**UNIVERSIDAD DE VALLADOLID ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Electrónica Industrial y Automática Taller de Robótica Industrial**

**Entrega 1: Simulador con MATLAB del robot construido.**

**Curso 2021 – 2022**

Imagen que contiene lego, juguete

Descripción generada automáticamente

**Hernández Jiménez, Daniel**

**Salán Conde, Christian**

**Redondo Quintero, David**

**Valladolid, abril 2022**

**INDICE DE CONTENIDOS**

[1 INTRODUCCIÓN 2](#_Toc101713918)

[2 ESTRUCTURA GEOMÉTRICA DEL ROBOT 3](#_Toc101713919)

[2.1 Dimensiones 3](#_Toc101713920)

[2.1 Articulaciones y eslabones 5](#_Toc101713921)

[2.3 Base del robot 8](#_Toc101713922)

[2.2 Actuador 8](#_Toc101713923)

[2.3 Volumen de trabajo del robot 10](#_Toc101713924)

[3 PARAMETRIZACIÓN DENAVIT-HARTENBERG 12](#_Toc101713925)

[3.1 Modelo cinemático directo 12](#_Toc101713926)

[3.2 Modelo cinemático inverso 13](#_Toc101713927)

[4 ESTUDIO DE LAS SINGULARIDADES DEL ROBOT 14](#_Toc101713928)

[5 SOFTWARE DESARROLLADO 16](#_Toc101713929)

[5.1 DIAGRAMA DE BLOQUES 16](#_Toc101713930)

[5.2 INTERFAZ 17](#_Toc101713931)

[6 BIBLIOGRAFÍA 18](#_Toc101713932)

# 1 INTRODUCCIÓN

En este trabajo se procede a diseñar un robot con elementos de Bioloid, un kit para construir robots que incluye servomotores, elementos estructurales y sensores entre otros.

El robot que se va a construir se trata de un brazo manipulador que cuenta con 5 grados de libertad del tipo PRRRR (Prismatic joint & Revolute joints).

Como herramienta del robot se incorporará una garra con sensor infrarrojo capaz de manipular objetos y distinguir colores.

Los servomotores del kit Bioloid Dynamixel AX-12 serán los encargados de realizar la función de articulaciones.

Se va a desarrollar una interfaz mediante la funcionalidad AppDesigner de MATLAB. En esta vendrán incluidos elementos como movimiento manual de las articulaciones individualmente (cinemática directa), visualizar robot en un punto del espacio (cinemática inversa), matriz de transformación homogénea de la base al efector final y representación 3D del robot en posiciones comandadas.

# 2 ESTRUCTURA GEOMÉTRICA DEL ROBOT

Se ha diseñado un robot manipulador de pequeño tamaño debido a las limitaciones de carga de los servomotores de los que se dispone. Según las especificaciones de los servomotores Dynamixel AX-12, su par máximo es de 0.2 N·m. Estos motores cuentan con dos modos de movimiento, uno con limitación de 0º a 300º y otro con movimiento ilimitado.

## 2.1 Dimensiones

Imagen que contiene lego, juguete

Descripción generada automáticamente

Ilustración 1 Modelo 3D del robot manipulador

Comenzaremos a dimensionar el robot partiendo desde la base en la posición correspondiente a q1 = 0.

S0 a S1:

Eslabón 1: 76.2 mm

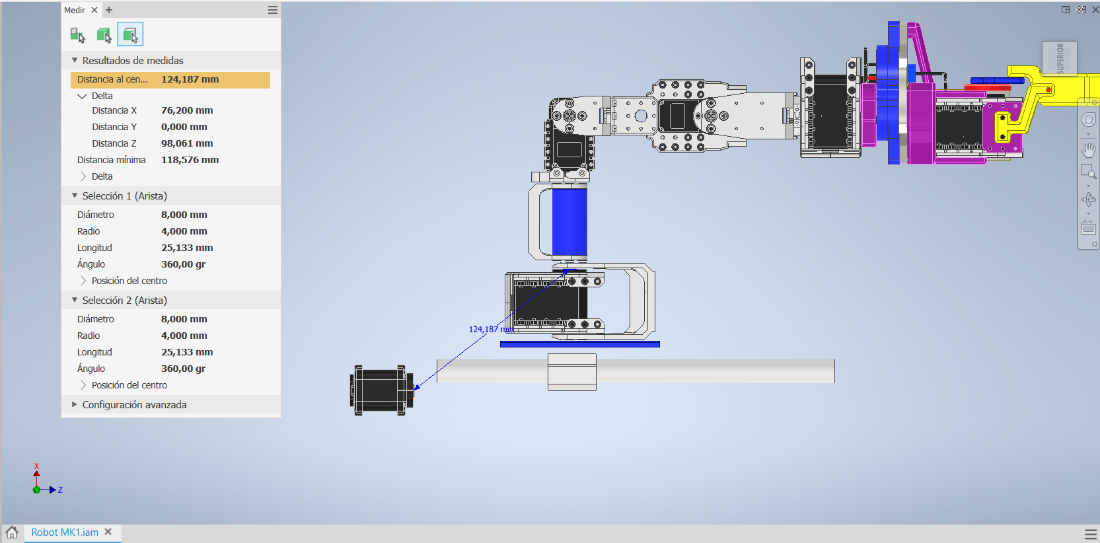


Ilustración 2 Longitud del eslabón 1

S1 a S2:

