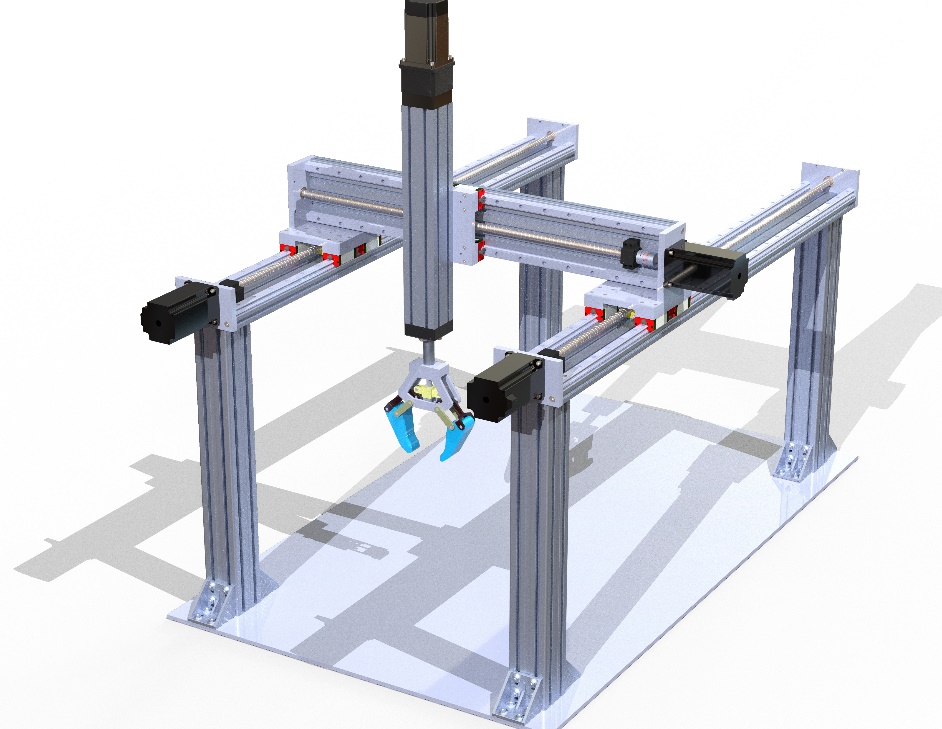
**DISEÑO DE UN ROBOT CARTESIANO**

**MEMORIA DE CÁLCULO**

**ÍNDICE**

[INTRODUCCIÓN 8](#_Toc172594679)

[CAPÍTULO I 9](#_Toc172594680)

[Planteamiento del problema 9](#_Toc172594681)

[Justificación 9](#_Toc172594682)

[Objetivos 9](#_Toc172594683)

[1.1.1. Objetivo general 9](#_Toc172594684)

[1.1.2. Objetivos específicos 9](#_Toc172594685)

[CAPÍTULO II 10](#_Toc172594686)

[2. REVISIÓN DE LITERATURA Y FUNDAMENTOS TEÓRICOS 10](#_Toc172594687)

[2.1. Generalidades de un Robot Cartesiano 10](#_Toc172594688)

[2.1.1. Configuración 11](#_Toc172594689)

[2.1.1.1. Tipos de robot 11](#_Toc172594690)

[2.1.1.2. Configuración cartesiana (PPP) 13](#_Toc172594691)

[2.1.2. Morfología del robot 14](#_Toc172594692)

[2.1.2.1. Estructura Mecánica 15](#_Toc172594693)

[2.1.2.2. Sistemas de transmisión 16](#_Toc172594694)

[2.1.2.3. Sistemas de impulsión 17](#_Toc172594695)

[2.1.2.4. Sistema sensorial 17](#_Toc172594696)

[2.1.3. Control y programación 19](#_Toc172594697)

[2.1.3.1. Software de programación: 19](#_Toc172594698)

[2.1.3.2. Paquetes de Software 19](#_Toc172594699)

[2.1.3.3. Simulación y modelado. 19](#_Toc172594700)

[2.2. Cinemática Directa 20](#_Toc172594701)

[2.2.1. Descripción de la posición 20](#_Toc172594702)

[2.2.2. Descripción de la orientación 21](#_Toc172594703)

[2.2.3. Coordenadas y matrices de transformación homogénea 23](#_Toc172594704)

[2.2.4. Parámetros Denavit – Hartenberg 23](#_Toc172594705)

[2.3. Cinemática Inversa 25](#_Toc172594706)

[2.4. Dinámica de robot 26](#_Toc172594707)

[2.4.1. Formulación Newton Euler 27](#_Toc172594708)

[2.5. Parte mecánica del robot cartesiano 29](#_Toc172594709)

[2.5.1. Selección de materiales 29](#_Toc172594710)

[2.5.2. Selección y diseño de elementos del robot cartesiano 29](#_Toc172594711)

[2.6. Normativas 30](#_Toc172594712)

[2.6.1. Normativa internacional ISO 10218: –1:2011 31](#_Toc172594713)

[2.6.2. Normativa americana ANSI/RIA R15.06–1992. 32](#_Toc172594714)

[2.6.3. Normativa americana ANSI/RIA R15.06–1999. 32](#_Toc172594715)

[2.6.4. ISO 9283:1998. Manipulación de robots. 32](#_Toc172594716)

[2.6.5. Normativas Europeas 32](#_Toc172594717)

[CAPÍTULO III 34](#_Toc172594718)

[3.1. Alternativas 34](#_Toc172594719)

[3.1.1. Alternativa 1 34](#_Toc172594720)

[3.1.2. Alternativa 2 35](#_Toc172594721)

[3.1.3. Alternativa 3. 36](#_Toc172594722)

[3.1.4. Evaluación de alternativas 36](#_Toc172594723)

[3.2. Análisis del Robot Cartesiano 37](#_Toc172594724)

[3.2.1. Análisis Cinemático 37](#_Toc172594725)

[3.2.1.1. Parámetros Denavit – Hartenberg aplicado al modelo del robot cartesiano 38](#_Toc172594726)

[3.2.1.2. Cinemática directa mediante matrices de transformación homogénea 39](#_Toc172594727)

[3.2.1.3. Matriz Jacobiana geométrica 44](#_Toc172594728)

[3.2.2. Análisis Dinámico 44](#_Toc172594729)

[3.2.3. Modelo dinámico 46](#_Toc172594730)

[3.2.4. Función de control 47](#_Toc172594731)

[3.2.5. Valores de kp y kd 50](#_Toc172594732)

[3.3. Diseño mecánico 51](#_Toc172594733)

[3.3.1. Análisis estático 51](#_Toc172594734)

[3.3.1.1. Análisis del gripper 51](#_Toc172594735)

[3.3.1.2. Trabajo necesario de elevación 52](#_Toc172594736)

[3.3.1.3. Trabajo necesario para el movimiento transversal 52](#_Toc172594737)

[3.3.2. Análisis de esfuerzos viga transversal 54](#_Toc172594738)

[3.3.2.1. Análisis con la carga en el centro 54](#_Toc172594739)

[3.3.2.2. Análisis en el extremo de la viga 57](#_Toc172594740)

[3.3.3. Análisis esfuerzos viga longitudinal 59](#_Toc172594741)

[3.3.3.1. Análisis en posición intermedia de la viga 61](#_Toc172594742)

[3.3.4. Cálculo de Tornillo de Potencia 63](#_Toc172594743)

[3.4. Componentes electrónicos 65](#_Toc172594744)

[3.4.1. Cálculo del torque del motor de pasos 65](#_Toc172594745)

[3.4.2. Descripción de los elementos de control del sistema 66](#_Toc172594746)

[3.4.2.1. Selección del controlador para el motor de pasos 66](#_Toc172594747)

[3.4.3. Descripción de los sensores del sistema 68](#_Toc172594748)

[3.4.3.1. Selección de sensores de final de carrera 68](#_Toc172594749)

[CAPÍTULO IV 71](#_Toc172594750)

[4. RESULTADOS 71](#_Toc172594751)

[4.1. Modelamiento en SOLIDWORKS 71](#_Toc172594752)

[4.2. Resultados del análisis cinemático en MATLAB 71](#_Toc172594753)

[4.3. Validación mediante software (CAE) 73](#_Toc172594754)

[4.4. Simulación ROS – Gazebo 76](#_Toc172594755)

[4.5. Costos 77](#_Toc172594756)

[CAPÍTULO V 78](#_Toc172594757)

[5. Conclusiones y recomendaciones 78](#_Toc172594758)

[5.1. Conclusiones 78](#_Toc172594759)

[5.2. Recomendaciones 78](#_Toc172594760)

[BIBLIOGRAFÍA 79](#_Toc172594761)

[ANEXOS 81](#_Toc172594762)

**ÍNDICE DE FIGURAS**

[Fig. 1. Robot Cartesiano 10](#_Toc172594804)

[Fig. 2. Mini robot cartesiano 11](#_Toc172594805)

[Fig. 3. Robots cartesianos grandes 12](#_Toc172594806)

[Fig. 4. Robots cartesianos de alta velocidad 12](#_Toc172594807)

[Fig. 5. Robots cartesianos de alta precisión 12](#_Toc172594808)

[Fig. 6. Manipulador cartesiano 13](#_Toc172594809)

[Fig. 7. Espacio de trabajo manipulador cartesiano 13](#_Toc172594810)

[Fig. 8. Configuraciones cartesianas 14](#_Toc172594811)

[Fig. 9. Configuración de robots industriales 15](#_Toc172594812)

[Fig. 10. Sensores externos 18](#_Toc172594813)

[Fig. 11. Sensores internos 18](#_Toc172594814)

[Fig. 12. Descripción de la posición 20](#_Toc172594815)

[Fig. 13. Posición de un sistema o con respecto a otro m 21](#_Toc172594816)

[Fig. 14. Sistemas de referencia coincidentes en el origen 22](#_Toc172594817)

[Fig. 15. Metodología Denavit - Hartenberg 24](#_Toc172594818)

[Fig. 16. Configuración mecánica del robot cartesiano 28](#_Toc172594819)

[Fig. 17. Perfiles de aluminio extruido 29](#_Toc172594820)

[Fig. 18. Componentes del robot cartesiano. 30](#_Toc172594821)

[Fig. 19. Perfil C de aluminio 34](#_Toc172594822)

[Fig. 20. Carrito de desplazamiento 34](#_Toc172594823)

[Fig. 21. C beam con carrito 35](#_Toc172594824)

[Fig. 22. Sistema por correa 35](#_Toc172594825)

[Fig. 23. Desplazamiento por rieles lineales 36](#_Toc172594826)

[Fig. 24. Esquema para el análisis cinemático 37](#_Toc172594827)

[Fig. 25. Longitudes consideradas 38](#_Toc172594828)

[Fig. 26. Dimensiones generales 44](#_Toc172594829)

[Fig. 27. Diagrama de análisis 51](#_Toc172594830)

[Fig. 28. Peso del gripper 52](#_Toc172594831)

[Fig. 29. Diagrama de peso transversal 53](#_Toc172594832)

[Fig. 30. Sección transversal del perfil 54](#_Toc172594833)

[Fig. 31. Viga analizada 54](#_Toc172594834)

[Fig. 32. Diagrama cortante 56](#_Toc172594835)

[Fig. 33. Análisis en el extremo de la viga 57](#_Toc172594836)

[Fig. 34. Partes de movimiento transversal 59](#_Toc172594837)

[Fig. 35. Diagrama sección transversal 59](#_Toc172594838)

[Fig. 36. Diagrama longitudinal 60](#_Toc172594839)

[Fig. 37. Ubicación del perfil 61](#_Toc172594840)

[Fig. 38. Diagrama de flexión de la viga 61](#_Toc172594841)

[Fig. 39. Gráfica de posición de articulaciones 71](#_Toc172594842)

[Fig. 40. Gráfica de velocidades de las articulaciones 72](#_Toc172594843)

[Fig. 41. Gráfica de fuerza 72](#_Toc172594844)

[Fig. 42. Material se 6063-T5 73](#_Toc172594845)

[Fig. 43. Mallado de viga 73](#_Toc172594846)

[Fig. 44. Calidad de malla 73](#_Toc172594847)

[Fig. 45. Análisis de la deformación 74](#_Toc172594848)

[Fig. 46. Análisis del factor de seguridad 74](#_Toc172594849)

[Fig. 47. Enlace de estudio 75](#_Toc172594850)

[Fig. 48. Análisis de viga longitudinal 75](#_Toc172594851)

[Fig. 49. Factor de seguridad de la viga longitudinal 76](#_Toc172594852)

[Fig. 50.Entorno de simulación de GAZEBO 76](#_Toc172594853)

**ÍNDICE DE TABLAS**

[Tabla 1. Tipos de Transmisiones 17](#_Toc172594854)

[Tabla 2. Tipos de sensores internos de los robots 18](#_Toc172594855)

[Tabla 3. Parámetros Denavit - Hartenberg 25](#_Toc172594856)

[Tabla 4. Evaluación de alternativas 37](#_Toc172594857)

[Tabla 5. Requerimientos de diseño 51](#_Toc172594858)

[Tabla 6. Tabla de pesos movimiento vertical 52](#_Toc172594859)

[Tabla 7. Tabla de pesos - movimiento transversal 53](#_Toc172594860)

[Tabla 8. Propiedades del perfil de aluminio 54](#_Toc172594861)

[Tabla 9. Resultados de la viga a flexión 55](#_Toc172594862)

[Tabla 10. Resultados análisis viga transversal 57](#_Toc172594863)

[Tabla 11. Resultados análisis viga transversal 61](#_Toc172594864)

[Tabla 12. Costos aproximados 77](#_Toc172594865)

# INTRODUCCIÓN

Las industrias en la actualidad buscan el aumento de la productividad mediante la automatización de los procesos y la reducción de costos en la mano de obra. Dentro de estas aplicaciones se encuentra el robot cartesiano que es una máquina utilizada para diversas tareas como levantamiento de pesos, mecanizado, pintura, entre otros. Estos robots poseen movimientos lineales a lo largo de los ejes X, Y, Z, y debido a esto puede realizar movimientos precisos en un espacio tridimensional. Las partes de las que está constituida el robot son la estructura mecánica, transmisiones, sistemas de accionamiento, sistema de control, entre otros.

En el presente proyecto se procederá a realizar el diseño y simulación de un robot cartesiano. El objetivo principal será diseñar y simular el robot cartesiano, por otra parte, en los objetivos específicos se realizará la definición del robot a diseñar donde se definirá los rangos de movimiento en cada uno de los ejes. También se realizará el estado del arte para obtener información sobre las simulaciones y componentes que tiene este robot. Además, se realizará el diseño mecánico de los elementos del robot cartesiano que comprenderá la selección de los componentes y el diseño de elementos principales. Para finalizar se efectuará la selección de los componentes electrónicos y de control para el robot, además de ello se efectuará la simulación del robot en donde se simulará las diversas tareas en ROS 2 y Gazebo.

Finalmente, se realizará el modelado del robot cartesiano en el software SolidWorks, en donde se dibujará cada una de las piezas mecánicas para su posterior ensamblaje con las relaciones de posición respectiva. Una vez concluido este procedimiento se procederá a la generación de los planos de cada uno de los elementos constitutivos del robot cartesiano. En el plano explosionado se mostrará cada uno de las piezas mecánicas del robot con su respectiva lista de materiales para su posterior ensamblaje, en los planos de fabricación se mostrará cada una de las medidas generales, detalles, tolerancias geométricas y superficiales que tendrán cada una de las piezas mecánicas.

# CAPÍTULO I

## Planteamiento del problema

En la industria manufacturera la precisión y la eficiencia en la manipulación de materiales es de vital importancia debido a que permitirá optimizar los procesos productivos. Los bloques de 20 Kg son utilizados para múltiples aplicaciones y su manipulación de manera manual puede ser difícil y tardía, además de ello puede ocasionar un riesgo ergonómico para los trabajadores. En ese contexto surge la necesidad de diseñar un robot cartesiano para ordenar en una fila boques de 20 kg, considerando un área de trabajo de 1250x600x600 mm.

## Justificación

El diseño de un robot cartesiano permitirá manipular los bloques de 20 Kg de una manera mas eficiente y segura, a su vez se podrá reducir los costos operativos debido a que el robot cartesiano podrá operar de manera continua y con gran precisión, además de ello ocasionará que se reduzca los errores que se ocasionaban cuando se manipulaba los bloques de manera manual.

## Objetivos

### Objetivo general

* Diseñar y simular un robot cartesiano para ordenar una fila de bloques de 20 Kg.

### Objetivos específicos

* Diseñar un robot cartesiano para un área de trabajo de 1250x600x600 mm.
* Realizar el estado del arte.
* Realizar el diseño de los elementos principales y la selección de los componentes mecánicos constitutivos del robot.
* Realizar la selección de los componentes electrónicos y de control del robot
* Construir y simular el robot cartesiano en el entorno ROS 2.

# CAPÍTULO II

# REVISIÓN DE LITERATURA Y FUNDAMENTOS TEÓRICOS

## Generalidades de un Robot Cartesiano

En el trabajo de Jara y Pacheco se menciona que los robots cartesianos son un tipo de robot de tipo industrial y que sus 3 ejes de control principales forman ángulos rectos con respecto a los otros, además de esos son lineales. Por otra parte, mediante esta configuración mecánica se puede simplificar las ecuaciones de control que se obtienen de los brazos robóticos. Otra denominación con la que se conocen a los robots cartesianos es robots pórticos, esto debido a que tiene el eje principal limitad y a su vez apoyado en los extremos, además de ello son relativamente grandes.(Jara & Pacheco, 2014)

Una de las aplicaciones en donde comúnmente se utiliza este tipo de robots es en máquinas de control numérico. Las aplicaciones más comunes son las que se utilizan en las máquinas de dibujo y fresado, donde una pluma o taladro se moviliza a lo largo de un plano x-y ,además de eso la herramienta tiene un movimiento de elevación y descenso en la superficie para obtener un diseño preciso.(Jara & Pacheco, 2014)

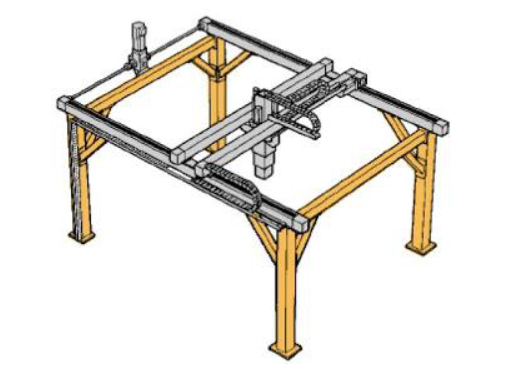


Fig. 1. Robot Cartesiano

Las características del robot cartesiano son las siguientes:

* Cuenta con una estructura con 3 articulaciones prismáticas.
* Las coordenadas (X, Y, Z) se utilizan para especificar una posición en el espacio.
* Tiene una precisión uniforme en toda la zona de operación del robot.
* Es ampliamente útil cuando se trata de seguir una trayectoria fija.
* Tiene una estructura solida donde la distribución de cargas no representa problema alguno.
* No es útil cuando se quiere acceder a puntos en espacios cerrados.(Jara & Pacheco, 2014)

### Configuración

#### Tipos de robot

Los robots cartesianos son comúnmente conocidos como robots rectilíneos, estos robots ofrecen una gran diversidad de características y formatos que son adaptadas a las distintas necesidades en el campo industrial. Estos robots son caracterizados debido a su estructura que puede tener de 2 a 6 ejes deslizantes y que se van a movilizar de manera perpendicular, por lo cual les da la posibilidad de tener una amplia variedad de movimientos y tener múltiples aplicaciones. (Castellanos, 2024)

Los robots cartesianos se pueden clasificar de acuerdo con diversas características como la capacidad de carga, velocidad, tamaño, etc. A continuación, se presentarán algunos de los robots cartesianos más usuales.(Castellanos, 2024)

