lineales y angulares del extremo del robot. Esta matriz mapea la cinemática del robot y permite calcular la velocidad del extremo en función de las velocidades de las articulaciones. [10] [4]

En cuanto a la operación del Jacobiano, es crucial tener en cuenta la topología del robot, lo que implica comprender la disposición y la conexión de las articulaciones y enlaces. Matemáticamente, el cálculo del Jacobiano suele implicar dos etapas: primero se calculan las componentes lineales y luego las componentes angulares de la velocidad del extremo del robot como se muestra en la ecuación 3.19.

$$\begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} z_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \quad 3.19$$

$J_{P_i}$  y  $J_{O_i}$  Representan las operaciones del Jacobiano para la velocidad lineal y angular respectivamente. Estas son matrices de tamaño  $(3 \times n)$  que describen la relación entre la velocidad lineal y angular del efector final y las velocidades de las articulaciones. La expresión  $Z_i - 1$  representa las componentes rotacionales a lo largo del eje de cada articulación. Estas componentes se extraen típicamente de las tres primeras filas de la tercera columna de una matriz de transformación homogénea, cuando se considera el eje z de esa matriz.  $P_e$  representa la posición

del efector final con respecto al origen, mientras que  $P_{i-1}$  es el vector posición de cada articulación con respecto al marco de referencia ubicado en  $i - 1$ .

Esta operación permite expresar la velocidad del extremo del robot, tanto en términos de su movimiento lineal como angular, como la derivada con respecto al tiempo.

Donde la sumatoria representa la propagación de velocidades, es decir, los términos de las sumatorias son la contribución de cada una de las velocidades de las articulaciones a la velocidad lineal y angular del efector final.

$$\dot{p}_e = \sum_{i=1}^n \frac{\partial p_e}{\partial q_i} \dot{q}_i = \sum_{i=1}^n J_{P_i} \dot{q}_i \quad 3.20$$

$$\omega_e = \omega_n = \sum_{i=1}^n \omega_{i-1,i} = \sum_{i=1}^n J_{O_i} \dot{q}_i \quad 3.21$$

En el contexto de la cinemática diferencial, esta derivada respecto al tiempo representa cómo las velocidades de las articulaciones se relacionan con las velocidades del extremo del robot como en se muestra en las siguientes ecuaciones.

Es posible obtener la matriz jacobiana diferenciando la función cinemática directa del manipulador con respecto a las variables conjuntas, como se muestra en la ecuación 3.22, 2.23, 3.24, 3.25, 3.26, 3.27 que representan la siguiente expresión y se refiere a las coordenadas lineales y angulares del manipulador. [7]

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_y}{\partial q_1} & \dots & \frac{\partial f_y}{\partial q_n} \end{bmatrix}$$

Esta relación se encuentra descrita por lo general en coordenadas cartesianas, no se tienen en cuenta las fuerzas o pares que actúan sobre el brazo articular como lo son gravedad y momentos ejercidos por las demás articulaciones. [5]

De esta forma la matriz Jacobiana directa nos permite encontrar las velocidades del efector final a partir de las velocidades articulares del mismo, por otra parte, la matriz Jacobiana inversa permite conocer las velocidades de cada articulación necesarias para obtener las velocidades determinadas en el extremo del robot [4].

$$\dot{x}_{ext}^0 = \frac{\partial f_x}{\partial q_1} + \frac{\partial f_x}{\partial q_2} + \frac{\partial f_x}{\partial q_3} + \dots + \frac{\partial f_x}{\partial q_n} \quad 3.22$$

$$\dot{y}_{ext}^0 = \frac{\partial f_y}{\partial q_1} + \frac{\partial f_y}{\partial q_2} + \frac{\partial f_y}{\partial q_3} + \dots + \frac{\partial f_y}{\partial q_n} \quad 3.23$$

$$\dot{z}_{ext}^0 = \frac{\partial f_z}{\partial q_1} + \frac{\partial f_z}{\partial q_2} + \frac{\partial f_z}{\partial q_3} + \dots + \frac{\partial f_z}{\partial q_n} \quad 3.24$$

$$\alpha'_{ext}{}^0 = \frac{\partial f_x}{\partial q_1} + \frac{\partial f_x}{\partial q_2} + \frac{\partial f_x}{\partial q_3} + \dots + \frac{\partial f_x}{\partial q_n} \quad 3.25$$

$$\beta'_{ext}{}^0 = \frac{\partial f_x}{\partial q_1} + \frac{\partial f_x}{\partial q_2} + \frac{\partial f_x}{\partial q_3} + \dots + \frac{\partial f_x}{\partial q_n} \quad 3.26$$

$$\gamma'_{ext}{}^0 = \frac{\partial f_x}{\partial q_1} + \frac{\partial f_x}{\partial q_2} + \frac{\partial f_x}{\partial q_3} + \dots + \frac{\partial f_x}{\partial q_n} \quad 3.27$$

### **3.4 Planeación de trayectorias**

Para que un sistema robot pueda llevar a cabo cierta tarea en específica, es decir, realizar de forma especial y única una trayectoria por medio de un camino determinado, según sea la necesidad de la aplicación en la industria, debe satisfacerse una serie de parámetros iniciales y finales de los cuales dependerá el camino del efector final del brazo robot. Esto con el fin de ejecutar un algoritmo de movimiento capaz de realizar la trayectoria planificada.

Para que un sistema robot pueda realizar una tarea específica a lo largo de una trayectoria determinada, se requiere una planificación cuidadosa, comprensión de la cinemática, control de movimiento preciso y retroalimentación en tiempo real. Estos factores son esenciales para garantizar el éxito de la aplicación industrial y la seguridad de las operaciones.

La finalidad del planeador de trayectorias es producir las variables de referencia de entrada para el sistema de control de movimiento, que garantice que el robot realizara esa trayectoria anteriormente programada. [4]

Es crucial desarrollar algoritmos de planificación de trayectorias que sean eficientes y reduzcan la complejidad, lo que implica generar trayectorias que no sean demasiado exigentes en términos de programación. Estos algoritmos generalmente se definen como funciones dependientes del tiempo, permitiendo muestrear la posición del robot en diferentes instantes de tiempo. Se puede realizar mediante dos metodologías las cuales son.

Trayectoria punto a punto: Esta técnica implica definir los parámetros iniciales y finales de la trayectoria, permitiendo al robot moverse directamente desde un punto de partida a un punto final de manera óptima.

Trayectoria continua a través de una serie de puntos: En este método, el robot sigue una serie de puntos continuos, lo que ofrece mayor flexibilidad en términos de la trayectoria, permitiendo una secuencia de movimientos más suaves y ajustados.

Para este caso se estudiará la implementación para llevar a cabo la planificación de trayectorias por medio de la metodología de punto a punto.

#### ***3.4.1 planificación de trayectoria punto a punto***

En la planificación de trayectorias para movimientos angulares, se busca encontrar una solución que permita moverse desde una posición inicial  $q_0$  a una posición final  $q_f$  en un tiempo determinado  $t_f$ , mientras se minimiza la energía disipada por el motor que impulsa la rotación.

Para lograr esto, se recurre a métodos de optimización que utilizan derivadas para determinar los movimientos óptimos. La idea es encontrar la trayectoria que requiere la menor energía para el motor y, al mismo tiempo, logra el movimiento deseado dentro del tiempo especificado.

El polinomio cúbico para la planificación de trayectorias, como se describe en la ecuación 3.28, es una herramienta común en este contexto. Estos polinomios permiten definir trayectorias suaves y controladas para moverse de manera eficiente y con menor consumo de energía entre dos puntos angulares como se muestra en la figura 18.

Restringiendo igualmente que la velocidad en  $t_0 = t_f = 0$  y  $a_0 = q_0$ ,  $a_1 = 0$  posición y velocidad respectivamente, se pueden determinar los coeficientes  $a_2$  y  $a_3$  a partir de las ecuaciones 3.31 y 3.32. [12]

$$q(t) = a_3 * t^3 + a_2 * t^2 + a_1 * t + a_0 \quad 3.28$$

Donde  $a_0$  es la posición inicial del efector final y  $a_1$  es la velocidad inicial que para muchos casos es igual a cero. Los coeficientes  $a_2$ ,  $a_3$  son dependientes de las condiciones iniciales  $q_0$ ,  $q_f$ ,  $t_f$ . [8]

Si se deriva el polinomio cúbico, que representa una función de posición angular en función al tiempo para obtener la velocidad, se obtiene la ecuación 3.29. se observa un efecto parabólico en la figura 19.

$$q'(t) = 3 * a_3 * t^2 + 2 * a_2 * t + a_1 \quad 3.29$$

Derivando de nuevo se obtiene la aceleración que el efector final llevará en toda la trayectoria, así se obtiene la ecuación 3.30, demuestra un movimiento lineal o una desaceleración en todo el camino en la figura 20.

$$q''(t) = 6 * a_3 * t + 2 * a_2 \quad 3.30$$

$$a_2 = \frac{3}{t_f^2} (q_f - q_0) \quad 3.31$$

$$a_3 = -\frac{2}{t_f^3} (q_f - q_0) \quad 3.32$$

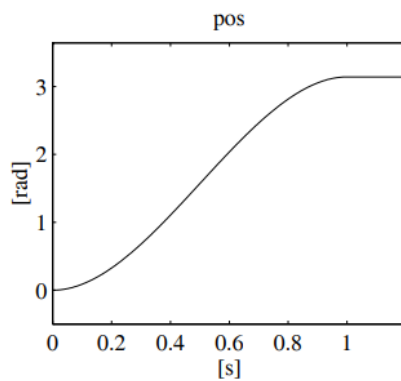


Figura 18. Perfil de velocidad para la ecuación de posición del efector final. [4]

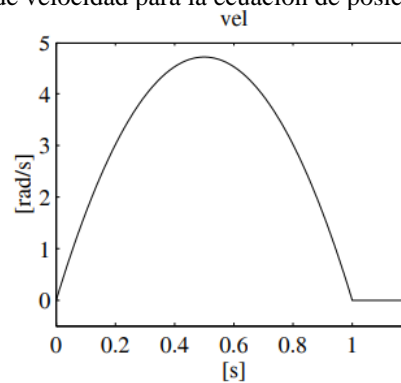


Figura 19. Perfil de velocidad del efector final. [4]

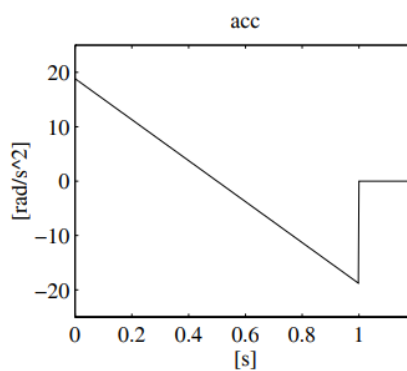


Figura 20. Perfil de aceleración del efector final. [4]

### 3.5 Dinámica

El modelo dinámico es esencial en la simulación y el control de sistemas como un brazo robot, este modelo proporciona una representación matemática de cómo las fuerzas, posiciones, velocidades y aceleraciones se relacionan entre sí en el sistema. Esto es crucial para el control preciso del robot y la garantía de que pueda seguir y ejecutar trayectorias deseadas de manera eficiente y segura, igualmente, proporciona información valiosa para el diseño mecánico, la planificación de trayectorias y el control del movimiento. [4]

Al considerar el movimiento dinámico de un brazo robot, es fundamental tener en cuenta propiedades como el momento de inercia y el centro de masas. El momento de inercia se refiere a la resistencia que un cuerpo rígido presenta ante cambios en su velocidad rotacional y se calcula teniendo en cuenta la distribución de masa en torno a un eje de rotación.

Además, sabiendo que un cuerpo rígido puede rotar alrededor de varios ejes, estos dos conceptos (centro de masas y momento de inercia) involucrados a la programación de brazos robot forman lo que se conoce como el tensor de inercia.

El tensor de inercia es una representación matricial que contiene la distribución de masa alrededor de los diferentes ejes de rotación de un cuerpo rígido. Esta matriz contiene información detallada sobre cómo la masa está distribuida y cómo influye en los movimientos rotacionales alrededor de los diferentes ejes. Que en un conjunto matricial forman una matriz de 3x3, cuya diagonal principal se conoce como momentos diagonales como se muestra en la ecuación 3.33. [8]

$$I_{l_i} = \begin{bmatrix} \int (r_{iy}^2 + r_{iz}^2) \rho dV & -\int r_{ix} r_{iy} \rho dV & -\int r_{ix} r_{iz} \rho dV \\ * & \int (r_{ix}^2 + r_{iz}^2) \rho dV & -\int r_{iy} r_{iz} \rho dV \\ * & * & \int (r_{ix}^2 + r_{iy}^2) \rho dV \end{bmatrix} \quad 3.33$$

Los elementos de la ecuación (3.33) son,  $r$  representa las componentes de un vector posición de cada articulación,  $\rho$  representa la densidad del elemento y  $dV$  un diferencial de volumen. Lo que en resumidas cuentas su puede expresar el tensor de inercia como el momento de inercia aplicado a cada eje de rotación de todos los enlaces en 3.34.

$$I_{l_i} = \begin{bmatrix} I_{l_{ixx}} & -I_{l_{ixy}} & -I_{l_{ixz}} \\ * & I_{l_{iyy}} & -I_{l_{iyz}} \\ * & * & I_{l_{izz}} \end{bmatrix} \quad 3.34$$



La ecuación (3.34) es definida positiva y se utiliza al momento de hallar la componente rotacional de la energía cinética del sistema.

El modelo dinámico de un manipulador se puede obtener a partir de dos métodos, uno de ellos es el método de Lagrange, que se basa en hallar el modelo dinámico encontrando las energías involucradas en el sistema y las derivadas de la suma de ellas respecto a posiciones y velocidades del manipulador. La segunda opción es el método de Newton-Euler, está basado en generalizar el concepto de la aceleración. [9]

### **3.5.1 Método de Lagrange**

Determinar la dinámica de un articulador por medio del método de Lagrange, consiste en hallar las derivadas de la ecuación lagrangiana (3.35), respecto a las variables conjuntas, tanto en sus posiciones como velocidades, partiendo del principio de conservación de la energía, donde  $T$  es la energía cinética y  $U$  la energía potencial de todo un sistema.

$$L = T - U \tag{3.35}$$

Luego la ecuación lagrangiana quedará por medio de las derivadas parciales con respecto a posiciones y velocidades de las variables conjuntas para cada articulación desde  $i=0$  hasta  $n$  articulaciones como se muestra en la ecuación (3.36)

$$\frac{d}{dt} \frac{\partial L}{\partial q'_i} - \frac{\partial L}{\partial q_i} = \xi_i \tag{3.36}$$

Donde  $\xi_i$  representa los torques que son ejecutados por las fuerzas generalizadas que actúan sobre cada una de las articulaciones del manipulador, por ejemplo, torque generado por la acción de la gravedad, fuerzas centrífugas y de coriolis, dependiendo de la morfología del manipulador robot. [4] [10]

### **3.5.2 Operación de la energía cinética**

Definida como un medio de la masa por la cantidad de movimiento de esa masa o cuerpo rígido por la velocidad que lleva en ese momento. En la ecuación 3.37 se observa la definición de energía cinética de todo un manipulador robot.

Donde los términos  $K_i$  representan la energía cinética de cada enlace, la energía cinética del enlace  $i$  está definido por las masas de cada enlace y su respectiva velocidad lineal y angular, igualmente depende del momento de inercia del enlace  $i$ , como se muestra en la ecuación 3.38.