Eslabón 2: 96.300mm

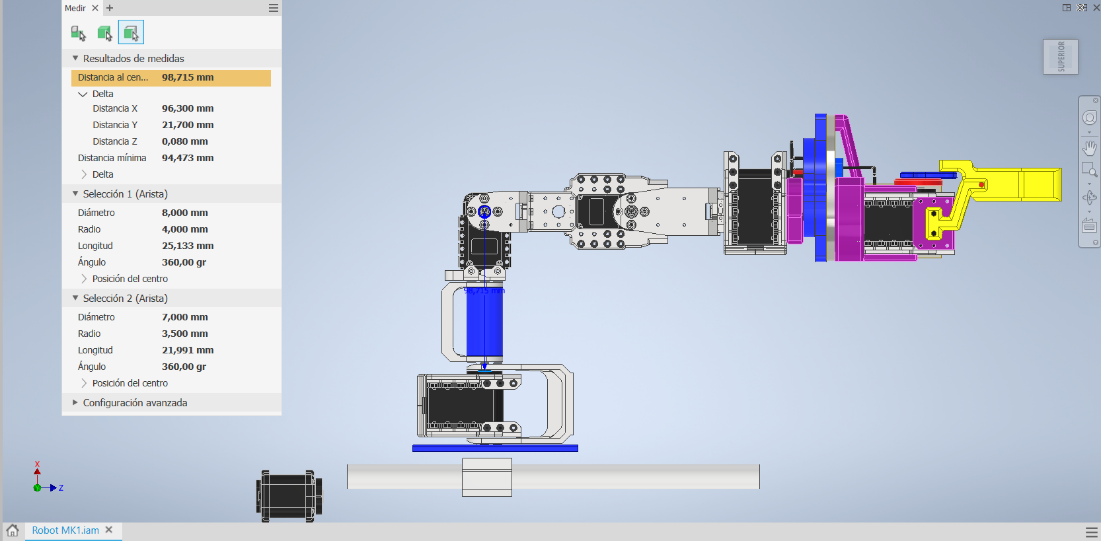


Ilustración 3 Longitud del eslabón 2

S2 a S3:

Distancia entre ejes Z: 5.327mm (tercer eje hacia fuera)

Distancia horizontal: 88.981mm

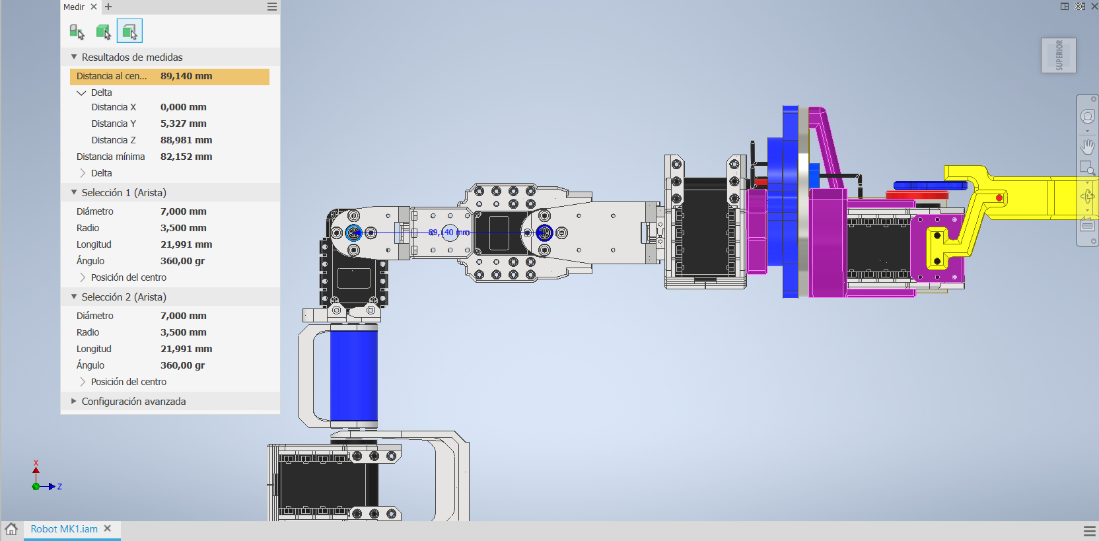


Ilustración 4 Longitud del eslabón 3

S3 a S4:

Altura: 14.5mm

Distancia horizontal: 96.092mm

Distancia entre ejes Z: 6.194mm (cuarto eje hacia dentro)

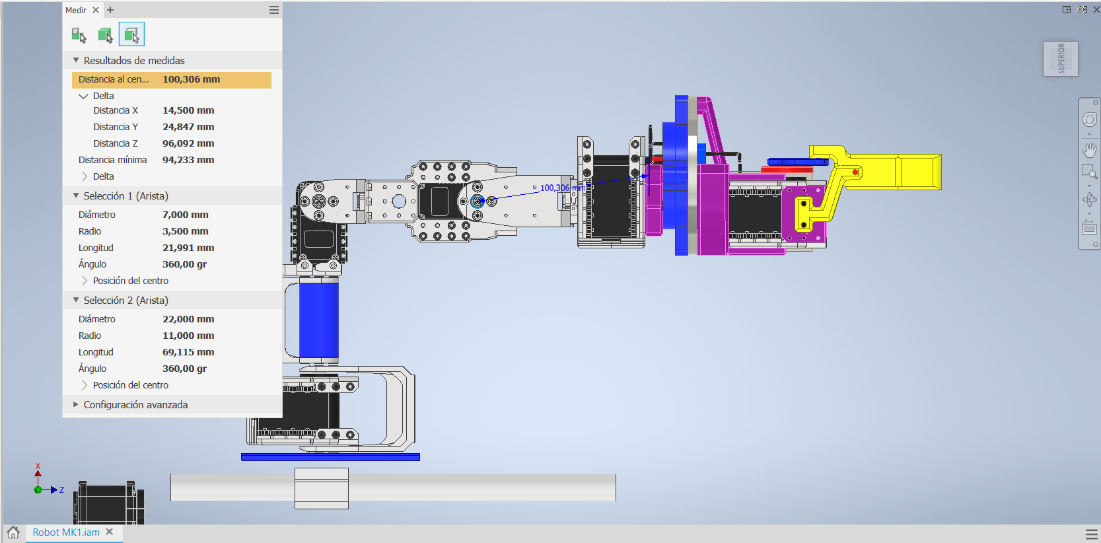


Ilustración 5 Longitud del eslabón 4

S4 a S5:

Altura: 1.853mm

Distancia horizontal: 164.394mm

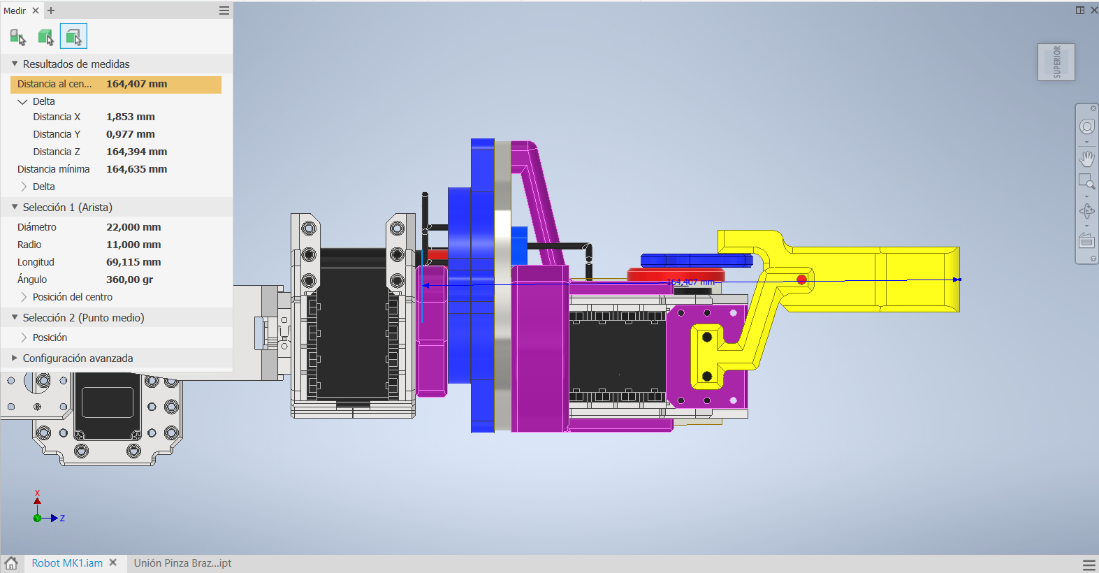


Ilustración 6 Longitud del eslabón 5

## 2.1 Articulaciones y eslabones

El robot será PRRRR y cuenta con 6 eslabones. La primera articulación (q1) se trata de una articulación prismática. El movimiento de q1 se conseguirá mediante el modo de rotación ilimitada que proporciona el AX-12 en conjunto con un tornillo sin fin.