* **Mini robot cartesiano:** Son diseñados para tareas que requieren una gran precisión, son compactos y además de ello son utilizados en el campo de la microelectrónica. Debido a que son de pequeño tamaño pueden operar en espacios pequeños con alto nivel de detalle.(Castellanos, 2024)

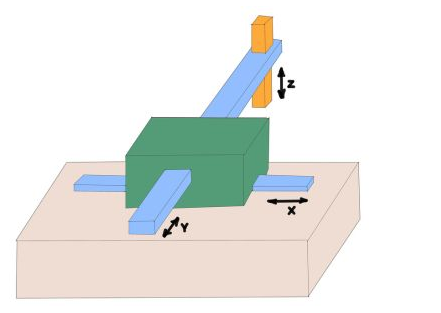


Fig. 2. Mini robot cartesiano

* **Robots cartesianos grandes:** Estos robots están diseñados para elevar cargas más pesadas a comparación de los otros robots cartesianos, son comúnmente utilizados en la manipulación de materiales en la industria Siderurgia.(Castellanos, 2024)



Fig. 3. Robots cartesianos grandes

* **Robots cartesianos de alta velocidad:** Son caracterizados por su rapidez, lo que los hacen adecuados para las aplicaciones que demanden de alta velocidad como es el caso de la peletización y el envasado. Debido a su movimientos repetitivos y rápidos permite aumentar la productividad de una manera significativa.(Castellanos, 2024)

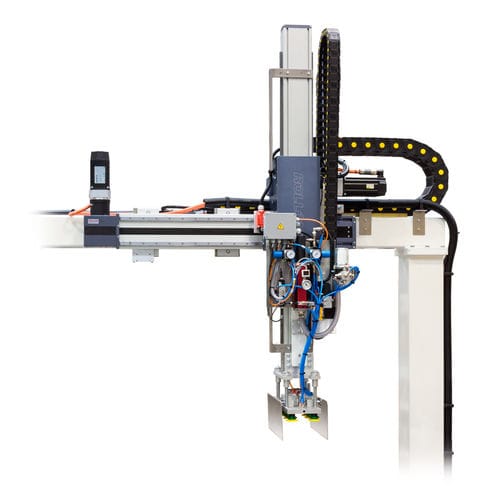


Fig. 4. Robots cartesianos de alta velocidad

* **Robots cartesianos de alta precisión:** Estos robots están caracterizados por realizar movimientos con gran precisión y son utilizados comúnmente para aplicación que necesitan de gran exactitud como es el caso de la producción de componentes ópticos. (Castellanos, 2024)



Fig. 5. Robots cartesianos de alta precisión

#### Configuración cartesiana (PPP)

Por lo general los manipuladores se encuentran dentro de una categoría geométrica: esférica (RRP), articulada (RRR), SCARA (RRP), cilíndrica (RPP) y cartesiana (PPP), cada una de ellas corresponden a robots seriales. Para el caso de estudio se mencionará los aspectos más representativos de la configuración cartesiana PPP.(Moran, 2016)

La configuración cartesiana (PPP) corresponde a un manipulador que disponga de sus tres primeras articulaciones prismáticas, donde su estructura estará representada como se indica en la Figura 6.(Moran, 2016)

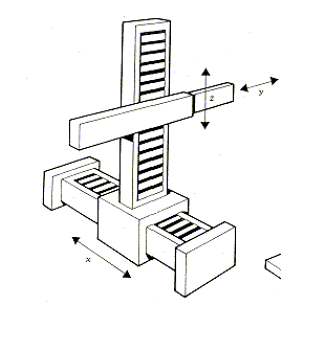


Fig. 6. Manipulador cartesiano

Para este tipo de manipulador las variables que van a contener las articulaciones corresponden a las de las coordenadas cartesianas del efector final, además de ello un uso común de este tipo de robot es en las aplicaciones de montaje de mesa. El espacio de trabajo que presenta esta configuración se muestra mediante la Figura 7.(Moran, 2016)

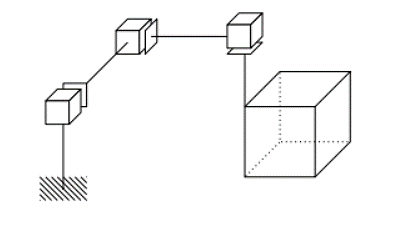


Fig. 7. Espacio de trabajo manipulador cartesiano

En la configuración cartesiana (PPP) existen diversas combinaciones o configuraciones que se basan de acuerdo a la aplicación para las cuales van a ser destinadas. A continuación, se muestran algunas de las configuraciones que puede tener un robot cartesiano.(Badillo, 2016)

* **Posicionador lineal:** Para este caso el posicionador lineal se encuentra por abajo o encima del espacio de trabajo en la dirección x/z. Una de las aplicaciones para la que está destinado este tipo de robot es para el transporte de cargas en recorridos cortos.(Badillo, 2016)
* **Robot lineal:** Este tipo de robot se encuentra cerca al espacio de trabajo y está destinado de manera específica para movilizar carga a alta velocidad en trayectos cortos.(Badillo, 2016)
* **Robot portal:** Este tipo de configuración se encuentra en el espacio de trabajo y está destinado para el transporte de cargas en trayectos largos.(Badillo, 2016)
* **Robot portal pared:** Este tipo de configuración está destinada para trabajar sobre superficies verticales.(Badillo, 2016)

A continuación, en la Figura 8 se muestra cada una de las configuraciones mencionadas anteriormente.

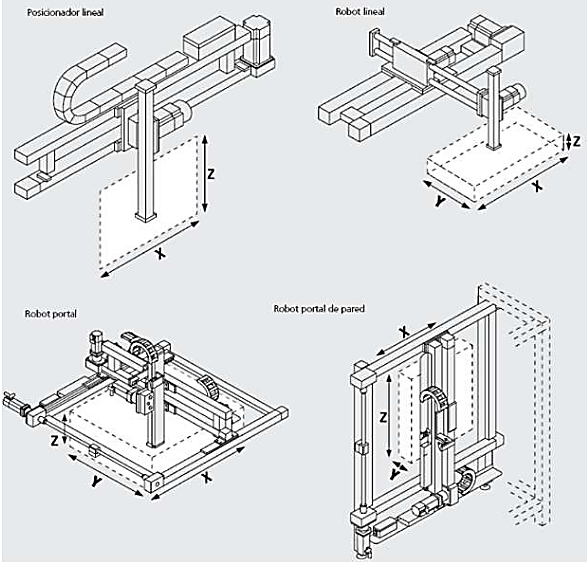


Fig. 8. Configuraciones cartesianas

### Morfología del robot

La morfología del robot hace referencia a la parte física del robot donde se describe cada una sus partes de las cuales está constituida. En el caso de la configuración de los robots industriales se asemeja a la del cuerpo humano debido a que está constituida de brazos y cuerpo, usualmente en la parte fija de la mesa se halla el cuerpo, por otra parte, el brazo es el que se encarga de realizar las tareas que son asignadas. Las partes de las cuales está constituida el robot se muestran a continuación.(Godoy & Quintero, 2017)

* Estructura mecánica.
* Transmisiones y Reducciones.
* Sistemas de accionamiento.
* Sistema sensorial.
* Sistema de control.
* Elementos terminales.

#### Estructura Mecánica

La estructura mecánica hace referencia al tipo de articulación que tiene el robot y además de ello al tipo de movimiento que genera. Con referencia a lo anterior se puede decir que los grados de libertad hacen referencia a los movimientos que tienen las articulaciones como lo son el cuerpo y brazo del robot. Con respecto al movimiento relativo se puede decir que son realizadas por las articulaciones del robot industrial con respecto las uniones contiguas, estos movimientos pueden ser rotacionales, lineales o también una combinación de ambos. En la siguiente figura se muestra los diferentes tipos de configuraciones que pueden tener los robots industriales.(Godoy & Quintero, 2017)

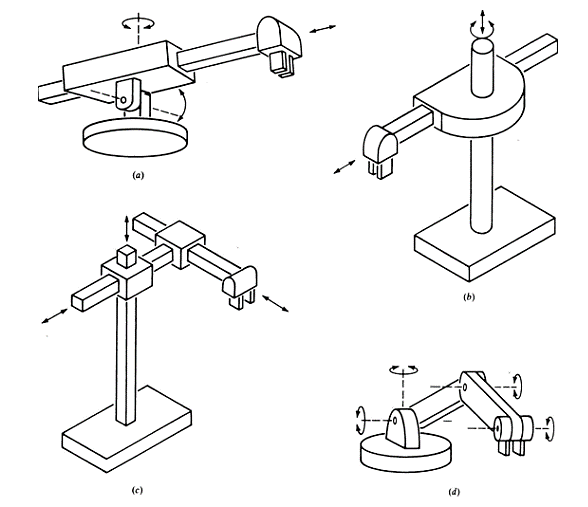


Fig. 9. Configuración de robots industriales

Mediante la Figura 9 se puede decir que el apartado a) corresponderá a la configuración esférica en donde el brazo telescópico podrá bajar o elevarse en torno a un pivote horizontal. El pivote estará sobre una mesa giratoria y además de ello esta mezcla de articulaciones permitirá al robot el desplazamiento de su brazo en un espacio esférico. Por otra parte, el apartado b) corresponderá a la configuración cilíndrica que usa un dispositivo cilíndrico que se moverá a través de la columna, así mismo se encontrará unido a un dispositivo que realizará un movimiento radial con referencia a la columna y mediante esta combinación de movimientos se podrá generar un espacio de trabajo en forma de un cilindro. (Godoy & Quintero, 2017)

Así mismo se puede decir que el apartado c) corresponderá a la configuración cartesiana la cual permite la posibilidad de deslizar sus articulaciones en línea recta con referencia a los ejes coordenados x, y, z, además de ello en su espacio de trabajo presenta una configuración en forma de un poliedro. Por otra parte, el apartado b) corresponderá a la configuración de tipo antropomórfica la cual se asemeja a un brazo humano y debido a esto estará conformado por brazo y antebrazo, además de ello están unidas a 2 articulaciones giratorias que son el hombro y el codo.(Godoy & Quintero, 2017)

Con respecto a las ventajas y desventajas que presentan cada una de estas configuraciones está relacionada de manera directa con su geometría. Una de las ventajas en cuanto los robots cartesianos es la repetitividad en los movimientos, esto debido a que su estructura es fija, por otra parte, una de las desventajas es su alcance en donde las configuraciones antropomórficas y esféricas son las más adecuadas. Así mismo en lo referente al levantamiento de maquinaria y materiales, la configuración cartesiana resulta ser la más indicada.(Godoy & Quintero, 2017)

#### Sistemas de transmisión

Son un conjunto de dispositivos que son los encargados de trasmitir la potencia que se ha generado por los actuadores a las distintas articulaciones o elementos del robot. Para determinar la transmisión correcta se tiene que tener en consideración el tipo de función que va a realizar el robot. Por lo general los sistemas de transmisión son los encargados de reducir la velocidad que se genera por los motores, esto se lo hace con la finalidad de disminuir las inercias en los extremos del robot. Además de ello, se utilizan los sistemas de transmisión para la transformación de un movimiento circular a lineal.(Godoy & Quintero, 2017)

En la tabla 1 se muestra los sistemas de transmisión que son comúnmente utilizados en la industria, en dicha tabla se presentará las desventajas y ventajas de cada uno de los tipos de transmisiones, así mismo se mostrará la entrada y salida del sistema.(Godoy & Quintero, 2017)

Tabla 1. Tipos de Transmisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Entrada/Salida** | **Denominación** | **Desventajas** | **Ventajas** |
| **Circular/circular** | Cable  Correa dentada  Cadena  Engranaje  Paralelogramo | Deformabilidad  -  Ruido  Holguras  Giro ilimitado | -  Distancia grande  Distancia grande  Pares altos  - |
| **Circular/lineal** | Cremallera  Tornillo sin fin | Rozamiento  Rozamiento | Holgura media  Poca holgura |
| **Lineal/circular** | Cremallera  Paral articulado | Rozamiento  Control difícil | Holgura media  - |

#### Sistemas de impulsión

Los sistemas de impulsión son los que se encarga de realizar acciones físicas que son ordenas por un sistema de control. Esta acción puede corresponder a un movimiento circular o lineal, o también la combinación de ambos. En un proceso bajo control, para la selección de la clase de actuador a utilizar se debe de considerar la acción que se va a efectuar como también la velocidad a la que se va a realizar. Por lo general se utilizan 3 tipos de actuadores los cuales se van a describir a continuación.(Jara & Pacheco, 2014)

* **Sistemas de impulsión hidráulicos:** Los actuadores hidráulicos son los que generalmente utilizan un fluido a una determinada presión para que el robot puede mover sus articulaciones. Para el caso de los robots de tamaño grande se utilizan los actuadores hidráulicos debido a que estos brindan una mayor resistencia mecánica y también una mayor velocidad.(Jara & Pacheco, 2014)
* **Sistemas de impulsión eléctricos:** Se los denomina de esta manera debido a que los impulsores eléctricos utilizan energía eléctrica para que el robot pueda mover sus articulaciones y realizar las operaciones para la que ha sido designado. Los robots de tamaño mediano son los que utilizan actuadores eléctricos debido a que no requieren de una alta velocidad en comparación a los robots diseñados para operar con actuadores hidráulicos.(Jara & Pacheco, 2014)

#### Sistema sensorial

Para que los robots tengan una relación de manera directa con su entorno, es necesario el uso de dispositivos que sean capaces de enviar la información para su correcto funcionamiento, estos dispositivos son denominados como sensores. A continuación, se muestran 2 clases de sensores.

* Sensores externos: En esta categoría de sensores se encuentran los sensores de velocidad, tacto, posición, fuerza, distancia o de visión artificial.



Fig. 10. Sensores externos

* Sensores internos: Están divididos en tres categorías correspondientes a: posición, velocidad, presencia. Los sensores de velocidad son los que controlan la velocidad del actuador, los sensores de posición son los encargados de proporcionar la información sobre el movimiento de las articulaciones en un espacio de trabajo determinado, mientras tanto los sensores de presencia son los encargados de limitar el movimiento de las articulaciones, además de ellos son los encargos de detectar la presencia de los objetos.

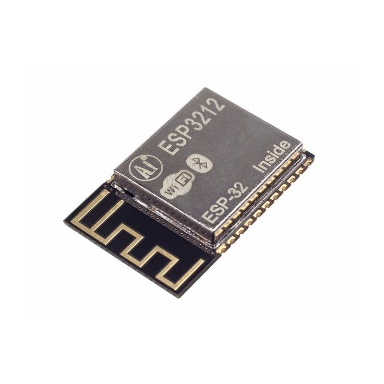


Fig. 11. Sensores internos

A continuación, mediante la Tabla 2 se puede apreciar los diferentes tipos de sensores comúnmente utilizados en la robótica.

Tabla 2. Tipos de sensores internos de los robots

|  |  |  |
| --- | --- | --- |
| **Sensores internos de los robots** | | |
| **Posición** | Analógicos | Potenciómetros  Resolver  Sincro  Inductivos  LVDT |
| Digítales | Encoders Absolutos  Encoders Incrementales Regla Óptica |
| **Velocidad** | Tacogenerador | |
| **Presencia** | Inductivo Capacitivo  Efecto Hall Célula Reed  Ultrasónico Contacto | |

### Control y programación

Para realizar el control de un robot cartesiano es necesario el uso de softwares y herramientas especificas donde se puede programar el código de las trayectorias del robot, la cual se comprobará mediante la ejecución de la misma. A continuación, se mostrará algunas opciones para el control y programación del robot.(Val, 2019)

#### Software de programación:

* LabVIEW: Es un software comúnmente utilizado para ingeniería en donde se puede desarrollar múltiples tareas, además de ello es un entorno de desarrollo integrado (IDE). Este software permite al usurario controlar al robot mediante señales digitales y comando, así mismo se puede realizar la crea con de interfaces.(Val, 2019)
* MATLAB: Es un lenguaje de programación y entorno de desarrollo integrado (IDE) que incluye herramientas para la generación y seguimiento de trayectorias, como la función bsplinepolytraj para generar trayectorias polinómicas con B-splines.(Val, 2019)

#### Paquetes de Software

* Robot Operating System (ROS): Es un conjunto de herramientas y bibliotecas para programar robots, incluyendo robots cartesianos. Permite controlar y programar robots de manera distribuida y escalable. (Mepal, 2020)
* Python: Es un lenguaje de programación ampliamente utilizado en robótica, que incluye bibliotecas como pyserial para comunicarse con el robot y numpy para realizar cálculos numéricos. (Contreras, 2021)

#### Simulación y modelado.

* Simulink: Es un entorno de simulación y modelado de sistemas dinámicos, que permite simular y probar algoritmos de control y trayectorias antes de implementarlos en el robot.(Val, 2019)
* Simscape: Es una herramienta de modelado y simulación de sistemas físicos, que incluye componentes de robótica y mecanismo para modelar y simular el comportamiento de robots cartesianos.(Val, 2019)

## Cinemática Directa

La cinemática es la encarga de estudiar el movimiento de un robot en relación con un sistema de referencia y permite el reconocimiento de la orientación y la posición del elemento a analizar sin tomar en consideración los momentos o fuerzas que originan dicho movimiento. (Carvajal & Godoy, 2019)

La cinemática directa es la encargada de determinar la posición del elemento final con relación a un sistema de referencia, esto cuando se conoce los parámetros geométricos del robot, así como los valores de las articulaciones. Debido a que los elementos de los que está constituido el robot se movilizan con respecto a un sistema de referencia, es necesario asignar un sistema de referencia correspondiente a cada uno de sus eslabones en movimiento. Mediante la cinemática directa se trata de establecer una matriz de transformación homogénea que de la posibilidad de relacionar el sistema de coordenadas ligado a cada eslabón con respecto al sistema de coordenadas de referencia.(Carvajal & Godoy, 2019)

Existen diversos métodos para determinar esta matriz, uno de ellos es el método desarrollado por Denavit-Hartenberg. Mediante este método se podrá relacionar los parámetros relativos al tamaño y la forma del eslabón y los parámetros que relacionan la posición relativa de un eslabón con respecto a su eje Para poder realizar este método es necesario conocer los siguientes fundamentos teóricos que se van a desarrollar a continuación.(Carvajal & Godoy, 2019)

### Descripción de la posición

Los robots han de ser referenciados en el espacio tridimensional, es decir, que para poder localizar un cuerpo rígido en el espacio es necesario disponer con una herramienta que nos permita su localización mediante puntos. Un robot se encuentra referenciado en un sistema tridimensional y su posición está representada por un vector PM con 3 posiciones con respecto al sistema de referencia M, como se puede apreciar mediante la Figura 12.(Carvajal & Godoy, 2019)



Fig. 12. Descripción de la posición

Para el análisis de la orientación y la posición del robot, se lo ha de representar como un cuerpo sólido; de tal manera que se pueda establecer un sistema de referencia mediante ejes ortogonales. Al tener asociado un objeto el sistema de referencia O, el vector representa la posición de origen del sistema O con respecto a M como se puede evidenciar mediante la Figura 13.(Carvajal & Godoy, 2019)

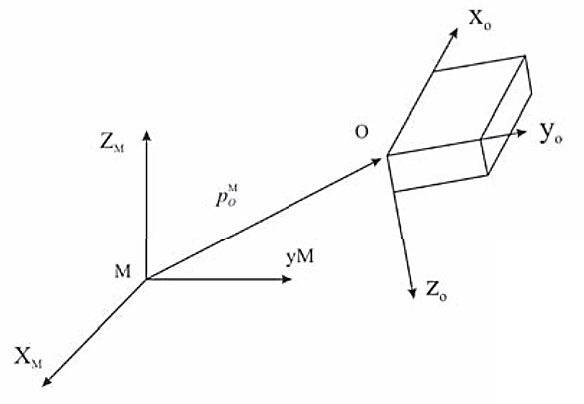


Fig. 13. Posición de un sistema o con respecto a otro m

Existen 3 formas de las cuales se puede representar la posición como se muestran a continuación.

