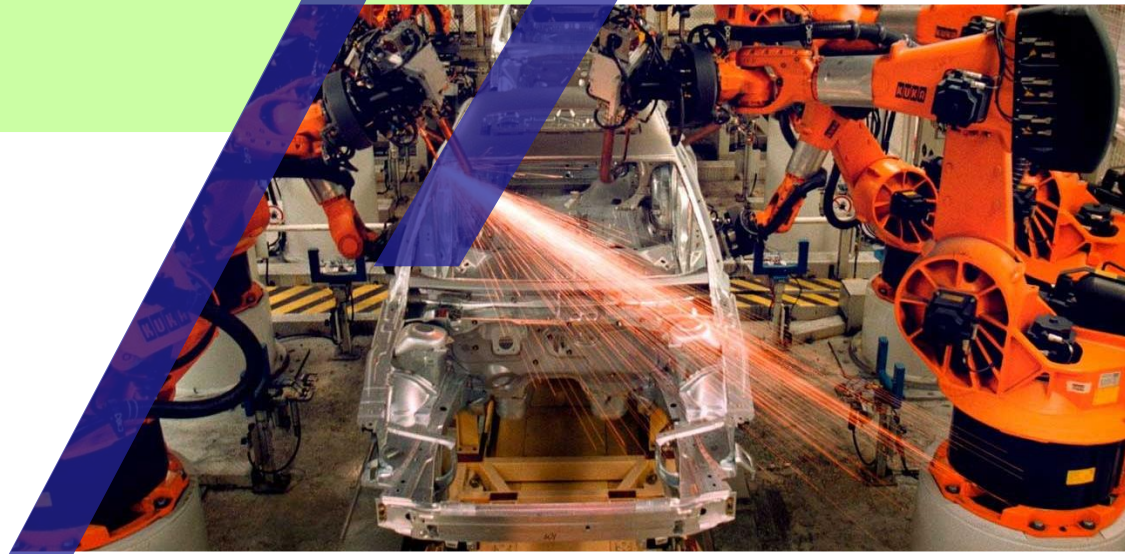




UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA MECÁNICA

INFORME N°3

“Análisis cinemático de un manipulador robótico cilíndrico de 3 grados de libertad”



Curso: Dinámica de Sistema Multicuerpo ([MT516](#))

Elaborado por:

Belleza Martínez Angel Steven
Castro Suazo, Fidel Angel
Llontop Herrera, Marco Leandro

Código:

20200070G
20200062D
20202011H

Revisado por:

Ing. Calle Flores Iván

Fecha de entrega:

12 de diciembre de 2022

1. RESUMEN	2
2. SISTEMAS COORDENADOS SEGÚN DENAVIT HARTENBERG	3
3.1. Convención modificada Waldron - Paul:	3
3.2. Convención modificada Khalil – Dombre:	5
3. HOME POSITION DEL ROBOT	8
4. ECUACIONES DE CINEMÁTICA DIRECTA.....	9
5. ESPACIO DE TRABAJO DEL ROBOT (WORKSPACE).....	12
6. ECUACIONES DE CINEMÁTICA INVERSA.....	17
6.1. Método Algebraico	17
6.2. Método Geométrico	18
7. PROGRAMA ARDUINO + MATLAB	20
7.1. Cinemática Directa.....	20
7.2. Cinemática Inversa.....	28
8. DIAGRAMA DE GANTT	35
9. BIBLIOGRAFÍA	36
10. ANEXOS	37
10.1. Mediciones de los parámetros constantes	37
10.2. Rango de movimiento de los mecanismos (Restricciones Reales)	38
10.3. Cálculo de Factores de Transmisión de las juntas.....	39

En esta tercera parte del proyecto de MT516, se realiza el análisis cinemático de un manipulador robótico cilíndrico de 3 grados de libertad. Esto comprende la asignación de sistemas coordenados según Denavit-Hartenberg con su tabla de parámetros. Asimismo, se muestran las ecuaciones de cinemática directa y cinemática inversa.

Por otro lado, se implementó un programa “Arduino + Matlab” para mostrar la cinemática directa e inversa para comprobar el correcto funcionamiento del robot en tiempo real.

Finalmente, se adjunta el diagrama de Gantt actualizado con la distribución de tareas y fechas designadas durante el desarrollo de este tercer trabajo. Concluida esta parte, el robot queda listo para su aplicación en una tarea de interés.

2.1. Convención modificada Waldron - Paul:

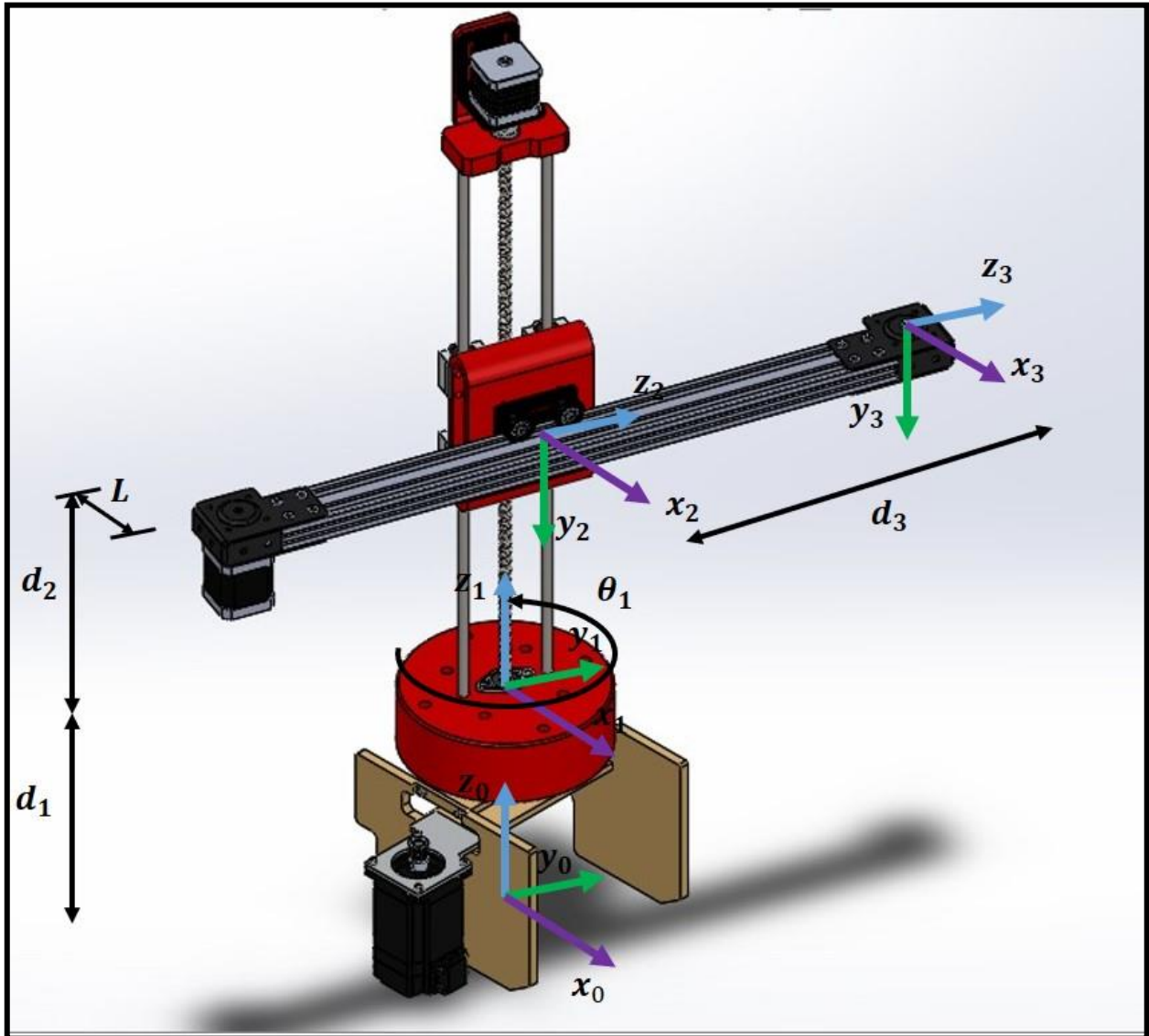
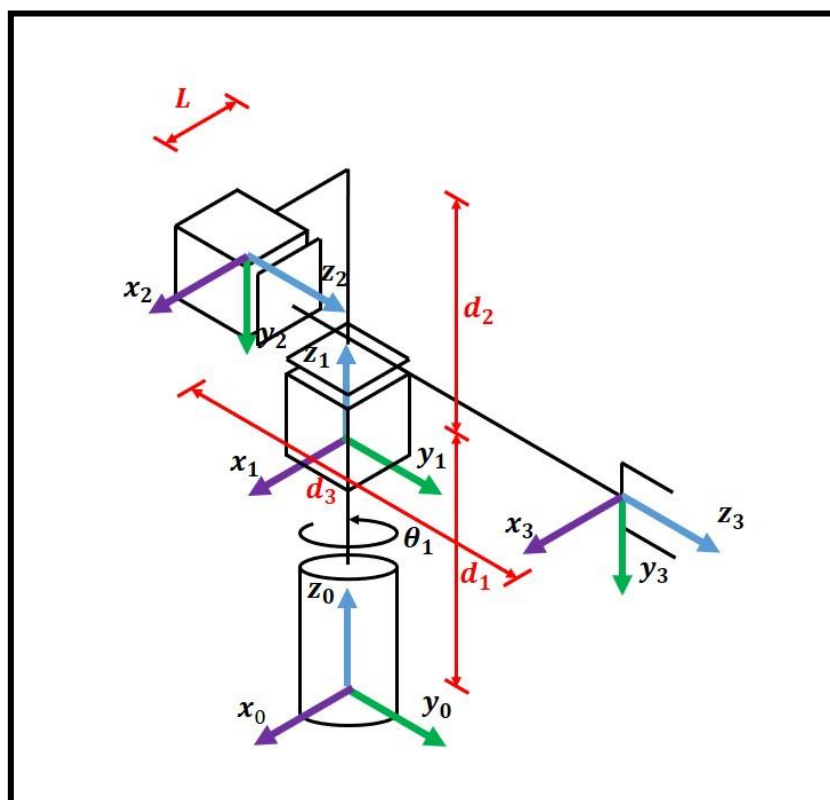