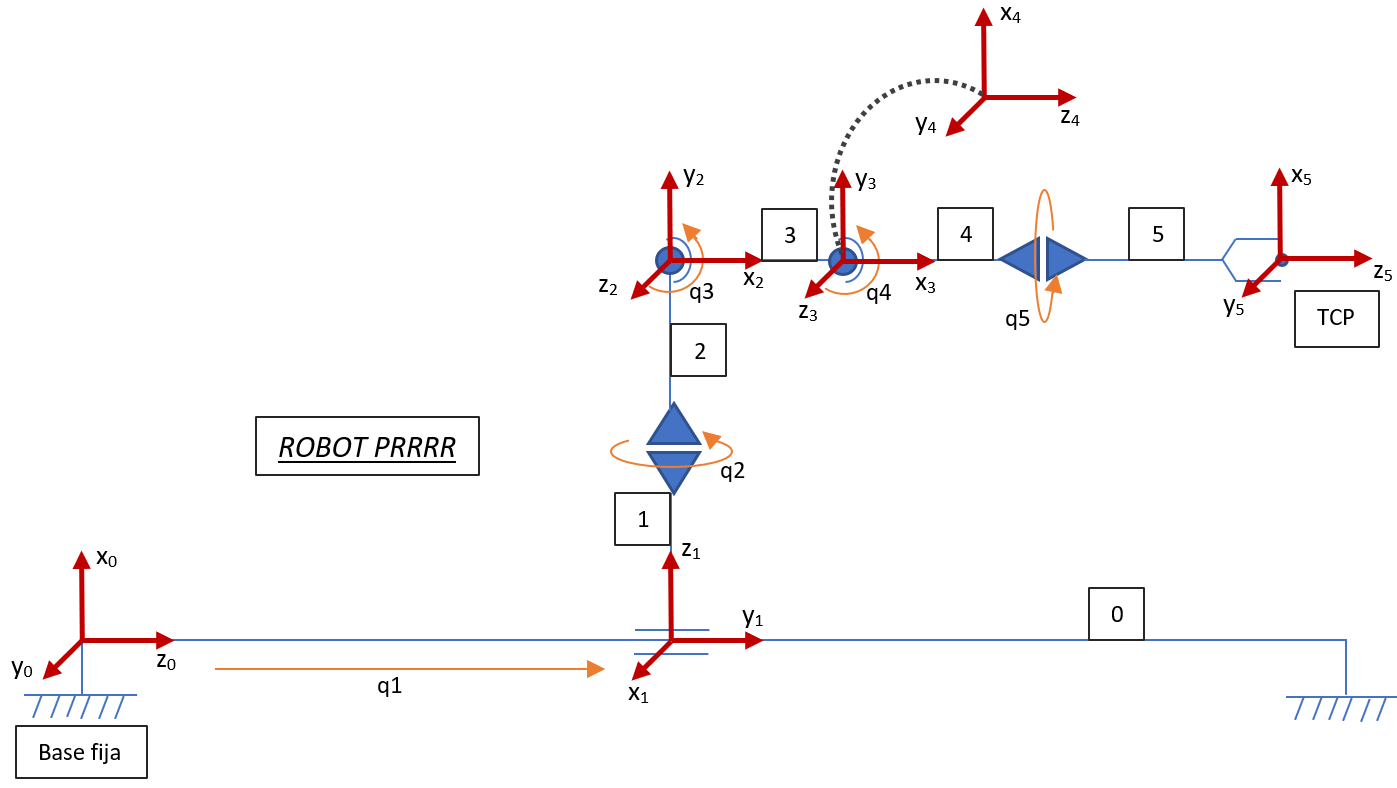


Ilustración 7 Esquema del robot manipulador

Por tanto, la base del robot será la estructura donde se sitúe el servomotor que da movimiento a la articulación q1.

Las demás articulaciones serán de revolución y en este caso se hará uso del modo de movimiento con limitación 0º a 300º del servomotor, excepto en la última articulación (muñeca). En esta, se ha diseñado un sistema de rotación infinita para facilitar un agarre más cómodo de los objetos y evitar dañar y enredar los cables de señal PWM y alimentación de los servomotores.

El mecanismo para lograr una rotación ilimitada es un sistema de 3 escobillas, de las cuales una es utilizada para la señal PWM del servomotor de la pinza, otra para la alimentación positiva y la tercera para la alimentación negativa.