Acoplado la metodología para determinar las ecuaciones que definen la morfología del robot, como lo son el método D-H y Jacobiano, la energía cinética de cada enlace del manipulador se puede expresar por la ecuación (3.35). [4]

$$T = \sum_{i=0}^n K_i \tag{3.37}$$

$$K_i = \frac{1}{2} m_i v_i^2 + \frac{1}{2} I \omega_i^2 \tag{3.38}$$

$$K_i = \frac{1}{2} m_i q'^T J_p^i J_p^i q' + \frac{1}{2} q'^T J_0^i R_i I_i R_i^T J_0^i q' \tag{3.39}$$

### 3.5.3 Operación de la energía potencial

Definida como  $mgh$  donde  $m$  es la masa,  $g$  es la gravedad y  $h$  la altura relativa a un punto de referencia. Aplicando esta teoría a un cuerpo rígido, para este caso el enlace  $i$ , queda la energía potencial como en la ecuación (3.40), donde  $\mathbf{g}$  y  $\mathbf{p}_i$  son matrices y la gravedad esta aplicada al eje vertical de sistema de referencia, las demás componentes serán cero, así mismo  $\mathbf{p}_i$  es un vector de componentes  $x, y, z$ . [4]

$$U = \sum_{i=1}^n m_i \mathbf{g} \mathbf{P}_i \tag{3.40}$$

### 3.5.4 Ecuación de movimiento

Para poder modelar un sistema es necesario contar con una ecuación de movimiento, esta ecuación determina la dinámica del sistema y es bastante útil al momento de realizar el control. En robótica se puede determinar esta ecuación de movimiento a partir del método de Lagrange o el método de Newton-Euler, para así poder realizar el correspondiente análisis dinámico del sistema manipulador. Al operar la ecuación lagrangiana se puede reorganizar los términos derivados de la velocidad, posición y gravedad como se puede observar en la ecuación (3.41) y la estructura dinámica del sistema robot general ahora es: [11]

$$\tau = M(\theta)\theta'' + V(\theta, \theta') + G(\theta) + F(\theta') \tag{3.41}$$

Donde  $\tau$  es el vector de torques generalizado de  $n$  elementos,  $M(\theta)$  es la matriz de masas y de inercias del manipulador y es una matriz de  $(n \times n)$  elementos, depende de las variables generalizadas y relaciona las posiciones angulares con las distancias a las articulaciones.  $V(\theta)$  es un vector de  $n$  elementos y representa los torques generados por las fuerzas centrífugas y de coriolis,  $G(\theta)$  es un vector de  $n$  elementos y representa los torques generados por las otras articulaciones e involucra en todos los términos la constante de la gravedad,  $F(\theta')$  es el vector que representa los torques generados por las fuerzas de fricción y solo depende de la variable articular, es decir que para modelar el sistema dinámico se pueden desacoplar. [8]. Los miembros de las matrices de masas, se determinan a partir de las derivadas parciales de la ecuación lagrangiana con respecto a cada una de las variables generalizadas, al igual que para el vector de fuerzas de coriolis y centrífugas y la matriz de gravedad. Es decir, por ejemplo, para un manipulador de 2 articulaciones rotacionales (RR), para encontrar la ecuación del torque, se debe de derivar la ecuación lagrangiana con respecto a posición y velocidad de ambas articulaciones para encontrar dichas matrices. [12]

### **3.6 Control de articulaciones en sistemas robots**

Con el fin de realizar una tarea controlada, en el caso de la robótica se desea llevar a cabo un control de posición o control de movimiento, control de trayectoria, control de fuerza. Se aplican leyes de control como lo son el control centralizado o acoplado, que es cuando el control de una articulación tiene en cuenta los efectos de los torques generados por las demás articulaciones, el control desacoplado o descentralizado es un tipo de control por articulación, un control utilizado en los sistemas MIMO (múltiples entradas y múltiples salidas) un control independientemente de las demás articulaciones, el cual se puede ver como un sistema SISO (única entrada única salida). [5] [4]

Finalmente, se conocen aplicadas o desarrolladas varias leyes de control basado en el modelo dinámico, técnicas de control por medio del par computado. Nuevas técnicas de control como control adaptativo o control con aprendizaje, el cual contiene en la estrategia de control un algoritmo de aprendizaje. Técnicas de control en el espacio de los estados viendo el sistema como uno solo, control en el lugar geométrico de las raíces y control en el espacio cartesiano, técnicas de control de esfuerzos y finalmente el control PID ampliamente utilizado en la industria. [8]

En esta amplia gama de control implementables en la industria robot entran igualmente los controles lineales y no lineales, los cuales se modelan con las técnicas matemáticas como ecuaciones diferenciales lineales, leyes de Newton, ecuación de Lagrange y método de Euler. Frecuentemente, se encuentran aplicados los controles sobre los servo-motores que son los actuadores encargados de realizar el trabajo, los controles en tiempo discreto, los cuales se encuentran implementados en los robots como el robot PUMA 560. [11]

#### ***3.6.1 control PD con precompensación o alimentación anticipada***

En este documento se planea desarrollar una ley de control PD (proporcional- derivativo), con acción de pre-compensación, la acción de pre-compensación en la ley de control se debe a que se usa el modelo dinámico explícitamente a la planeación de la trayectoria. Viendo el sistema MIMO del brazo robot como varios sistemas SISO independientes desacoplados, pero relacionados teniendo en cuenta el modelo dinámico del manipulador y fuera del ciclo del servo, un control no lineal de segundo orden y con retro alimentación de posición y velocidad. El cual tendrá como referencia o señal de entrada la planeación de las trayectorias en los splines cúbicos como se muestra en la figura 21 del modelo de un control de trayectoria.

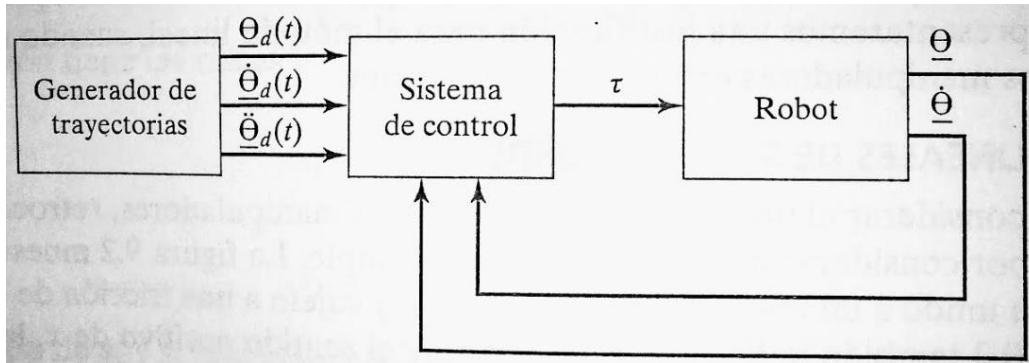


Figura 21. La ley de control PD con precompensación se puede describir como se muestra en la ecuación (3.42). [11]

$$\tau = K_p \bar{q} + K_v \dot{\bar{q}} + M(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + g(q_d) \quad 3.42$$

Donde  $K_p$  y  $K_v$  en un sistema MIMO se puede ver como una matriz de  $n \times n$  con una diagonal principal definida positiva y en los sistemas SISO son ganancias de posición y velocidad aplicadas a las señales de retroalimentación y referencia que actúan como señal de error de posición y velocidad respectivamente sobre cada articulación. El esquema de control en lazo cerrado de la ecuación 3.42 se muestra en la figura 22.

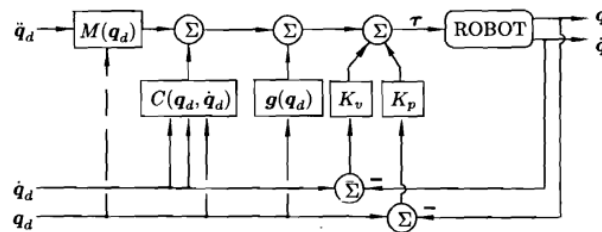


Figura 22. Control PD con precompensación. [5]

El comportamiento en lazo cerrado se representa mediante la ecuación (3.43).

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = K_p \bar{q} + K_v \dot{\bar{q}} + M(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + g(q_d) \quad 3.43$$

Con el fin de calcular el punto de equilibrio, se asume como torque de control  $\tau_c$ , este será el torque aplicado a la articulación, es decir, que la entrada de control para este caso sería:

La ecuación 3.43 queda entonces como aparece en la ecuación (3.44)

$$\tau_r + K_d(q_d - q_r) + K_v(q'_d - q'_r) = \tau_c \quad 3.44$$

Donde  $\tau_r$  representa el torque de alimentación anticipada que se encuentra compuesto por la matriz de masas, vector de coriolis y fuerzas centrifugas y vector de gravedad.

Si se obtiene un modelo dinámico bien concreto y que se acerca lo más posible a la exactitud del modelo se puede considerar a  $\tau_r$  como  $\theta''$ , con el fin de aplicar un doble integrador, así como en la figura 23. La ecuación (3.41) que da entonces como en (3.45). [8] [5]

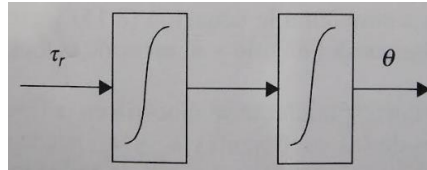


Figura 23. Control de doble integrador. [11]

$$\tau_c = \theta'' + K_v e' + K_p e \quad 3.45$$

Donde  $e = \theta_d - \theta$  Para la ecuación de equilibrio se asume que no hay control es decir  $\tau_c = 0$ . La ecuación 3.45 queda:

$$e'' + K_v e' + K_p e = 0 \quad 3.46$$

Para ajustar los valores de  $K_v, K_p$  se toma la función de transferencia de la ecuación (3.45) por medio de la transformada de Laplace.

$$\tau_c(s) = S^2 e(s) + K_v S e(s) + K_p e(s) \quad 3.47$$

$$\frac{e(s)}{\tau(s)} = \frac{1}{s^2 + K_v S + K_p}$$

El cual el denominador de la ecuación (3.47) se toma como el polinomio característico del sistema de control.

La ubicación de las ganancias proporcionales y derivativas para que el sistema permanezca estable, se pueden definir a partir del tipo de respuesta que se obtiene en los sistemas de segundo orden, teniendo en cuenta  $\xi$  como el factor de amortiguamiento y  $\omega_n$  como la frecuencia natural. El polinomio deseado es de la forma como se muestra en la ecuación (3.48) y dependerá de las especificaciones de control.

$$p(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \quad 3.48$$

Si lo que se desea es tener una respuesta suave sin oscilaciones se debe seleccionar el factor de amortiguamiento crítico, tal que se obtengan poles reales iguales, es decir,  $\xi = 1$ . [13]

La ecuación de equilibrio para que los polos del sistema queden fijos queda entonces:

$$s^2 + 2\xi\omega_n s + \omega_n^2 = S^2 + K_v S + K_p = 0 \quad 3.49$$

Igualando las dos ecuaciones para encontrar los valores de  $K_v$  y  $K_p$ . [11] [4]

$$\omega_n = \sqrt{K_p}$$
$$2\xi\sqrt{K_p} = K_v \quad 3.50$$

$$K_v = 2\sqrt{K_p}$$
$$t_s = \frac{5}{\xi\omega_n} \quad 3.51$$

Con la ecuación 3.51 se establece el tipo de respuesta que se desea y el tiempo de respuesta, con eso se determina la frecuencia natural.

## **4 Desarrollo metodológico**

Primero que todo se acude al sitio en donde se desea implantar el brazo robot, con el fin de recolectar la mayor cantidad de datos sobre las especificaciones de diseño, como alcance del efector final, posiciones a adoptar, ubicación espacial del brazo, trayectorias de pintura a llevar a cabo, definición de grados de libertad, etc. Luego se proponen las ideas de diseño y así se escoge y se procede a realizar el diseño CAD en solidworks. Con el fin de poderlo importar a Simscape de MATLAB para realizar la planificación de trayectorias y modelar el sistema dinámico del robot por medio del método de Lagrange. Finalmente, se procede a definir una ley de control capaz de realizar las trayectorias de pintura de forma controlada y adecuada.

A continuación, se demuestra un cuadro en la figura 24 el cual permite visualizar como se planea cumplir con cada uno de los objetivos de forma esquemática.



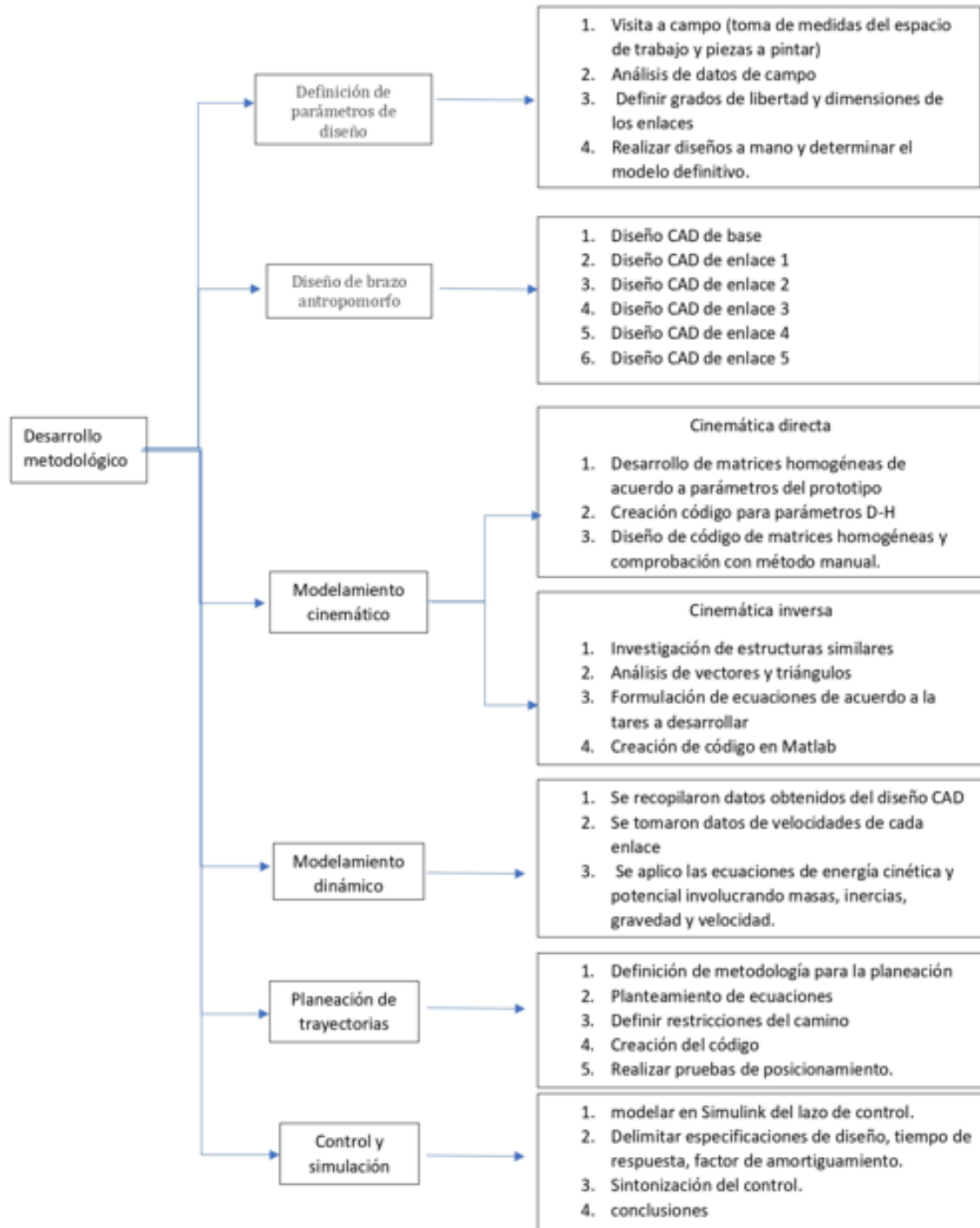


Figura 24. planteamiento del desarrollo metodológico.