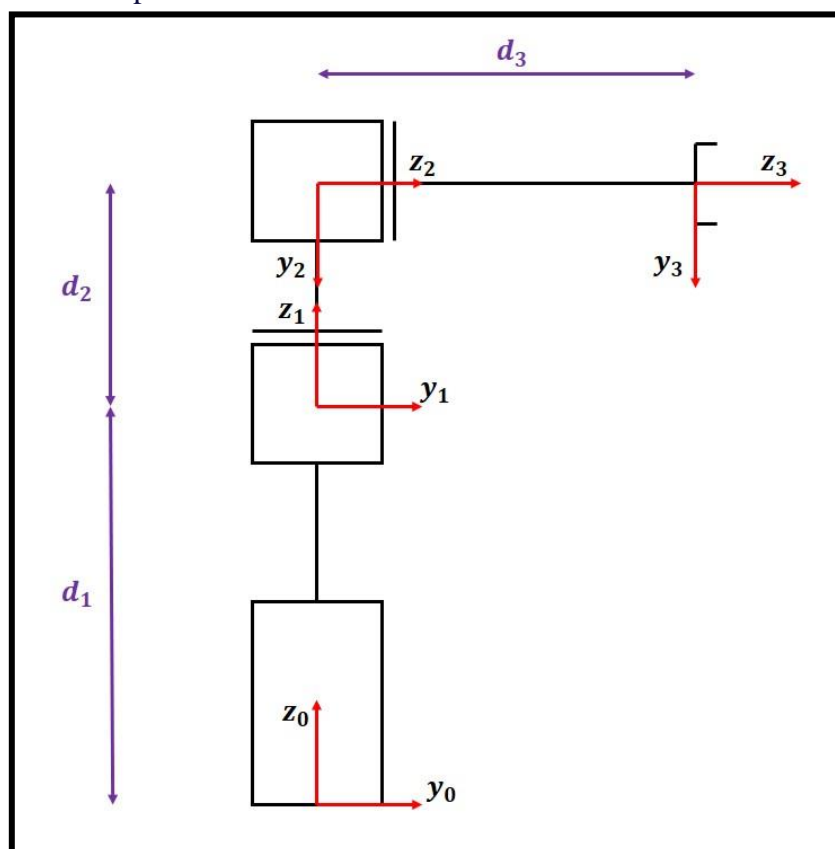
Tabla de parámetros:

link	θ_i	d_i	a_i	α_i
1	θ_1	$d_1 = 190.8\text{mm}$	0	0
2	0	d_2	$L=70\text{ mm}$	-90°
3	0	d_3	0	0

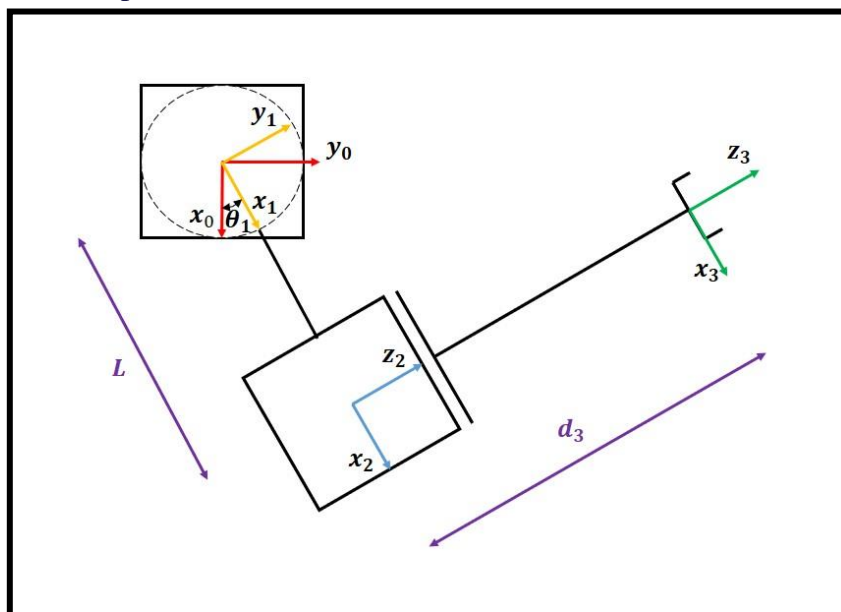
Vista simbólica isométrica:



Vista simbólica de perfil:



Vista simbólica superior:



2.1. Convención modificada Khalil – Dombre:

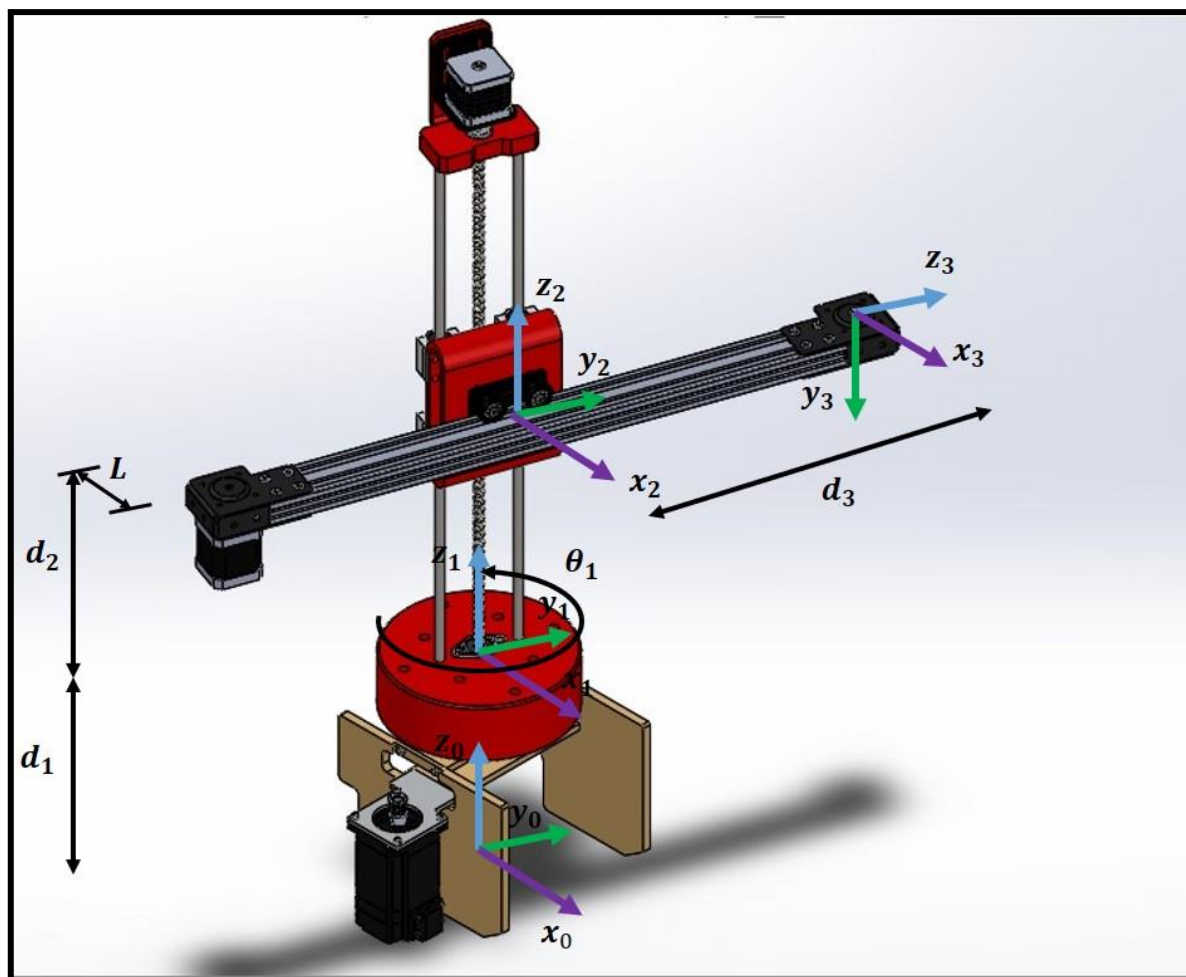
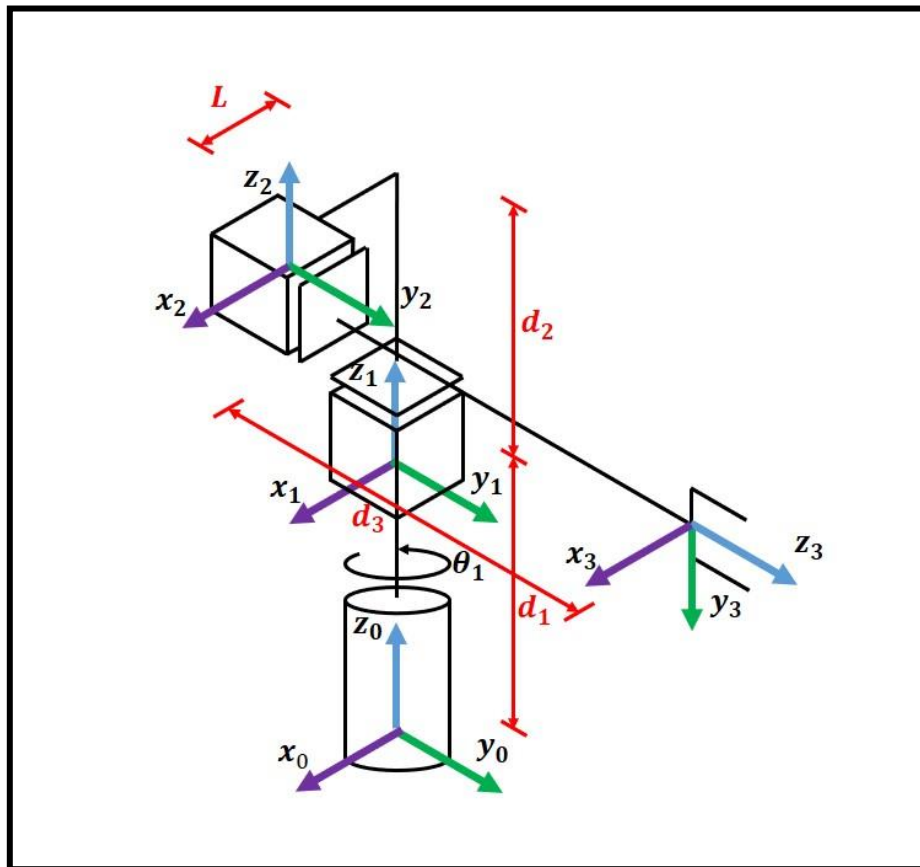