* Coordenadas cartesianas: La forma en la que se representa el vector es de la forma .
* Coordenadas esféricas: La forma en la que se representa el vector es de la forma .
* Coordenadas cilíndricas: La forma en la que se representa el vector es de la forma

### Descripción de la orientación

Cuando se trata de un robot no es suficiente identificar o conocer su posición, también es necesario determinar y conocer su orientación específicamente del elemento terminal que es el que interactúa con el resto de los objetos. Una orientación en el espacio tridimensional está definida mediante 3 grados de libertad o 3 componentes linealmente independientes.(Carvajal & Godoy, 2019)

La orientación de un cuerpo con respecto a un sistema de referencia M está dada por la orientación relativa de los ejes del sistema de referencia O asociado a este con respecto al sistema M. esto se puede observar mediante la Figura 14.(Carvajal & Godoy, 2019)

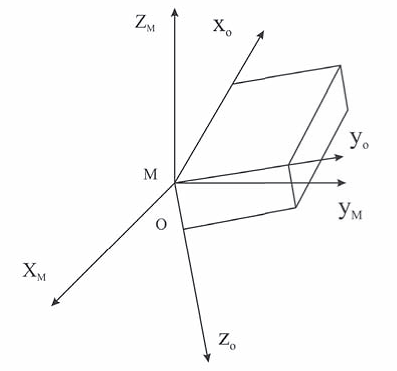


Fig. 14. Sistemas de referencia coincidentes en el origen

Las matrices de rotación son la metodología más utilizada para realizar la descripción de la orientación

tridimensional que se encuentra en un sistema de referencia rotado, sistema móvil (u, v, w) a un sistema de coordenadas de referencia fijo (x, y, z). Un vector que se encuentre en el plano se puede representar como se muestra a continuación.(Carvajal & Godoy, 2019)

Realizando las transformaciones respectivas se obtiene:

R corresponde a la matriz de rotación que define la orientación del sistema (u, v, w) con respecto al sistema (x, y, z), esta matriz es ortogonal y por lo tanto la inversa R-1 es igual a la traspuesta RT: R-1 =RT. Con esta matriz se realiza la representación de la orientación de sistemas girados únicamente sobre uno de sus ejes principales. Las siguientes matrices representan las rotaciones sobre los ejes (x, y, z).(Carvajal & Godoy, 2019)

### Coordenadas y matrices de transformación homogénea

Para la orientación y posición de un objeto en el espacio se las realiza mediante las matrices de transformación homogéneas que son cuadradas 4 x 4 y compuesta por 4 submatrices de distinto tamaño. Para la configuración cartesiana en la robótica donde se realizan movimientos traslacionales se las representa mediante la siguiente matriz que se muestra a continuación.(Carvajal & Godoy, 2019)

Por otra parte, cuando se desee conocer los componentes del vector r (u, v, w) se lo podrá obtener mediante la siguiente forma.

### Parámetros Denavit – Hartenberg

Para el desarrollo de esta metodología se define el origen de los sistemas de coordenadas de las articulaciones. Se denota el eje i como el eje de conexión de los elementos i-1 a i; para definir el marco del elemento i. La metodología para la cinemática directa basada en la sistemática Denavit- Hartenberg, se realiza mediante 9 pasos que se muestran a continuación.(Carvajal & Godoy, 2019)

1. Definir las direcciones de los ejes
2. Definir el origen del sistema de coordenadas de la base del eje
3. Definir el origen en la intersección con la normal común entre los ejes y
4. Escoger el eje a lo largo de la normal común a los ejes y con dirección de la articulación
5. Escoger el eje , para que con se cumpla la regla de la mano derecha.
6. Escoger el sistema de coordenadas de la herramienta donde sea normal a
7. Para construir una tabla con los parámetros de
8. Con estos parámetros se calculan las matrices de transformación homogéneas.
9. Una vez obtenidos los parámetros representar el modelo de la cinemática directa para la posición y orientación del sistema de coordenadas de la herramienta.(Carvajal & Godoy, 2019)

Mediante la Figura 15 se puede establecer los sistemas de coordenadas correspondientes a un robot cartesiano.

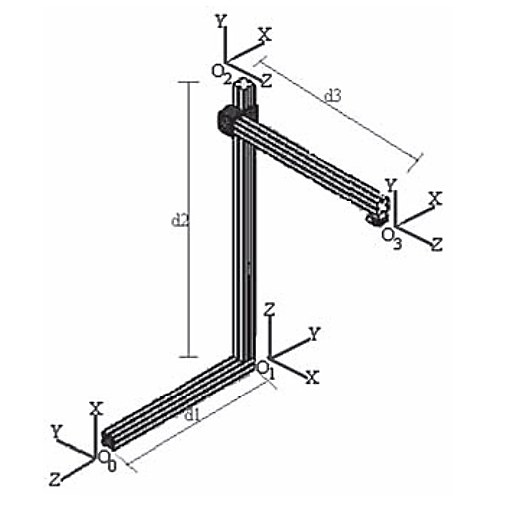


Fig. 15. Metodología Denavit - Hartenberg

Luego de haber determinado los marcos de los elementos, la posición y orientación de cada uno de ellos, es posible establecer los parámetros.(Carvajal & Godoy, 2019)

Ángulo entre los ejes y alrededor del eje

Puede tomarse positivo cuando la rotación se hace en sentido horario

Ángulo entre los ejes y alrededor de , puede tomarse positivo cuando la rotación se hace en sentido horario

: Coordenada de a lo largo de Zi-1

: Distancia entre y a lo largo de

Y a continuación en la Tabla 3. se muestran los parámetros Denavit - Hartenberg que representan la cinemática del robot.

Tabla 3. Parámetros Denavit - Hartenberg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Eslabón** |  |  |  |  |
| **1** | 90 | 90 | d1 | 0 |
| **2** | 90 | -90 | d2 | 0 |
| **3** | 0 | 0 | d3 | 0 |

Además, mediante la ecuación es posible determinar la matriz para cada eslabón como se muestra a continuación.

## Cinemática Inversa

Mediante la cinemática inversa es posible encontrar los valores de las articulaciones del robot para una posición y orientación específica. La metodología para resolver este problema a partir de la matriz de transformación homogénea se muestra continuación.(Carvajal & Godoy, 2019)

Para este caso en especial se tiene que . Después de haber determinado la matriz se tendrá que obtener la inversa de cada una de las matrices , para estos casos la inversa esta dada por la siguiente matriz.(Carvajal & Godoy, 2019)

Además, para determinar la cinemática inversa se utiliza el siguiente procedimiento que se muestra con las respectivas ecuaciones.

La configuración del robot cartesiano no presenta alto grado de complejidad para determinar la cinemática inversa, simplemente es necesario saber cuáles son las coordenadas en el plano cartesiano en las cuales se encuentra ubicado el objeto y con estos valores se determina qué distancia se debe desplazar cada uno de los eslabones.(Carvajal & Godoy, 2019)

## Dinámica de robot

La dinámica está destinada a analizar la relación que existe entre el origen del movimiento y las fuerzas que actúan sobre en un cuerpo. Debido a esto el modelo dinámico del robot tiene por finalidad conocer la relación que existe entre las fuerzas que actúan sobre un robot y el origen de su movimiento. Esta relación se consigue mediante las siguientes relaciones matemáticas que se muestran a continuación.(Muñoz, 2017)

* Los pares y las fuerzas aplicadas en las articulaciones o también en la parte final del robot específicamente en sus extremos.
* Las dimensiones y características del robot como sus longitudes, inercias y las masas de cada uno de los elementos.
* La ubicación del robot determinado por las variables articulares y sus respectivas derivadas como la aceleración y la velocidad.(Muñoz, 2017)

La obtención de este modelo matemático cuando se trata de mecanismos de uno o dos grados de libertad no resulta ser tan complejo, pero cuando se incrementa los grados de libertad su planteamiento resulta ser más difíciles de realizar. Por lo tanto, la generación de un modelo matemático para un robot resulta ser uno de los aspectos más difíciles de la robótica, y es comúnmente realizado con los siguientes fines que se muestran a continuación.(Muñoz, 2017)

* Para simular el movimiento que va a tener el robot.
* Para evaluar la resistencia de la estructura del robot.
* Para determinar las dimensiones de los actuadores.
* Con el fin de evaluar y diseñar el control dinámico del robot.(Muñoz, 2017)

Es necesario mencionar que el modelo completo del robot debe de contemplar el modelo dinámico de sus elementos como los eslabones y las barras, pero además de ello debe de contener el modelo dinámico de los actuares, sistemas de transmisión y de los equipos electrónicos de mando, esto debido a que con estos elementos se agregan nuevos rozamientos, inercias, etc.(Muñoz, 2017)

Mediante la segunda ley de Newton y su equivalente la ley de Euler que está destinada para movimiento de rotación se puede obtener el modelo dinámico para un robot de estructura mecánica rígida. A continuación, se presenta las ecuaciones, mencionadas anteriormente.(Muñoz, 2017)

Además de ellos existen diferentes algoritmos computacionales para obtener el modelo dinámico de un robot, mediante diversas investigaciones se han determinado diversas formulaciones alternativas que están basadas en la mecánica Newtoniana y Lagrangiana, con la finalidad de obtener un modelo matemático manejable y eficiente.(Muñoz, 2017)

Los métodos que más se han utilizado para la obtención del modelo matemático de un robot son el algoritmo de Lagrange- Euler, Gibbs Apple, Newton Euler y trabajos virtuales. A continuación, se presentará la formulación de Newton Euler.(Muñoz, 2017)

### Formulación Newton Euler

Mediante la formulación lagrangiana nos permite conseguir un modelo dinámico de un robot y a su vez esta formulación conduce a un algoritmo que tiene un costo computacional de un orden de O(n4). Esto quiere decir que el número de operaciones que se va a efectuar crece en función del número de grados de libertad del robot elevada a la cuarta potencia, esto se hace desde el punto de vista de la eficiencia y además de ello no se puede utilizar para robots que tengan más de 4 grados de libertad.(Muñoz, 2017)

El algoritmo de Newton – Euler está basado en operaciones vectoriales donde se incluye productos vectoriales y escalares entre cantidades vectoriales, además de productos de matrices con vectores, siendo más eficiente cuando se realiza la comparación con las operaciones matriciales de la formulación Lagrangiana.(Muñoz, 2017)

A continuación, en la siguiente Figura se muestra la configuración mecánica del robot cartesiano

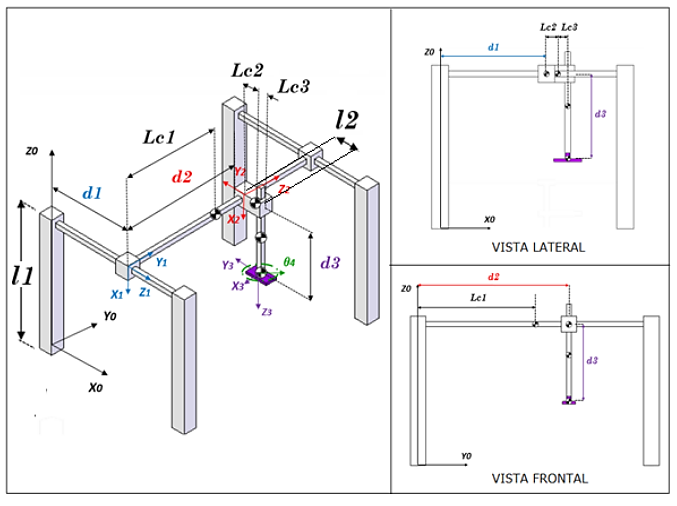


Fig. 16. Configuración mecánica del robot cartesiano

Una vez que se haya aplicado el algoritmo de Newton-Euler, deducido en base a la Figura 4 se puede apreciar que se obtuvieron las siguientes ecuaciones que corresponden al modelo dinámico del robot cartesiano las cuales se muestran a continuación.(Muñoz, 2017)

Ecuación 1:

Ecuación 2:

Ecuación 3:

Ecuación 4:

Estas ecuaciones corresponden a la representación del modelo dinámico del robot cartesiano, además de ello se recomienda la validación de dicha formulación matemática mediante la utilización de softwares especializados para el caso de estudio.(Muñoz, 2017)

## Parte mecánica del robot cartesiano

### Selección de materiales

En lo referente a los materiales empleados para la fabricación de los robots cartesianos cabe destacar que el material mas empleado es el aluminio tanto para los perfiles de la estructura como para los soportes codos, chapas, etc. Además, es importante mencionar que en los topes mecánicos y los soportes para los finales de carrera es comúnmente utilizado el plástico ABS. En lo referente al acero se utilizan por lo general para la fabricación de los ejes de trasmisión de potencia debido a que son elementos mecánicos que deben soportar motores esfuerzos de torsión y flexión.



Fig. 17. Perfiles de aluminio extruido

### Selección y diseño de elementos del robot cartesiano

En general se conoce que este tipo de robots tienen 3 grados de libertad, donde en el eje X se encuentran 2 guías por donde se desplazará el eslabón vertical, por otra parte, la transmisión de movimiento correspondiente al eje X se lo realiza mediante una correa dentada. Así miso para el movimiento en el eje Z se dispone de tornillos de bolas y guías. Finalmente, el desplazamiento en el eje Y se realiza con un actuador cilindro neumático.(Godoy & Quintero, 2017)

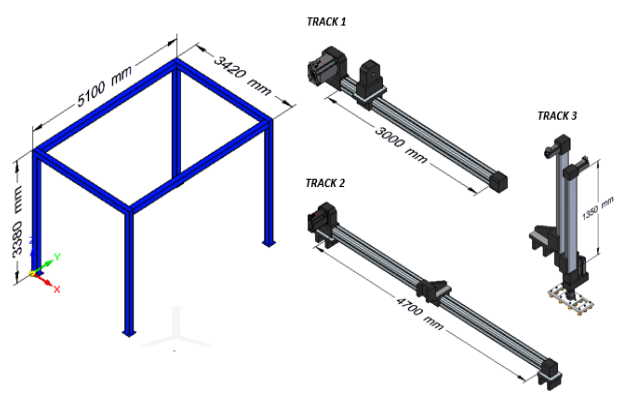


Fig. 18. Componentes del robot cartesiano.

El diseño del robot cartesiano está comprendido por la selección materiales para la construcción de la estructura. Además de eso tiene actuadores que están conectados a guías lineales y rieles que permiten el movimiento controlado a lo largo de cada eje. Por otra parte, también cuentan con motores que están acoplados a husillos de bolas, correas dentadas etc., con la finalidad de poder convertir el movimiento rotativo en lineal. Con respecto a las especificaciones para el diseño se desarrollarán de manera más completa en el capítulo 3, donde se podrá evidenciar el procedimiento y análisis realizados para el diseño del robot cartesiano.(Godoy & Quintero, 2017)

## Normativas

En los inicios de los años novena era escasas las normativas relacionadas con el empleo y la instalación de los robots. Los principales factores que han llevado a la demora para formular y utilizar una normalización para los robots son los siguientes que se muestran a continuación.(Sellés, 2016)

* Existía una tendencia de enfrentarse a problemas de mercado y técnicos, antes que cualquier otro.
* No existía la suficiente experiencia para solucionar los accidentes que se produjeran con los robots.
* Existía dificultad para concordar criterios entre los diferentes usuarios de los distintos países.
* Dificultad para preparar la documentación adecuada referente a la normativa, de la misma manera se presentaban problemas para proponer los procedimientos de evaluación respectivos.(Sellés, 2016)

En la actualidad existen diversas normativas legales y a continuación en los siguientes apartados se describen las normativas vigentes más importantes y utilizadas en los robots.(Sellés, 2016)

### Normativa internacional ISO 10218: –1:2011

Es una versión corregida en el año 2014 y está destinada a los dispositivos robóticos y a los robots, y en específico muestra los requisitos de seguridad para los robots de tipo industrial. Es una parte de la norma ISO 10218 y muestra las reglas para el uso robots, los requisitos para el diseño seguro, y las medias de protección que se deben de utilizar para mantener un correcto funcionamiento. En este parte de la norma se muestra los peligros básicos asociados con los robots y da a conocer los requisitos para disminuir estos peligros. (Sellés, 2016)

En esta parte de la norma no se trata de robot en forma completa. En este apartado se excluye el ruido producido por el robot debido a que no se considera un peligro significativo del mismo. Así mismo se menciona que este apartado no es utilizable para robots industriales, a excepción de los principios de seguridad si se los puede utilizar para estos robots. A continuación, se muestran normas que son indispensables.(Sellés, 2016)

* **ISO 9283:1998.** Robots manipuladores industriales. Criterios de análisis de prestaciones y métodos de ensayo relacionados.
* **ISO 10218-2.** Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración.
* **ISO 12100.** Seguridad de las máquinas. Conceptos básicos, principios generales para el diseño. Evaluación y reducción del riesgo.
* **ISO 13849–1:2006.** Seguridad de las máquinas. Partes de los sistemas de mando relativas a la seguridad. Parte 1: Principios generales para el diseño.
* **ISO 13850.** Seguridad de las máquinas. Parada de emergencia. Principios para el diseño.
* **IEC 60204–1.** Seguridad de las máquinas. Equipo eléctrico de las máquinas. Parte 1: Requisitos generales.
* **IEC 62061:2005.** Seguridad de las máquinas. Seguridad funcional de sistemas de mando eléctricos, electrónicos y electrónicos programables relativos a la seguridad.(Sellés, 2016)

### Normativa americana ANSI/RIA R15.06–1992.

Es una normativa que ha sido realizada por el Instituto Nacional de Normalización de los Estados Unidos. Fue una normativa publicada en el año 1986 y fue revisada en el año 1992. Es una normativa corta, pero presenta distintas características que son muy destacables. Un ejemplo de ellos es la inclusión del apartado donde se encuentra la definición de los riesgos, además de ellos en sus capítulos se muestran la probabilidad de que aparezca un accidente y el daño físico que puede ocasionar sobre una persona, todo esto dependiendo de la experiencia del operador.(Sellés, 2016)

### Normativa americana ANSI/RIA R15.06–1999.

Es una normativa que fue aprobada en el año de 1999 por el Instituto Nacional de Normalización de los Estados Unidos. Esta normativa nos brinda especificaciones acerca de la fabricación de los robots industriales, su instalación y las formas de cómo se puede mejorar la seguridad de los operadores, así mismo muestra lo métodos para mejorar los sistemas de los robots. (Sellés, 2016)

### ISO 9283:1998. Manipulación de robots.

Es una norma internacional en donde se muestra los métodos para ensayar determinados componentes que permitirá comprobar el correcto funcionamiento de las características propias de los robots manipuladores. Además de ello muestra las recomendaciones para la realización de los ensayos del comportamiento de dichos robots. Por otra parte, se muestra ejemplos prácticos que permitirán conocer cómo deben realizaros los informes y como se deben de interpretar los resultados obtenidos. También contiene una sección con información acerca de la fabricación y el diseño, y además de ello dedica un parte para para el diseño de sistemas robotizados con respecto a la ergonomía, control y aspectos mecánicos.(Sellés, 2016)

### Normativas Europeas

En referencia a las normativas europeas se pueden mostrar los tipos que se muestran a continuación.

* **EN 60204–1.** Esta norma está destinada para el equipo eléctrico de las máquinas en referencia a los aspectos de protección personal, HMI, conexión, equipos, cableados y marcado. (Sellés, 2016)
* **EN 418.** Esta norma está destinada a la conformidad con los requisitos básicos de las máquinas. En específico está destinada a evitar los riesgos que pueden aparecer y afectar a los operadores, además menciona que el parado de emergencia debe estar disponible y funcionar en cualquier momento.(Sellés, 2016)
* **EN 292.** Conceptos básicos. Principios generales de diseño.
* **EN 294.** Distancias de seguridad para impedir que las extremidades superiores alcancen zonas peligrosas.
* **EN 954–1.** Partes de los sistemas de control relativas a la seguridad.
* **EN 1050.** Principios para la evaluación del riesgo.
* **EN 1037.** Prevención de una puesta en marcha accidental.
* **EN 1088.** Dispositivos de enclavamiento asociados a resguardos.
* **EN 60947–5–1.** Aparatos electromecánicos para circuitos de control.(Sellés, 2016)

# CAPÍTULO III

1. **MARCO METODOLÓGICO**

## Alternativas

Se considera entonces un robot cartesiano para realizar la tarea de apilar bloques de 20 kg, para ello el robot se maneja en 3 ejes con un alcance de 1.2m de manera longitudinal y 0.6m de manera vertical y transversal. De esta manera se consideran diversas alternativas de robots cartesianos ya sea tomando en cuenta el material de construcción, así como también el tipo de perfil que se utilizaría.