#### 4.1 Definición de los parámetros de diseño

La fase inicial del proyecto consistió en una visita a la empresa, en donde se tomaron las especificaciones de diseño, se midió el espacio de la cabina, los cuales eran 3x3x6 metros, en cuestión de obstáculos se contó con gran espacio, solo se interpusieron los tubos entre sí, cuyas distancias podían oscilar entre los 0.2 a 0.6 metros. Las dimensiones de las piezas aplicar la pintura electrostática eran de 1 metro de largo y un diámetro de 0.0127 metros, para lo cual con dos trayectorias de pintura por dos lados opuestos quedara completamente recubierta la pieza. Los tiempos para la realización del proceso se definieron a partir de requerimientos de producción que oscilan entre 5 a 10 segundos por pasada, la posición del efector final para la aplicación de la pintura es perpendicular a la pieza siempre de forma horizontal de tal manera que se recubra la pieza por dos trayectorias, la ubicación espacial del brazo robot se definió basándonos en los grados de libertad, luego teniendo en cuenta el criterio de proporcionalidad de los grados de libertad con respecto a la tarea a realizar que se expuso en el inciso (3.1) y también teniendo en cuenta que para posicionar un objeto en 3D un manipulador debe de tener por lo menos 6 (DOF), 3 para posicionamiento y 3 para la orientación. Se estiman los diferentes desplazamientos de forma progresiva desde la base al efector final, de manera que se cumplan con todos los movimientos necesarios y se determinaron 5 (DOF), con el fin de tener más flexibilidad y menos complejidad al momento de realizar la planeación de trayectorias y el control. A continuación, se muestra en la figura 25 las condiciones actuales del espacio de trabajo.



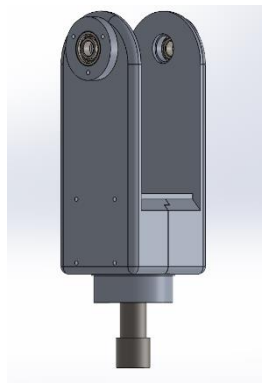
*Figura 25. espacios de trabajo para el proceso de la aplicación de pintura electrostática.*

## 4.2 Diseño del brazo antropomorfo

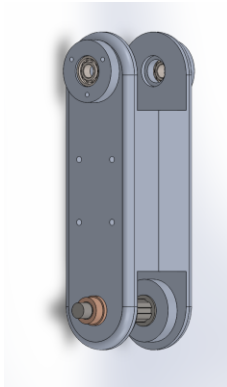
En la segunda etapa del proyecto se presentaron bosquejos a mano alzada del diseño, para definir un modelo que se acople a las especificaciones de las tareas del trabajo, necesidades, eficiencia para llevar a cabo las trayectorias. Teniendo en cuenta las necesidades y eficiencia en la ejecución de las trayectorias, de la siguiente manera se seleccionaron los grados de libertad.

- ✓ Movimiento hacia adelante para acercarse a la pieza.
- ✓ Movimiento vertical para desplazarse hacia arriba y hacia abajo (posiblemente para la tarea de pintura).
- ✓ Una articulación para girar sobre la base del robot
- ✓ Otra articulación para girar el efector final.

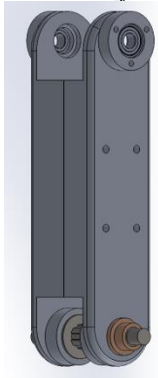
Teniendo en cuenta estas especificaciones y que para ubicar un elemento en un espacio 3D se requieren como mínimo 6 DOF, 3 para posicionar y 3 para orientar el elemento. Se pudo optimizar y definir un brazo antropomorfo de 5 DOF, tomando la articulación 3 como un codo que permita el movimiento hacia arriba y hacia abajo dándole más espacio de trabajo a esa articulación. De esta forma se inició con el diseño en solidworks como se muestra en las siguientes figuras en la que se contemplan el brazo robot.



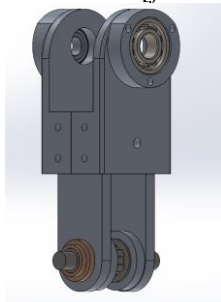
*figura 26. Enlace 1 (Base) [fuente propia]*



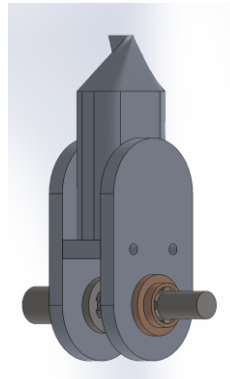
*figura 27. Enlace 2 [fuente propia]*



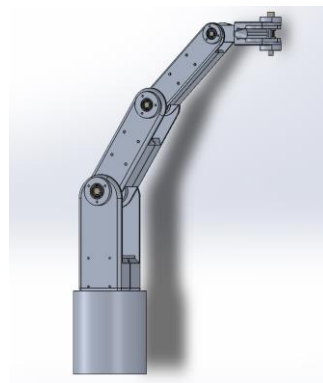
*figura 28. Enlace 3 [fuente propia]*



*figura 29. Enlace 4 [fuente propia]*



*figura 30. enlace 5 [fuente propia]*



*figura 31. Ensamble final del brazo robot. [fuente propia]*

La definición de las dimensiones de los enlaces, ya teniendo un diseño realizado, se determinaron sobre un eje de coordenadas desde el cual se ubicó el origen en coordenadas  $(x, y, z)$ . A una distancia del origen en la base del robot de 0.450 metros en  $x$  se encuentra ubicada la pieza a pintar y a una altura de un metro en  $Z$ , esa será la posición más alta del robot. Como se muestra en la figura 32. Se muestra un vector  $r$ , el vector suma de  $L2$ ,  $L3$ , y  $L4$  para estimar sus longitudes, teniendo en cuenta que el enlace 4 siempre este horizontalmente al suelo.

En la figura 33 se observa el brazo robot en el plano  $(x, y)$  desde arriba, definiendo también la longitud de  $L5$  como 0.145 metros. Luego, para la determinación de los enlaces 2, 3 y 4 se plantea la figura 34 asignándole al enlace 4 un valor de 0.165 metros, con la condición de siempre permanecer en coordenadas de  $x$ , es decir, perpendicular a la pieza a pintar y paralelo al suelo. Ya habiendo escogido la distancia a la que se quiere ubicar en  $x$  y la distancia desde  $L1$  al inicio de la

pieza que fue de 0.5 metros, se calcula el máximo valor de  $h$  que es igual a 0.512 metros. Sabiendo que estos dos enlaces hombro y antebrazo, son proporcionales, se toma a  $L_2$  como 0.3 y  $L_3$  como 0.305 metros para que no queden totalmente extendidos cuando se requiera la posición más alta.

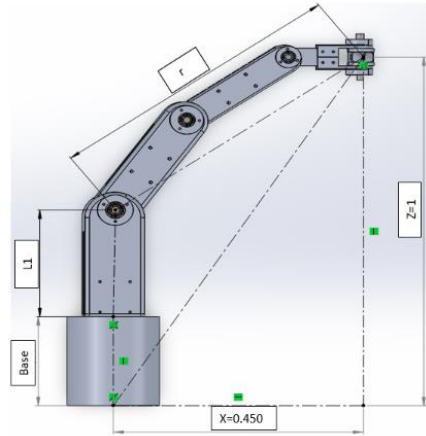


Figura 32. disposición inicial para iniciar la trayectoria. [fuente propia]

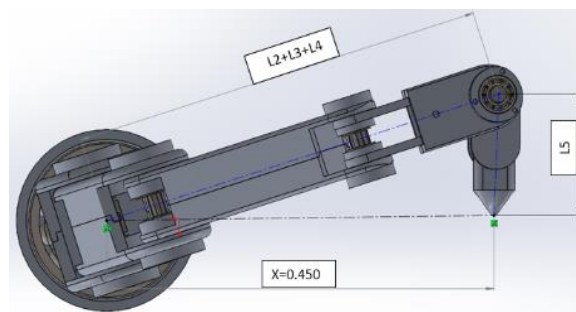


Figura 33. definición de  $L_5$  usando vista superior. [fuente propia]

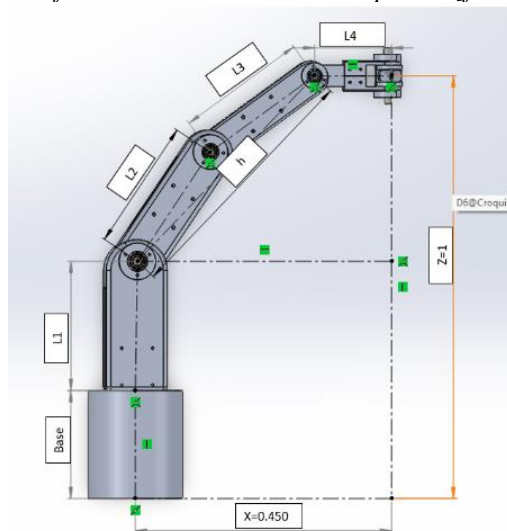
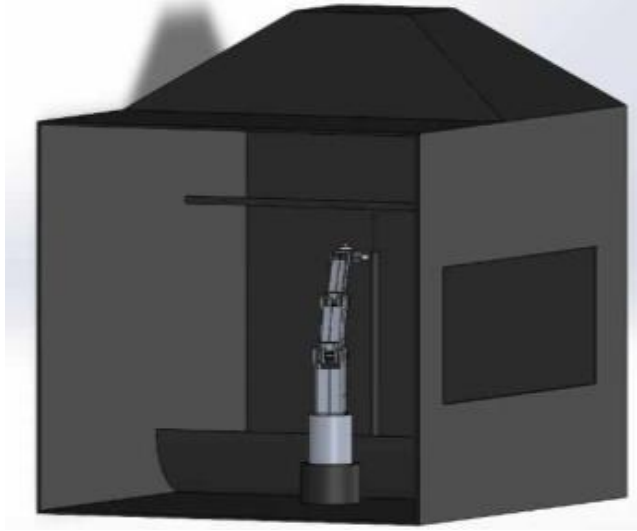


Figura 34. vector resultante para cálculo de  $L_2$  y  $L_3$ . [fuente propia]



*Figura 35. disposición del robot dentro de la cabina*

Una vez se diseñó y desarrollo el brazo robot por completo, se involucró la cabina para establecer el espacio de trabajo de la estación robot de pintura automatizada, donde se ejecutará la simulación, como se muestra en la figura 35. Se realizó un algoritmo para aproximar muy bien el espacio de trabajo del brazo robot teniendo en cuenta su morfología dentro de la cinemática inversa.

### 4.3 Exportación de diseño mecánico a Simulink de MATLAB

Una vez concebido un diseño mecánico, por medio de solidworks se hace uso de la herramienta Simscape multibody, ubicada en la barra de herramienta en la pestaña herramientas, para realizar la exportación del modelo robot a MATLAB. Se crea una carpeta para guardar la ubicación de la exportación en formato xml. Estando en el comand window de MATLAB se hace uso del comando `smimport('Robot05.slx')` para completar el proceso. Luego se abrirá un documento en Simulink con un diagrama de bloques de Simscape de un sistema mecánico como se observa en la figura 36.

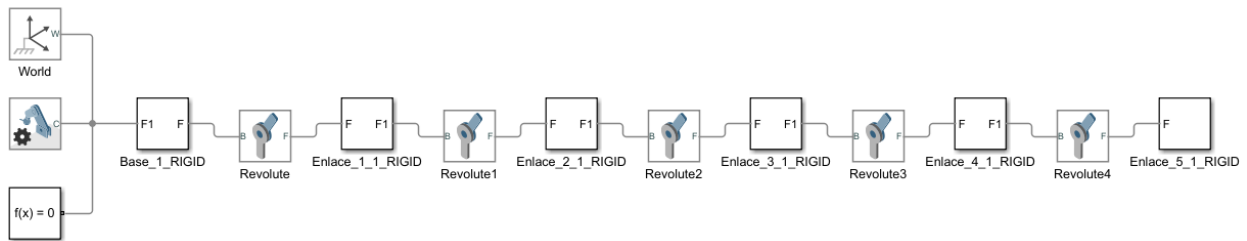







Figura 36. (Diagrama de bloques del sistema dinámico brazo robot antropomorfo en Simulink). [fuente propia]

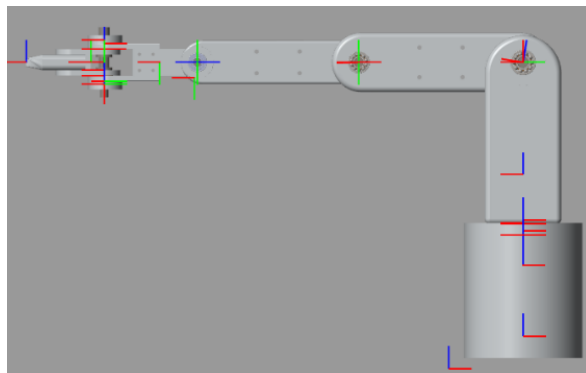
En donde el bloque de  World es el sistema de referencia general en Simulink, el cuadro de configuración de mecanismo  es el bloque donde se configura la acción de la fuerza de gravedad en el eje vertical de su sistema de referencia y el cuadro de solver function , debe de estar es todo modelamiento de un sistema mecánico Simscape para poder comenzar una simulación.

El cuadro que relaciona cada uno de los enlaces entre si es , que contiene los marcos de referencia de los elementos que componen un solo enlace con respecto al enlace anterior y al marco de referencia Word. Estos sistemas de referencia deben de coincidir con la convención Denavit Hartenberg. El cuadro  representa una articulación de revoluta, en la cual se configuran los parámetros de entrada como posición o torque y de salida como posición, velocidad angular, y aceleración según sea requerida la necesidad.

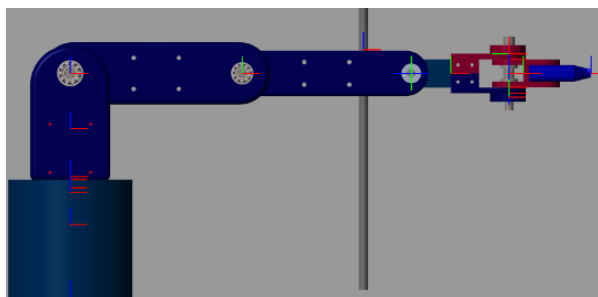
Configurando los bloques de articulación de revoluta tenemos para las especificaciones a la entrada un torque que vendrá dado por la dinámica del sistema y a la salida censaremos la posición y la velocidad angular.



En la figura 37 se observa como los sistemas de referencia no aparecen como se espera en la convención Denavit Hartenberg con coordenadas en  $x, z$  y en la figura 38 como se corrigen los sistemas de referencia adecuadamente según la convención para una sincronizada simulación.



*Figura 37. sistemas de referencia desincronizados. [fuente propia]*



*Figura 38. marcos de referencia sincronizados con parámetros D-H. [fuente propia]*

#### 4.4 Modelamiento cinemático directo e inverso

Para la etapa del modelamiento cinemático se tuvo en cuenta el algoritmo de Denavit Hartenberg como punto de partida, para eso realizamos la metodología de los parámetros D-H como se explica en el apartado (3.3.7), se obtuvieron los parámetros D-H de la estructura como muestra la figura 39, a partir de la ecuación (3.18) que se plasmó mediante una función de MATLAB, figura 40.

```
R5GDL:: 5 axis, RRRRR, stdDH, slowRNE
```

j	theta	d	a	alpha	offset
1	q1	398	0	1.5708	0
2	q2	0	305	0	0
3	q3	0	300	0	0
4	q4	0	150	-1.5708	0
5	q5	0	145	0	0

Figura 39. parámetros D-H robot antropomorfo 5 DOF

```
function [Y]=FDH(a,alpha,d,theta)
[Y]=[cos(theta) -sin(theta)*cos(alpha) sin(theta)*sin(alpha) a*cos(theta);sin(theta) cos(theta)*cos(alpha) -cos(theta)*sin(alpha) a*sin(theta);0 sin(alpha) cos(alpha) d;0 0 0 1]
end
```

Figura 40. función D-H para modelo cinemático del robot [fuente propia]

Teniendo estos parámetros definidos se hizo uso de la herramienta de MATLAB para modelar un robot articular sencillo obteniendo la figura 41 que muestra posición y orientación de cada articulación en la posición cero o inicial.