Desde uno de los lados de la muñeca tenemos la pieza donde podemos encontrar las pistas sobre las cuales deslizan las escobillas, como se puede observar en la siguiente imagen quedan diferenciadas en color dorado, plata y cobre:

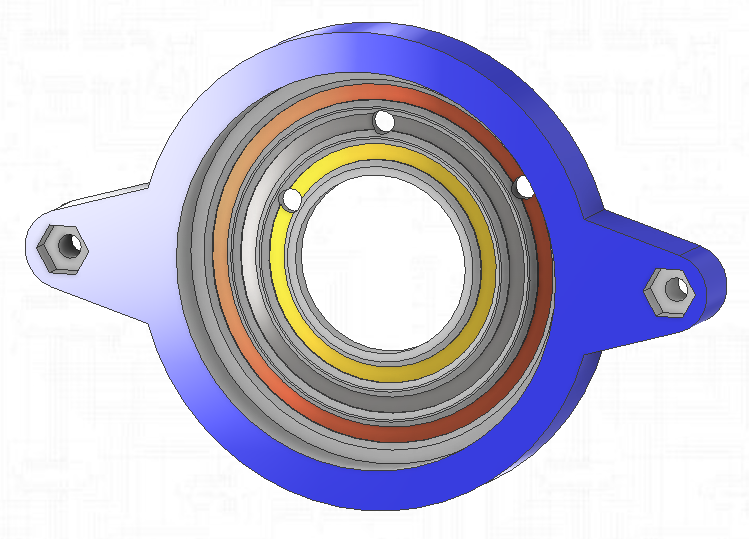


Ilustración 8 Mecanismo rotación ilimitada parte 1

Además, en cada una de las pistas tenemos un agujero para poder realizar la conexión soldando un pequeño tramo de cable y al final un conector del tipo “dupont”.

Desde el otro lado de la muñeca tenemos los 3 agujeros correspondientes a cada una de las escobillas, con su orificio de salida hacia la pinza para poder realizar por clavija “dupont”, la conexión del segundo tramo de cable con el servomotor encargado de accionar la pinza:

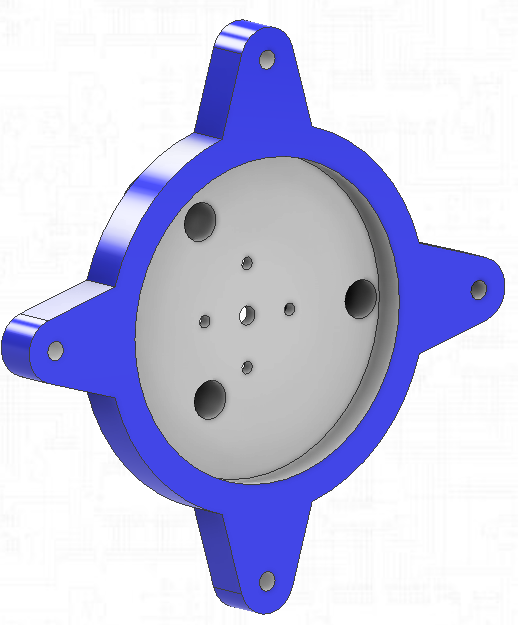


Ilustración 9 Mecanismo rotación ilimitada parte 2

La tercera articulación (q3) será la articulación que más par de inercia deberá vencer debido a que soporta el peso de los eslabones 3,4 y 5, los servomotores de q4, q5 y el efector final. Por estos motivos, se ha incluido un segundo servomotor que trabaja en conjunto con el de la articulación q3 para aumentar el par que puede vencer esa articulación y así dotar de una mayor robustez al robot.

## 2.3 Base del robot

La base de este robot es móvil, en concreto es del tipo prismática. Se moverá mediante un tornillo sin fin conectado a un servomotor AX-12, y se ayudará de dos guías de perfil circular junto con 2 rodamientos lineales para su desplazamiento.

## 2.2 Actuador

El actuador es una pinza accionada por un servomotor modelo AX-12. Se utiliza el propio movimiento rotativo del brazo circular del servo para desplazar dos brazos (indicados en color azul en la imagen) que se encargarán de mover cada una de las garras de la pinza.

El rango de apertura va desde 0 mm hasta 26 mm.

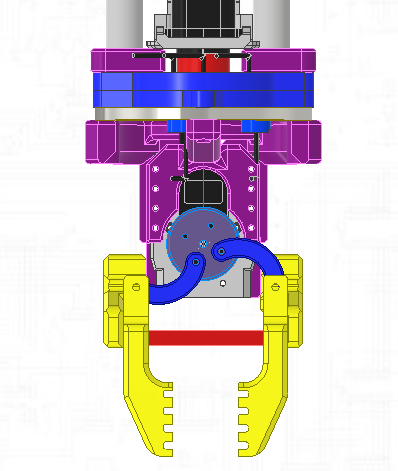


Ilustración 10 Actuador en posición abierta

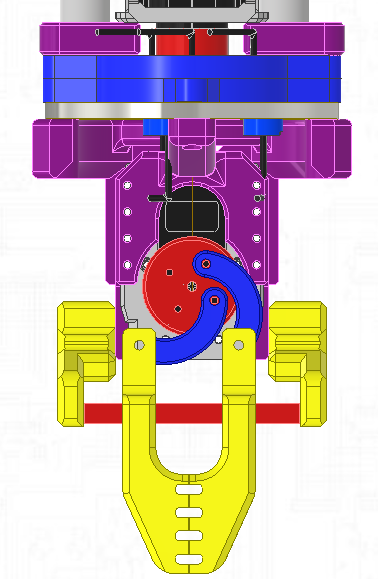


Ilustración 11 Actuador en posición cerrada

## 2.3 Volumen de trabajo del robot

Para poder calcular el volumen de trabajo del robot en *Matlab* se ha utilizado un método llamado *Método Monte Carlo* (Abdolmalaki, 2017).