### Alternativa 1

|  |  |
| --- | --- |
| **Componentes** | |
| Sistema de transmisión | Por husillo |
| Material de perfil | Aluminio 6063-T5 |
| Tipo de perfil | Tipo C 80 x 40 |
| Tipo de ranura | Ranura en V |

Se considera un perfil de aluminio tipo C con ranura en V.

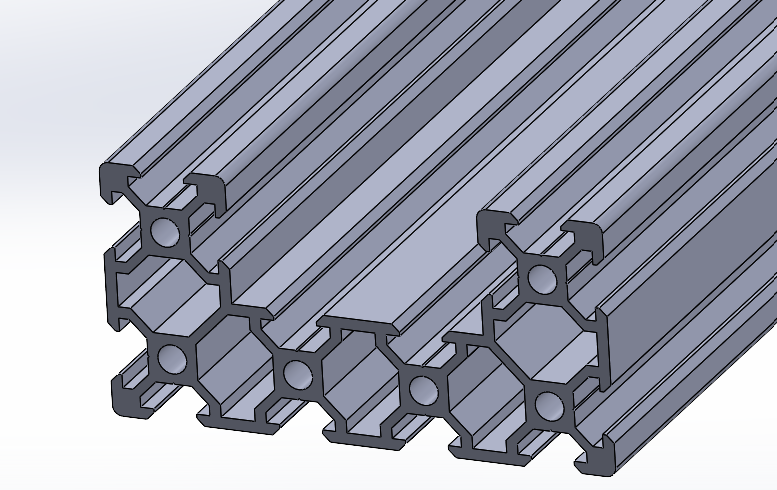


Fig. 19. Perfil C de aluminio

Para este tipo de perfiles se requiere un desplazamiento lineal por medio de un carrito con ruedas que se apoyan en el perfil.

Imagen que contiene Icono

Descripción generada automáticamente

Fig. 20. Carrito de desplazamiento

Por lo que el sistema acoplado seria:

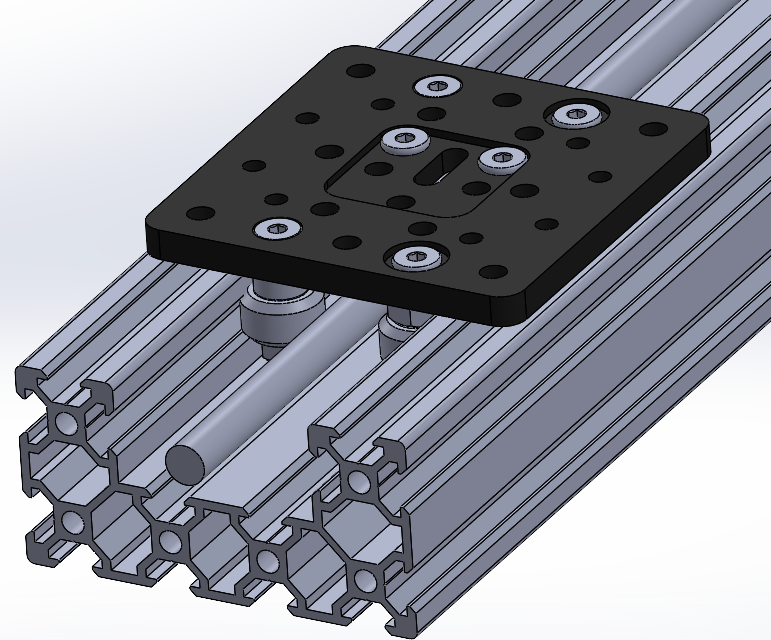


Fig. 21. C beam con carrito

### Alternativa 2

|  |  |
| --- | --- |
| **Componentes** | |
| Sistema de transmisión | Por correa |
| Material de perfil | Aluminio 6063-T5 |
| Tipo de perfil | 20 x 40 |
| Tipo de ranura | Ranura en V |
| Polea dentada | GT2-32 D.I 5mm |
| Correa | GT2-6 abierta |

La segunda alternativa va enfocada a un desplazamiento rápido sin mucha perdida.



Fig. 22. Sistema por correa

El actuador lineal V-slot con transmisión por correa dentada es una excelente opción para movimientos rápidos y precisos en proyectos de automatización, robótica y sliders para cámaras.

### Alternativa 3.

|  |  |
| --- | --- |
| **Componentes** | |
| Sistema de transmisión | Por husillo |
| Material de perfil | Aluminio 6063-T5 |
| Tipo de perfil | 30 x 150 |
| Tipo de ranura | Ranura en T |
| Guías lineales | HGR 20 |
| Husillo de bolas | SFU 2510 |

Debido al peso que supone la carga de trabajo del robot también se considera el uso de rieles lineales y husillos como medio desplazamiento.

Imagen que contiene guitarra

Descripción generada automáticamente

Fig. 23. Desplazamiento por rieles lineales

### Evaluación de alternativas

Todas estas alternativas son analizadas y evaluadas para determinar cuales es la más efectiva para ser diseñada y modelada de acuerdo con criterios de evaluación que son útiles en este proyecto.

* Facilidad de diseño y construcción
* Operabilidad y confiabilidad
* Apto para el peso de aplicación
* Costo de inversión

Con la evaluación de estos 5 factores, se pretende obtener un parámetro de medición que facilite tener en cuenta la mejor alternativa que brinde las mejores prestaciones para la aplicación anteriormente determinada.

Tabla 4. Evaluación de alternativas

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Alt 1** | | **Alt 2** | | **Alt 3** | |  |
| **Ítems** | **Peso** | **Factor** | **Calc.** | **Pond.** | **Calf.** | **Pond.** | **Calf.** | **Pond.** | **Parámetros cualitativos** |
| 1 | 25% | Facilidad de diseño y construcción | 5 | 1,25 | 3 | 0,75 | 5 | 1,25 | 5: fácil,  3: Medio,  1: Difícil |
| 2 | 25% | Operabilidad y confiabilidad | 3 | 0,75 | 3 | 0,75 | 3 | 0,75 | 5: fácil,  3: Medio,  1: Difícil |
| 3 | 25% | Apto para el peso de aplicación | 3 | 0,75 | 1 | 0,25 | 5 | 1,25 | 5: fácil,  3: Medio,  1: Difícil |
| 4 | 25% | Costo de inversión | 1 | 0,25 | 3 | 0,75 | 1 | 0,25 | 5: fácil,  3: Medio,  1: Difícil |
| Evaluación | | | 3 | | 2,5 | | **3,5** | |  |

Dada la evaluación expuesta en la tabla anterior se tiene que la mejor alternativa a ser desarrollada es la 3 por lo tanto es la que se selecciona para su diseño.

## Análisis del Robot Cartesiano

### Análisis Cinemático

Con el fin de desarrollar el análisis cinemático primero se plantea el diseñar un modelo conceptual aproximado para establecer las posibles dimensiones que se tendrían en el robot cartesianos, tomando en cuenta los sistemas de referencia tanto inercial como móvil. Esto nos permitirá ejecutar el análisis y diseño del robot de una manera practica y ordenada.



Fig. 24. Esquema para el análisis cinemático

Es así como definimos nuestro sistema de referencia inercial en la parte inferior de una columna del lado izquierdo.

#### Parámetros Denavit – Hartenberg aplicado al modelo del robot cartesiano

Este método permite determinar la relación cinemática entre dos cuerpos sólidos como los eslabones, utilizando la menor cantidad de parámetros. Se analiza los parámetros de Denavit – Hartenberg respecto al sistema de referencia inercial propuesto anteriormente y a la cadena cinemática alineada con el eje x inercial.

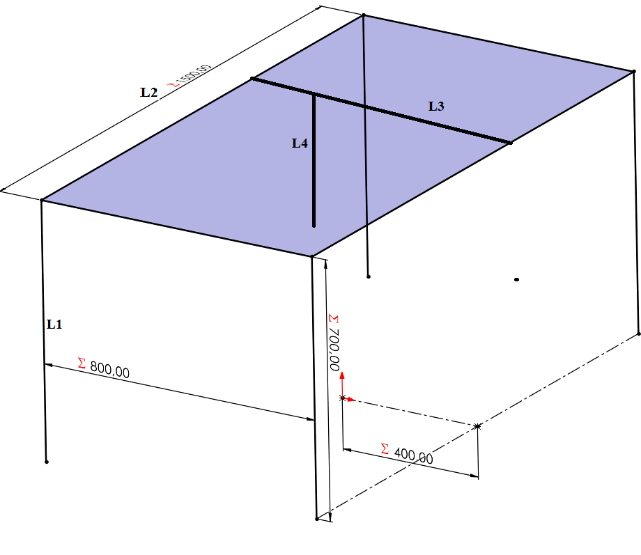


Fig. 25. Longitudes consideradas

Por lo tanto, la tabla que determina los parámetros Denavit – Hartenberg para este tipo de robot en el modelo planteado es la siguiente:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 | 0 |  | 0 |
| 0 |  | 0 | -π/2 |
| -π/2 |  | 0 | -π/2 |
| π/2 |  | 0 | -π/2 |
| 0 |  | 0 |  |

Donde el valor de L1, es el valor de altura desde el centro del sistema de referencia inercial al centro de la base de la primera articulación, mientras que L2,L3,L4, son valores que miden de la distancia que existe entre el extremo y el centro de la articulación prismática. Por otro lado, los valores de , , y son los valores de la posición de las articulaciones en el espacio.

Dado que el primer marco móvil se encuentra alineado con el marco de referencia inercial no existe rotación ni traslación en *z* y tampoco existirá traslación y rotación en x. Para el segundo marco móvil, este debe ser situado en la primera viga longitudinal, es así como es necesario trasladarnos *L1* unidades en dirección al eje +*z* y para el caso de la rotación en el eje *x* este rotaria 90 grados en sentido negativo, esto debido a que alineamos al eje *z* con el eje lineal de la viga por donde se desplazaría la articulación.

Para el tercer marco este se debe analizar tomando en cuenta la siguiente posición de la articulación, por lo tanto, primero debemos rotar en el eje *z* 90 grados en sentido negativo y con una traslación de L2 unidades y tomando en cuenta la posición de la siguiente articulación se rota en el eje *x* 90 grados en sentido negativo. Para el cuarto marco de referencia se rota 90 grados en el eje *z*, y se traslada en ese mismo eje L3 unidades y por último rotando -90 grados en el eje x consiguiendo así preparar el marco móvil para la siguiente posición en el efector final. La quinta posición del marco de referencia sucede en el último elemento colocado de manera vertical y únicamente tendría una traslación en *z* de L4 unidades.

Estos valores se dejarán expresados en forma de variables con el fin de determinar las ecuaciones necesarias como el modelo dinámico o la función de control.

#### Cinemática directa mediante matrices de transformación homogénea

Una vez determinado los parámetros Denavit – Hartenberg para cada marco de referencia del sistema se determina la cinematica directa a través de las matrices de transformación homogénea

Donde se debe determinar la cinematica directa hasta el quinto marco móvil. Es decir:

Donde la matriz de transformación homogénea en el marco 1 es:

Utilizando los valores de

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| = 0 | d = 0 | a = | = 0 |

Entonces la matriz de transformación homogénea del primer marco de referencia es:

Para el marco 2 es:

Utilizando los valores de

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 |  | 0 | -π/2 |

Entonces la matriz de transformación homogénea del segundo marco de referencia es:

Para el marco 3 es:

Utilizando los valores de

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| -π/2 |  | 0 | -π/2 |

Entonces la matriz de transformación homogénea del segundo marco de referencia es:

Para el marco 4 es:

Utilizando los valores de

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| π/2 |  | 0 | -π/2 |

Entonces la matriz de transformación homogénea del segundo marco de referencia es:

El procedimiento se realiza de la misma forma hasta llegar el ultimo marco de referencia por lo que todas operaciones se lo realizan en el Software MATLAB teniendo así la siguiente matriz de transformación homogénea.

#### Matriz Jacobiana geométrica

Definimos la relación dinámica para los dos sistemas diferentes sistemas de coordenadas.

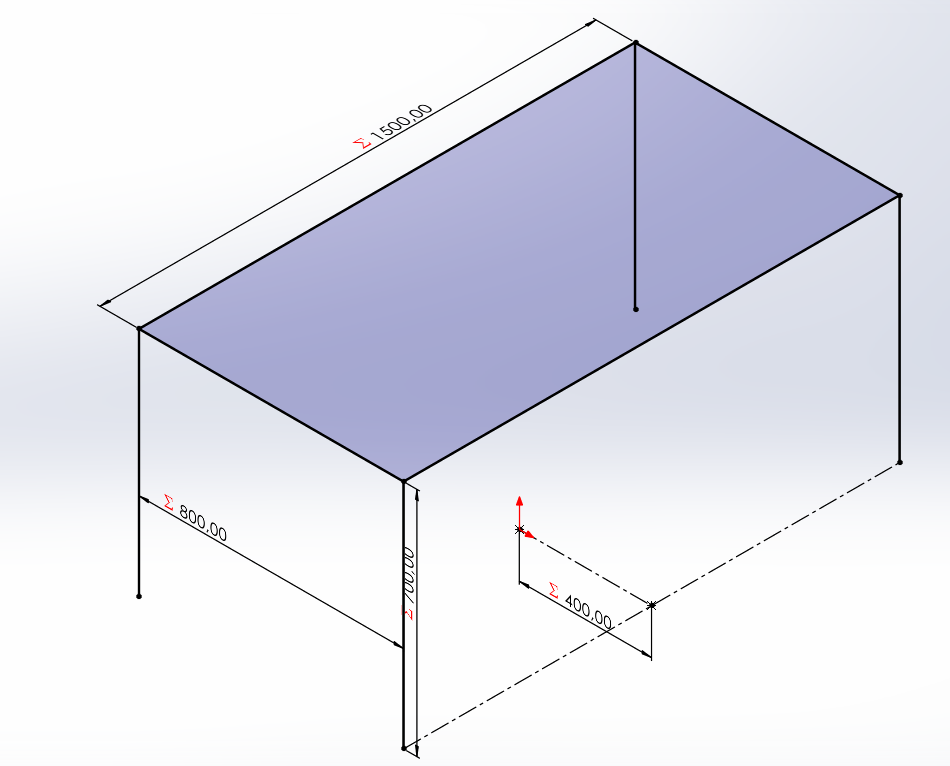


Fig. 26. Dimensiones generales

Para el desarrollo del robot y cumplir el área útil de trabajo propuesto se consideran las dimensiones que muestran la imagen anterior.

### Análisis Dinámico

Con el fin de realizar el análisis dinámico, partimos de la ecuación dinámica de un robot:

Donde:

Corresponde a la matriz de inercia [kg m2]

: Vector que representa la segunda derivada de la posición de las articulaciones [rad/s2]

Matriz de efectos centrífugos y de Coriolis [kg m2/s]

Vector que representa la primera derivada de la posición de las articulaciones [rad/s]

: Matriz de efectos gravitacionales [N m]

: Vector de momentos [N m].

Para el caso de la matriz de inercia se relaciona la masa de cada una de las articulaciones mediante la siguiente ecuación:

Momentum lineal:

Momentum Angular:

Donde el segundo termino va a ser cero debido a que las articulaciones que tenemos son prismáticas y no van a ejercer movimientos de rotación por lo tanto únicamente relacionamos cada masa de las articulaciones por la respectiva matriz jacobiana.

Matriz de efectos centrífugos y de Coriolis relaciona los efectos que resultan de un cuerpo al encontrarse en un medio que rota o tiene un movimiento rotacional. Dado que el sistema de un robot cartesiano no posee movimientos rotacionales sino únicamente movimientos lineales de sus articulaciones, se puede concluir que la matriz de efectos centrífugos no afectaría este sistema.

Como indica la ecuación anterior, se deriva la matriz jacobiana la cual para este caso ha resultado de ser una matriz de números constantes en cada marco, por lo tanto, al derivar dicha matriz su resultado sería una matriz de ceros, indicando así que el sistema no es afectado por el efecto de Coriolis.

Matriz de efectos gravitacionales indica que articulación se ve afectada en su accionamiento por la aceleración gravitacional.

Este cálculo se lo realiza hasta la i-ésimo masa la cual corresponde a la respectiva articulación.

Este vector indica que la gravedad únicamente afecta a nuestra tercera articulación, porque el eje de traslación de esta articulación se encuentra alineada al eje Z, mientras que las dos primeras articulaciones se encuentran en un movimiento en el plano XY.

Para obtener el modelo dinámico del sistema se debe resolver la ecuación de tercer orden.

### Modelo dinámico

Para determinar el modelo matemático del sistema se hace uso de la ecuación dinámica, empleando espacio de estados. El espacio n-dimensional cuyos ejes de coordenadas están formados por el eje xn donde son variables de estados. El análisis en el espacio de estados se centra la atención en los tres tipos de variables que aparecen en el modelado de los sistemas dinámicos; las variables de entrada, las variables de salida y las variables de estado.

Donde:

: Vector de estado

matriz de estado

: vector de no linealidades

B: matriz de control

: vector de entradas

Lo dejamos en función de espacio de estados

Al realizar esta operación lograremos obtener el modelo dinámico de nuestro sistema.

### Función de control

Para definir el control se propone hacerlo mediante un control Diferencial – Proporcional, para eso definimos la función de error.

Donde

: corresponde a la matriz de la posición y velocidad de la articulación deseada.

Para garantizar la estabilidad asintótica del sistema la dinámica del error debe evolucionar idealmente como el producto entre una matriz de ganancia por la función de error.

Nuestra señal de control será una señal PD que se mapea a cada instancia del sistema.

Para que el sistema sea estable se propone mediante una función candidata de Lyapunov. A partir de la teoría clásica de la mecánica, sabemos que un sistema vibratorio es estable si su energía total (una función definida positiva) es continuamente decreciente (lo que significa que la derivada de tiempo de la energía total debe ser definida negativa) hasta que se alcanza un estado de equilibrio.

Donde *V* va a ser definida positiva.

La cual se considera la derivada ideal para el caso de la función candidata de Lyapunov y se igual a la derivada real de la misma.

Entonces a partir de:

Donde:

Despejando *u* que será la función de control de manera generalizada.

A partir entonces de la ecuación del modelo dinámico:

Sustituimos estos términos en la función de control

Al operar la anterior ecuación se llega a una ecuación mucho más simplificada.

Por lo tanto, la función de control es:

Donde *kp* corresponde a la ganancia proporcional y *kd* corresponde a la ganancia diferencial.

### Valores de kp y kd

Se evalúa las funciones de error con el fin de saber si el sistema va a ser sobre amortiguado o críticamente amortiguado de acuerdo con los valores de las ganancias.

Simplificando estas operaciones:

Comportamiento sobre amortiguado

Comportamiento críticamente amortiguado

## Diseño mecánico

### Análisis estático

Tabla 5. Requerimientos de diseño

|  |  |  |
| --- | --- | --- |
| **REQUERIMIENTOS DE DISEÑO** | | |
| Peso de carga | 20 | kg |
| Área de trabajo | 1250 x 600 x 600 | mm |
| Material de perfil | Aluminio | |

Para empezar con el desarrollo del análisis estático se lo realiza a partir del bloque a cargar hasta llegar al apoyo de las columnas. Es así como se realizar un diagrama para representar únicamente las fuerzas o pesos que se vayan a aplicar y nos facilita el análisis de las fuerzas y la determinación de la resultante. Esto es esencial para resolver problemas de estática y dinámica.