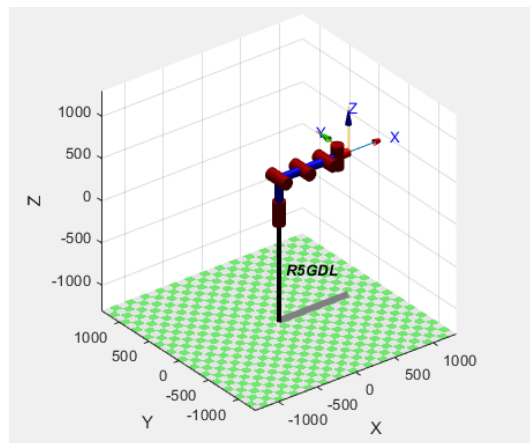


Figura 41. posición y orientación inicial del robot [fuente propia]

#### 4.5 Cinemática directa

Para la cinemática directa se pudo hacer uso de la función D-H planteada en MATLAB con el fin de obtener la matriz de transformación homogénea en términos de las variables conjuntas, que nos entregó orientación y posición de la articulación  $i$  con respecto al marco de referencia en el origen, figura 42. Para nuestro robot obtuvimos la posición y orientación del efector final vista desde el origen en la articulación  $i$ , haciendo uso de la matriz de transformación homogénea tomando de la columna número 4 las filas 1 a 3, que representan la componente traslacional de la posición del efector final como en el apartado (3.3.5) y como aparece en la figura 43.

```

%% PARAMETROS DENAVID HARTENBERG
%% a alpha d theta
A01=FDH(0,pi/2,L1,th1);
A12=FDH(L2,0,0,th2);
A23=FDH(L3,0,0,th3);
A34=FDH(L4,0,0,th4);
A45=FDH(L5,-pi/2,0,th5);
%% matriz de transformacion homogenea
T01=A01;
T02=A01*A12;
T03=A01*A12*A23;
T04=A01*A12*A23*A34;
T05=A01*A12*A23*A34*A45;
size(T05)

```

Figura 42. matriz de transformación homogénea con función D-H en MATLAB [fuente propia]

La función de cinemática directa para cada una de las articulaciones en términos de  $(x, y, z)$  del robot antropomorfo, se alcanza con la ecuación (3.17)

```

%% CINEMATICA DIRECTA
M=eye(4)
xx=[M(1,4) T01(1,4) T02(1,4) T03(1,4) T04(1,4) T05(1,4)]
yy=[M(2,4) T01(2,4) T02(2,4) T03(2,4) T04(2,4) T05(2,4)]
zz=[M(3,4) T01(3,4) T02(3,4) T03(3,4) T04(3,4) T05(3,4)]

```

Figura 43. Cinemática directa del robot antropomorfo [fuente propia]

#### 4.6 Cinemática inversa

El desarrollo de la cinemática inversa se empezó definiendo condiciones de trabajo como:

- ✓ Trayectoria máxima del objetivo a pintar en este caso un tubo de un metro la cual se ve en el eje de las z
- ✓ Distancia en el eje de las x de la base al tubo
- ✓ Longitud de enlace uno y cinco
- ✓ Ángulo de aplicación de pintura

Ya con estos datos definidos se empezó por analizar el triángulo que se forma entre la distancia entre la base y el tubo

- ✓ longitud y posición del enlace cinco

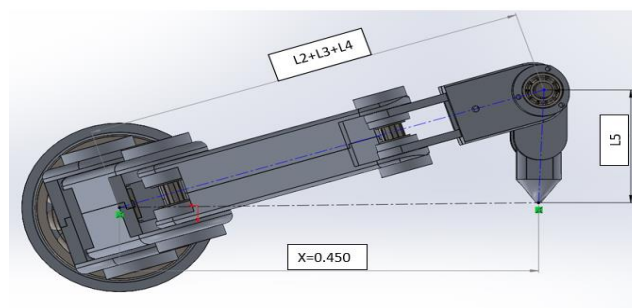


Figura 44. Triangulo de la posición inicial de enlace 1 y 5 [fuente propia]

Mirando la figura 44 vector  $L_2 + L_3 + L_4$ ,  $L_5$  y  $x$  forman un triángulo del cual ya tenemos la longitud del enlace cinco y la distancia en  $x$  del objetivo y con la función de tangente a la menos uno obtenemos el ángulo en enlace uno.

$$G_1 = \tan^{-1}\left(\frac{L_5}{x}\right) \quad 4.1$$

$$G_5 = -\left(\tan^{-1}\left(\frac{x}{L_5}\right) + \frac{\pi}{2}\right) \quad 4.2$$

Vector resultante  $L_2 + L_3 + L_4 = r$

$$r = \sqrt{x^2 + L_5^2} \quad 4.3$$

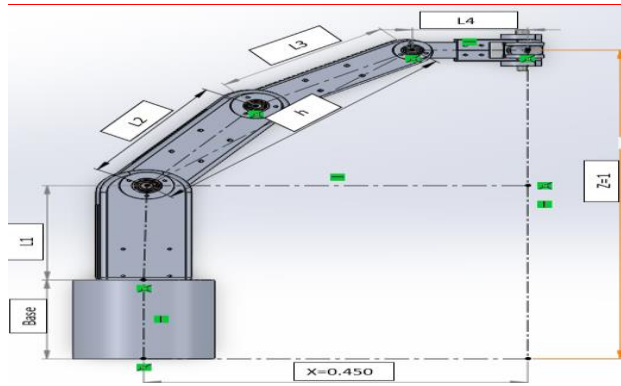


Figura 45. Posición inicial para cinemática inversa [fuente propia]

Para definir grados del enlace cuatro utilizamos una de las condiciones de trabajo que era que la pistola quedará en paralelo al suelo lo cual se plasma en el eje  $x$  y con el eje del enlace dos como el punto de referencia de nuestro análisis y la longitud del tubo a pintar la vamos a dividir en dos para tomar una parte positiva y negativa en el eje  $z$ .

El valor en el eje  $x$  se obtiene de la resta del vector resultante  $r$  y la longitud del enlace cuatro.

$$x_1 = r - L_4 \tag{4.4}$$

Ya con la distancia final en  $x$  y la altura en  $z$  sacamos la hipotenusa el cual va ser el vector entre el enlace 2 Y 3.

$$h = \sqrt{x_1^2 - z^2} \tag{4.5}$$

Con el vector resultante y las distancias de los enlaces  $L_2$  y  $L_3$  se aplica ley de cosenos para despejar los ángulos de  $G_2$  y  $G_3$

$$G_3 = \text{Cos}^{-1}\left(\frac{x_1^2 + z^2 - L_2^2 - L_3^2}{-2L_2L_3}\right) - \pi \tag{4.6}$$

$$\beta = \text{tan}^{-1}\left(\frac{z}{x_1}\right) \tag{4.7}$$

Donde beta es el angulo que se forma entre la razón de la suma vectorial de  $L_2$ ,  $L_3$  y el eje horizontal  $x$ .

$$G_2 = \beta + \text{Cos}^{-1}\left(\frac{x_1^2 + z^2 + L_2^2 - L_3^2}{2L_2h}\right) \tag{4.8}$$

$$G_4 = -(G_2 + G_3) \tag{4.9}$$

```
%% CINEMATICA INVERSA
x=0.450                                %%posicion final en 1x
zmax=sqrt((L2+L3-0.0001)^2-(sqrt(x^2+L5^2)-L4)^2) %%altura maxima
z=0.450                                %%posicion inicial en z
zfin2=-z;                               %%posicion final en z
zfin3=z;                                %%posicion inicial en z
r=sqrt(x^2+L5^2)
x1=r-L4;
G1=atan(L5/x)
h=sqrt(x1^2+z^2)                        %%vector pocision de 2-4
error=(L2+L3)-sqrt((sqrt(x^2+L5^2)-L4)^2+z^2) %%error de rango de pocision
```

Figura 46. cinemática inversa en MATLAB y función de error de posición

## 4.7 Cinemática diferencial (Jacobiano)

La cinemática diferencial relaciona el cambio de la posición con el paso del tiempo en el brazo robot, se demostró como se propagan las velocidades en el manipulador relacionándolas entre sí por medio de jacobiano. Tomamos la cinemática directa obtenida en la sección anterior que está en términos de las variables conjuntas (theta) con respecto al tiempo. Tomando las componentes traslacionales y rotacionales de cada articulación se pueden tomar sus derivadas y así obtener el jacobiano de un manipulador con las ecuaciones 3.22 al 3.27 y operando en MATLAB se obtuvieron funciones que quedaron de la siguiente manera, figura 47.

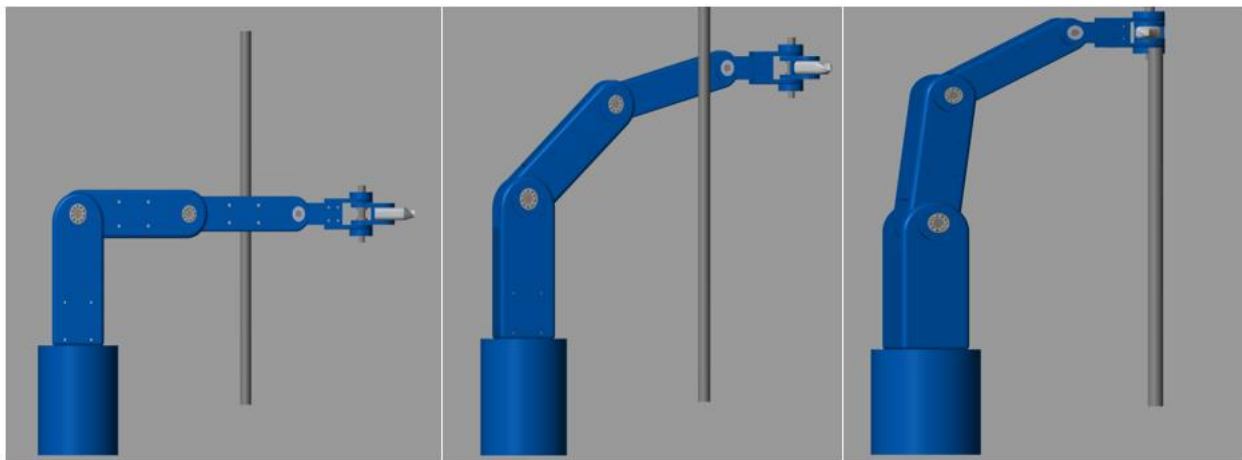
```
%% JACOBIANO O PROPAGACION DE VELOCIDADES
%%Vector de velocidades thp
Z0=[0;0;1]
P0=[0;0;0]
Z1=T01(1:3,3)
P1=T01(1:3,4)
Z2=T02(1:3,3)
P2=T02(1:3,4)
Z3=T03(1:3,3)
P3=T03(1:3,4)
Z4=T04(1:3,3)
P4=T04(1:3,4)
P5=T05(1:3,4)
J10=[diff(A01(1:3,4),th1)]
jaco10=simplify([J10;Z0])
J12=[diff(A12(1:3,4),th2)]
jaco20=simplify([diff(T02(1:3,4),th1) diff(T02(1:3,4),th2);Z0 Z1])
J23=[diff(A23(1:3,4),th3)]
jaco30=simplify([diff(T03(1:3,4),th1) diff(T03(1:3,4),th2) diff(T03(1:3,4),th3);Z0 Z1 Z2])
J34=[diff(A34(1:3,4),th4)]
jaco40=simplify([diff(T04(1:3,4),th1) diff(T04(1:3,4),th2) diff(T04(1:3,4),th3) diff(T04(1:3,4),th4);Z0 Z1 Z2 Z3])
J45=[diff(A45(1:3,4),th5)]
JACOBIANO50=simplify([diff(T05(1:3,4),th1) diff(T05(1:3,4),th2) diff(T05(1:3,4),th3) diff(T05(1:3,4),th4) diff(T05(1:3,4),th5);Z0 Z1 Z2 Z3 Z4])
```

Figura 47. operación del Jacobiano por medio de derivadas parciales [fuente propia]

Donde las matrices de las variables  $Z$  representan las componentes rotacionales sobre el eje  $z$  en cada una de las matrices de transformación homogénea, las matrices de las variables  $P$  son las componentes traslacionales de la matriz de transformación homogénea de cada articulación, las matrices de las variables  $J$ , representan las derivadas parciales de las componentes traslacionales de las matrices de transformación homogénea con respecto a cada variable conjunta y las variables “jaco” son el jacobiano de  $(6 \times n)$  de cada articulación con respecto al origen.

#### 4.8 Planeación de trayectorias

En la etapa de planeación, en la primera trayectoria, la cual sería la de posicionarse desde un punto inicial en cero hasta el punto superior del objetivo a pintar, figura 48, se tuvieron ya definidos los ángulos para determinar la posición de cada articulación con base en la posición y orientación del efector final, proporcionados por el análisis de la cinemática inversa, que en el código realizado corresponden a las posiciones finales de cada articulación. La posición inicial se definió como cero al estar el robot en su posición de reposo, igual que la velocidad y aceleración, el tiempo que es definido para la trayectoria, en este caso nosotros hemos definido un tiempo de cinco segundos para posicionarse. Ya con estos datos se logró obtener a través de los splines cúbicos, posiciones, velocidades y aceleraciones con respecto al tiempo con las ecuaciones ya parametrizadas (3.28) a (3.32), como se muestra en la figura 49.



*Figura 48. Posicionamiento del brazo para realización de la primera trayectoria de aplicación de pintura.*



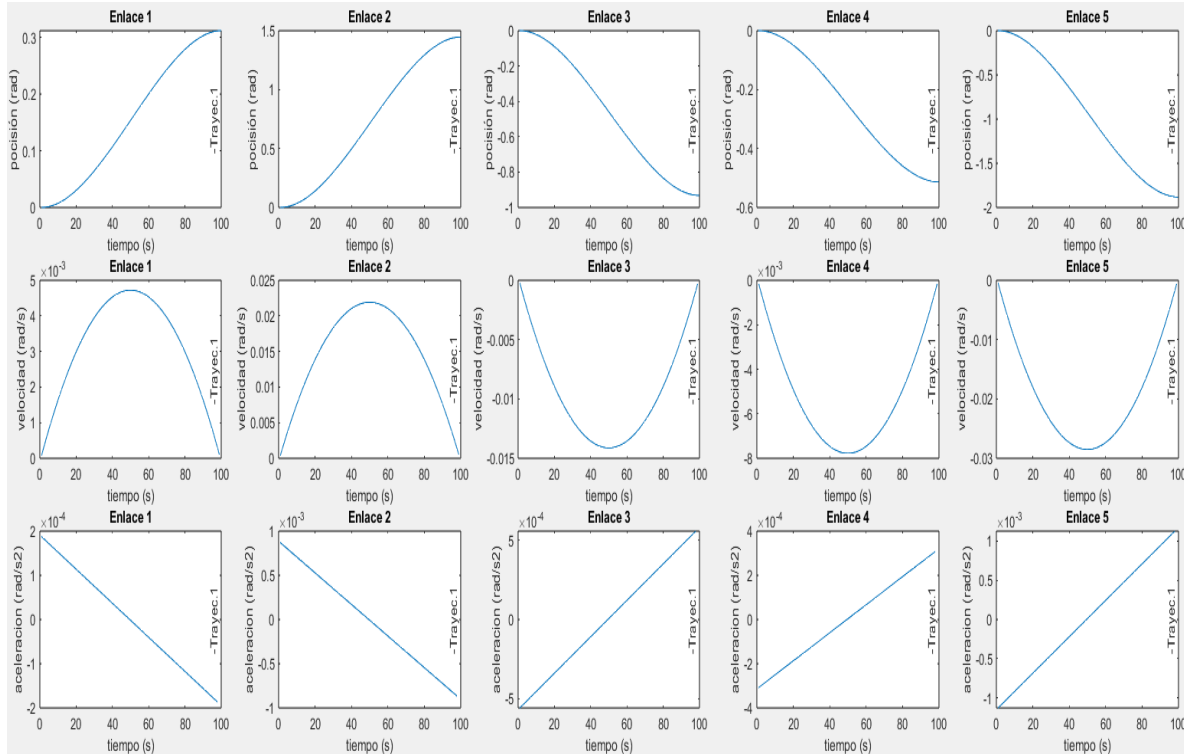


Figura 49. Posiciones, velocidades y aceleraciones de cada articulación, para la ubicación a pinar del brazo.