Sabiendo la longitud de los eslabones, los límites de las articulaciones y habiendo conseguido la matriz de transformación homogénea podemos hallar el espacio de trabajo generando n puntos aleatorios que respeten la geometría y los límites del robot.

Representando esos n puntos aleatorios en un gráfico *3D* gracias a la función *plot3D* de *Matlab*, obtenemos el espacio de trabajo del robot.

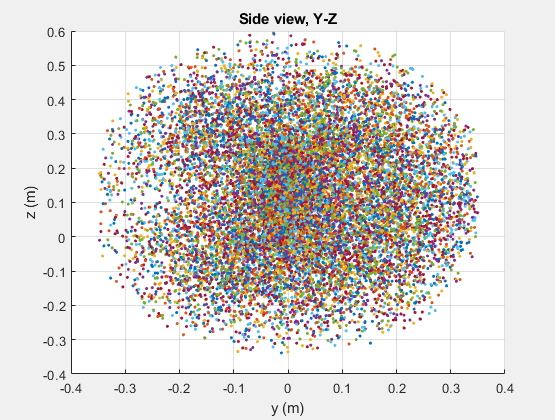


Ilustración 12 Espacio de trabajo representación plano Y-Z

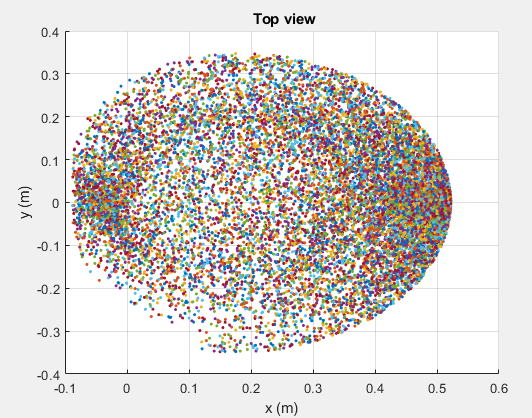


Ilustración 13 Espacio de trabajo representación plano X-Y

En las ilustraciones anteriores se pueden observar los límites de alcance del robot con 15000 puntos aleatorios calculados por *Matlab*.

En la Ilustración 13 se pude apreciar claramente como la concentración de puntos es mayor en la parte superior (donde el robot tiene menos restricciones de movimiento) y mucho menos en la parte central, donde se produce el desplazamiento del robot de forma lineal.

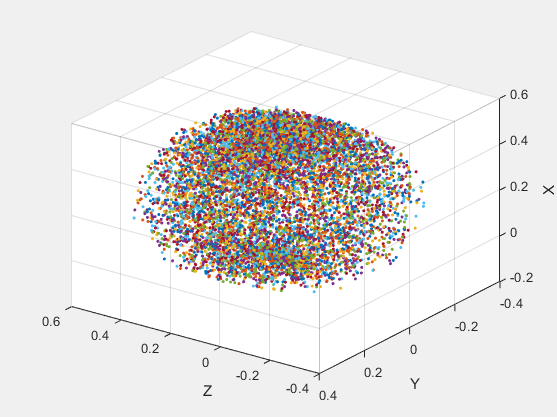


Ilustración 13 Espacio de trabajo representación plano X-Y-Z

# 3 PARAMETRIZACIÓN DENAVIT-HARTENBERG

Se ha obtenido la tabla de parámetros de Denavit-Hartenberg con la siguiente posición y orientación del robot:

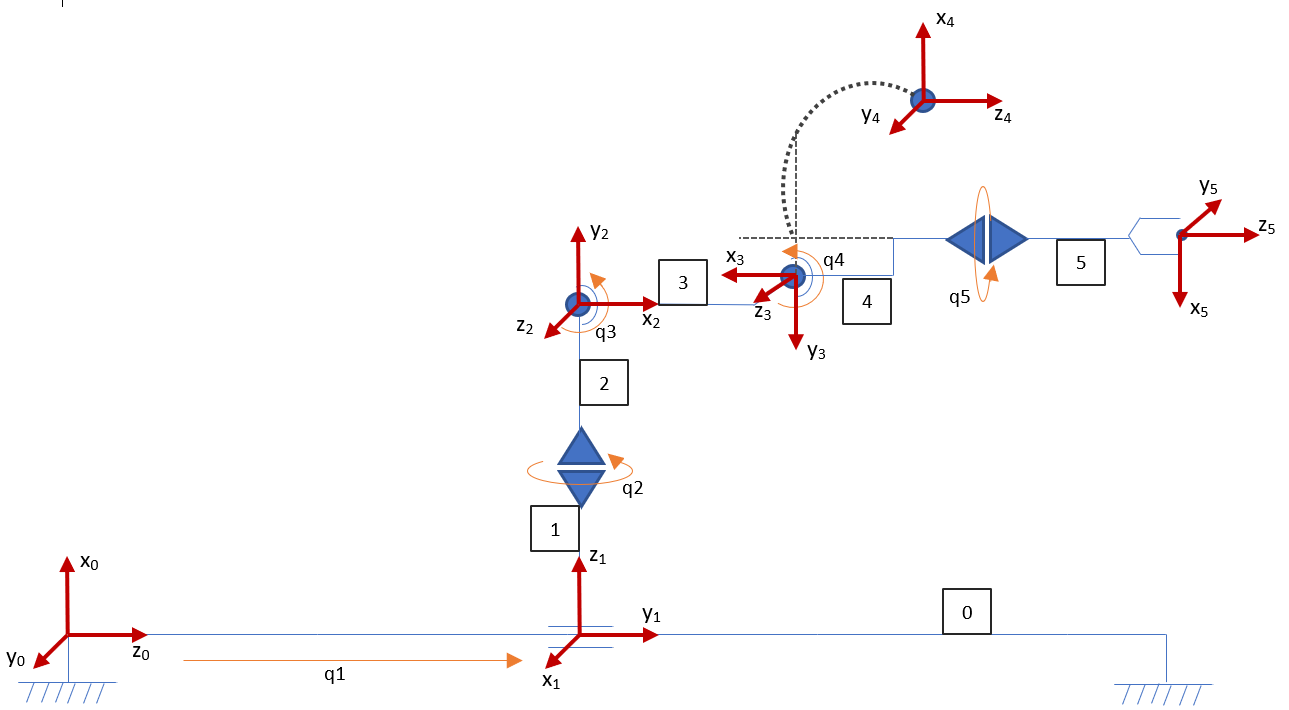


Ilustración 14 Representación del robot para obtener los parámetros D-H

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Articulación** | **theta** | **d** | **a** | **alpha** |
| 1 | π/2 | x + q1 | 0 | π/2 |
| 2 | π/2 + q2 | 0,1725 | 0 | π/2 |
| 3 | π + q3 | 0,0053 | 0,0889 | 0 |
| 4 | q4 - π/2 | -0,0062 | 0,0145 | π/2 |
| 5 | π + q5 | 0,2605 | -0,0019 | 0 |

Tabla 1 Parámetros D-H

## 3.1 Modelo cinemático directo

El modelo cinemático directo pasa del estado articular al espacio cartesiano.

Gracias a los parámetros D-H podemos obtener las ecuaciones cartesianas del TCP del robot con los valores de las articulaciones como variables de esas ecuaciones.

Lo que se hace es calcular la matriz de transformación homogénea (MTH) y las tres primeras filas de la cuarta columna corresponden a los valores de ‘x’, ‘y’ y ‘z’ respectivamente.

Para nuestra aplicación se han utilizado las funciones de la Robotics Toolbox de Matlab, en específico la función *rigidBodyTree()* y sus derivadas que facilitan bastante la construcción y el diseño del robot en simulación.

## 3.2 Modelo cinemático inverso

El modelo cinemático inverso pasa del estado cartesiano al espacio articular.

Una vez se han obtenido las ecuaciones de la cinemática directa, se pueden obtener las de cinemática inversa operándolas. La cinemática inversa tiene como incógnita los valores de las articulaciones del robot para unas coordenadas cartesianas que actúan como variables.

Es importante recalcar que, a diferencia del modelo cinemático directo, un punto en el espacio puede tener varias soluciones debido a que el robot puede llegar a ese punto con distintas orientaciones (sistema de 5 ecuaciones con 3 incógnitas en nuestro caso).

Debido a la complejidad de los cálculos, el modelo cinemático inverso se calculará con la función de *Matlab inverseKinematics()* pudiendo dedicar más potencia de cómputo para la posición del punto o la orientación. Normalmente se le suele dar la máxima importancia a llegar al punto de forma precisa y no tanta a la orientación.

# 4 ESTUDIO DE LAS SINGULARIDADES DEL ROBOT

Las singularidades pueden representar puntos donde el robot pierde movilidad (puntos no alcanzables), también puntos o configuraciones donde pequeñas velocidades lineales y angulares de TCP se obtengan con elevadísimas velocidades de articulación, puntos donde pequeñas fuerzas y pares aplicados al TCP se obtengan gracias a elevadísimos pares en las articulaciones (limitación física de los motores), puntos donde el robot va a trabajar en los límites del espacio de trabajo y puntos o configuraciones donde en ellos mismos o cercanos a ellos no exista solución para el problema cinemático inverso.