Gráfico de cajas y bigotes

Descripción generada automáticamente

Fig. 27. Diagrama de análisis

Selección de materiales

Lo cual es la fuerza necesaria de aplicar para elevar el bloque de 20 kg.

#### Análisis del gripper

Para saber que garra o efector final se utilizara es necesario considerar el peso del elemento que se vaya a elevar. En este tomando en cuenta el peso del objeto de 20 kg es necesario determinar un gripper con una fuerza de agarra mayor a este peso.

#### Trabajo necesario de elevación

Se toma en cuenta todos los pesos de los elementos que existan para elevarlos.

Diagrama

Descripción generada automáticamente con confianza media

Fig. 28. Peso del gripper

Tabla 6. Tabla de pesos movimiento vertical

|  |  |  |
| --- | --- | --- |
| **PESOS** | | |
| Bloque | 20 | kg |
| Gripper | 0.509 | kg |

Se determina el trabajo requerido determinando la altura de elevación que se requiere.

#### Trabajo necesario para el movimiento transversal

Se considera todos los pesos de los elementos que se moverán transversalmente. Incluyendo el posible peso del motor que se considerara un 0.5% del peso total a izar.

Tabla 7. Tabla de pesos - movimiento transversal

|  |  |  |
| --- | --- | --- |
| **PESOS** | | |
| Bloque | 20 | kg |
| Gripper | 0.509 | kg |
| Actuador Lineal eléctrico | 4.29 | kg |
| **Total** | 24.799 | kg |

Diagrama

Descripción generada automáticamente

Fig. 29. Diagrama de peso transversal

Si analizamos el cuerpo para el movimiento transversal. Se lo hace considerando las fuerzas que se presentan en este.

Para determinar el trabajo necesario para desplazar el cuerpo en la longitud requerida se calcula de la siguiente manera:

### Análisis de esfuerzos viga transversal

#### Análisis con la carga en el centro

Se debe tomar en cuenta la geometría del perfil.

Diagrama

Descripción generada automáticamente

Fig. 30. Sección transversal del perfil

Tabla 8. Propiedades del perfil de aluminio

|  |  |  |
| --- | --- | --- |
| **Perfil de aluminio** | | |
| I1 | [mm4] | 2625780 |
| I2 | [mm4] | 117202,8 |
| Sc | [mm] | 75 |
| St | [mm] | 75 |
| A | [mm2] | 1256,81 |

Se analiza la flexión a este elemento que se encuentra de manera transversal a nuestra distribución del robot. Dado que el peso del sistema de izaje se encuentra apoyado en una superficie se transforma esta en una carga distribuida a lo largo de esta tomando en cuenta la longitud que posee.

Gráfico

Descripción generada automáticamente

Fig. 31. Viga analizada

Para este análisis se tiene los siguientes resultados:

Tabla 9. Resultados de la viga a flexión

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados de la viga** | | | |
| Momento de inercia | I1 | [mm4] | 2625780 |
| Momento de inercia | I2 | [mm4] | 117202.8 |
| Momento de inercia | Leff | [mm4] | 2625780 |
| Máx. dist. al límite |  | [mm] | 75 |
| Factor de seguridad | 98.3108 | | |
| Límites de estiramiento |  | [N/mm2] | 145 |
| Módulo E |  | (N/mm2] | 69000 |
| Material | 6063T5 | | |
| Máx. deflexión | S1 | [mm] | 0.02948 E-15 |
| Máx. momento flector | Mbl | [Nm] | 2.8296 E-15 |
| Máx. deflexión | S2 | [mm] | 22.59823 E3 |
| **Máx. momento flector** | Mb2 | [Nm] | 51.637 |
| Máx. res. esfuerzo | Res. | [N/mm2] | 14.749 |
| Máx. deflexión | Sres | [mm] | 22.59823 E3 |
| Máx. momento flector | Mbres | [Nm] | 51.637 |
| Escala de línea defl. | 5177.4:1 | | |
| Escala de línea de momento flector | 4.5316:1 | | |

En base al factor de seguridad expresado en la tabla se puede comprobar que el perfil resistirá adecuadamente el peso de carga y de los elementos de izaje. Debido a que el factor de seguridad es demasiado alto se entiende que el sistema está sobredimensionado, esto se puede entender dado que la orientación del perfil hace que las cargas se encuentren actuando en el eje con mayor inercia o masa.

Como se observa en el diagrama, el momento flector máximo es de 51.637 Nm.

Diagrama

Descripción generada automáticamente

Fig. 32. Diagrama cortante

Para determinar el esfuerzo cortante máximo se toma en cuenta el valor máximo de fuerza cortante de 121.93 N.

Donde:

V: Carga aplicada

Q: Primer momento de área

I: Momento de Inercia

t: Espesor de pared

Donde:

: es la distancia desde el eje neutro de la sección completa hasta el centroide del área A′

A’: es el área de la sección transversal por encima (o por debajo) del punto donde se mide el esfuerzo cortante.

Entonces,

Ahora se determina los esfuerzos principales en el punto a partir de la ecuación:

Para obtener un factor de seguridad se debe relacionar este esfuerzo principal con el permisible del material.

Factor de seguridad

#### Análisis en el extremo de la viga

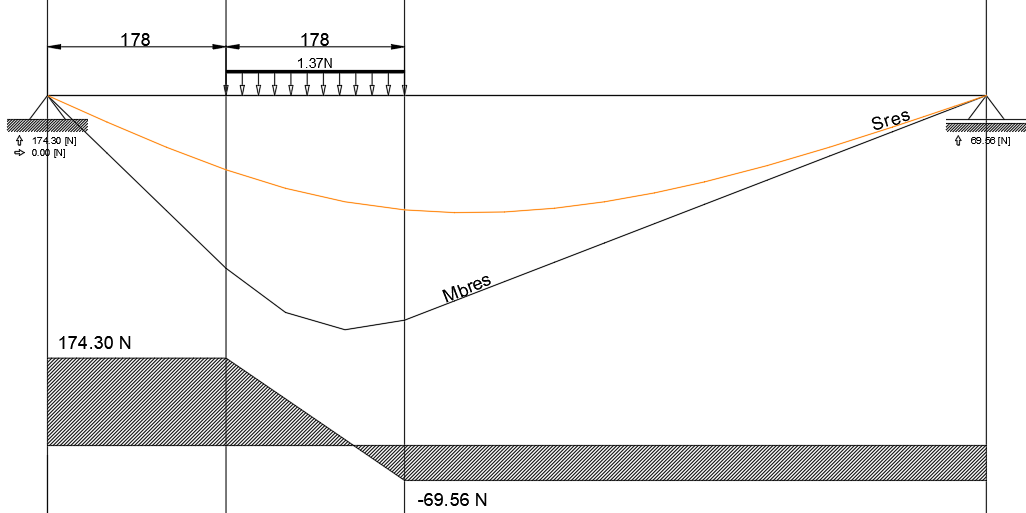


Fig. 33. Análisis en el extremo de la viga

Al realizar este cálculo se tiene como resultado los siguientes valores:

Tabla 10. Resultados análisis viga transversal

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados de la viga** | | | |
| Momento de inercia | I1 | [mm4] | 2625780 |
| Momento de inercia | I2 | [mm4] | 117202.8 |
| Momento de inercia | Leff | [mm4] | 2625780 |
| Máx. dist. al límite |  | [mm] | 75 |
| Factor de seguridad | 120.54 | | |
| Límites de estiramiento |  | [N/mm2] | 145 |
| Módulo E |  | (N/mm2] | 69000 |
| Material | 6063 T-5 | | |
| Máx. deflexión | S1 | [mm] | 0.023446 E-15 |
| Máx. momento flector | Mbl | [Nm] | 2.4705 E-15 |
| Máx. deflexión | S2 | [mm] | 17.52103 E-3 |
| **Máx. momento flector** | Mb2 | [Nm] | 42.112 |
| Máx. res. esfuerzo | Res. | [N/mm2] | 1.2029 |
| Máx. deflexión | Sres | [mm] | 17.52103 E-3 |
| Máx. momento flector | Mbres | [Nm] | 42.112 |
| Escala de línea defl. | 6678:1 | | |
| Escala de línea de momento flector | 5.5566:1 | | |

Para determinar el esfuerzo cortante máximo se toma en cuenta el valor máximo de fuerza cortante de 174.30 N.

Entonces,

Ahora se determina los esfuerzos principales en el punto a partir de la ecuación:

Para obtener un factor de seguridad se debe relacionar este esfuerzo principal con el permisible del material.

Factor de seguridad

Al tener un factor elevado en el segundo análisis se recomienda en este caso elegir el menor pues será el más crítico del robot, por lo tanto, se considera que en la viga transversal el factor de seguridad es de 4

Este valor indica que el perfil resistirá adecuadamente la carga apoyada en este.

### Análisis esfuerzos viga longitudinal

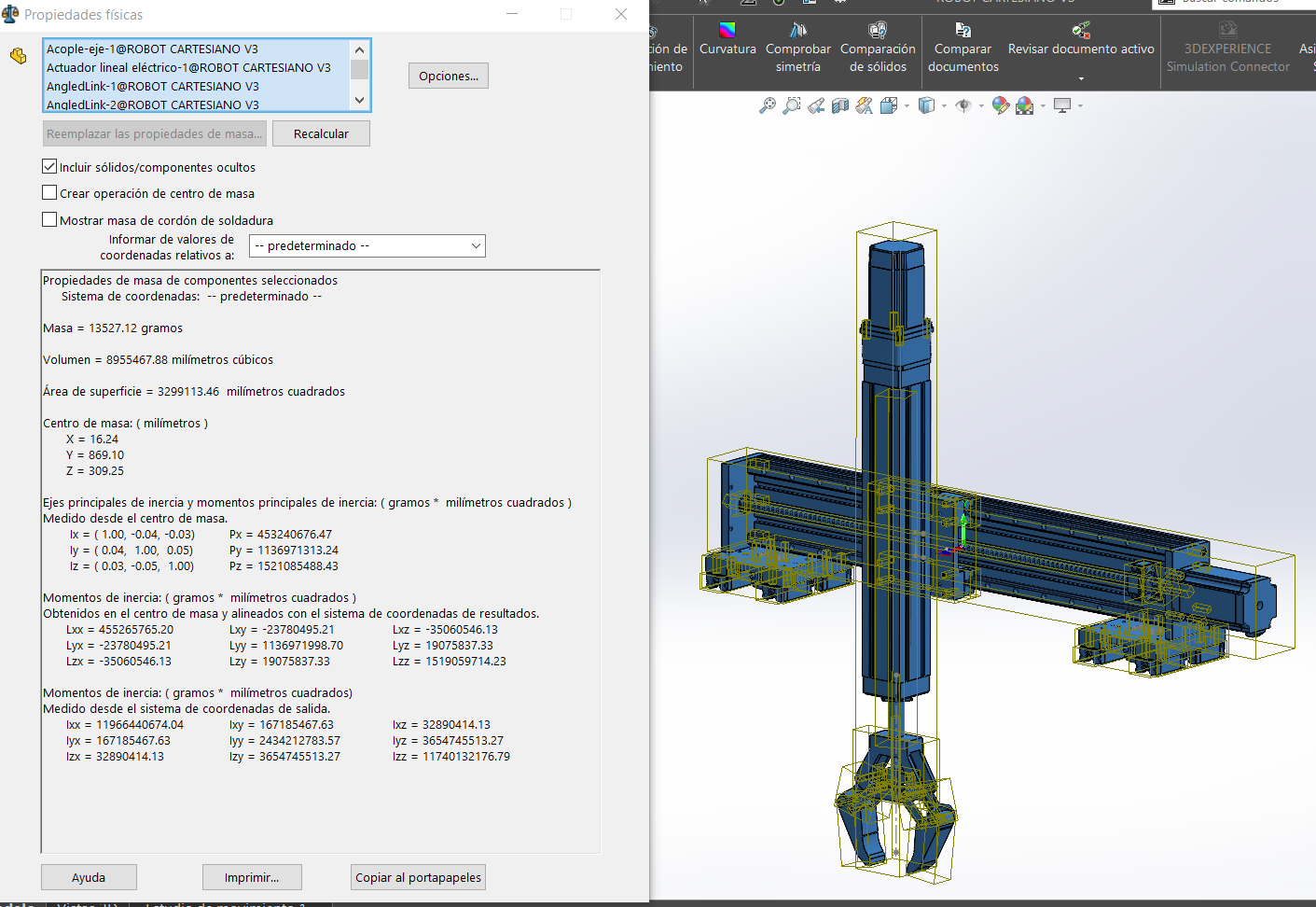


Fig. 34. Partes de movimiento transversal

Se considera el peso de los elementos seleccionados para continuar con el estudio estático del sistema.

Por lo tanto, para su análisis se los considera como un cuerpo apoyado y fijo en sus extremos

Rectángulo

Descripción generada automáticamente con confianza baja

Fig. 35. Diagrama sección transversal

La longitud total que este cuerpo va a recorrer en su desplazamiento es de 1.25 m.

Consideramos en primera instancia que nuestra parte vertical se encuentra en el centro y determinamos que carga vamos a tener en los carritos que desplaza la estructura.

Entonces la fuerza que ejercerá en cada lado de la viga es de:

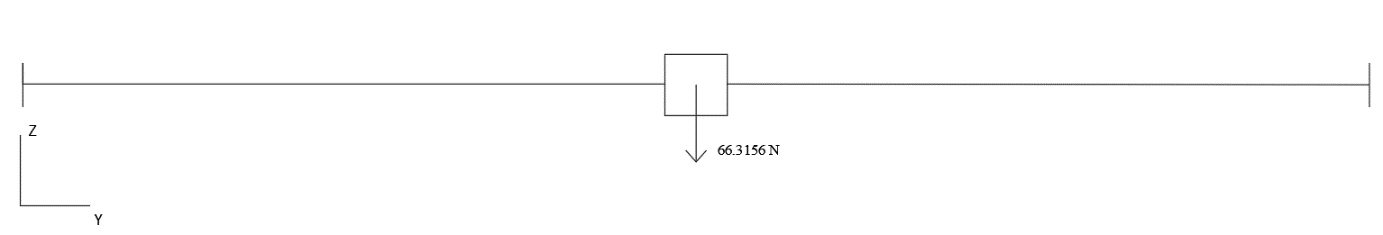


Fig. 36. Diagrama longitudinal

Para encontrar el trabajo realizado en un costado se determinará con el peso total del equipo en un solo lado del robot.

Por lo tanto, el trabajo requerido para poder mover el cuerpo transversal en cada lado es de 16.57 Nm.

#### Análisis en posición intermedia de la viga

Diagrama, Esquemático

Descripción generada automáticamente

Fig. 37. Ubicación del perfil

|  |  |  |
| --- | --- | --- |
| **Perfil de aluminio** | | |
| I1 | [mm4] | 2625780 |
| I2 | [mm4] | 117202,8 |
| Sc | [mm] | 15 |
| St | [mm] | 15 |
| A | [mm2] | 1256,81 |

Entonces el diagrama correspondiente a la ubicación del perfil y a la carga presente en todo el cuerpo se presenta en la siguiente figura:

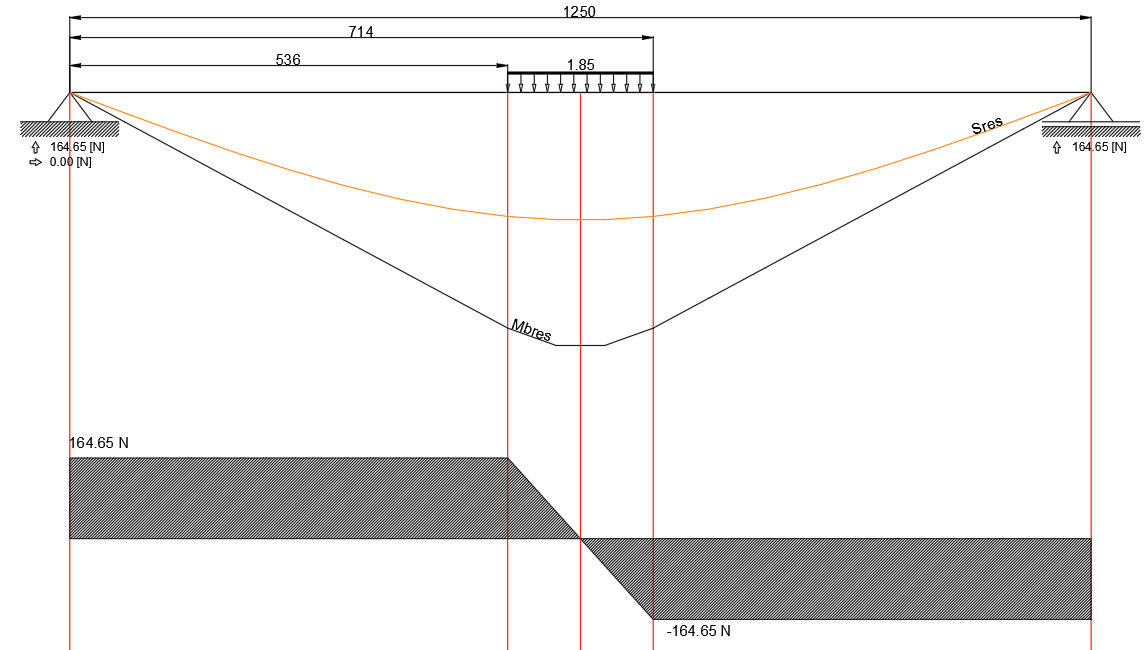


Fig. 38. Diagrama de flexión de la viga

Al realizar este cálculo se tiene como resultado los siguientes valores:

Tabla 11. Resultados análisis viga transversal

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultados de la viga** | | | |
| Momento de inercia | I1 | [mm4] | 2625780 |
| Momento de inercia | I2 | [mm4] | 117202,8 |
| Momento de inercia | Leff | [mm4] | 117203 |
| Máx. dist. al límite |  | [mm] | 15 |
| Factor de seguridad | 120.54 | | |
| Límites de estiramiento |  | [N/mm2] | 145 |
| Módulo E |  | (N/mm2] | 69000 |
| Material | 6063T5 | | |
| Máx. deflexión | S1 | [mm] | 1,640688 |
| **Máx. momento flector** | **Mbl** | **[Nm]** | **95,579** |
| Máx. deflexión | S2 | [mm] | 8,715833 E-18 |
| Máx. momento flector | Mb2 | [Nm] | 0,0108 E-12 |
| Máx. res. esfuerzo | Res. | [N/mm2] | 12,232 |
| Máx. deflexión | Sres | [mm] | 1,640688 |
| Máx. momento flector | Mbres | [Nm] | 95,579 |
| Escala de línea defl. | 6678:1 | | |
| Escala de línea de momento flector | 5.5566:1 | | |

Tomando como el momento flector máximo en el sentido del eje 1 corresponde a MB1.

Para determinar el esfuerzo cortante máximo se toma en cuenta el valor máximo de fuerza cortante de 164.65 N.

Entonces

Ahora se determina los esfuerzos principales en el punto a partir de la ecuación:

Para obtener un factor de seguridad se debe relacionar este esfuerzo principal con el permisible del material.

Factor de seguridad

Esto indica que tenemos una seguridad alta en este perfil por lo tanto resistiría aun más peso si fuese necesario o podríamos ocupar otro perfil de menor dimensión o espesor para soportar dichos pesos.

### Cálculo de Tornillo de Potencia

El número de tornillos de potencia que se utilizarán en el proyecto son 4, de los cuales 2 son para el movimiento longitudinal, 1 para el movimiento transversal y 1 para el movimiento vertical. Un tornillo de potencia es un dispositivo que se utiliza para cambiar el movimiento angular a movimiento lineal y por lo general, para transmitir potencia.