Los códigos realizados en MATLAB para esta planeación de cada articulación se muestran en la figura 50 a la figura 54. En donde se definieron los tiempos para cada planeación con las variables  $t_{fin1}$ ,  $t_{fin2}$ ,  $t_{fin3}$  y  $t_{fin4}$  y se implementa la técnica de los splines cúbicos para obtener posiciones, velocidades y aceleraciones en función del tiempo.

```

%% Planeacion con cinematica inversa y splines cubicos
%% tiempo inicial
tfin1=5
tfin2=10
tfin3=5
tfin4=10
%% planeacion 01 para G1
qfin=G1
a01=0 %%posicion inicial
a11=0 %%velocidad inicial
a21=3*(qfin-a01)/tfin1^2
a31=-2*(qfin-a01)/tfin1^3
t01=linspace(0,tfin1,100)'
for n=1: 100
    t=t01(n,1);
    q(n)=a01+a11*t+3*(qfin-a01)/tfin1^2*t^2+a31*t^3;
    qp(n)=a11+2*a21*t+3*a31*t^2;
    qpp(n)=2*a21+6*a31*t;
end
q1=q'
qp1=qp'
qpp1=qpp'
    
```

Figura 50. planeación 1 para la articulación 1 [fuente propia]

```

%%% planeacion 01 para G2
qfin7=G2
a02=0      %%posicion inicial
a12=0      %%velocidad inicial
a22=3*(qfin7-a02)/tfin1^2
a32=-2*(qfin7-a02)/tfin1^3
for n=1:100
    t=t01(n,1);
    q21(n)=a02+a12*t+3*(qfin7-a02)/tfin1^2*t^2+a32*t^3;
    qp21(n)=a12+2*a22*t+3*a32*t^2;
    qpp21(n)=2*a22+6*a32*t;
end
q2_1=q21'
qp2_1=qp21'
qpp2_1=qpp21'

```

Figura 51. planeación 1 para la articulación 2 [fuente propia]

```

%%% planeacion 01 para G3
qfin8=G3
a05=0      %%posicion inicial
a15=0      %%velocidad inicial
a25=3*(qfin8-a05)/tfin1^2
a35=-2*(qfin8-a05)/tfin1^3
for n=1:100
    t=t01(n,1);
    q31(n)=a05+a15*t+3*(qfin8-a05)/tfin1^2*t^2+a35*t^3;
    qp31(n)=a15+2*a25*t+3*a35*t^2;
    qpp31(n)=2*a25+6*a35*t;
end
q3_1=q31'
qp3_1=qp31'
qpp3_1=qpp31'

```

Figura 52. planeación 1 para la articulación 3 [fuente propia]

```

%%% planeacion 01 para G4
qfin9=G4
a04=0      %%posicion inicial
a14=0      %%velocidad inicial
a24=3*(qfin9-a04)/tfin1^2
a34=-2*(qfin9-a04)/tfin1^3
for n=1:100
    t=t01(n,1);
    q4(n)=a04+a14*t+3*(qfin9-a04)/tfin1^2*t^2+a34*t^3;
    qp4(n)=a14+2*a24*t+3*a34*t^2;
    qpp4(n)=2*a24+6*a34*t;
end
q4_1=q4'
qp4_1=qp4'
qpp4_1=qpp4'

```

Figura 53. planeación 1 para la articulación 4 [fuente propia]

```

%%% planeacion 01 para G5
qfin3=G5      %%posicion final
a05=0      %%posicion inicial
a15=qp1(100,1) %%velocidad inicial
a35=-2*(qfin3-a05)/tfin1^3
a25=3*(qfin3-a05)/tfin1^2
for n=1:100
    t=t01(n,1);
    q5(n)=a05+a15*t+3*(qfin3-a05)/tfin1^2*t^2+a35*t^3;
    qp5(n)=a15+2*a25*t+3*a35*t^2;
    qpp5(n)=2*a25+6*a35*t;
end
q5_1=q5'
qp5_1=qp5'
qpp5_1=qpp5'

```

Figura 54. planeación 1 para la articulación 5 [fuente propia]

Para la segunda trayectoria figura 55, se tomó la posición del robot cuando está en la parte superior del tubo orientado de tal forma para iniciar el proceso de pintura, en este punto inicia la segunda planeación correspondiente a la primera simulación de la aplicación de la pintura, el robot va de la parte superior de la pieza a la parte más baja. Las articulaciones uno y cinco en esta trayectoria permanecen en reposo por condiciones de posición de trayectoria.

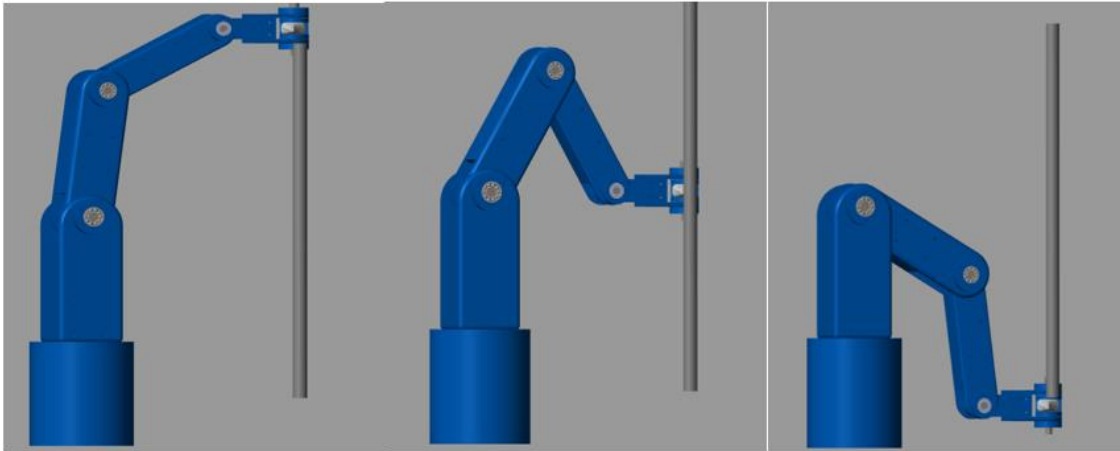


Figura 55 Posición inicial, intermedia y final, segunda trayectoria [fuente propia]

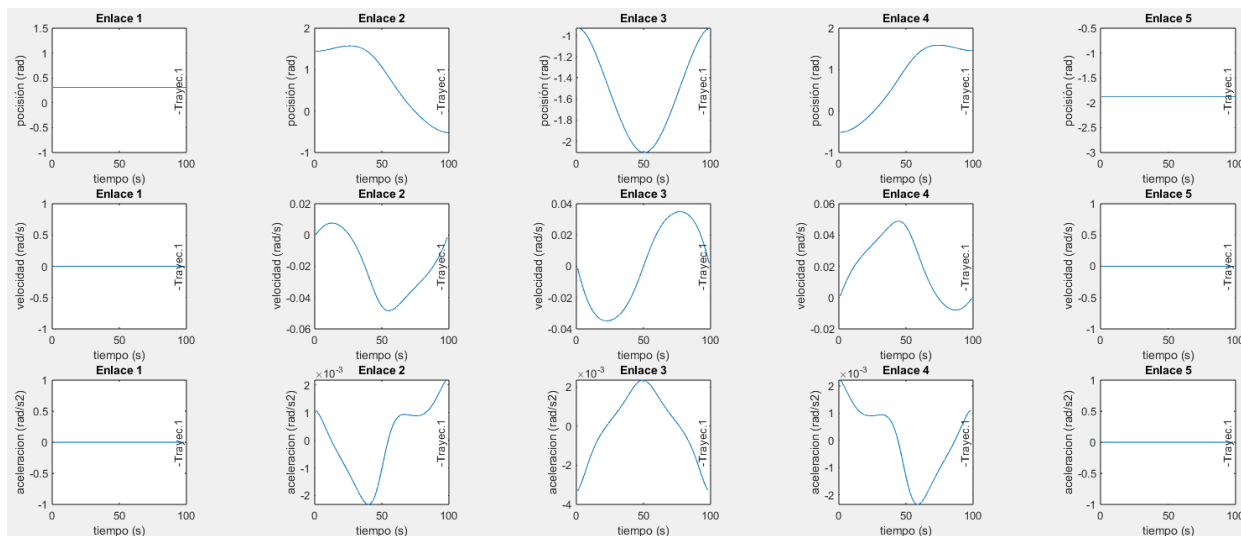


Figura 56 Graficas de posición, velocidad y aceleración de cada articulación para la figura 55. [fuente propia]

el código para esta planeación se muestra a continuación:

```

%%% planeacion 02 para G1
qfin2=q1(100)                %%posicion final
a01=q1(100)                  %%posicion inicial
a11=qp1(100)                 %%velocidad inicial
a31=-2*(qfin2-a01)/tfin2^3
a21=3*(qfin2-a01)/tfin2^2
t02=linspace(0,tfin2,100)'
for n=1:100
    t2=t02(n,1);
    q12(n)=a01+a11*t2+3*(qfin2-a01)/tfin2^2*t2^2+a31*t2^3;
    qp12(n)=a11+2*a21*t2+3*a31*t2^2;
    qpp12(n)=2*a21+6*a31*t2;
end
q1_2=q12'
qp1_2=qp12'
qpp1_2=qpp12'

```

Figura 57. planeación 2 para la articulación 1 [fuente propia]

```

%%% planeacion 02 para G2
tfin22=tfin2*26/100
qfin22=G22(26);                %%posicion final
a02=G22(1);                    %%posicion inicial
a12=qp2_1(100);                %%velocidad inicial
a33=-2*(qfin22-a02)/tfin22^3;
a22=3*(qfin22-a02)/tfin22^2;
for n=1:26
    t2=t02(n,1);
    q22(n)=a02+a12*t2+3*(qfin22-a02)/tfin22^2*t2^2+a33*t2^3;
    qp22(n)=a12+2*a22*t2+3*a33*t2^2;
    qpp22(n)=2*a22+6*a33*t2;
end
q2_2=q22';
qp2_2=qp22';
qpp2_2=qpp22';

```

Figura 58. planeación 2 para la articulación 2 [fuente propia]

```

%%% planeacion 02 para G3
tfin223=tfin2*50/100
qfin33=G33(50);                %%posicion final
a03=G33(1);                    %%posicion inicial
a13=qp3_1(100);                %%velocidad inicial
a33=-2*(qfin33-a03)/tfin223^3;
a23=3*(qfin33-a03)/tfin223^2;
for n=1:50
    t2=t02(n,1);
    q3_2(n)=a03+a13*t2+3*(qfin33-a03)/tfin223^2*t2^2+a33*t2^3;
    qp3_2(n)=a13+2*a23*t2+3*a33*t2^2;
    qpp3_2(n)=2*a23+6*a33*t2;
end
q3_2=q3_2';
qp3_2=qp3_2';
qpp3_2=qpp3_2';

```

Figura 59. planeación 2 para la articulación 3 [fuente propia]

```

%%% planeacion 02 para G4
tfin222=tfin2*74/100
qfin222=G44(74);                %%posicion final
a02=G44(1);                    %%posicion inicial
a12=qp4_1(100);                %%velocidad inicial
a33=-2*(qfin222-a02)/tfin222^3;
a22=3*(qfin222-a02)/tfin222^2;
for n=1:74
    t2=t02(n,1);
    q222(n)=a02+a12*t2+3*(qfin222-a02)/tfin222^2*t2^2+a33*t2^3;
    qp222(n)=a12+2*a22*t2+3*a33*t2^2;
    qpp222(n)=2*a22+6*a33*t2;
end
q4_21=q222';
qp4_21=qp222';
qpp4_21=qpp222';

```

Figura 60. planeación 2 para la articulación 4 [fuente propia]

```
%%% planeacion 02 para G5
qfin4=G5;          %%posicion final
a05=G5;           %%posicion inicial
a15=qp1(100,1);  %%velocidad inicial
a35=-2*(qfin4-a05)/tfin2^3;
a25=3*(qfin4-a05)/tfin2^2;
for n=1:100
    t2=t02(n,1);
    q52(n)=a05+a15*t2+3*(qfin4-a05)/tfin2^2*t2^2+a35*t2^3;
    qp52(n)=a15+2*a25*t2+3*a35*t2^2;
    qpp52(n)=2*a25+6*a35*t2;
end
q5_2=q52'
qp5_2=qp52'
qpp5_2=qpp52'
```

Figura 61. planeación 2 para la articulación 5 [fuente propia]

En la trayectoria tres, los enlaces dos, tres y cuatro permanecieron en reposo. La posición inicial de la planeación se tomó de la orientación de la posición final de la planeación anterior y la final corresponde a la ubicación del efector final a 180 grados mirando hacia la pieza. Figura 62

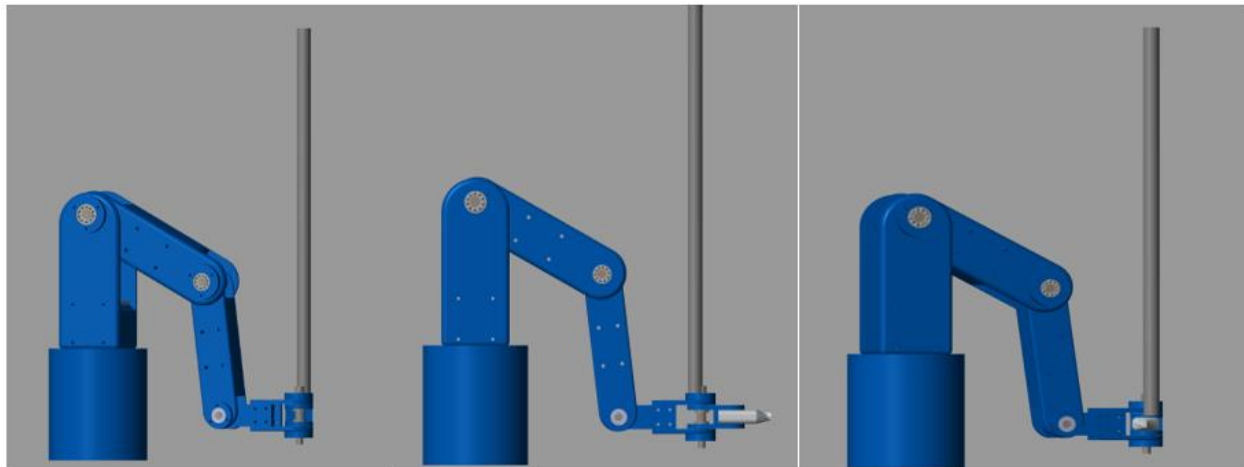


Figura 62. Posición inicial y final de tercera trayectoria [fuente propia]

Posteriormente como se esperaba las articulaciones dos, tres y cuatro se tornaron en reposo asegurando el posicionamiento eficiente del robot en el tiempo establecido, las gráficas de posición, velocidad y aceleración de las articulaciones se muestra en la figura 63 y los códigos implementados en MATLAB en la figura 64 a la figura 68.