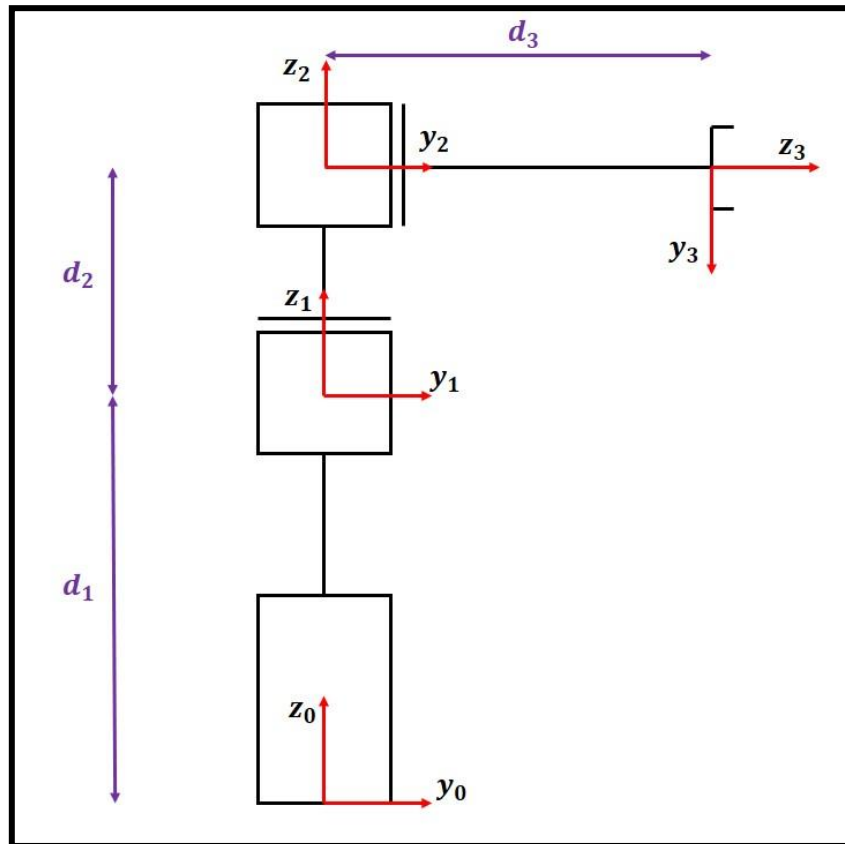
Tabla de parámetros:

link	α_i	a_i	θ_i	d_i
1	0	0	θ_1	$d_1=190.8\text{mm}$
2	0	$L=70\text{ mm}$	0	d_2
3	-90°	0	0	d_3

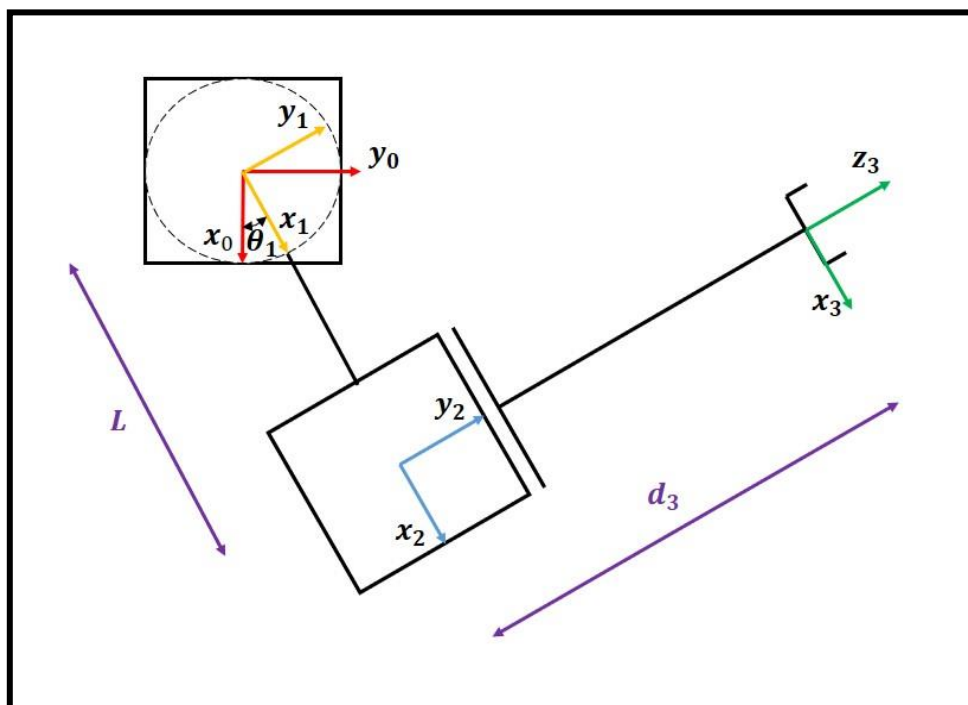
Vista simbólica isométrica:



Vista simbólica de perfil:

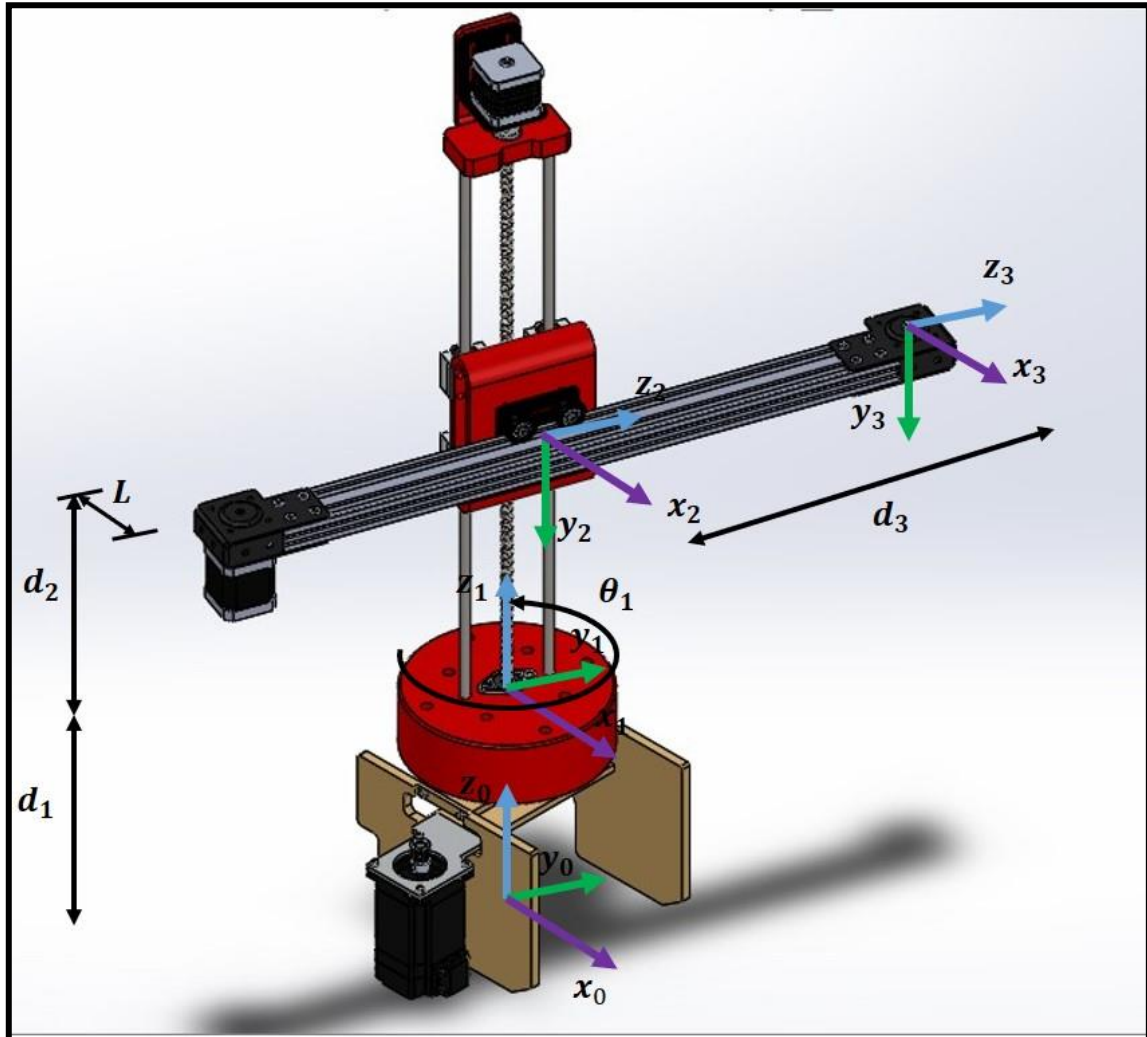


Vista simbólica superior:



3. HOME POSITION DEL ROBOT

De acuerdo al gráfico mostrado, podemos establecer los valores iniciales de las variables de RPP (revolución – prismática – prismática).



Link	θ_i	d_i	a_i	α_i
1	$\theta_1 = 0^\circ$ (inicial)	$d_1 = 190.8 \text{ mm}$	0	0
2	0	$d_2 = 240 \text{ mm}$ (inicial)	$L = 70 \text{ mm}$	-90°
3	0	$d_3 = 246 \text{ mm}$ (inicial)	0	0

Coordenadas del Home Position

$$(x_o, y_o, z_o) = (70, 246, 430.8) \text{ mm}$$

Parámetros:

$$\begin{aligned}d_1 &= 190.8 \text{ mm} \\-120^\circ &\leq \theta_1 \leq 120^\circ \\107 \text{ mm} &\leq d_2 \leq 380 \text{ mm} \\136.5 \text{ mm} &\leq d_3 \leq 355.5 \text{ mm} \\a_2 &= L = 70 \text{ mm}\end{aligned}$$

De la tabla de parámetros DH Waldron – Paul, las matrices de transformación de cada junta son:

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) * \cos(\alpha_i) & \sin(\theta_i) * \sin(\alpha_i) & a_i * \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) * \cos(\alpha_i) & -\cos(\theta_i) * \sin(\alpha_i) & a_i * \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) * \cos(0) & \sin(\theta_1) * \sin(0) & 0 * \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) * \cos(0) & -\cos(\theta_1) * \sin(0) & 0 * \sin(\theta_1) \\ 0 & \sin(0) & \cos(0) & 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} \cos(0) & -\sin(0) * \cos(-90) & \sin(0) * \sin(-90) & 70 \text{ mm} * \cos(0) \\ \sin(0) & \cos(0) * \cos(-90) & -\cos(0) * \sin(-90) & 70 \text{ mm} * \sin(0) \\ 0 & \sin(-90) & \cos(-90) & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} 1 & 0 & 0 & 70 \text{ mm} \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos(0) & -\sin(0) * \cos(0) & \sin(0) * \sin(0) & 0 * \cos(0) \\ \sin(0) & \cos(0) * \cos(0) & -\cos(0) * \sin(0) & 0 * \sin(0) \\ 0 & \sin(0) & \cos(0) & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De este modo, se obtienen las ecuaciones o matrices de cinemática directa:

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 70 \text{ mm} * \cos(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 70 \text{ mm} * \sin(\theta_1) \\ 0 & -1 & 0 & 190.8 \text{ mm} + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 70 \text{ mm} * \cos(\theta_1) - d_3 * \sin(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 70 \text{ mm} * \sin(\theta_1) + d_3 * \cos(\theta_1) \\ 0 & -1 & 0 & 190.8 \text{ mm} + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De la tabla de parámetros DH Khalil-Dombre, las matrices de transformación de cada junta son:

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \sin(\theta_i) * \cos(\alpha_i) & \cos(\theta_i) * \cos(\alpha_i) & -\sin(\alpha_i) & -\sin(\alpha_i) * d_i \\ \sin(\theta_i) * \sin(\alpha_i) & \cos(\theta_i) * \sin(\alpha_i) & \cos(\alpha_i) & \cos(\alpha_i) * d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) * \cos(0) & \cos(\theta_1) * \cos(0) & -\sin(0) & -\sin(0) * 190.8 \text{ mm} \\ \sin(\theta_1) * \sin(0) & \cos(\theta_1) * \sin(0) & \cos(0) & \cos(0) * 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} \cos(0) & -\sin(0) & 0 & 70mm \\ \sin(0) * \cos(0) & \cos(0) * \cos(0) & -\sin(0) & -\sin(0) * d_2 \\ \sin(0) * \sin(0) & \cos(0) * \sin(0) & \cos(0) & \cos(0) * d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} 1 & 0 & 0 & 70mm \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos(0) & -\sin(0) & 0 & 0 \\ \sin(0) * \cos(-90) & \cos(0) * \cos(-90) & -\sin(-90) & -\sin(-90) * d_3 \\ \sin(0) * \sin(-90) & \cos(0) * \sin(-90) & \cos(-90) & \cos(-90) * d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De este modo, se obtienen las ecuaciones o matrices de cinemática directa:

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 190.8 \text{ mm} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 70mm * \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 70mm * \sin(\theta_1) \\ 0 & 0 & 1 & 190.8 \text{ mm} + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 70mm * \cos(\theta_1) - d_3 * \sin(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 70mm * \sin(\theta_1) + d_3 * \cos(\theta_1) \\ 0 & -1 & 0 & 190.8 \text{ mm} + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rango de movimiento de los mecanismos (Restricciones Reales)

Junta de revolución – Junta 1

Es evidente que la junta de revolución se puede identificar con la variable θ_1 , a la cual tiene un rango de 240° para prevenir enredos de los cables. Entonces, se puede decir que:

$$-120^\circ < \theta_1 < 120^\circ$$



Límite inferior

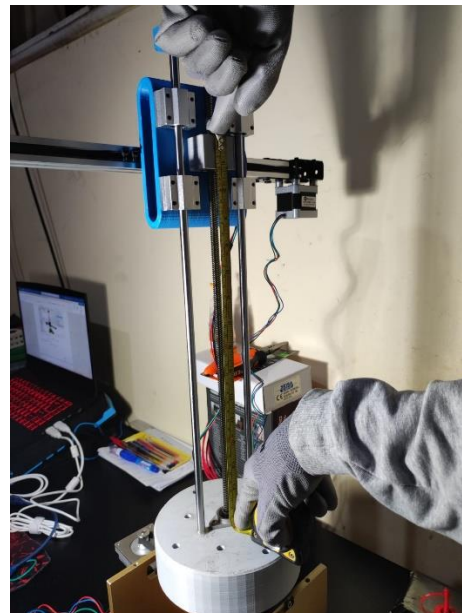


Límite superior

Junta prismática – Junta 2



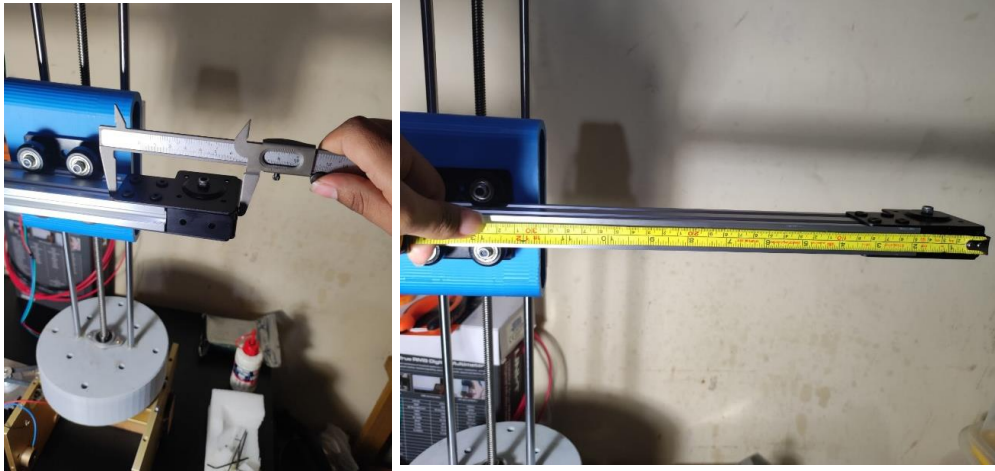
Límite inferior



Límite superior

$$107 \text{ mm} (122 - 15) < d_2 < 380 \text{ mm} (395 - 15)$$

Junta prismática – Junta 3



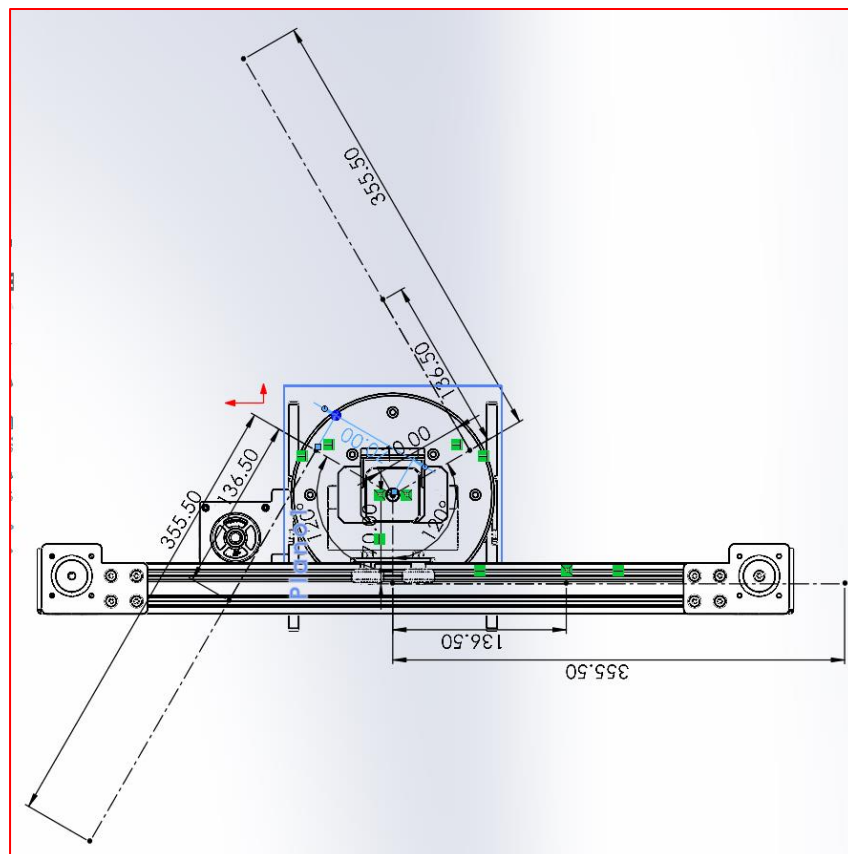
Límite inferior

Límite superior

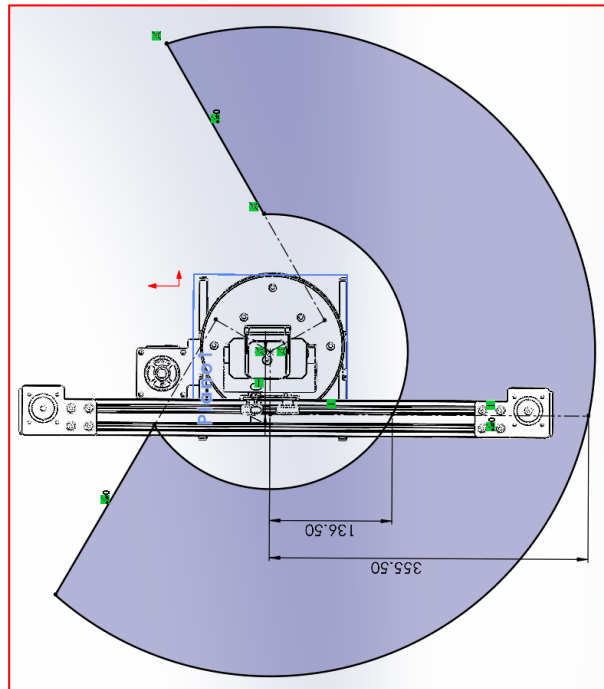
$$136.5 \text{ mm} (50.5 + 86) < d_3 < 355.5 \text{ mm} (50.5 + 305)$$

Construcción del Workspace

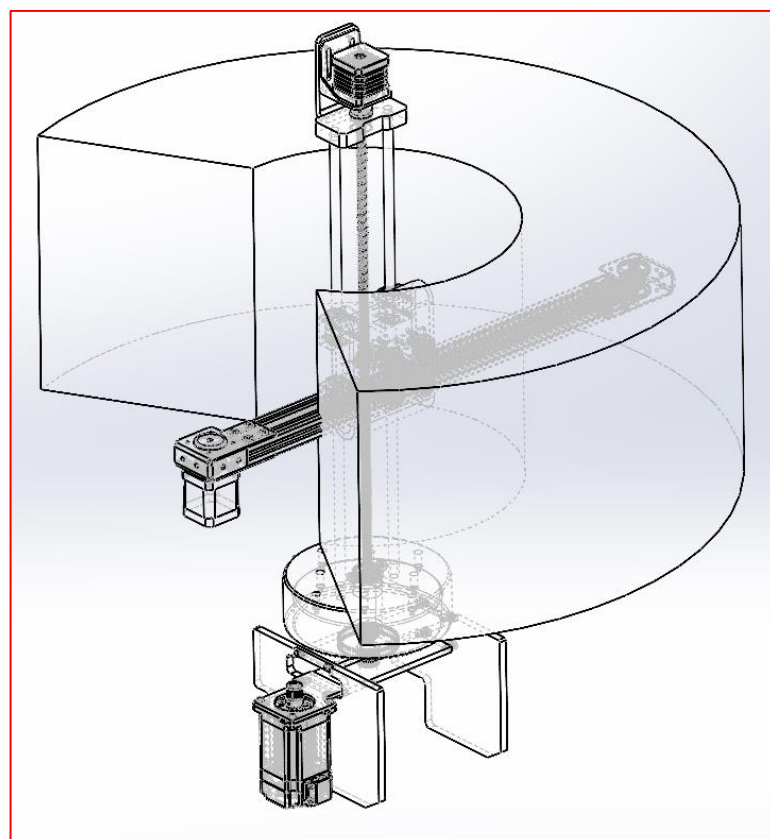
- ✚ Usamos el rango de movimiento de la junta 1 y junta 3, el valor del parámetro constante “L” y la posición inicial del robot para hallar puntos de trabajo límites:

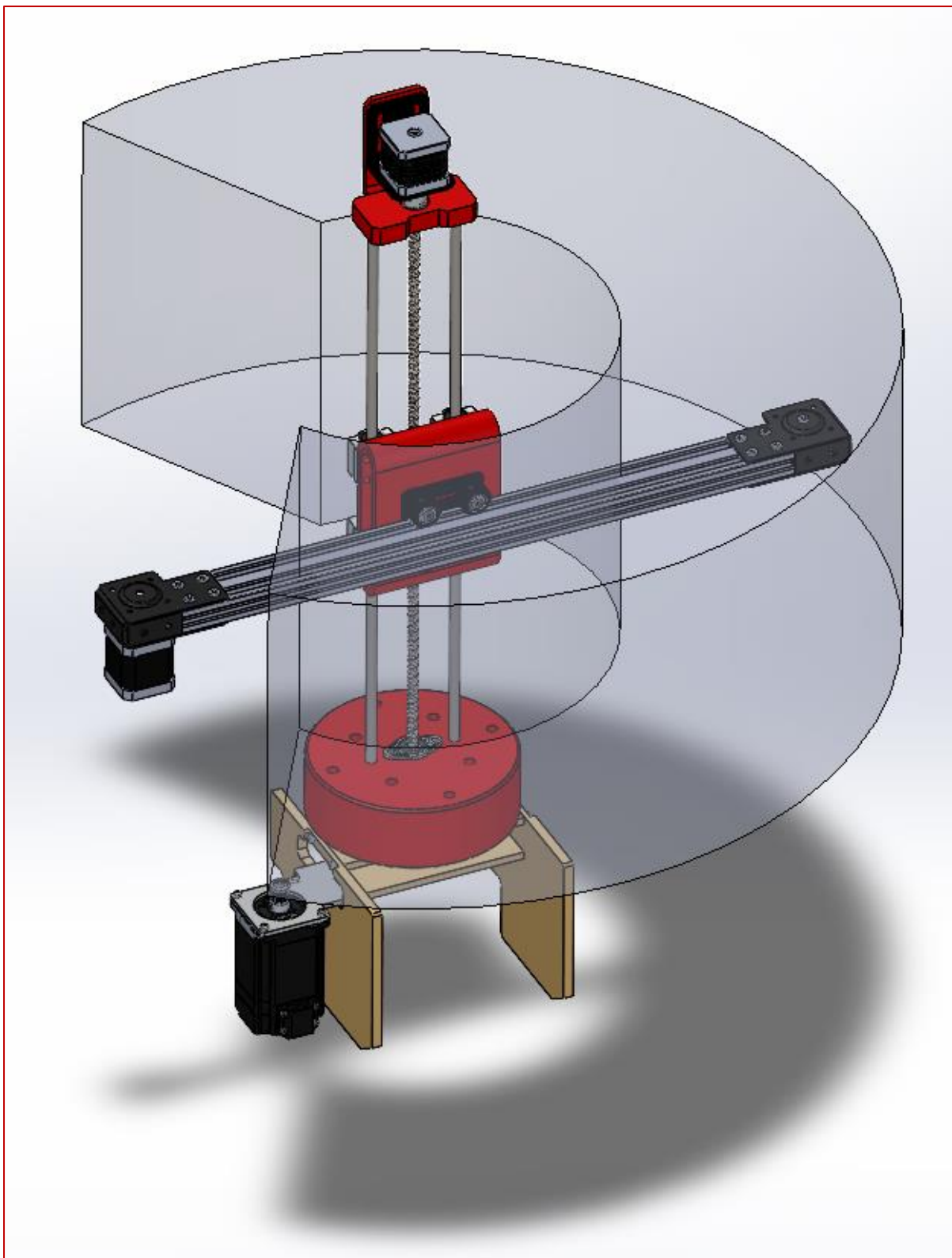


- Con ayuda de dichos puntos y tomando en cuenta el movimiento giratorio y lineal del robot obtenemos la siguiente sección de trabajo:



- Usando los límites de la junta 2, se completa espacio de trabajo del robot :

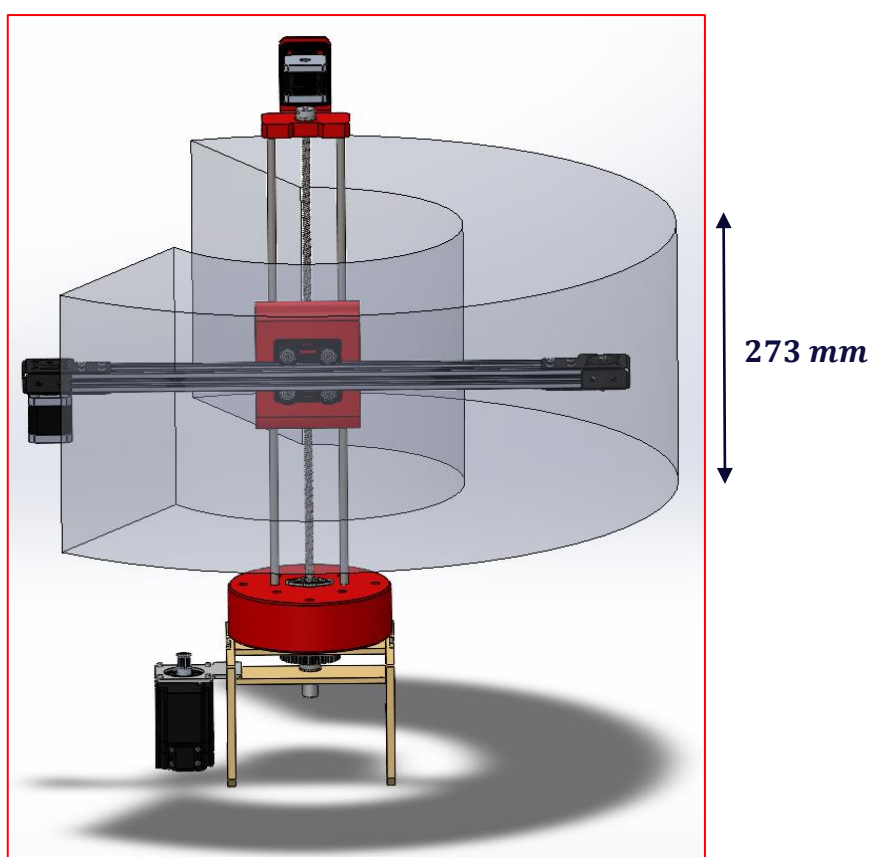
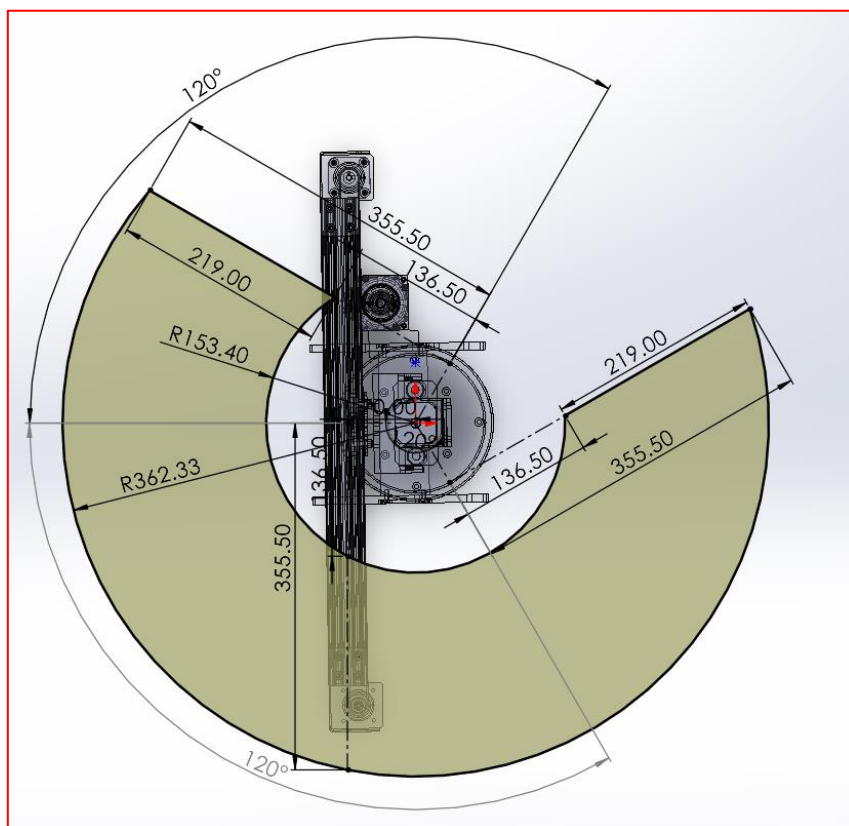




$$\text{Volumen de Trabajo} = 61607.06 \text{ cm}^3 = 0.0616 \text{ m}^3$$

Volumen = 61607059.19 milímetros cúbicos

Medidas del Work-Space



6.1. Método Algebraico:

$$T_3^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & a_2 * \cos(\theta_1) - d_3 * \sin(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & a_2 * \sin(\theta_1) + d_3 * \cos(\theta_1) \\ 0 & -1 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De la ecuación de cinemática directa:

$$a_2 * \cos(\theta_1) - d_3 * \sin(\theta_1) = x \dots (1)$$

$$a_2 * \sin(\theta_1) + d_3 * \cos(\theta_1) = y \dots (2)$$

$$d_1 + d_2 = z \dots (3)$$

Despejando (3):

$$d_2 = z - d_1$$

De (1) y (3):

$$x^2 + y^2 = (a_2 * \cos(\theta_1) - d_3 * \sin(\theta_1))^2 + (a_2 * \sin(\theta_1) + d_3 * \cos(\theta_1))^2$$

$$x^2 + y^2 = a_2^2 * \cos^2(\theta_1) + d_3^2 * \sin^2(\theta_1) - 2 * a_2 * d_3 * \cos(\theta_1) * \sin(\theta_1) + a_2^2 * \sin^2(\theta_1) + d_3^2 * \cos^2(\theta_1) + 2 * a_2 * d_3 * \cos(\theta_1) * \sin(\theta_1)$$

$$x^2 + y^2 = a_2^2 + d_3^2$$

$$d_3 = \sqrt{x^2 + y^2 - a_2^2}$$

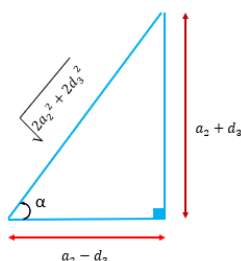
También:

$$x + y = (a_2 + d_3) * \cos(\theta_1) + (a_2 - d_3) * \sin(\theta_1)$$

$$\frac{x + y}{\sqrt{(a_2 + d_3)^2 + (a_2 - d_3)^2}} = \frac{(a_2 + d_3) * \cos(\theta_1) + (a_2 - d_3) * \sin(\theta_1)}{\sqrt{2a_2^2 + 2d_3^2}}$$