A continuación, se presentan los cálculos en base a al diseño en ingeniería mecánica de Shigley. Se determina la torsión requerida para superar la fricción de la rosca y mover la carga.

Donde:

F: fuerza sobre el tornillo

dm: Diámetro de paso

f: fricción

l: avance

Se determina el diámetro de paso mediante:

El valor *l,* corresponde a el mismo paso es decir 10 mm por vuelta.

El valor de *f* corresponde a un factor de fricción de pares roscados que se determina en la tabla 8-5 de Shigley.

Tabla

Descripción generada automáticamente con confianza media

El valor de la fuerza sobre el tornillo se determina considerando una vez que el robot comience a desplazar la placa, pues es aquí en donde el tornillo ejerce fuerza para vencer el peso de los objetos. Entonces, la fuerza que se considerara es la sumatoria de la fuerza debida a los pesos, la fricción de los carros y la fuerza de elevación.

Donde:

Wc: corresponde al peso del objeto de levantar.

Wt: peso de los elementos estructurales

Fr: fuerza de fricción de los carros

Entonces el valor de torque necesario para mover la carga calculada es:

Se calcula el esfuerzo cortante nominal en torsión t del cuerpo del tornillo

Donde:

dr: es el diámetro menor

Cálculo del esfuerzo cortante nominal de torsión

Cálculo del esfuerzo axial en el cuerpo del tornillo debido a la carga F

## Componentes electrónicos

### Cálculo del torque del motor de pasos

El torque para el eje z se toma al elevar una cara de 20 kg

La distancia es tonada del radio del tornillo de potencia

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

El motor seleccionado es un motor de pasos nema 23 serie WS23–0150–30–4 el cual cumplen con el torque necesario para elevar la carga de 20kg

Gráfico, Gráfico de líneas

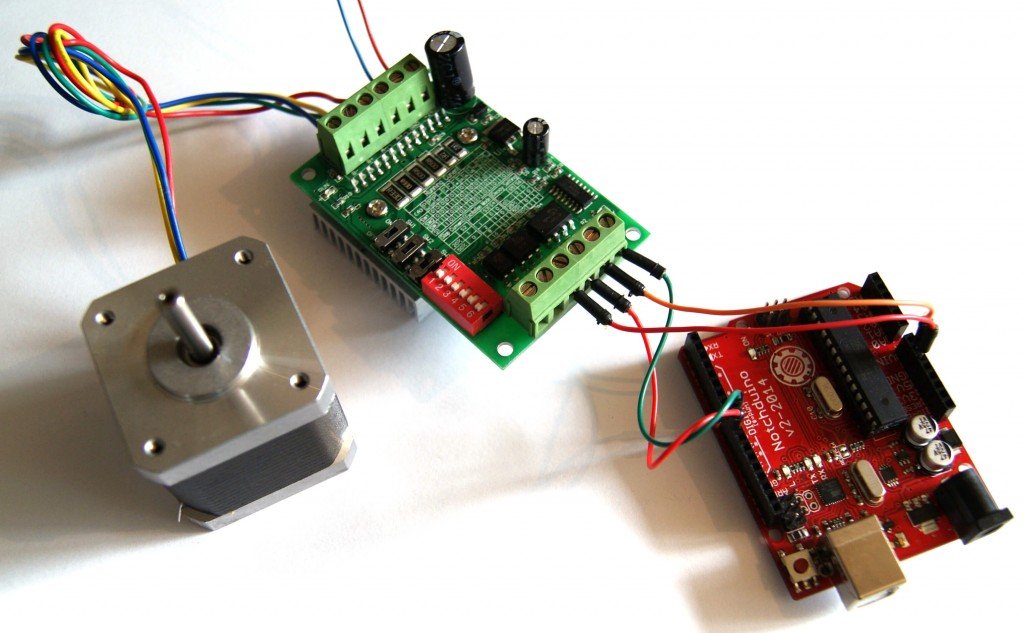
Descripción generada automáticamente

### Descripción de los elementos de control del sistema

#### Selección del controlador para el motor de pasos

**TB6560 Controlador Motor a Pasos**

La tarjeta con el integrado TB6560 es un controlador de motores. Funciona al proporcionar un pulso digital de trabajo y dirección. Podemos usar la tarjeta para el control de un motor a pasos como el NEMA 23 HS8401



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Las Entradas a 5V “EN- EN+ CW- CW+ CLK- CLK+” en caso de requerir operarlo a más de 5V, requiere colorar una resistencia de 1k para 12V y 2.4 para 24V máximo. El + significa entrada el – significa salida, en este caso conectado a GND

### Descripción de los sensores del sistema

#### Selección de sensores de final de carrera

El tipo de sensor de carrera es uno de accionamiento por muelle en el cual cambia de giro al motor para que retome a un punto indicado en la programación



Datos tecnicos del sensor:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Controlador lógico programable (PLC) con expansión**

**SIMATIC S7-1200**



|  |
| --- |
| **Características** |
| 6ES7214-1AG40-0XB0 |
| Alimentación 20,4-28,8VDC |
| 14 DI 24 V DC; 10 DO 24VDC; 2 AI 0-10 V DC |
| Interfaz Ethernet |
| servidor web integrado |

Diagrama

Descripción generada automáticamente

|  |  |
| --- | --- |
|  | |
| **1** | |  |  | | --- | --- | |  | Alimentación de sensores 24 V DC |   Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores |
| **2** | Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M". |
|  | El borne L1 o N (L2) se puede conectar a una fuente de tensión de hasta 240 V AC. El borne se puede considerar L2 y no es necesario que esté puesto a tierra. No se necesita polarización para los bornes L1 y N (L2). |
|  | Información acerca del puerto Ethernet de la CPU. |

**Actuadores neumáticos tipo pinzas de mordaza**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Estas pinzas vienen articuladas con sensores rasantes instalados en la propia pinza el Angulo de apertura es de 30 grados

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama

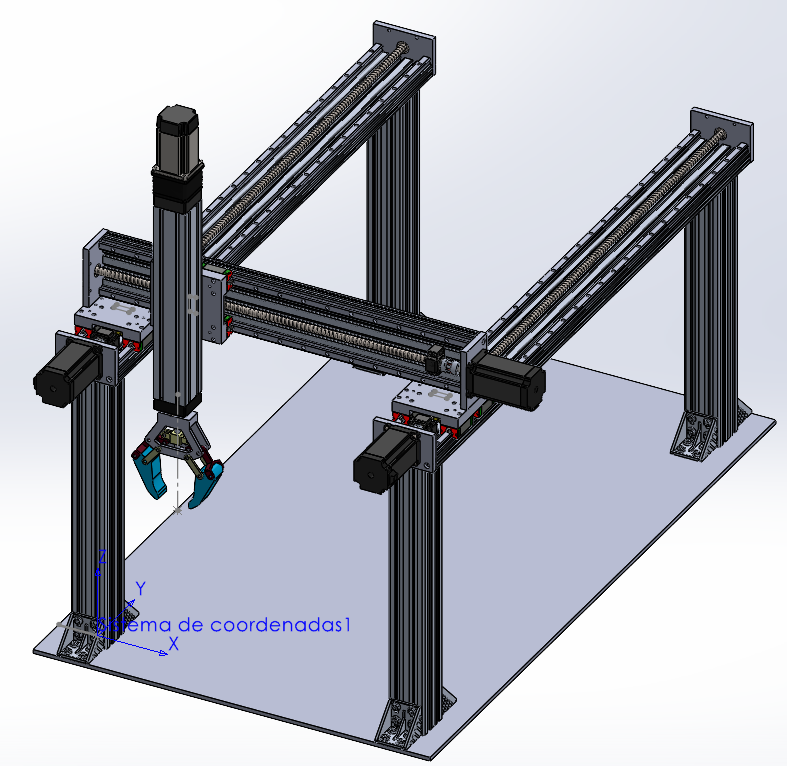
Descripción generada automáticamente

# CAPÍTULO IV

# RESULTADOS

## Modelamiento en SOLIDWORKS

A continuación, se desarrollará el robot cartesiano en el Software SolidWorks, para el cual en primera instancia se llevará a cabo el modelamiento de cada una de las piezas de manera individual, luego se procederá a realizar el ensamble de cada una de las piezas constitutivas del robot, después de haber realizado todo este procedimiento se tendrá la siguiente Figura como se muestra a continuación:



## Resultados del análisis cinemático en MATLAB

Como resultado en el análisis de la cinemática y dinámica en Matlab obtenemos graficas tanto de posición y velocidad del robot cartesiano.

Gráfico

Descripción generada automáticamente

Fig. 39. Gráfica de posición de articulaciones

Está gráfica representa como las articulaciones de desplazan al realizar la trayectoria establecida. Para la articulación 1, la cual es la articulación que se desplaza longitudinalmente tiene un rango de movimiento entre los intervalos que se pueden ver en las líneas de color azul y en un tiempo establecido. De igual forma también se encuentran establecidas las dos articulaciones siguientes.

Diagrama

Descripción generada automáticamente

Fig. 40. Gráfica de velocidades de las articulaciones

En el gráfico generado por MATLAB, se presenta una representación clara de las velocidades de las articulaciones de un robot cartesiano en función del tiempo. La articulación 1, representada por una línea de color azul, muestra una velocidad máxima de 0.025 m/s, indicando movimientos relativamente suaves y lentos. En contraste, la articulación 2, destacada en rojo, alcanza una velocidad significativamente mayor de 0.25 m/s, lo que sugiere movimientos más rápidos y dinámicos en comparación con las otras articulaciones. Finalmente, la articulación 3, marcada en amarillo, presenta una velocidad de 0.15 m/s, ubicándose entre las velocidades de las articulaciones 1 y 2. Este gráfico proporciona una visión detallada de cómo varían las velocidades de las distintas articulaciones del robot a lo largo del tiempo, permitiendo una evaluación precisa del comportamiento dinámico del sistema robótico.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Fig. 41. Gráfica de fuerza

Esta grafica presenta la fuerza necesaria que necesitaría ser aplicada en el efector final para poder levantar el cuerpo y su peso.

## Validación mediante software (CAE)

Para validar el diseño estructural por medio de elementos finitos se lo hará mediante el software de ANSYS. Para ello primero creamos nuestro material que vamos a utilizar, el cual es el aluminio 6063 T5 propio de la viga.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Fig. 42. Material se 6063-T5

Se debe realizar la importación del modelo de la viga y analizar la calidad de malla pues una malla de alta calidad con elementos bien formados puede representar mejor la geometría y las condiciones de contorno del modelo, lo que conduce a resultados más precisos.

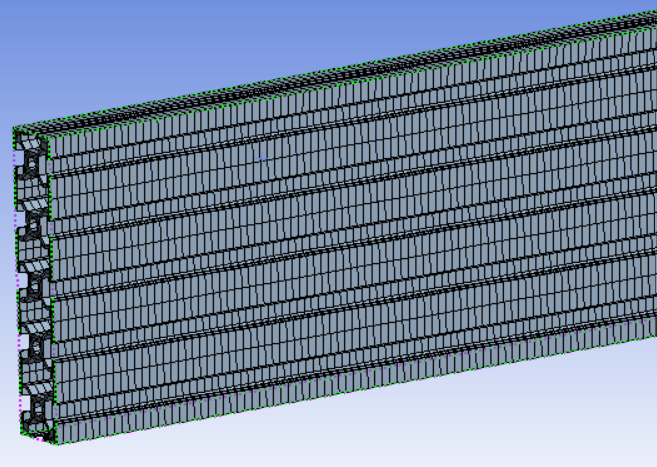


Fig. 43. Mallado de viga

Entonces valorando la calidad de malla con el método de Element Qualy los resultados que se obtienen son:

Tabla

Descripción generada automáticamente

Fig. 44. Calidad de malla

Lo cual muestra una malla adecuada para obtener resultados altamente confiables.

Gráfico

Descripción generada automáticamente con confianza media

Fig. 45. Análisis de la deformación

La simulación de ANSYS ha demostrado que la viga de aluminio 6063 T5 presenta una deformación máxima de 0.021 mm bajo las condiciones de carga y restricciones especificadas. Esta pequeña deformación indica que el material y el diseño de la viga son adecuados para soportar las cargas aplicadas sin sufrir deformaciones significativas. El aluminio 6063 T5, conocido por su buena resistencia mecánica y a la corrosión, ha mostrado ser una elección eficiente y confiable para aplicaciones estructurales donde se requieren alta precisión y durabilidad. Estos resultados validan el uso del aluminio 6063 T5 para la fabricación de vigas en aplicaciones similares, asegurando un rendimiento estructural seguro y eficiente.

Imagen que contiene Diagrama

Descripción generada automáticamente

Fig. 46. Análisis del factor de seguridad

Además, el análisis ha revelado un factor de seguridad de 15, lo que significa que la viga puede soportar hasta 15 veces la carga aplicada antes de alcanzar el punto de fallo. Este alto factor de seguridad proporciona una gran margen de confiabilidad y garantiza que la viga no solo es adecuada para las condiciones de carga actuales, sino que también tiene una capacidad considerable para soportar cargas adicionales imprevistas o condiciones extremas.

Para analizar el elemento de la viga transversal se estudia utilizando la carga misma estudiada anteriormente y enlazando las propiedades del material del estudio de la viga transversal.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fig. 47. Enlace de estudio

De igual forma se evalúa la calidad de malla de esta viga obteniendo valores similares a los obtenido anteriormente y luego se analiza tanto la deformación máxima que sufre la viga y su factor de seguridad correspondiente.

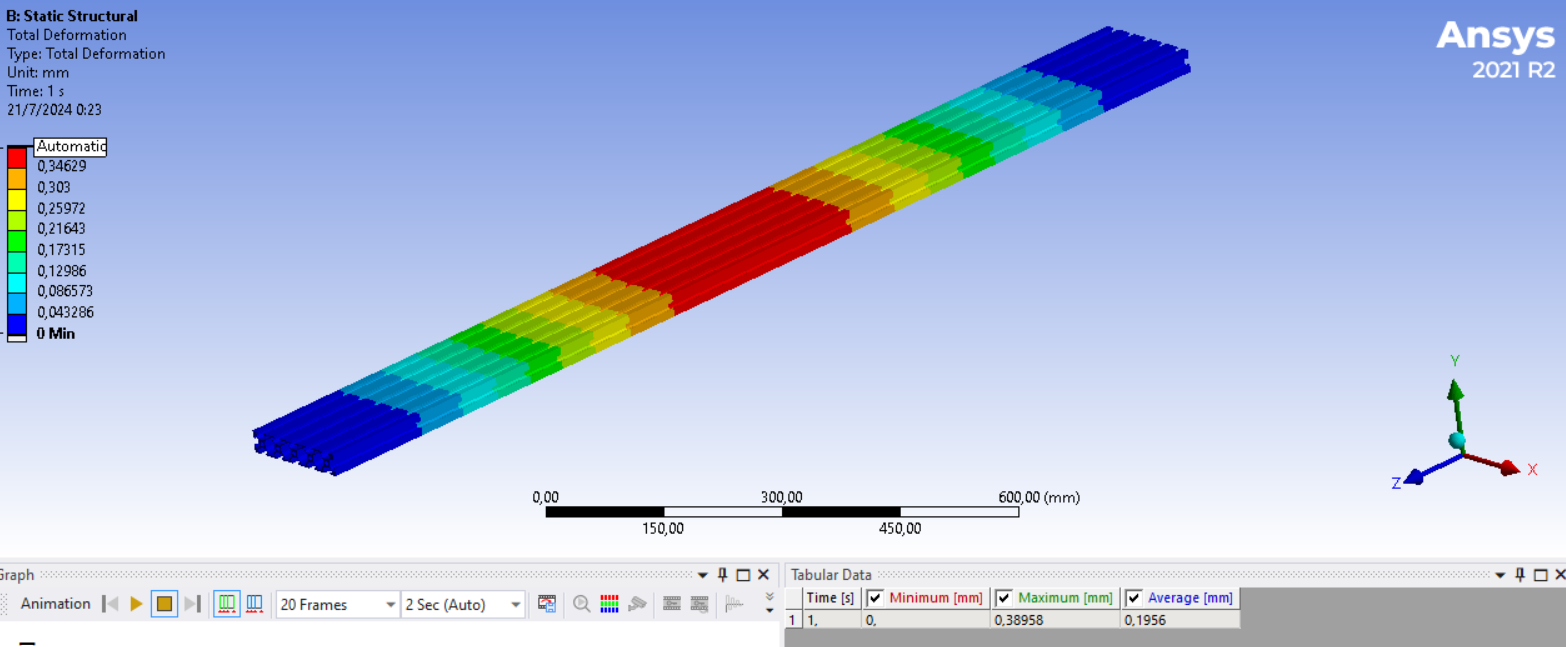


Fig. 48. Análisis de viga longitudinal

En la simulación de ANSYS realizada para una viga de aluminio 6063 T5 con una longitud de 1500 mm y fijada en ambos extremos, se obtuvo una deformación máxima de 0.389 mm bajo las condiciones de carga aplicadas. Este resultado indica que, aunque la viga experimenta una cierta deformación, esta es relativamente pequeña en comparación con la longitud total de la viga. La relación entre la deformación máxima y la longitud de la viga (0.389 mm / 1500 mm) es baja, lo que sugiere que el material y el diseño de la viga son adecuados para soportar las cargas aplicadas sin comprometer significativamente su integridad estructural.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Fig. 49. Factor de seguridad de la viga longitudinal

Además, el análisis ha revelado un factor de seguridad de 15, lo que significa que la viga puede soportar hasta 15 veces la carga aplicada antes de alcanzar el punto de fallo. Este alto factor de seguridad proporciona un amplio margen de confiabilidad y asegura que la viga no solo es adecuada para las condiciones de carga actuales, sino que también tiene una capacidad considerable para soportar cargas adicionales imprevistas o condiciones extremas.

## Simulación ROS – Gazebo

El sistema operativo robótico (ROS 2) desempeñó un papel fundamental en el diseño y funcionamiento de robots cartesianos fabricados con perfiles de aluminio 6063 T5 y servomotores. ROS 2 es una plataforma middleware ampliamente utilizada en robótica y tiene varias ventajas que pueden mejorar significativamente la funcionalidad y eficiencia de los robots. Estos son los principales beneficios de usar ROS 2 para este proyecto.