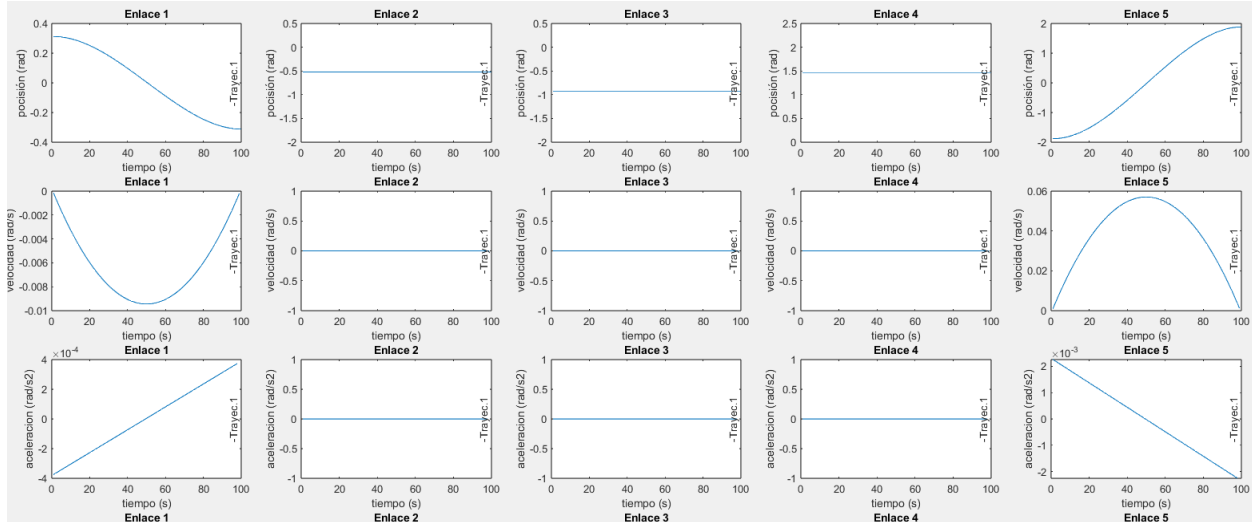


Figura 63. Graficas de posición, velocidad y aceleración de cada articulación en la tercera trayectoria. [fuente propia].

```

%%% planeacion 03 para G1
qfin1_3=-q1(100)
a01=q1(100)      %%posicion inicial
a11=0            %%velocidad inicial
a21=3*(qfin1_3-a01)/tfin3^2
a31=-2*(qfin1_3-a01)/tfin3^3
t03=linspace(0,tfin3,100)'
for n=1:100
    t3=t03(n,1);
    q13(n)=a01+a11*t3+3*(qfin1_3-a01)/tfin3^2*t3^2+a31*t3^3;
    qp13(n)=a11+2*a21*t3+3*a31*t3^2;
    qpp13(n)=2*a21+6*a31*t3;
end
q1_3=q13'
qp1_3=qp13'
qpp1_3=qpp13'
    
```

Figura 64. planeación 3 para la articulación 1 [fuente propia]

```

%%% planeacion 03 para G2
qfin10=G2(100)
a02=G2(100)      %%posicion inicial
a12=0            %%velocidad inicial
a22=3*(qfin10-a02)/tfin3^2
a32=-2*(qfin10-a02)/tfin3^3
for n=1:100
    t3=t03(n,1);
    q23(n)=a02+a12*t3+a22*t3^2+a32*t3^3;
    qp23(n)=a12+2*a22*t3+3*a32*t3^2;
    qpp23(n)=2*a22+6*a32*t3;
end
q2_3=q23'
qp2_3=qp23'
qpp2_3=qpp23'
    
```

Figura 65. planeación 3 para la articulación 2 [fuente propia]

```

%%% planeacion 03 para G3
qfin11=G3(100)
a03=G3(100)      %%posicion inicial
a13=0            %%velocidad inicial
a23=3*(qfin11-a03)/tfin3^2
a33=-2*(qfin11-a03)/tfin3^3
for n=1:100
    t3=t03(n,1);
    q33(n)=a03+a13*t3+a23*t3^2+a33*t3^3;
    qp33(n)=a13+2*a23*t3+3*a33*t3^2;
    qpp33(n)=2*a23+6*a33*t3;
end
q3_3=q33'
qp3_3=qp33'
qpp3_3=qpp33'

```

Figura 66. planeación 3 para la articulación 3 [fuente propia]

```

%%% planeacion 03 para G4
qfin12=G4(100)
a04=G4(100)      %%posicion inicial
a14=0            %%velocidad inicial
a24=3*(qfin12-a04)/tfin3^2
a34=-2*(qfin12-a04)/tfin3^3
for n=1:100
    t3=t03(n,1);
    q43(n)=a04+a14*t3+a24*t3^2+a34*t3^3;
    qp43(n)=a14+2*a24*t3+3*a34*t3^2;
    qpp43(n)=2*a24+6*a34*t3;
end
q4_3=q43'
qp4_3=qp43'
qpp4_3=qpp43'

```

Figura 67. planeación 3 para la articulación 4 [fuente propia]

```

%%% planeacion 03 para G5
qfin5_3=-q5_1(100)
a05=q5_1(100)    %%posicion inicial
a15=0            %%velocidad inicial
a25=3*(qfin5_3-a05)/tfin3^2
a35=-2*(qfin5_3-a05)/tfin3^3
for n=1:100
    t3=t03(n,1);
    q53(n)=a05+a15*t3+a25*t3^2+a35*t3^3;
    qp53(n)=a15+2*a25*t3+3*a35*t3^2;
    qpp53(n)=2*a25+6*a35*t3;
end
q5_3=q53'
qp5_3=qp53'
qpp5_3=qpp53'

```

Figura 68. planeación 3 para la articulación 5 [fuente propia]

La última planeación, correspondió a la segunda simulación de la aplicación de la pintura, en donde se pinta el tubo por el lado opuesto al de la segunda trayectoria, de abajo hacia arriba. Así, de esta manera termino el brazo en la parte superior, como se muestra en la figura 69. Los códigos en MATLAB se muestran en la figura 70 a la figura 73.

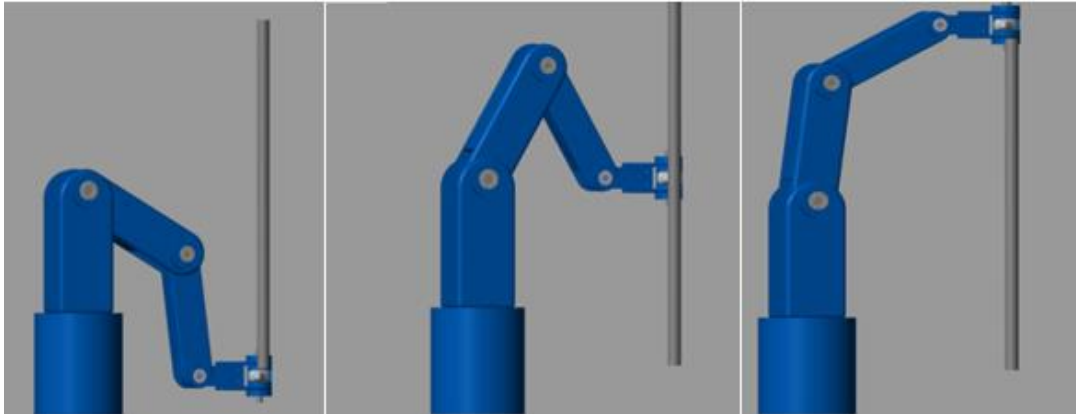


Figura 69. cuarta planeación, segunda aplicación de pintura.

```

%% planeacion 04 para G1
qfin13=q1_3(100,1)      %%posicion final
a01=q1_3(100,1)        %%posicion inicial
a11=qp1_3(100,1)       %%velocidad inicial
a31=-2*(qfin13-a01)/tfin4^3
a21=3*(qfin13-a01)/tfin4^2
t04=linspace(0,tfin4,100)'
for n=1:100
    t4=t04(n,1);
    q14(n)=a01+a11*t4+a21*t4^2+a31*t4^3;
    qp14(n)=a11+2*a21*t4+3*a31*t4^2;
    qpp14(n)=2*a21+6*a31*t4;
end
q1_4=q14'
qp1_4=qp14'
qpp1_4=qpp14'
    
```

Figura 70. planeación 4 articulación 1 [fuente propia]

```

%% planeacion 04 para G2
tfin222=tfin4*74/100
qfin222=q2_4(74);      %%posicion final
a02=q2_4(1);          %%posicion inicial
a12=qp2_3(100);       %%velocidad inicial
a33=-2*(qfin222-a02)/tfin222^3;
a22=3*(qfin222-a02)/tfin222^2;
for n=1:74
    t2=t02(n,1);
    q222(n)=a02+a12*t2+3*(qfin222-a02)/tfin222^2*t2^2+a33*t2^3;
    qp222(n)=a12+2*a22*t2+3*a33*t2^2;
    qpp222(n)=2*a22+6*a33*t2;
end
q2_41=q222';
qp2_41=qp222';
qpp2_41=qpp222';
    
```

Figura 71. planeación 4 para la articulación 2 [fuente propia]

```

%% planeacion 04 para G3
tfin223=tfin4*50/100
qfin33=q3_4(50);      %%posicion final
a03=q3_4(1);          %%posicion inicial
a13=qp3_3(100);       %%velocidad inicial
a33=-2*(qfin33-a03)/tfin223^3;
a23=3*(qfin33-a03)/tfin223^2;
for n=1:50
    t2=t02(n,1);
    q3_23(n)=a03+a13*t2+3*(qfin33-a03)/tfin223^2*t2^2+a33*t2^3;
    qp3_23(n)=a13+2*a23*t2+3*a33*t2^2;
    qpp3_23(n)=2*a23+6*a33*t2;
end
q3_41=q3_23';
qp3_41=qp3_23';
qpp3_41=qpp3_23';
    
```

Figura 72. planeación 4 para la articulación 3 [fuente propia]



```

%%% planeacion 04 para G4
tfin22=tfin2*26/100
qfin22=q4_4(26);          %%posicion final
a02=q4_4(1);             %%posicion inicial
a12=qp4_3(100);         %%velocidad inicial
a33=-2*(qfin22-a02)/tfin22^3;
a22=3*(qfin22-a02)/tfin22^2;
for n=1:26
    t2=t02(n,1);
    q22(n)=a02+a12*t2+3*(qfin22-a02)/tfin22^2*t2^2+a33*t2^3;
    qp22(n)=a12+2*a22*t2+3*a33*t2^2;
    qpp22(n)=2*a22+6*a33*t2;
end
q4_41=q22';
qp4_41=qp22';
qpp4_41=qpp22';

```

Figura 73. planeación 4 para la articulación 4 [fuente propia]

```

%%% planeacion 04 para G5
qfin14=q5_3(100,1);     %%posicion final
a05=q5_3(100,1);       %%posicion inicial
a15=qp5_3(100,1);     %%velocidad inicial
a35=-2*(qfin14-a05)/tfin4^3;
a25=3*(qfin14-a05)/tfin4^2;
for n=1:100
    t4=t04(n,1);
    q54(n)=a05+a15*t4+a25*t4^2+a35*t4^3;
    qp54(n)=a15+2*a25*t4+3*a35*t4^2;
    qpp54(n)=2*a25+6*a35*t4;
end
q5_4=q54';
qp5_4=qp54';
qpp5_4=qpp54';

```

Figura 74. planeación 4 para la articulación 5 [fuente propia]

Por medio de esta metodología de aplicación de trayectorias de punto a punto usando los polinomios cúbicos (splines cúbicos), podemos obtener las gráficas de posición, velocidad y aceleración que servirán como parámetros de entrada para la adecuación del control se muestra en la figura 75.

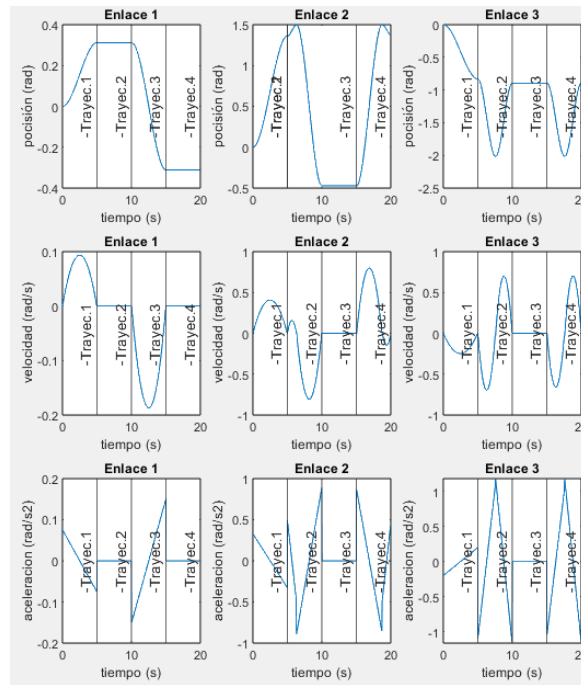


Figura 75. Posiciones, velocidades y aceleraciones de las articulaciones 1,2 y 3 [fuente propia]

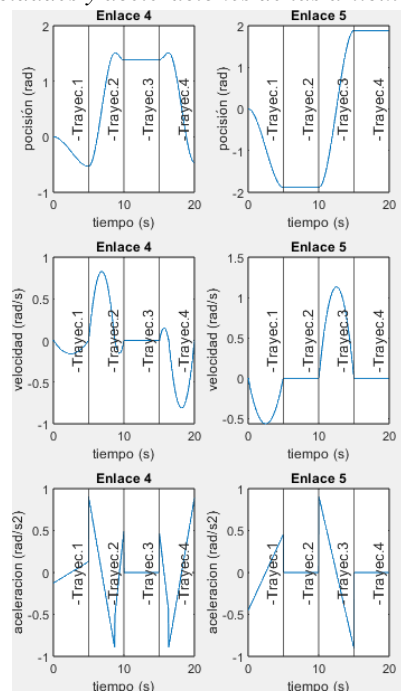


Figura 76. Posiciones, velocidades y aceleraciones de las articulaciones 4 y 5 [fuente propia]

### 4.9 Modelamiento dinámico de robot antropomorfo

Para la quinta etapa del proyecto, realizar el modelo dinámico del robot antropomorfo, se utilizó el método de Lagrange, el cual consiste en realizar el análisis sobre las energías involucradas en él moviendo. En este caso tenemos la energía cinética K y la energía potencial U. Conociendo los valores de las masas de cada enlace y los tensores de inercia, los cuales se calculan en los parámetros físicos en solidworks, jacobianos y distancias a centro de masas, se puede determinar la energía cinética y potencial de una articulación rotacional con la ecuación (3.35) y (3.36) se muestra cómo se opera con MATLAB en las siguientes figuras.

```
%% DINAMICA
%%energia cinetica
%%masa
m1=8.464755 %Kg
m2=5.997012
m3=3.12
m5=1.08
m4=1.605285
%%gravedad m/s2
g=9.81
```

Figura 77. masas de enlaces y constante de gravedad [fuente propia]

```
%%distacias a centro de masas (metros)
Lcm5=0.036017
Lcm4=0.022668
Lcm3= 0.113531
Lcm2=0.0122910
Lcm1=0.001629
```

Figura 78. distancias a centros de masas de cada enlace [fuente propia]

```
%%MATRICES DE INERCIA (metros)
I1=[0 0 1;0 -1 0;1 0 0];
I2=[1 0 0;0 1 0;0 0 1];
I3=[1 0 0;0 1 0;0 0 1];
I4=[1 0 0;0 1 0;0 0 1];
I5=[1 0 0;0 0 -1;0 1 0];
```

Figura 79. tensores de inercia [fuente propia]

```
%%ENERGIA CINETICA
k1=1/2.*m1*qp_1.*jaco10(1:3).*jqp1(1:3)*qp_1+1/2*qp_1.*jqp1(4:6).*A01(1:3,1:3)*I1*A01(1:3,1:3).*jaco10(4:6)*qp_1
k2=1/2*m2*qp_2.*jaco20(1:3,1:2).*jaco20(1:3,1:2)*qp_2+1/2*qp_2.*jaco20(4:6,1:2).*A12(1:3,1:3)*I2*A12(1:3,1:3).*jaco20(4:6,1:2)*qp_2
k3=1/2*m3*qp_3.*jaco30(1:3,1:3).*jaco30(1:3,1:3)*qp_3+1/2*qp_3.*jaco30(4:6,1:3).*A23(1:3,1:3)*I3*A23(1:3,1:3).*jaco30(4:6,1:3)*qp_3
k4=1/2*m4*qp_4.*jaco40(1:3,1:4).*jaco40(1:3,1:4)*qp_4+1/2*qp_4.*jaco40(4:6,1:4).*A34(1:3,1:3)*I4*A34(1:3,1:3).*jaco40(4:6,1:4)*qp_4
k5=1/2*m5*qp_5.*JACOBIAN050(1:3,1:5).*JACOBIAN050(1:3,1:5)*qp_5+1/2*qp_5.*JACOBIAN050(4:6,1:5).*A45(1:3,1:3)*I5*A45(1:3,1:3).*JACOBIAN050(4:6,1:5)*qp_5
```

figura 80. operación de la energía cinética del manipulador con la ecuación (3.35) [fuente propia]

```
%%Energia potencial
U1=m1*g*T01(3,4);
U2=m2*g*T02(3,4);
U3=m3*g*T03(3,4);
U4=m4*g*T04(3,4);
U5=m5*g*T05(3,4);
%%Lagrangeano cita
```