$$\sin(\theta_1 + \alpha) = \frac{x + y}{\sqrt{2a_2^2 + 2d_3^2}}$$

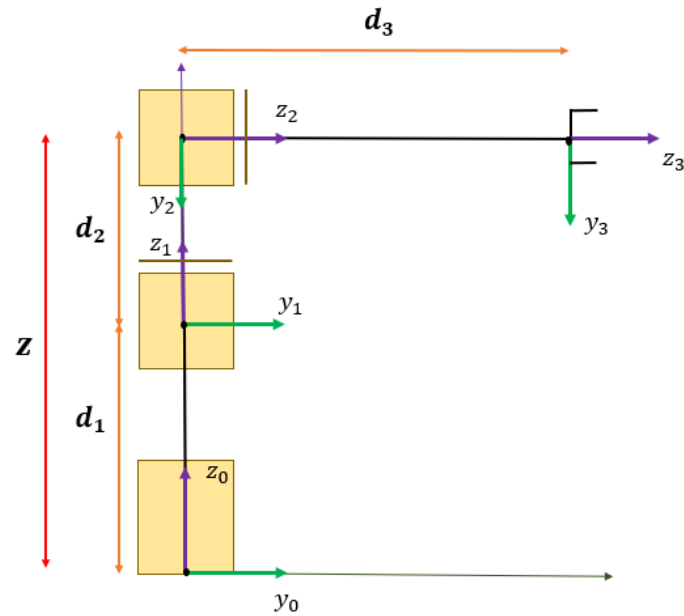
$$\alpha = \text{atan}\left(\frac{a_2 + d_3}{a_2 - d_3}\right)$$



$$\theta_1 = \text{atan2}(-x, y) + \text{atan2}(70 \text{ mm}, d_3)$$

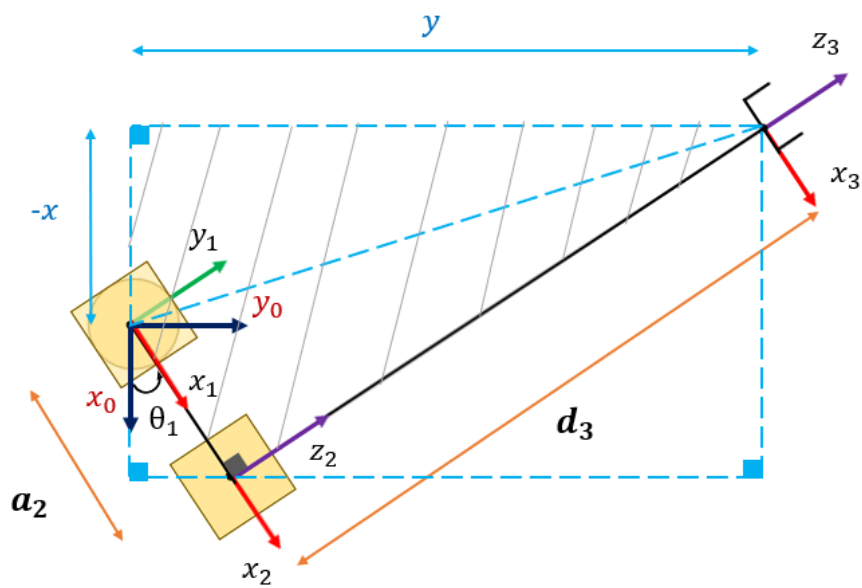
6.2. Método Geométrico:

Cálculo de d_2 :



$$d_1 + d_2 = z \quad \Rightarrow \quad d_2 = z - d_1$$

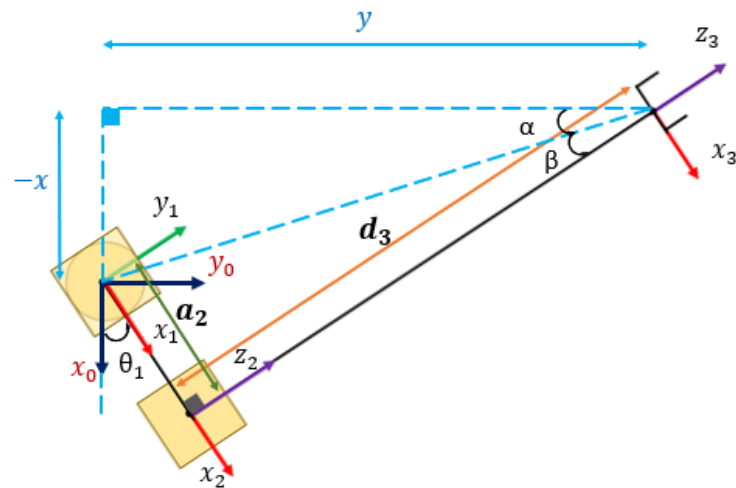
Cálculo de d_3 :



$$x^2 + y^2 = a_2^2 + d_3^2$$

$$d_3 = \sqrt{x^2 + y^2 - a_2^2}$$

Cálculo de θ_1 :



Se obtiene:

$$\theta_1 = \alpha + \beta$$

$$\alpha = \text{atan2}(-x, y)$$

$$\beta = \text{atan2}(a_2, d_3)$$

Reemplazando:

$$\theta_1 = \text{atan2}(-x, y) + \text{atan2}(a_2, d_3)$$

$$\theta_1 = \text{atan2}(-x, y) + \text{atan2}(70 \text{ mm}, d_3)$$

7. PROGRAMA ARDUINO + MATLAB

7.1. Cinemática Directa:

```
function [T01,T02,T03]=cinematica_directa(q1,q2,q3, DATA)
% 1.1. SISTEMA COORDENADO {1}
theta1 = q1;
d1      = DATA.d1;
a1      = 0;
alpha1  = 0;
A1 = matriz_homogenea_DH(theta1,d1,a1,alpha1);
T01 = A1;
% 1.2. SISTEMA COORDENADO {2}
theta2 = 0;
d2      = q2;
a2      = 70/1000;
alpha2  = -pi/2;
A2 = matriz_homogenea_DH(theta2,d2,a2,alpha2);
T02 = A1*A2;
% 1.3. SISTEMA COORDENADO {3}
theta3 = 0;
d3      = q3;
a3      = 0;
alpha3  = 0;
A3 = matriz_homogenea_DH(theta3,d3,a3,alpha3);
T03 = T02*A3;

%-----%
% B. VALORES INICIALES DEL ROBOT (HOME POSITION)
%-----%
% B.1. VALORES INICIALES DE LAS JUNTAS
q1 = 0;           % th1 revolucion (sexag)
q2 = 240;         % d2 prismatico 120mm (por seguridad no empieza en 0mm)
q3 = 246;         % d3 prismatico 15mm
set(handles.q1, 'String',round(q1*1000)/1000);
set(handles.q2, 'String',round(q2*1000)/1000);
set(handles.q3, 'String',round(q3*1000)/1000);

% B.2. ESTRUCTURA PARA ALMACENAR LAS CONSTANTES DEL ROBOT
DATA.d1 = 190.8/1000; % En metros
DATA.a1 = 0;
DATA.a2 = 70/1000;
DATA.a3 = 0;
handles.DATA = DATA; % Lo guardamos para poder usarlo en
                     % otras funciones de la GUIDE
% B.3. VARIABLES PARA ALMACENAR EL ULTIMO VALOR DE LAS JUNTAS
handles.q1_old = q1;
handles.q2_old = q2;
handles.q3_old = q3;

%-----%
% C. HALLAMOS LA CINEMATICA PARA LA CONF. INICIAL
%-----%
q1_rad = q1*pi/180; % A radianes
q2_metros = q2/1000; % A metros
q3_metros = q3/1000;
[T01,T02,T03] = cinematica_directa(q1_rad,q2_metros,...
    q3_metros, DATA);
x = T03(1,4);
y = T03(2,4);
```

```

z = T03(3,4);
set(handles.x, 'String',round(x*1000)/1000);
set(handles.y, 'String',round(y*1000)/1000);
set(handles.z, 'String',round(z*1000)/1000);

%-----%
% CONFIGURACION DE LA FIGURA PARA EL ROBOT
%-----%
view(3)
grid on
axis([-0.5 0.5 -0.5 0.5 0 0.6])
xlabel('x (m)')
ylabel('y (m)')
% SISTEMA INERCIAL - TIERRA
T0 = eye(4);
plot_frame(T0,'frame','0', 'length', 0.1);
%-----%
% PLOTEAMOS LOS SISTEMAS COORDENADOS
h1 = plot_frame(T01,'frame','1', 'length', 0.1);
h2 = plot_frame(T02,'frame','2', 'length', 0.1);
h3 = plot_frame(T03,'frame','3', 'length', 0.1);
handles.h1 = h1;
handles.h2 = h2;
handles.h3 = h3;
%-----%
% PLOTEAMOS LOS ESLABONES
% ESLABON 1
h_link1 = plot_link(T01,T0,DATA.a1, 'k');
% ESLABON 2
h_link2 = plot_link_DH(T02,T01,'k');
% ESLABON 3
h_link3 = plot_link(T03,T02,DATA.a3, 'k');
% GUARDAMOS LOS HANDLES
handles.h_link1 = h_link1;
handles.h_link2 = h_link2;
handles.h_link3 = h_link3;
%-----%
% 1. LEEMOS EL VALOR DE LOS ANGULOS (caja de texto)
%-----%
q1 = get(handles.q1, 'String'); % esta en sexag
q1 = str2double(q1);
q2 = get(handles.q2, 'String'); % esta en mm
q2 = str2double(q2);
q3 = get(handles.q3, 'String'); % esta en mm
q3 = str2double(q3);
disp('---')
fprintf('q1 (sexag):%2.4f\n', q1)
fprintf('q2 (mm):%2.4f\n', q2)
fprintf('q3 (mm):%2.4f\n', q3)

%-----%
% 2. VERIFICAMOS LIMITACIONES
%-----%
if(q1<-120 || q1>120)
    errordlg('q1 fuera de rango (-120,120)', 'q1')
    return
end
if(q2<107 || q2>380)