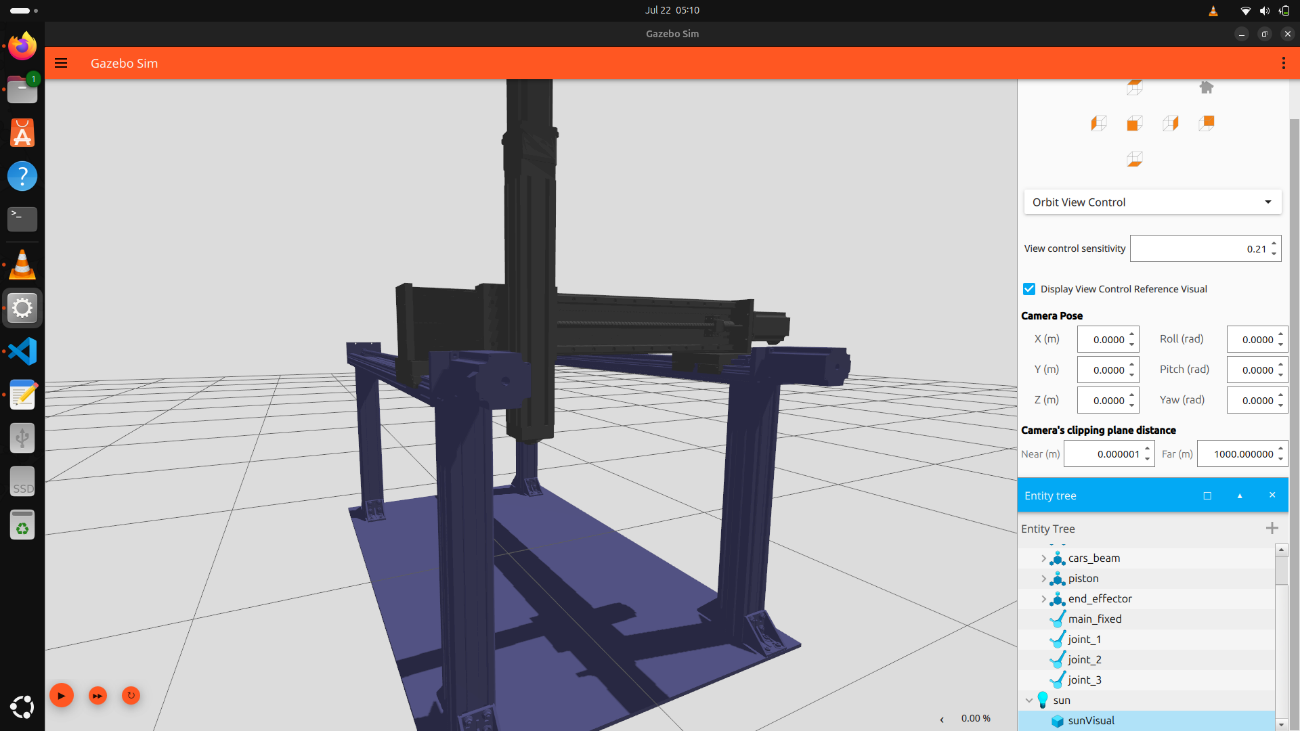


Fig. 50.Entorno de simulación de GAZEBO

Para que se pueda simular un entorno en gazebo se debe realizar un código XML en formato sdf. Estos archivos describen los modelos y el mundo en el que se simulan. Los archivos SDF son muy detallados y pueden describir la geometría, la física, los sensores, las luces, y otros aspectos del entorno simulado. Y XML es un lenguaje de marcado que define un conjunto de reglas para codificar documentos en un formato que es legible tanto por humanos como por máquinas.

## Costos

El diseño de un robot cartesiano fabricado con perfiles de aluminio 6063 T5 y equipado con servomotores ha dado lugar a un análisis detallado de los costes implicados. Los perfiles de aluminio 6063 T5, seleccionados por su excelente relación resistencia-peso y resistencia a la corrosión, representaron una parte importante del coste total debido a su calidad y durabilidad. Por otro lado, los servomotores, imprescindibles para el movimiento preciso y controlado del robot, también contribuyeron considerablemente al presupuesto, debido a la necesidad de componentes de alta precisión y fiabilidad.

Tabla 12. Costos aproximados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Materiales** | **Unidades** | **Cantidad** | **Costo unitario** | **Sub. Total** |
| Perfil de aluminio T-slot 6063 T5 30x180 | 1500 mm | 2 | 79,5 | 159 |
| Perfil de aluminio T-slot 6063 T5 30x180 | 1000 mm | 1 | 53,00 | 53 |
| Perfil de aluminio 6063 T5 80x80 | 500 mm | 1 | 59,35 | 59,35 |
| Bloque HGH20CA | U | 12 | 31,8 | 381,6 |
| Riel lineal / Guía lineal HGR20 | 1500 mm | 4 | 95,4 | 381,6 |
| Riel lineal / Guía lineal HGR20 | 1000 mm | 2 | 63,6 | 127,2 |
| Acople flexible de aluminio tipo araña - 10 mm a 14mm | U | 3 | 30 | 90 |
| Soporte final BK12 | U | 3 | 25,57 | 76,71 |
| Montaje de aluminio para tuerca SFU | U | 3 | 15,4 | 46,2 |
| SFU1610 Tornillo de bolas 1500 mm | U | 2 | 145 | 290 |
| SFU1610 Tornillo de bolas 1000 mm | U | 1 | 130 | 130 |
| Kit Servomotor + Cableado + Driver RTA 750W (3000-6000rpm, 2.390N.m) | U | 1 | 2000 | 2000 |
| Actuador lineal eléctrico | U | 1 | 160 | 160 |
| **Total** | | | | 3954,66 |

# CAPÍTULO V

# Conclusiones y recomendaciones

## Conclusiones

Se concluye que el robot cartesiano puede soportar cargas más de 20 kg pues los factores de seguridad calculados son mayores lo cual indicaría que para la aplicación de 20 kg estaría siendo sobredimensionado. Además, el robot cartesiano hace uso de dos husillos para el movimiento transversal para que exista un desplazamiento eficiente.

El área de trabajo establecida fue el punto de partida para saber las longitudes generales del robot en ancho, largo y profundidad. Y cuyo modelamiento 3D fue clave para poder exportar dicho modelo al entorno visual de GAZEBO por medio de ROS

El análisis cinemático y dinámico fue desarrollado en el software MATLAB pues permitió resolver las matrices y cadenas cinemáticas del robot además de ofrecer un código base de trayectoria que permite visualizar como se desplaza las articulaciones en el entorno visual de GAZEBO.

ROS 2 fue necesario para desarrollar todo el entorno visual mediante lenguaje XML en un archivo tipo SDF. Aquí se plantea y se configura el mundo de simulación y que valores de masa e inercias van a tener los cuerpos. Además, también plantea todas las características del mundo como la luz, sombras etc.

# Recomendaciones

Para poder transportar 20 kg se deben elegir materiales resistentes y a la vez ligeros, como aleaciones de aluminio o aceros especiales. El diseño del robot debe ser robusto para reducir la flexión y la vibración durante el funcionamiento. Un diseño bien pensado incluirá refuerzos estratégicos y una base sólida para garantizar la estabilidad. Además, es importante tener en cuenta los factores de seguridad y dimensiones de los componentes estructurales para soportar una carga mayor a 20 kg y así evitar daños mecánicos en circunstancias imprevistas.

# BIBLIOGRAFÍA

Badillo, M. (2016). *Control de trayectorias de un robot cartesiano* [Universidad Politécnica de Tulancingo]. http://www.upt.edu.mx/contenido/Investigacion/Contenido/TESIS/MAC/2016/MAC\_T\_2016\_02\_MBN.pdf

Carvajal, J., & Godoy, R. (2019). Proyecto mecatrónico de brazo robot cartesiano integrado a una celda de almacenamiento y recuperación automatizada AS / RS de un Sistema Flexible de Manufactura FMS. *ITECKNE: Innovación e Investigación en Ingeniería*, *6*(1), 35-49. https://dialnet.unirioja.es/servlet/articulo?codigo=4991533

Castellanos, P. (2024). *Robots cartesianos: Definición, tipos y funcionamiento en la Industria*. Manufactura Latam. https://www.manufactura-latam.com/es/noticias/robots-cartesianos-definicion-tipos-y-funcionamiento-en-la-industria

Contreras, K. (2021). *Patogenicidad bacteriana II*. 28. https://fmed.uba.ar/sites/default/files/2020-02/Seminario%203%202020.pdf

Godoy, R., & Quintero, W. (2017). *Diseño y modelamiento de un robot cartesiano para el posicionamiento de piezas* [Universidad de la Salle]. https://ciencia.lasalle.edu.co/ing\_automatizacion/50

Jara, M., & Pacheco, M. (2014). *Diseño e implementación de un robot cartesiano, para el montaje de tapa y/o pasador, en el proceso de paletizado* [bachelorThesis]. http://dspace.espoch.edu.ec/handle/123456789/3265

Mepal, P. (2020, noviembre 3). *ROS – Un sistema operativo diseñado para robots* [Autracen]. Autracen Soluciones Industriales 4.0. https://www.autracen.com/blog/viajes-1/ros-un-sistema-operativo-disenado-para-robots-67

Moran, Y. (2016). *Estudio cinemático y dinámico de un robot dedicado a la reparación de fallas menores en tuberías hidrosanitarias*. https://repository.unimilitar.edu.co/bitstream/handle/10654/15189/FallaMoranYeudielFerney\_SacristanGutierrezMariaCamila2016.pdf?sequence=1&isAllowed=y

Muñoz, J. (2017). *Propuesta para la clasificación y paletizado automático de productos en la planta de grupo familia Medellín* [Universidad Tecnológica de Pereira]. https://hdl.handle.net/11059/7825

Sellés, R. (2016). *Diseño, cálculo y construcción de un robot cartesiano para manipulación* [Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València]. https://riunet.upv.es/handle/10251/74157

Val, C. (2019). *Robot Cartesiano de tres ejes controlado por computadora* [Pontífica Universidad Católica Argentina]. https://repositorio.uca.edu.ar/bitstream/123456789/11220/1/robot-cartesiano-tres-ejes.pdf

# ANEXOS

ANEXO 1 – CÓDIGO MATLAB

clc; clear variables

%% Cargar directorios

addpath("HTM/");

addpath("../ROS/");

%% Definir los atributos del robot (simbólicos o numéricos)

% Variables de las articulaciones

theta = sym("theta\_", [3, 1]);

dtheta = sym("dtheta\_", [3, 1]);

ddtheta = sym("ddtheta\_", [3, 1]);

% Valores deseados de las articulaciones

dtheta\_d = sym("dtheta\_d\_", [3, 1]);

ddtheta\_d = sym("ddtheta\_d\_", [3, 1]);

% Variables del efector final

X = [sym("x"); sym("y"); sym("z")];

dX = [sym("dx"); sym("dy"); sym("dz")];

ddX = [sym("ddx"); sym("ddy"); sym("ddz")];

% Valores deseados del efector final

dX\_d = [sym("dx\_d"); sym("dy\_d"); sym("dz\_d")];

ddX\_d = [sym("ddx\_d"); sym("ddy\_d"); sym("ddz\_d")];

% Parámetros físicos

g = [0 0 -9.80665]';

m = sym("m\_", [3, 1]);

m\_p = sym("m\_p");

L = sym("l\_", [4, 1]);

dt = 0.01;

% Parámetros de control

syms e\_theta\_1(t) e\_theta\_2(t) e\_theta\_3(t) e\_x(t) e\_y(t) e\_z(t) t

e = [ e\_theta\_1(t); e\_theta\_2(t); e\_theta\_3(t); diff([e\_theta\_1(t); e\_theta\_2(t); e\_theta\_3(t)])];

de = diff(e);

e\_c = [e\_x(t); e\_y(t); e\_z(t); diff([e\_x(t); e\_y(t); e\_z(t)])];

de\_c = diff(e\_c);

K = sym("k\_", [6, 6]);

K\_c = sym("k\_c\_", [6, 6]);

%% Cinemática del robot

% Parámetros de Denavit-Hartenberg

DH = denavitHartenberg(theta, L);

% Cinemática directa usando matrices de transformación homogéneas

H = forwardKinematicsHTM(DH, 5);

%% Cinemática diferencial del robot

% Matriz Jacobiana geométrica usando matrices de transformación homogéneas

J\_g = jacobianMatrixHTM(DH, theta, 5);

% Verificar si se puede realizar la cinemática inversa numérica

if isnumeric(theta)

% Cinemática inversa usando la matriz Jacobiana geométrica

theta\_htm = inverseKinematicsHTM(zeros(3, 1), L, H(1 : 3, 4), eye(3), 5, dt);

% Cinemática inversa usando matrices de transformación homogéneas

[n, ~] = size(theta);

legends = string(zeros(1, n));

figure()

hold on;

for i = 1 : n

plot(theta\_htm(i, :), 'LineWidth', 1)

legends(1, i) = string(append("$\theta\_", num2str(i), "$"));

end

title("Inverse Kinematics using Geometric Jacobian Matrix", 'Interpreter', 'latex')

xlabel("Samples $\left[ k \right]$", 'Interpreter', 'latex')

ylabel("Amplitude $\left[ meters \right]$", 'Interpreter', 'latex')

legend(legends, 'Interpreter', 'latex', 'FontSize', 12)

grid on;

hold off;

end

%% Dinámica del robot

% Invertible por la izquierda:

% filas > columnas

%

% Invertible por la derecha:

% filas < columnas

%

% Invertible por ambos lados:

% filas = columnas

% Matriz de masa usando matrices de transformación homogéneas

D = massMatrixHTM(m, DH, theta);

% Matriz de efectos centrífugos y de Coriolis usando matrices de transformación homogéneas

C = centrifugalCoriolis(D, theta, dtheta);

% Efectos gravitacionales usando matrices de transformación homogéneas

G = gravitationalEffectsHTM(m, g, DH, theta);

% Torques generados por las placas

tau\_p = m\_p \* transpose(J\_g) \* g;

% Modelo dinámico en el espacio de estados

A = [zeros(3, 3) eye(3)

zeros(3, 3) -D \ C];

x = [theta; dtheta];

Phi = [zeros(3, 1); D \ (tau\_p - G)];

B = [zeros(3, 3) zeros(3, 3)

zeros(3, 3) inv(D)];

u = [zeros(3, 1); sym("tau\_", [3, 1])];

% Model

dx = (A \* x) + Phi + (B \* u);

%% Dinámica del error de las articulaciones

% Control de articulaciones

u = pinv(B) \* ([dtheta\_d; ddtheta\_d] + (K \* e) - (A \* x) - Phi);

% Dinámica del error

f\_e = subs(de - [dtheta\_d; ddtheta\_d] + (A \* x) + Phi + (B \* u), [dtheta(1) - dtheta\_d(1); dtheta(2) - dtheta\_d(2); dtheta(3) - dtheta\_d(3)], -diff(e(1 : 3, :)));

% Simplificación de la dinámica del error

f\_e = subs(f\_e, [K(4, 2), K(4, 3), K(4, 5), K(4, 6), K(5, 1), K(5, 3), K(5, 4), K(5, 6), K(6, 1), K(6, 2), K(6, 4), K(6, 5)], zeros(1, 12)) == 0;

%% Control

% Parámetros físicos

m\_p = 20;

m = [7.538; 4.211; 1.202];

L = [0.755 0.1269 0.083 0.083]';

% Parámetros de las articulaciones

theta\_0 = zeros(3, 1);

dtheta\_0 = zeros(3, 1);

%% Ganancias

% % Subamortiguado

% k\_d = 2;

% k\_p = (k\_d^2 / 4) + 7;

% Críticamente amortiguado

k\_d = 7;

k\_p = (k\_d^2 / 4);

% % Sobreamortiguado

% k\_d = 2;

% k\_p = (k\_d^2 / 4) - 1;

% Matrices de ganancias

K = [zeros(3, 3) zeros(3, 3)

k\_p 0 0 k\_d 0 0

0 k\_p 0 0 k\_d 0

0 0 k\_p 0 0 k\_d];

% K\_c = [zeros(3, 3) zeros(3, 3)

% k\_p 0 0 k\_d 0 0

% 0 k\_p 0 0 k\_d 0

% 0 0 k\_p 0 0 k\_d];

%% Planificación de trayectorias (para las articulaciones)

% Parámetros para las placas

d = 2.54 / 200; % Plate's thickness. 1/2" to meters

d\_r = 1 / 100; % Base de la placa. 1 centímetro a metros

h\_s = 1 / 100; % Altura de seguridad

n = 7; % Número de placas

% Coordenadas euclidianas

x\_1 = 0.2;

y\_1 = 0.2;

z\_1 = 0.2;

% Coordenadas euclidianas

x\_2 = 0.4;

y\_2 = 0.4;

% Inicializar variable para almacenar las trayectorias

theta\_d = theta\_0;

theta\_d\_gz = zeros(3, 7 \* n);

% Itera a través de todas las placas

for i = 1 : n

% Punto de partida: (x\_1, y\_1, z\_1)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_1 y\_1 z\_1]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 6) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

% Bajar para una placa: (x\_1, y\_1, z\_1 - i \* (d + d\_r) - h\_s)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_1 y\_1 z\_1 - i \* (d + d\_r) - h\_s]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 5) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

%{

\*\*\* EL AGENTE DEBE SER ACTIVADO EN ESTE PUNTO PARA TRANSPORTAR LA i-ésima PLACAE \*\*\*

%}

% Subir con la placa: (x\_1, y\_1, z\_1)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_1 y\_1 z\_1]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 4) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

% Moverse a la parte superior de la columna de placas: (x\_2, y\_2, z\_1)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_2 y\_2 z\_1]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 3) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

% Bajar para alcanzar el punto más cercano de descarga de las placas: (x\_2, y\_2, z\_1 + (i - n) \* (d + d\_r) + h\_s)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_2 y\_2 z\_1 + (i - n) \* (d + d\_r) + h\_s]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 2) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

% Alcanzar el punto de descarga de las placas: (x\_2, y\_2, z\_1 + (i - n) \* (d + d\_r))

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_2 y\_2 z\_1 + (i - n) \* (d + d\_r)]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 1) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

%{

\*\*\* EL IMÁN DEBE SER DESACTIVADO EN ESTE PUNTO PARA DESCARGAR LA i-ésima PLACA \*\*\*

%}

% Moverse a la parte superior de la columna de placas: (x\_2, y\_2, z\_1)

theta\_d = [theta\_d(:, :) inverseKinematicsHTM(theta\_d(:, end), L, [x\_2 y\_2 z\_1]', eye(3), 5, 0.001)]; % For the controller (full trajectory)

theta\_d\_gz(:, 7 \* i - 0) = theta\_d(:, end); % For Gazebo (partial trajectory, only keypoints)

end

% Calcular las derivadas de las trayectorias

dtheta\_d = gradient(theta\_d);

ddtheta\_d = gradient(dtheta\_d);

% Guardar el comando para Gazebo con los puntos calculados

file = fopen('../ROS/gazebo\_command.txt', 'w');

fprintf(file, append('gz topic -t "/model/euclidian\_robot/joint\_trajectory" -m gz.msgs.JointTrajectory -p ', "'joint\_names: ", '"joint\_1"; joint\_names: "joint\_2"; joint\_names: "joint\_3";'));

% Iterar a través de todos los puntos

for i = 1 : 7 \* n

% Agregar datos al comando

fprintf(file, "points { positions: %f; positions: %f; positions: %f; time\_from\_start { sec: %d; nsec: 0 } }; ", theta\_d\_gz(1, i), theta\_d\_gz(2, i), theta\_d\_gz(3, i) - 1.1075, i);

end

fprintf(file, "'");

% Cerrar archivo

fclose(file);

%% Preliminares de la simulación

% Tiempo de simulación (segundos)

t = 1500;

% Número de muestras para la simulación

[~, k] = size(theta\_d);

% Transformar en series temporales

theta\_d = timeseries( theta\_d, linspace(0, t, k));

dtheta\_d = timeseries(dtheta\_d, linspace(0, t, k));

ddtheta\_d = timeseries(ddtheta\_d, linspace(0, t, k));

%% Simulación (modelo de articulaciones)

% Simulate

sim("joints\_2022b.slx");

%% Gráficas

% Posiciones de las articulaciones

figure()

hold on;

plot(x.time, reshape(x.data(1, :, :), 1, []), 'LineWidth', 1)

plot(x.time, reshape(x.data(2, :, :), 1, []), 'LineWidth', 1)

plot(x.time, reshape(x.data(3, :, :), 1, []), 'LineWidth', 1)

plot(theta\_d, '--', 'LineWidth', 1)

title("Position of each Joint", 'Interpreter', 'latex')

xlabel("Time $\left[ sec \right]$", 'Interpreter', 'latex')

ylabel("Amplitude $\left[ meters \right]$", 'Interpreter', 'latex')

legend(["$\theta\_1 \left( t \right)$", "$\theta\_2 \left( t \right)$", "$\theta\_3 \left( t \right)$", "$\theta\_{d\_1} \left( t \right)$", "$\theta\_{d\_2} \left( t \right)$", "$\theta\_{d\_3} \left( t \right)$"], 'Interpreter', 'latex', 'FontSize', 12)

grid on;

hold off;