Figura 81. operación de la energía potencial del manipulador con la ecuación (3.36) [fuente propia]

El método para encontrar la ecuación de movimiento que modele los más cerca posible el comportamiento dinámico del robot es el lagrangeano y consiste en aplicar la ecuación (2.36) y la aplicación en código se muestra en la figura 82.

```
%%Lagrangeano cita
L=(k1+k2+k3+k4+k5)-(U1+U2+U3+U4+U5);
```

Figura 82. ecuación lagrangiana [fuente propia]

Ahora, tomando las derivadas parciales con respecto a posición angular, velocidad angular y con respecto al tiempo de la ecuación lagrangiana, obtenemos los toques que representas las acciones de las fuerzas generalizadas sobre el robot, como se muestra en las siguientes figuras.

```
%%Derivada con respecto a pocisiones
DL1=diff(L,th1);
DL2=diff(L,th2);
DL3=diff(L,th3);
DL4=diff(L,th4);
DL5=diff(L,th5);
```

Figura 83. derivada del lagrangeano respecto a posiciones angulares [fuente propia]

```
%%Derivada con respecto a velocidades
DLP1=diff(L,th1p);
DLP2=diff(L,th2p);
DLP3=diff(L,th3p);
DLP4=diff(L,th4p);
DLP5=diff(L,th5p);
```

Figura 84. derivada del lagrangeano respecto a velocidades angulares [fuente propia]

```
%%Derivada con respecto al tiempo
Lt11=subs(DLP1,th1pp);
Lt2=diff(DLP2)
Lt22=subs(Lt2,th2pp);
Lt3=diff(DLP3)
Lt33=subs(Lt3,th3pp);
Lt4=diff(DLP4)
Lt44=subs(Lt4,th4pp);
Lt5=diff(DLP5)
Lt55=subs(Lt5,th5pp);
```

Figura 85. derivadas respecto al tiempo [fuente propia]

Simplificando se reemplazan los términos en la ecuación (3.36) se obtienen los torques generalizados sobre cada articulación y la ecuación de movimiento de la ecuación (3.41) queda.

```
%%torques generalizados cita
cita1=simplify(Lt11-DL1)
cita2=simplify(Lt22-DL2)
cita3=simplify(Lt33-DL3)
cita4=simplify(Lt44-DL4)
cita5=simplify(Lt55-DL5)
```

Figura 86. ecuación de movimiento para cada articulación [fuente propia]

#### 4.10 Acción de control PD con alimentación anticipada

Se plantea la ley de control PD con precompensación de la ecuación (3.42) en lazo cerrado por medio de un diagrama de bloques en Simulink para llevar a cabo la simulación del control, en donde se toma como referencia las variables de los splines cúbicos y retroalimentado con las posiciones y velocidades reales corregidas con ganancias de posición y velocidad respectivamente. Se tiene como alimentación anticipada al torque calculado con la ecuación de Lagrange y se plantea como una función en MATLAB como parámetro de entrada al control como se muestra en la figura 87.

El termino de alimentación anticipada es anteponer el control ante un comportamiento ya establecido el cual se regula por medio de unas ganancias basadas en pre-alimentación como se muestra en la figura 86, el cual representa el diagrama de bloques del sistema de control en lazo cerrado para la ecuación 3.43, luego a continuación se explican cada uno de los bloques y que función cumplen en el lazo de control.

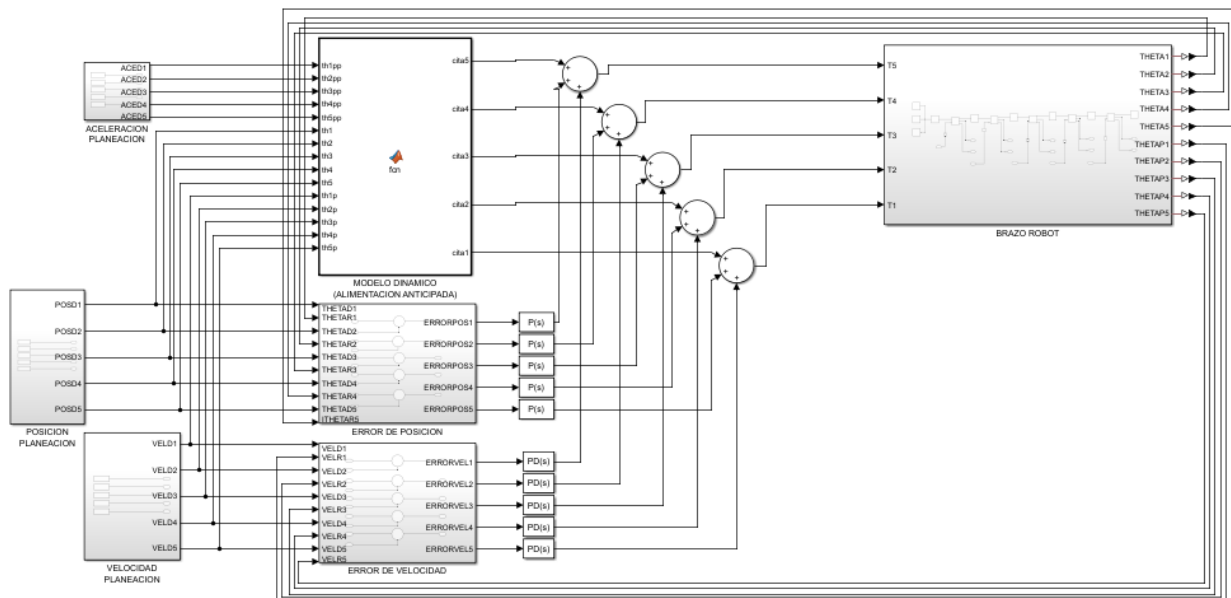
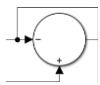




Figura 87. Diagrama de control PD con precompensación desacoplado y unificado [fuente propia]

En este caso, las variables  $Q$ ,  $Q_p$ ,  $Q_{pp}$ , representan respectivamente las posiciones, velocidades y aceleraciones generadas por los splines cúbicos para cada una de las articulaciones. Estas variables se utilizan como entrada para el control de pre-alimentación por medio de los subsistemas (POSICION PLANEACION), (VELOCIDAD PLANEACION) y (ACELERACION PLANEACION).

los símbolos  son los convertidores de variables físicas a variables en Simulink.



es el comparador de posición con la señal de referencia y de retroalimentación y la cual su

salida le aplicamos las ganancias  proporcional y  derivativa a la señal de salida del comparador de velocidad angular respectivamente.

En la siguiente figura el subsistema representa la parte del brazo robot, donde la retroalimentación continua permite ajustar y corregir el comportamiento del sistema en tiempo real representadas por las variables THETA y THETAP, posiciones y velocidades respectivamente. La eficacia del control dependerá de la precisión de la modelización de la planta y la calidad del algoritmo de control utilizado.

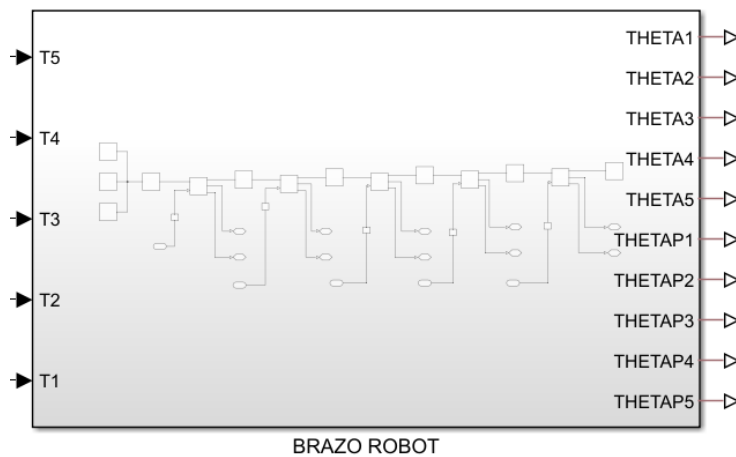


Figura 88. subsistema del brazo robot con entradas y salidas. [fuente propia]

La exportación del brazo robot de solidworks a Simscape de MATLAB se muestra en la figura 89, reduciendo los diferentes marcos de referencia de los demás elementos como, rodamientos, bujes, ejes, chapetas que componen en sistema del brazo robot.

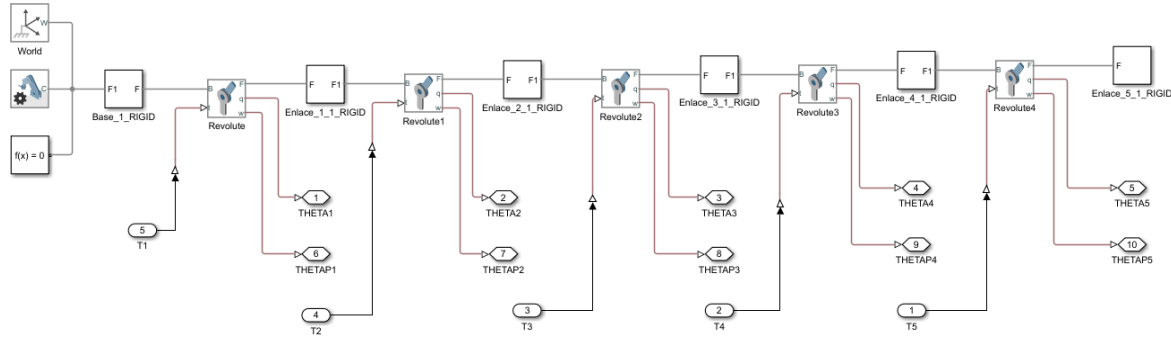


Figura 89. modelo robot exportado de solidworks

Para la alimentación anticipada, es decir, el modelo dinámico calculado con el lagrangeano, se realiza una función en MATLAB donde los parámetros de entrada son las posiciones, velocidades y aceleraciones obtenidas en los splines cúbicos como se muestra en la figura 90.

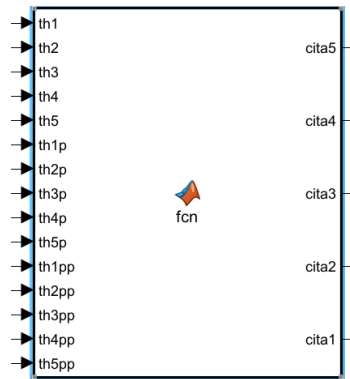


Figura 90. función de torques para la acción de precompensación [fuente propia]

Para la sintonización del control se hizo uso de las ecuaciones 3.50 y 3.51 basándonos en un tiempo de respuesta aceptable para el sistema, se asumió el factor de amortiguamiento  $\xi = 1$  para obtener un sistema críticamente amortiguado y un tiempo de respuesta de 0.1 segundos. Luego con la ecuación 3.51 se halló la frecuencia natural  $\omega_n = 10 \frac{rad}{s}$  y con lo cual se obtienen las ganancias a partir de la ecuación 3.50 se obtiene  $K_p = \omega_n^2 = 2500$  y la ganancia  $K_v = 2\sqrt{K_p} = 2\sqrt{2500} = 100$ , con lo que obtuvo la gráfica de error para la articulación uno en la figura 91.

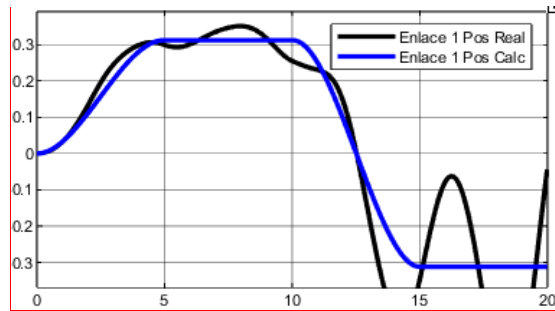


Figura 91. Error posición deseada en negro y posición real controlada de azul articulación 1 [fuente propia]

como lo que se requiere es una respuesta más rápida de 50 milisegundos y sabemos nuestro modelo dinámico que nos permite variar la frecuencia angular ahora a  $\omega_n = 100 \frac{rad}{s}$ . Con la ecuación (3.50) calculamos los valores de la ganancia proporcional y derivativa relacionadas al error de posición y velocidad respectivamente. Con lo que se obtuvieron las siguientes gráficas de error de posición dadas por los valores de  $K_p$  y  $K_v$ , en lo que se puede ver las gráficas de una respuesta de 0.1 segundos y una respuesta de 0.050 segundos, con las ganancias previamente calculadas en base al tiempo de respuesta que en este caso 2 es más que suficiente para estabilizar el sistema como se muestra en la figura 92.

se puede observar que el error de posición es menos significativo. Sin embargo, las gráficas con las ganancias de  $K_p = 10000$  y  $K_v = 200$  calculadas se obtiene un sistema controlado cumpliendo con las especificaciones de la trayectoria a seguir.

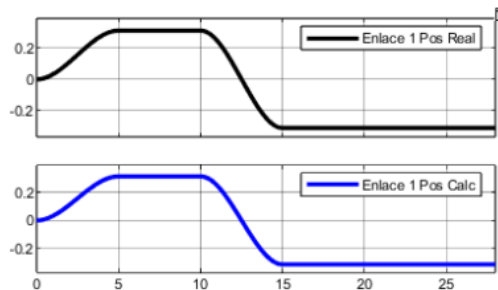


Figura 92. Error de posición angular en función del tiempo, articulación 1. [fuente propia]



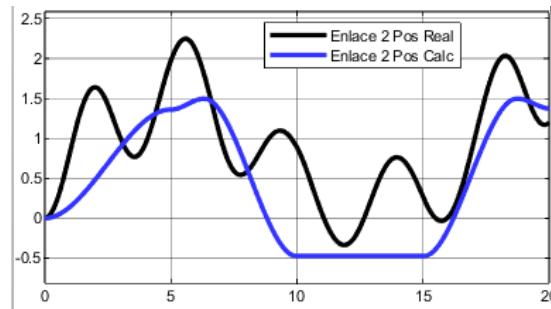


Figura 93. error de posición en la articulación 2. [fuente propia]

En la figura 93 se ve que la señal real de posición de la articulación 2 tiene bastante error aun con ganancias proporcionales igual a  $K_p = 2500$  y  $K_v = 100$ . calculadas con el mismo método que con la articulación 1.

Si realizamos ahora la prueba con un tiempo de respuesta igual a 50 milisegundos la gráfica del error de la posición dos medidas con la posición dos deseada es:

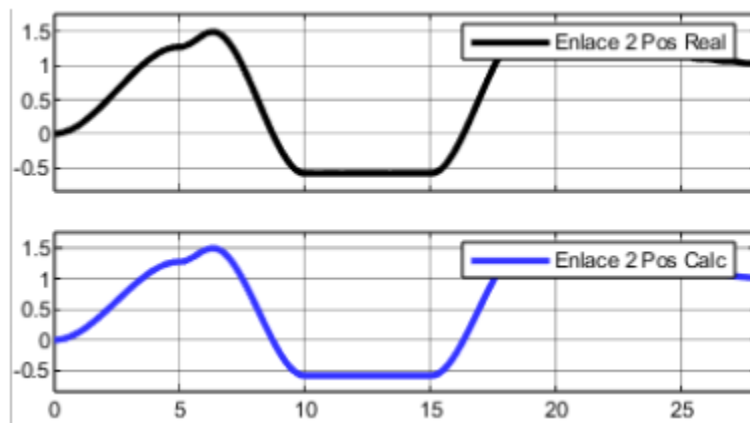


Figura 94 señal de error de posición angular vs tiempo, corregida con nuevas ganancias  $K_p = 10000$  y  $K_v = 200$ . [fuente propia]

En esta figura se ve la señal de error azul de posición más corregida y se acerca más a la deseada en negra, con menos error y atenuada para la articulación dos.