Para conocer las singularidades del robot será necesario calcular y operar el jacobiano o matriz jacobiana.

El jacobiano es una matriz que nos permite pasar de velocidades de articulación a velocidades del TCP.

Debido a que el brazo del robot tiene pequeños offsets en ciertas articulaciones que dificultan los parámetros D-H se hace una aproximación a la siguiente tabla D-H para poder obtener el jacobiano sin que surjan ecuaciones muy extensas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Articulación** | **theta** | **d** | **a** | **alpha** |
| 1 | π/2 | x + q1 | 0 | π/2 |
| 2 | π/2 + q2 | 0,1725 | 0 | π/2 |
| 3 | π + q3 | 0 | 0,0889 | 0 |
| 4 | q4 - π/2 | 0 | 0 | π/2 |
| 5 | π + q5 | 0,2605 | 0 | 0 |

Tabla 2 Tabla D-H aproximada

En este caso, el jacobiano se ha obtenido utilizando *Matlab* y quedaría tal que así:

Para poder obtener los puntos singulares del robot se debe igualar a cero el determinante de la matriz jacobiana. En este caso, la matriz jacobiana no es cuadrada, por lo que no podemos calcular su determinante. Lo que se debe hacer es buscar las combinaciones de filas y columnas posibles para calcular determinantes de 3x3 (rango máximo de la matriz) e igualar ese polinomio a cero.

Las combinaciones son de las columnas 123, 124, 125, 134, 135, 145, 234, 235, 245, 345.

Operando en Matlab obtenemos que los únicos determinantes no nulos son los de las combinaciones 124 y 134. Estos determinantes dan lugar a dos ecuaciones con 3 incógnitas. En ambas, las articulaciones (incógnitas) que aparecen son q2, q3 y q4. Por lo tanto, no se puede resolver la ecuación a no ser que demos valores al menos a una de las articulaciones.

# 5 SOFTWARE DESARROLLADO

En este apartado se explicará cómo se ha desarrollado el software del robot en *Matlab*. Podemos distinguir elementos en el software, la interfaz y los scripts. Mientras que la interfaz va a ser aquello con lo que el usuario interacciona directamente, los scripts son los programas que soportarán toda la carga computacional.

## 5.1 DIAGRAMA DE BLOQUES

Las funciones que se han desarrollado para este trabajo son las siguientes:

* calculaMTH(): Función con parámetros D-H predefinidos que obtiene y devuelve la matriz Denavit-Hartenberg del robot.

Llamada a la función

MTH simbólica

*calculaMTH()*

* getWorkspace(nPuntos, vista): Función que devuelve una gráfica con el espacio de trabajo del robot con diferentes opciones.
  + vista = 1: Gráfica 2D representada en el plano X-Y.
  + vista = 2: Gráfica 2D representada en el plano Y-Z.
  + vista ≠ 1, 2: Gráfico 3D del espacio de trabajo.

Gráfico espacio de trabajo

n\_puntos

vista

*getWorkspace(nPuntos, vista)*

* creaRobot(): Instancia un robot mediante la unión de varios cuerpos y se contemplan sus tipos de articulaciones, límites de estas, posiciones, etc.

Llamada a la función

Robot Instanciado

*creaRobot()*

* getMTH\_TCP(q1,q2,q3,q4,q5): Al mandar una configuración de articulaciones determinada, esta función devuelve la MTH del TCP en con esos valores.

Matriz de paso T0\_TCP

*getMTH\_TCP (q1, q2, q3, q4, q5)*

(q1, q2, q3, q4, q5)

* muestraCinematicaDirecta(q1,q2,q3,q4,q5): Resuelve el problema cinemático directo instanciando un robot con la función creaRobot() y mandándole la configuración de articulaciones dada y muestra el robot en esa posición.
* muestraCinemáticaInversa(x,y,z,angulo): Resuelve el problema cinemático inverso instanciando un robot con la función creaRobot() y mandándole la configuración cartesiana dada y muestra el robot en esa posición.

## 5.2 INTERFAZ

La interfaz ha sido realizada en el entorno AppDesigner de Matlab. Este entorno sirve de nexo entre el usuario y los scripts, haciendo más sencilla la manipulación y ejecución de los scripts.

Box and whisker chart

Description automatically generated with medium confidence

Ilustración 15 Representación de la interfaz en AppDesigner

Dentro de la interfaz tenemos métodos como Calcula\_Cinematica\_Directa que calcula la MTH a través de los scripts antes mencionados y la muestra por pantalla a través de los valores Q que hayamos ajustado con los sliders.

También se ha implementado una visualización 3D del robot. Las opciones de Cinemática Inversa y cálculo de trayectorias todavía no están programadas.

# 6 BIBLIOGRAFÍA

Abdolmalaki, R. Y. (2017). Development of direct kinematics and workspace representation for smokie robot manipulator & the barret wam. *arXiv*.

Documentación de Matlab.