% Velocidades de las articulaciones

figure()

hold on;

plot(x.time, reshape(x.data(4, :, :), 1, []), 'LineWidth', 1)

plot(x.time, reshape(x.data(5, :, :), 1, []), 'LineWidth', 1)

plot(x.time, reshape(x.data(6, :, :), 1, []), 'LineWidth', 1)

plot(dtheta\_d, '--', 'LineWidth', 1)

title("Velocity of each Joint", 'Interpreter', 'latex')

xlabel("Time $\left[ sec \right]$", 'Interpreter', 'latex')

ylabel("Amplitude $\left[ \frac{m}{s} \right]$", 'Interpreter', 'latex')

legend(["$\dot{\theta}\_1 \left( t \right)$", "$\dot{\theta}\_2 \left( t \right)$", "$\dot{\theta}\_3 \left( t \right)$", "$\dot{\theta}\_{d\_1} \left( t \right)$", "$\dot{\theta}\_{d\_2} \left( t \right)$", "$\dot{\theta}\_{d\_3} \left( t \right)$"], 'Interpreter', 'latex', 'FontSize', 12)

grid on;

hold off;

% Fuerzas del efector final

figure()

hold on;

plot(force\_c.time, reshape(force\_c.data(1, :, :), 1, []), '-r', 'LineWidth', 1)

plot(force\_c.time, reshape(force\_c.data(2, :, :), 1, []), '-g', 'LineWidth', 1)

plot(force\_c.time, reshape(force\_c.data(3, :, :), 1, []), '-b', 'LineWidth', 1)

title("Force at the End Effector", 'Interpreter', 'latex')

xlabel("Time $\left[ sec \right]$", 'Interpreter', 'latex')

ylabel("Amplitude $\left[ Newtons \right]$", 'Interpreter', 'latex')

legend(["$f\_x \left( t \right)$", "$f\_y \left( t \right)$", "$f\_z \left( t \right)$"], 'Interpreter', 'latex', 'FontSize', 12)

grid on;

hold off;

% Función de error

figure()

plot(e, 'LineWidth', 1)

title("Error Function", 'Interpreter', 'latex')

xlabel("Time $\left[ sec \right]$", 'Interpreter', 'latex')

ylabel("Amplitude", 'Interpreter', 'latex')

legend(["$e\_{\theta\_1} \left( t \right)$", "$e\_{\theta\_2} \left( t \right)$", "$e\_{\theta\_3} \left( t \right)$", "$\dot{e}\_{\theta\_1} \left( t \right)$", "$\dot{e}\_{\theta\_2} \left( t \right)$", "$\dot{e}\_{\theta\_3} \left( t \right)$"], 'Interpreter', 'latex', 'FontSize', 12)

grid on;

% Función de control

figure()

plot(force, 'LineWidth', 1)

title("Control Function", 'Interpreter', 'latex')

xlabel("Time $\left[ sec \right]$", 'Interpreter', 'latex')

ylabel("Amplitude $\left[ Newtons \right]$", 'Interpreter', 'latex')

legend(["$f\_{\theta\_1} \left( t \right)$", "$f\_{\theta\_2} \left( t \right)$", "$f\_{\theta\_3} \left( t \right)$"], 'Interpreter', 'latex', 'FontSize', 12)

grid on;

ANEXO 2 – CÓDIGO ROS 2

<?xml version="1.0"?>

<sdf version="1.10">

    <!--El nombre del mundo-->

    <world name="euclidian\_world">

        <!--Física del mundo

       Especifica el tipo y las propiedades del motor dinámico

        Tipo se refiere al motor dinámico (biblioteca de física); hay opciones como Ode, Bullet, Simbody y Dart-->

        <physics name="1ms" type="ignored">

            <!--Step

            Tamaño del paso para la simulación y factor para el tiempo real y el tiempo en la simulación-->

            <max\_step\_size>0.001</max\_step\_size>

            <real\_time\_factor>1.0</real\_time\_factor>

        </physics>

        <!--Abstract

        Interfaz de física abstracta diseñada para soportar la simulación y el desarrollo rápido de

        aplicaciones robóticas-->

        <plugin

            filename="gz-sim-physics-system"

            name="gz::sim::systems::Physics">

        </plugin>

        <!--Esto es responsable de crear modelos

        mover modelos, eliminarlos y muchos otros comandos del

        usuario.-->

        <plugin

            filename="gz-sim-user-commands-system"

            name="gz::sim::systems::UserCommands">

        </plugin>

        <!--Muestra

        la escena del mundo.-->

        <plugin

            filename="gz-sim-scene-broadcaster-system"

            name="gz::sim::systems::SceneBroadcaster">

        </plugin>

        <!-- Configura TODOS los complementos para la Interfaz Gráfica de Usuario -->

        <gui fullscreen="0">

            <!-- Camera -->

            <camera name="user\_camera">

                <projection\_type>orthographic</projection\_type>

            </camera>

            <!-- 3D scene -->

            <plugin filename="MinimalScene" name="3D View">

                <!-- Configura los atributos básicos en la vista 3D -->

                <gz-gui>

                    <title>3D View</title>

                    <!-- Si es verdadero, mostrará la barra de título azul sobre el complemento

                    con el nombre mencionado en la etiqueta <title> -->

                    <property type="bool" key="showTitleBar">false</property>

                    <!-- Es el estado del complemento; puede estar acoplado en su lugar usando "docked"

                    o puede estar flotante -->

                    <property type="string" key="state">docked</property>

                </gz-gui>

                <!-- Esta es la escena de renderizado -->

                <engine>ogre2</engine>

                <scene>scene</scene>

                <!-- Estos especifican el color ambiente y el color de fondo de la escena -->

                <ambient\_light>0.4 0.4 0.4</ambient\_light>

                <background\_color>0.8 0.8 0.8</background\_color>

            </plugin>

            <!-- Estos son responsables de mostrar la escena 3D de nuestro mundo -->

            <plugin filename="GzSceneManager" name="Scene Manager">

                <gz-gui>

                    <property key="resizable" type="bool">false</property>

                    <property key="width" type="double">7</property>

                    <property key="height" type="double">7</property>

                    <property key="state" type="string">floating</property>

                    <property key="showTitleBar" type="bool">false</property>

                </gz-gui>

            </plugin>

            <!-- Con esto podemos interactuar con el mundo, pero no con sus componentes -->

            <plugin filename="InteractiveViewControl" name="Interactive view control">

                <gz-gui>

                    <property key="resizable" type="bool">false</property>

                    <property key="width" type="double">5</property>

                    <property key="height" type="double">5</property>

                    <property key="state" type="string">floating</property>

                    <property key="showTitleBar" type="bool">false</property>

                </gz-gui>

            </plugin>

            <!-- Complemento de Ángulo de Vista -->

            <plugin filename="ViewAngle" name="View Angle">

            </plugin>

            <!-- Control del Mundo -->

            <plugin filename="WorldControl" name="World control">

                <gz-gui>

                    <title>World control</title>

                    <property type="bool" key="showTitleBar">false</property>

                    <property type="bool" key="resizable">false</property>

                    <property type="double" key="height">77</property>

                    <property type="double" key="width">177</property>

                    <property type="double" key="z">1</property>

                    <property type="string" key="state">floating</property>

                    <anchors target="3D View">

                        <line own="left" target="left" />

                        <line own="bottom" target="bottom" />

                    </anchors>

                </gz-gui>

                <!-- Si es verdadero, tendremos el botón de reproducir-pausar en la esquina inferior izquierda -->

                <play\_pause>true</play\_pause>

                <!-- Si es verdadero, mostrará un botón en la esquina inferior izquierda

                con la simulación paso a paso -->

                <step>true</step>

                <!-- Si es verdadero, la simulación se pausará al inicio de Gazebo -->

                <start\_paused>true</start\_paused>

                <service>/world/euclidian\_world/control</service>

                <!-- La etiqueta especifica el tema en el que se publican las estadísticas del mundo

                como el tiempo de simulación y el tiempo real -->

                <stats\_topic>/world/euclidian\_world/stats</stats\_topic>

            </plugin>

            <!-- Estadísticas del mundo. Responsable de mostrar las estadísticas del mundo -->

            <plugin filename="WorldStats" name="World stats">

                <gz-gui>

                    <title>World stats</title>

                    <property type="bool" key="showTitleBar">false</property>

                    <property type="bool" key="resizable">false</property>

                    <property type="double" key="height">110</property>

                    <property type="double" key="width">290</property>

                    <property type="double" key="z">1</property>

                    <property type="string" key="state">floating</property>

                    <anchors target="3D View">

                        <line own="right" target="right" />

                        <line own="bottom" target="bottom" />

                    </anchors>

                </gz-gui>

                <!-- Con estas etiquetas podemos elegir qué valores mostrar (expande la esquina inferior derecha para

                ver estos valores). Podemos elegir en qué <topic> se publicarán estos valores. Intentemos

                ejecutar el mundo y escuchar ese tema -->

                <sim\_time>true</sim\_time>

                <real\_time>true</real\_time>

                <real\_time\_factor>true</real\_time\_factor>

                <iterations>true</iterations>

                <!-- Podemos elegir en qué <topic> se publicarán los valores -->

                <topic>/world/euclidian\_world/stats</topic>

            </plugin>

            <!-- Árbol de entidades -->

            <plugin filename="EntityTree" name="Entity tree"></plugin>

        </gui>

        <!-- Configura la luz ambiental (como si fuera un sol). Especifica la fuente de luz en el mundo

        El <type> de la luz puede ser puntual, direccional o de spot -->

        <light

            type="directional" name="sun">

            <!-- Mostrar sombras -->

            <cast\_shadows>true</cast\_shadows>

            <!-- Configura la posición (primeros tres términos) y la orientación (rollo, inclinación, guiñada)

            de la luz con respecto al marco mencionado en el atributo relative\_to.

            Esto es ignorado por nosotros, por lo que la posición se configura con respecto al marco inercial -->

            <pose>0 0 10 0 0 0</pose>

            <!-- Color de la luz difusa -->

            <diffuse>0.8 0.8 0.8 1</diffuse>

            <!-- Color de la luz especular -->

            <specular>0.2 0.2 0.2 1</specular>

            <!-- Especifica las propiedades de atenuación de la luz -->

            <attenuation>

                <!-- Especifica el rango de la luz -->

                <range>1000</range>

                <!-- Especifica el factor de atenuación constante;

                1 nunca atenúa y 0 atenuación completa -->

                <constant>0.9</constant>

                <!-- Especifica el factor de atenuación lineal; 1 significa atenuar de manera

                uniforme a lo largo de la distancia. -->

                <linear>0.01</linear>

                <quadratic>0.001</quadratic>

            </attenuation>

            <!-- Especifica la dirección de la luz, aplicable

            solo a la luz puntual y direccional -->

            <direction>-0.5 0.1 -0.9</direction>

        </light>

        <!-- Para añadir un robot al mundo -->

        <model name="euclidian\_robot">

            <!-- Posición (x, y, z) y orientación (roll, pitch, yaw)

            con respecto al marco inercial-->

            <pose relative\_to="world">-0.5 -0.725 0 0 0 0</pose>

            <!-- Estructura principal -->

            <link name="main\_structure">

                <pose relative\_to="\_\_model\_\_">0 0 0 0 0 0</pose>

                <!-- Atributos físicos del enlace-->

                <inertial>

                    <!-- Masa del cuerpo (kilogramos)-->

                    <mass>65.057</mass>

                    <!-- Tensor de inercia (kg \* m²)-->

                    <inertia>

                        <ixx>15.105</ixx>

                        <ixy>0.0</ixy>

                        <ixz>0.0</ixz>

                        <iyy>30.128</iyy>

                        <iyz>0.0</iyz>

                        <izz>33.425</izz>

                    </inertia>

                </inertial>

                <!-- Configura los atributos de colisión -->

                <collision name="collision">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/frame.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                </collision>

                <!-- Configura los atributos visuales -->

                <visual name="visual">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/frame.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                    <!-- Color tutorial: https://classic.gazebosim.org/tutorials?tut=color\_model -->

                    <material>

                        <ambient>0.1 0.1 0.1 1</ambient>

                        <diffuse>0.1 0.1 0.3 1</diffuse>

                        <specular>0 0 0 0</specular>

                        <emissive>0 0 0 1</emissive>

                    </material>

                </visual>

            </link>

            <!-- Cars - beam joint -->

            <link name="cars\_beam">

                <pose relative\_to="main\_structure">0.015 0 0 0 0 0</pose>

                <!-- Atributos físicos del enlace-->

                <inertial>

                    <!-- Masa del cuerpo (kilogramos)-->

                    <mass>7.538</mass>

                    <!-- Tensor de inercia (kg \* m^2)-->

                    <inertia>

                        <ixx>0.0009</ixx>

                        <ixy>0.0</ixy>

                        <ixz>0.0</ixz>

                        <iyy>0.198</iyy>

                        <iyz>0.0</iyz>

                        <izz>0.198</izz>

                    </inertia>

                </inertial>

                <!-- Configura los atributos de colisión -->

                <collision name="collision">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/cars\_beam.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                </collision>

                <!-- Configura los atributos visuales -->

                <visual name="visual">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/cars\_beam.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                    <!-- Color tutorial: https://classic.gazebosim.org/tutorials?tut=color\_model -->

                    <material>

                        <ambient>0.1 0.1 0.1 1</ambient>

                        <diffuse>0.1 0.1 0.1 1</diffuse>

                        <specular>0 0 0 0</specular>

                        <emissive>0 0 0 1</emissive>

                    </material>

                </visual>

            </link>

            <!-- Piston -->

            <link name="piston">

                <pose relative\_to="cars\_beam">0.083 0 0 0 0 0</pose>

                <!-- Atributos físicos del enlace-->

                <inertial>

                    <!-- Masa del cuerpo (kilogramos)-->

                    <mass>4.211</mass>

                    <!-- Inertia tensor (kg \* m^2)-->

                    <inertia>

                        <ixx>0.0009</ixx>

                        <ixy>0.0</ixy>

                        <ixz>0.0</ixz>

                        <iyy>0.198</iyy>

                        <iyz>0.0</iyz>

                        <izz>0.198</izz>

                    </inertia>

                </inertial>

                <!-- Configura los atributos de colisión -->

                <collision name="collision">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/piston.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                </collision>

                <!-- Configura los atributos visuales -->

                <visual name="visual">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/piston.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                    <!-- Color tutorial: https://classic.gazebosim.org/tutorials?tut=color\_model -->

                    <material>

                        <ambient>0.1 0.1 0.1 1</ambient>

                        <diffuse>0.1 0.1 0.1 1</diffuse>

                        <specular>0 0 0 0</specular>

                        <emissive>0 0 0 1</emissive>

                    </material>

                </visual>

            </link>

            <!-- Piston -->

            <link name="end\_effector">

                <pose relative\_to="piston">0 0 0 0 0 0</pose>

                <!-- Atributos físicos del enlace-->

                <inertial>

                    <!-- Masa del cuerpo (kilogramos)-->

                    <mass>1.202</mass>

                    <!-- Inertia tensor (kg \* m^2)-->

                    <inertia>

                        <ixx>0.0009</ixx>

                        <ixy>0.0</ixy>

                        <ixz>0.0</ixz>

                        <iyy>0.198</iyy>

                        <iyz>0.0</iyz>

                        <izz>0.198</izz>

                    </inertia>

                </inertial>

                <!-- Configura los atributos de colisión -->

                <collision name="collision">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Importar malla desde un archivo STL -->

                        <mesh>

                            <uri>model://ROS/STL/end\_effector.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                </collision>

                <!-- Configura los atributos visuales -->

                <visual name="visual">

                    <!-- Configura el tipo de geometría -->

                    <geometry>

                        <!-- Import mesh from STL file -->

                        <mesh>

                            <uri>model://ROS/STL/end\_effector.stl</uri>

                            <!--<scale>0.001 0.001 0.001</scale>-->

                        </mesh>

                    </geometry>

                    <!-- Color tutorial: https://classic.gazebosim.org/tutorials?tut=color\_model -->

                    <material>

                        <ambient>0.1 0.1 0.1 1</ambient>

                        <diffuse>0.1 0.1 0.1 1</diffuse>

                        <specular>0 0 0 0</specular>

                        <emissive>0 0 0 1</emissive>

                    </material>

                </visual>

            </link>

            <!-- Fijación de la estructura principal -->

            <joint name="main\_fixed" type="fixed">

                <!-- Define la pose con respecto al movimiento que se realizará -->

                <pose relative\_to="\_\_model\_\_" />

                <!-- Establece quién lidera el movimiento y quién lo sigue -->

                <parent>world</parent>

                <child>main\_structure</child>

            </joint>

            <!-- Joint 1 (Car 1) -->

            <joint name="joint\_1" type="prismatic">

                <!-- Define la pose con respecto al movimiento que se realizará -->

                <pose relative\_to="main\_structure" />

                <!-- Establece quién lidera el movimiento y quién lo sigue. -->

                <parent>main\_structure</parent>

                <child>cars\_beam</child>

                <!-- Configura los atributos para el eje de actuación de la articulación -->

                <axis>

                    <!-- Configura el eje de actuación de la articulación -->

                    <xyz>0 1 0</xyz>

                    <!-- Amortiguación de la articulación

                    <dynamics>

                        <damping>0.05</damping>

                    </dynamics> -->

                </axis>

            </joint>

            <!-- Joint 2 (Car 2) -->

            <joint name="joint\_2" type="prismatic">

                <!-- Define la pose con respecto al movimiento que se realizará -->

                <pose relative\_to="cars\_beam" />

                <!-- Establece quién lidera el movimiento y quién lo sigue -->

                <parent>cars\_beam</parent>

                <child>piston</child>

                <!-- Configura los atributos para el eje de actuación de la articulación -->

                <axis>

                    <!-- Configura el eje de actuación de la articulación -->

                    <xyz>1 0 0</xyz>

                    <!-- Amortiguación de la articulación

                    <dynamics>

                        <damping>0.05</damping>

                    </dynamics> -->

                </axis>

            </joint>

            <!-- Joint 3 (Piston) -->

            <joint name="joint\_3" type="prismatic">

                <!-- Define la pose con respecto al movimiento que se realizará -->

                <pose relative\_to="cars\_beam" />

                <!-- Establece quién lidera el movimiento y quién lo sigue -->

                <parent>piston</parent>

                <child>end\_effector</child>

                <!-- Configura los atributos para el eje de actuación de la articulación -->

                <axis>

                    <!-- Configura el eje de actuación de la articulación -->

                    <xyz>0 0 -1</xyz>

                    <!-- Joint damping

                    <dynamics>

                        <damping>0.05</damping>

                    </dynamics> -->

                </axis>

            </joint>

            <!-- Controlador de trayectoria de las articulaciones -->

            <plugin

                filename="gz-sim-joint-trajectory-controller-system"

                name="gz::sim::systems::JointTrajectoryController">

                <!-- Controlador de trayectoria de la articulación 1 -->

                <joint\_name>joint\_1</joint\_name>

                <initial\_position>0</initial\_position>

                <position\_p\_gain>14.0</position\_p\_gain>

                <position\_d\_gain>14.0</position\_d\_gain>

                <!-- Controlador de trayectoria de la articulación 2 -->

                <joint\_name>joint\_2</joint\_name>

                <initial\_position>0</initial\_position>

                <position\_p\_gain>14.0</position\_p\_gain>

                <position\_d\_gain>14.0</position\_d\_gain>

                <!-- Controlador de trayectoria de la articulación 3 -->

                <joint\_name>joint\_3</joint\_name>

                <initial\_position>0</initial\_position>

                <position\_p\_gain>14.0</position\_p\_gain>

                <position\_d\_gain>14.0</position\_d\_gain>

            </plugin>

            <!-- Para las trayectorias

                gz topic -t "/model/euclidian\_robot/joint\_trajectory" -m gz.msgs.JointTrajectory -p 'joint\_names: "joint\_1"; joint\_names: "joint\_2"; joint\_names: "joint\_3"; points { positions: 0; positions: 0; positions: 0; time\_from\_start {  sec: 7; nsec: 0 } }; points { positions: 0; positions: -3; positions: 4; time\_from\_start {  sec: 14; nsec: 770000000 } }; points { positions: -3; positions: 4; positions: -5; time\_from\_start {  sec: 21; nsec: 770000000 } }'

            -->

        </model>

    </world>

</sdf>