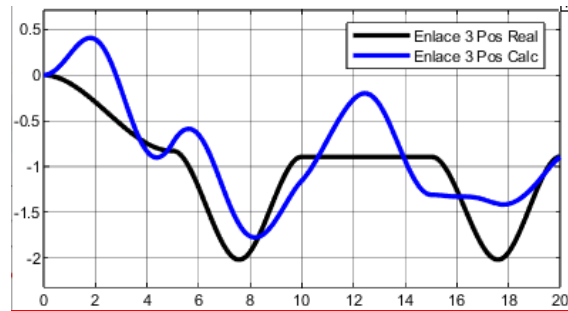


Figura 95. señal de error de posición en la articulación 3. [fuente propia]

En la figura 95 la trayectoria de la articulación 3 con ganancias  $K_p = 2500$  y  $K_v = 100$ , se comporta bien, solo hace falta atenuar un poco más la señal, eso lo logramos colocando las ganancias calculadas con la ecuación 3.50 para la articulación 1  $K_p = 10000$  y  $K_v = 200$ , y se obtiene la siguiente grafica de error para la articulación número 3.

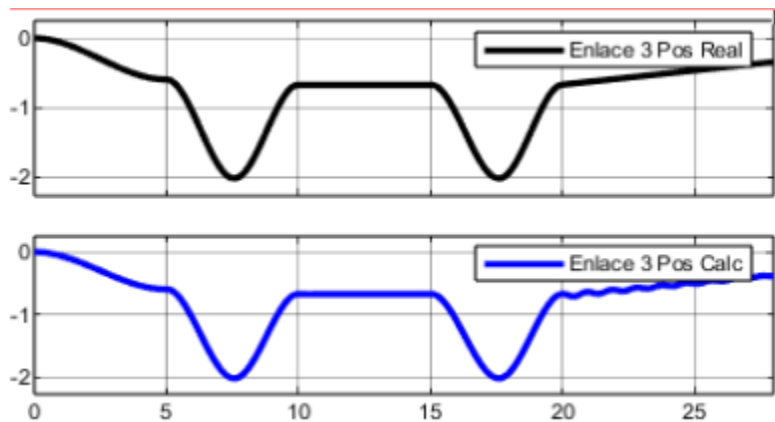


Figura 96. señal de error de posición para la articulación 3 con nuevas ganancias  $K_p$  y  $K_v$ . [fuente propia]

En la figura 97 sintonizamos el control para la articulación 4 con ganancias  $K_p = 2500$  y  $K_v = 100$  se obtiene la gráfica de error siguiente con un tiempo de respuesta de 0.1 segundos.

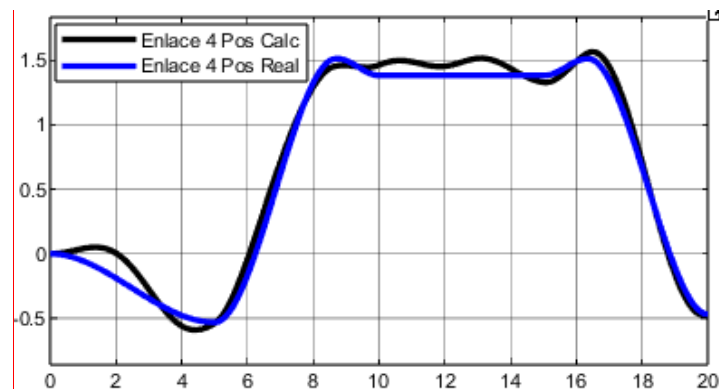


Figura 97. Error de posición para la articulación 4. [fuente propia]

si continuamos con un tiempo de respuesta de 50 milisegundos y con las ganancias calculadas se obtiene la siguiente grafica de error de posición la articulación número 4.

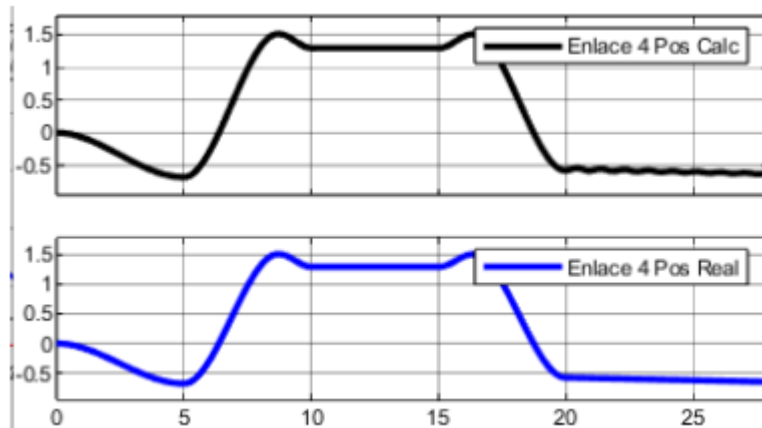


Figura 98. Grafica de error de posición vs tiempo. Sintonización para la articulación 4. [fuente propia]

En la cual se obtiene un error mínimo siguiendo la metodología de la ecuación (3.50). Para la sintonización del control de la articulación 5 se plantea la ganancia proporcional en  $K_p = 2500$  y  $K_v = 100$  con un tiempo de respuesta de 0.1 segundos y se obtiene la siguiente respuesta.

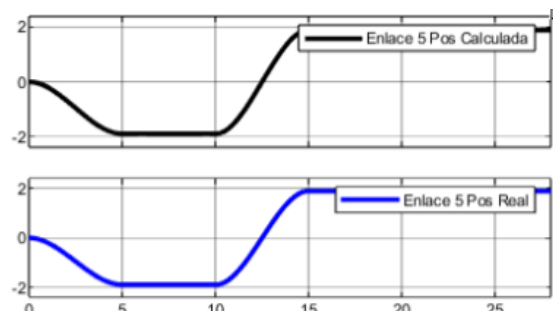


Figura 99. grafica de error de posición en la articulación 5. [fuente propia]

Es más que suficiente para mantener el sistema estable cada una de las ganancias calculadas anteriormente por medio de la metodología de la implementación de un control con alimentación anticipada.

El enlace para visualizar la simulación realizada es el siguiente.

[https://clipchamp.com/watch/oBnTsbqrKyN?fbclid=IwAR2e1P3Yui2uOpmNruCUi9Bhs6\\_\\_gYSWZyozbBrXD8008AQ6ek1in3Ud-0c](https://clipchamp.com/watch/oBnTsbqrKyN?fbclid=IwAR2e1P3Yui2uOpmNruCUi9Bhs6__gYSWZyozbBrXD8008AQ6ek1in3Ud-0c)

## 5 Análisis de resultados

- ✓ Con el diseño CAD de la morfología robot antropomorfo de 5 DOF, fue fundamental para el cálculo a través de MATLAB de los parámetros D-H así como para la cinemática total del sistema.
- ✓ Con la migración del diseño CAD a Simscape de MATLAB se pudo observar que no eran coherentes las posiciones del efector final de acuerdo al algoritmo de posicionamiento, esto debido a que si los marcos de referencia con los que el robot viene desde el software CAD no coinciden con los parámetros D-H, hay que hacer una reubicación de los marcos de referencia para que coincidan los parámetros D-H.
- ✓ Debido a que las trayectorias de pintura debían ser líneas rectas, la estrategia que se utilizó para realizar la planeación de las trayectorias fue una de las más óptimas, ya que los splines cúbicos siguen una trayectoria en línea recta de un punto inicial a un punto final y la realiza de forma suave como se observa en la figura 11, 12 y 13. De este modelamiento se calcula derivando con respecto al tiempo las velocidades y aceleración de toda la planeación. Esto se calcula para que sea la referencia en el control.
- ✓ Uno de los controles considerados más robustos son los controles basados en el modelo dinámico, ya que involucra las fuerzas aplicadas como, fuerza de gravedad, fuerza centrífuga y de coriolis, y fuerzas debido a las inercias de las masas de los enlaces. Por esta razón se implementó un control con alimentación anticipada para controlar de manera formidable la posición del efector final en toda la trayectoria con un tiempo de respuesta de 50 ms obteniendo una respuesta críticamente amortiguada acoplándose muy bien a las especificaciones del control.
- ✓ Los resultados obtenidos en la simulación durante la trayectoria, demostraron que el error de posición en cada una de las articulaciones, era mínimo y se podría considerar como un control viable, adaptándose el control PD con pre-compensación muy bien a las especificaciones de control y basándonos en el tiempo de respuesta y el modelo dinámico como alimentación anticipada.

## 6 Conclusiones

1. Se realizó el modelo 3D en el software solidworks con el fin de obtener datos como, longitud de los enlaces, longitud a centros de masas y momentos de inercia, a partir de la simulación de material que se adapte a las especificaciones de diseño.
2. Se hizo la migración del prototipo robot desde SolidWorks a Simscape de MATLAB teniendo en cuenta y precisando la coincidencia de los marcos de referencia del modelo 3D con los de los parámetros D – H. con esto se obtiene una ventaja y es tener un modelo a escala preciso en SolidWorks y como resultado de la importación a Simscape se tiene una modelo exactamente igual al diseño CAD con el que se puede trabajar como herramienta para la visualización y manipulación de una simulación de un control en MATLAB.
3. Se desarrolló la estrategia de control PD con alimentación anticipada, la cual se anticipa un modelo dinámico como entrada de control en el que se pueden tener en cuenta varias fuerzas externas como fricciones debido a los actuadores y como resultado tener un control más adaptable.
4. Se evaluó el error de posición durante toda la simulación, suponiendo una respuesta críticamente amortiguada para la pronta respuesta del control manteniendo estable la posición del efector final durante toda la trayectoria y basándonos en la evaluación del tiempo de respuesta igual a 68 milisegundos, asumiendo lo valores de la frecuencia natural en 74 rad/seg y un factor de amortiguamiento igual a uno. Esto es posible por la flexibilidad del control al contar con la dinámica del robot como alimentación anticipada.
5. La simulación se presenta como una herramienta integral para analizar, optimizar y validar el sistema antes de llevar a cabo la implementación física. En donde se evalúa el error de la posición de las articulaciones en función del tiempo de respuesta del control. Esto puede ahorrar tiempo y recursos, al tiempo que garantiza un proceso más eficiente y seguro.

## 7 Referencias

- [1] I. S. Departament, «IFR INTERNACIONAL FEDERATION ROBOTICS,» 2022. [En línea]. Available: <https://ifr.org/free-downloads>.
- [2] IFR statistical departament, «World Robotics Industrial Robots,» 2022. [En línea]. Available: <https://ifr.org/free-downloads>.
- [3] R. S.A.S, «ROBOTIKA S.A.S,» 2023. [En línea]. Available: <https://robotika.com.co/>.
- [4] B. Siciliano, L. Sciavicco, L. Villani y G. Oriolo, Robotics, Berlin: Springer-Verlag, 2010.
- [5] E. A. ing. Ruiz, Control de movimiento de Robots Manipuladores con estradas acotadas, San Luis Potosí: INSTITUTO POTOSINO DE INVESTIGACION CIENTIFICA Y TECNOLOGICA, A.C., 2006.
- [6] J. F. Reyes Cortes , MATLAB aplicado a Robótica y Mecatrónica, Mexico: Alfaomega Grupo Editor , 2013.
- [7] A. Barrientos , L. F. Peñin, C. Balaguer y R. Aracil, FUNDAMENTOS DE ROBÓTICA, Madrid, España: McGraw Hill, 1997.
- [8] A. O. Baturone, ROBÓTICA Manipuladores y robots móviles, Barcelona (España): MARCOMBO, 2001.
- [9] J. Angeles , Fundamental of Robotics mechanical system: theory methods and algorithms, Texas, USA: The university of Texas at Austin , 2002.
- [10] F. H. Ramirez Leyva , Robotica 2. Modelado Cinematico de Robots, Intituto de electronica y Mecatronica , 2012.
- [11] J. J. Craig , ROBÓTICA, México: PEARSON EDUCATION , 2006.
- [12] P. A. MAMANI APAZA, Estabilidad del Sistema de Control de un Manipulador Robotico, Arequipa, Peru: Universidad Nacional de San Agustin de Arequipa , 2017.
- [13] K. Ogata, Ingenieria de control moderna, Mexico: PRENTICE-HALL HISPANOAMERICANA, S.A., 1998.

- [14 F. Torres , G. Pomares, P. Gil, S. T. Puente y R. Aracil, Robots y Sistemas Sensoriales, Madrid:  
] Prentice Hall, 2002.
- [15 A. ROBOTICS, «ALL ROBOTICS,» PORTAFOLIO ROBOTICA, 15 Agosto 2021. [En línea]. Available:  
] <https://all-robotics.com/portfolio/portafolio-robotica/>. [Último acceso: 20 Septiembre 2021].
- [16 M. V. Isaac Wladimir, ANÁLISIS DE TÉCNICAS DE CONTROL PARA ROBOTS CON ARTICULACIONES  
] FLEXIBLES, Valencia (España): Universidad Politecnica de valencia , 2018.
- [17 C. R. CUARTAS SARMIENTO y C. E. ARENAS ROMERO, DISEÑO, CONSTRUCCIÓN Y PUESTA A PUNTO  
] DE UN PROTOTIPO DE SISTEMA ELECTROMECAÁNICO CONSTRUIDO, Bogotá : Universidad Antonio  
Nariño , 2020.
- [18 A. Valera Fernandez , Modelado y control en el espacio de estados, Valencia (España): Universidad  
] Politecnica de Valencia , 2016.
- [19 J. C. BORJAS CASTAÑEDA , CONTROLABILIDAD Y OBSERVABILIDAD NO LINEAL DE UN ROBOT  
] MANIPULADOR DE DOS GRADOS DE LIBERTAD, PERU: Universidad Nacional de Callao Peru.  
Facultad de Ingenieria Electrica y Electronica , 2019.
- [20 A. Rendon Trejo, J. Rodriguez Franco y A. Morales Alquieria , INTRODUCCION AL ALGEBRA LINEAL Y  
] DE MATRICES APLICACIONES CON EXCEL, Mexico: Universidad Autonoma Metropolitana, Unidad  
Xochimilco, 1998.
- [21 Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Robot Dynamics and Control, January 28,  
] 2004.
- [22 R. N. Jazar, Theory of Applied Robotics, Australia : Springer , 2007.  
]
- [23 K. ROBOTICS, «KUKA,» 2011. [En línea]. Available: [https://www.kuka.com/es-es/sectores/base-de-  
\] datos-de-soluciones/2023/03/faw-volkswagen-foshan](https://www.kuka.com/es-es/sectores/base-de-datos-de-soluciones/2023/03/faw-volkswagen-foshan).
- [24 A. GROUP, «ABB,» 2023. [En línea]. Available: <https://new.abb.com/products/robotics/robots>.  
]
- [25 FANUC, «FANUC,» 2023. [En línea]. Available: <https://www.fanuc.eu/es/es/robots>.  
]
- [26 C. ROBOTICS, «CAV ROBOTICS,» 2023. [En línea]. Available:  
] <https://www.cavrobotics.com.co/nosotros>.
- [27 DEMPRO, «DEMPRO Importadores y Distribuidores de Robots, Equipos, Herramientas y  
] Repuestos,» 2023. [En línea]. Available:  
[https://www.dempro.co/schmersal?gclid=Cj0KCQjwi7GnBhDXARIsAFLvH4mcjTGQOyZ\\_mjl9a4eCB1  
CWws\\_aGKlx0ph-0oHOYW1YBRifhnuOMiQaAs51EALw\\_wcB](https://www.dempro.co/schmersal?gclid=Cj0KCQjwi7GnBhDXARIsAFLvH4mcjTGQOyZ_mjl9a4eCB1CWws_aGKlx0ph-0oHOYW1YBRifhnuOMiQaAs51EALw_wcB).

[28 J. A. VIEDMA, Lecciones de Geometria intuitiva, Cali: NORMA.  
]

[29 J. Kadirimangalam y Sivraj P., Teaching and Learning Robotic Arm Model, india: Amrita School of  
] Engineering, Coimbatore,, 2017.