```

```

        errordlg('q2 fuera de rango (107mm,380mm)', 'q2')
        return
    end
    if(q3<136.5 || q3>355.5) % Prismático
        errordlg('q3 fuera de rango (136.5mm,355.5mm)', 'q3')
        return
    end

%-----%
% 3. ACTUALIZAMOS LA CINEMATICA DIRECTA
%-----%
q1_rad = q1*pi/180;
q2_metros = q2/1000;
q3_metros = q3/1000;
[T01,T02,T03] = cinematica_directa(q1_rad,q2_metros,...
    q3_metros, handles.DATA);
x = T03(1,4);
y = T03(2,4);
z = T03(3,4);
set(handles.x, 'String',round(x*1000)/1000);
set(handles.y, 'String',round(y*1000)/1000);
set(handles.z, 'String',round(z*1000)/1000);

%-----%
% PLOTEAMOS LOS SISTEMAS COORDENADOS
%-----%
%axes(handles.axes_robot)
plot_frame(handles.h1, T01);
plot_frame(handles.h2, T02);
plot_frame(handles.h3, T03);
% PLOTEAMOS LOS ESLABONES
T0 = eye(4);
plot_link(T01,T0,handles.DATA.a1, handles.h_link1);
% eslabon 2
plot_link_DH(T02,T01,handles.h_link2);
% eslabon 3
plot_link(T03,T02,handles.DATA.a3, handles.h_link3);
%-----%
% 1. LEEMOS EL VALOR DE LOS ANGULOS (caja de texto)
%-----%
q1 = get(handles.q1, 'String'); % esta en sexag
q1 = str2double(q1);
q2 = get(handles.q2, 'String'); % esta en mm
q2 = str2double(q2);
q3 = get(handles.q3, 'String'); % esta en mm
q3 = str2double(q3);
disp('---')
fprintf('q1 (sexag):%2.4f\n', q1)
fprintf('q2 (mm):%2.4f\n', q2)
fprintf('q3 (mm):%2.4f\n', q3)

%-----%
% 2. VERIFICAMOS LIMITACIONES
%-----%
if(q1<-120 || q1>120)
    errordlg('q1 fuera de rango (-120,120)', 'q1')
    return
end

```

```

if(q2<107 || q2>380)
    errordlg('q2 fuera de rango (107mm,380mm)', 'q2')
    return
end
if(q3<136.5 || q3>355.5) % Prismatico
    errordlg('q3 fuera de rango (136.5mm,355.5mm)', 'q3')
    return
end

%-----%
% 3. HALLAMOS LOS DELTAS DE CADA JUNTA
%-----%
% 3.1. PARA LA JUNTA 1 - MOTOR 1
q1_delta = q1 - handles.q1_old; % Esta en sexag
if(q1_delta>=0)
    %q1_delta = q1_delta;
    q1_dir = 1;
else
    q1_delta = -q1_delta;
    q1_dir = 0;
end
% 3.2. PARA LA JUNTA 2 - MOTOR 2
q2_delta = q2 - handles.q2_old; % En mm
if(q2_delta>=0)
    %J2_delta = q2_delta;
    q2_dir = 1;
else
    q2_delta = -q2_delta;
    q2_dir = 0;
end
% 3.3. PARA LA JUNTA 3 - MOTOR 3
q3_delta = q3 - handles.q3_old;
if(q3_delta>=0)
    %q3_delta = q3_delta;
    q3_dir = 1;
else
    q3_delta = -q3_delta;
    q3_dir = 0;
end
fprintf('q1_delta(sexag)=%2.4f, q1_dir:%d\n', q1_delta, q1_dir)
fprintf('q2_delta(mm)=%2.4f, q2_dir:%d\n', q2_delta, q2_dir)
fprintf('q3_delta(mm)=%2.4f, q3_dir:%d\n', q3_delta, q3_dir)

%-----%
% 4. APLICAMOS LA TRANSMISION PARA HALLAR EL GIRO
% QUE DEBE REALIZAR LOS MOTORES
%-----%
% FACTOR DE TRANSMISION DE LAS JUNTAS
k1 = 4.5; % Esto valor debe estar fundamentado en el informe
k2 = 45.7;
k3 = 9;
% DELTA EN EL EJE DEL ROTOR
delta_motor1 = k1*q1_delta; % Este delta ya es positivo
delta_motor2 = k2*q2_delta; % Ya es un angulo
delta_motor3 = k3*q3_delta;

%-----%
% 5. HALLAMOS LOS PASOS Y DIRECCION PARA CADA MOTOR

```



```

%-----%
% 5.1. MOTOR 1
N1 = 200; % Pasos por vuelta de mi motor
m1_pasos = N1*delta_motor1/360;
m1_pasos = round(m1_pasos); % Convertimos a entero
m1_dir = q1_dir;
fprintf('m1_pasos:%d, m1_dir:%d\n', m1_pasos,m1_dir)
% 5.2. MOTOR 2
N2 = 200; % Pasos por vuelta de mi motor
m2_pasos = N2*delta_motor2/360;
m2_pasos = round(m2_pasos); % Convertimos a entero
m2_dir = q2_dir;
fprintf('m2_pasos:%d, m2_dir:%d\n', m2_pasos,m2_dir)
% 5.3. MOTOR 3
N3 = 200; % Pasos por vuelta de mi motor
m3_pasos = N3*delta_motor3/360;
m3_pasos = round(m3_pasos); % Convertimos a entero
m3_dir = q3_dir;
fprintf('m3_pasos:%d, m3_dir:%d\n', m3_pasos,m3_dir)

%-----%
% 6. ENVIAMOS LOS ANGULOS AL ARDUINO
%-----%
% LO CONVERTIMOS EN UNA CADENA
cad = strcat(num2str(m1_pasos), ',', num2str(m1_dir), ...
            ',', num2str(m2_pasos), ',', num2str(m2_dir), ...
            ',', num2str(m3_pasos), ',', num2str(m3_dir));
fprintf('cadena a enviar: %s\n', cad)
% ENVIAMOS AL ARDUINO
fprintf(handles.puerto, '%s', cad);

%-----%
% ACTUALIZAMOS VARIABLES
%-----%
handles.q1_old = q1;
handles.q2_old = q2;
handles.q3_old = q3;

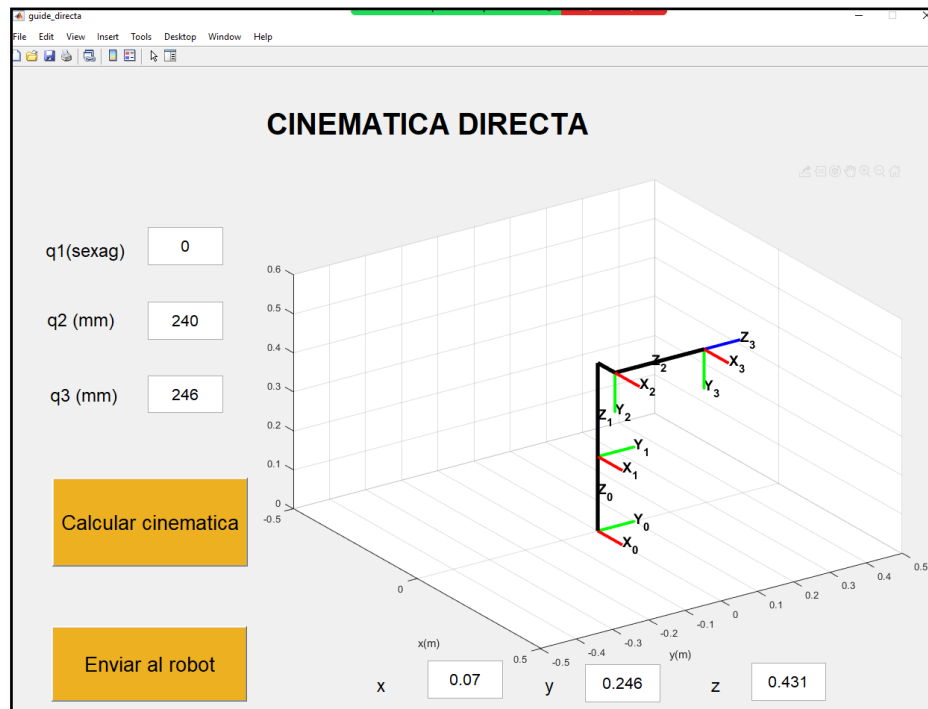
% ACTUALIZAMOS TODOS LOS DATOS EN EL SISTEMA
guidata(hObject, handles);
%-----%
% CERRAMOS EL PUERTO SERIAL
%-----%
s1 = handles.puerto;
fclose(s1);
disp('...cerrando ventana')

% Hint: delete(hObject) closes the figure
delete(hObject);

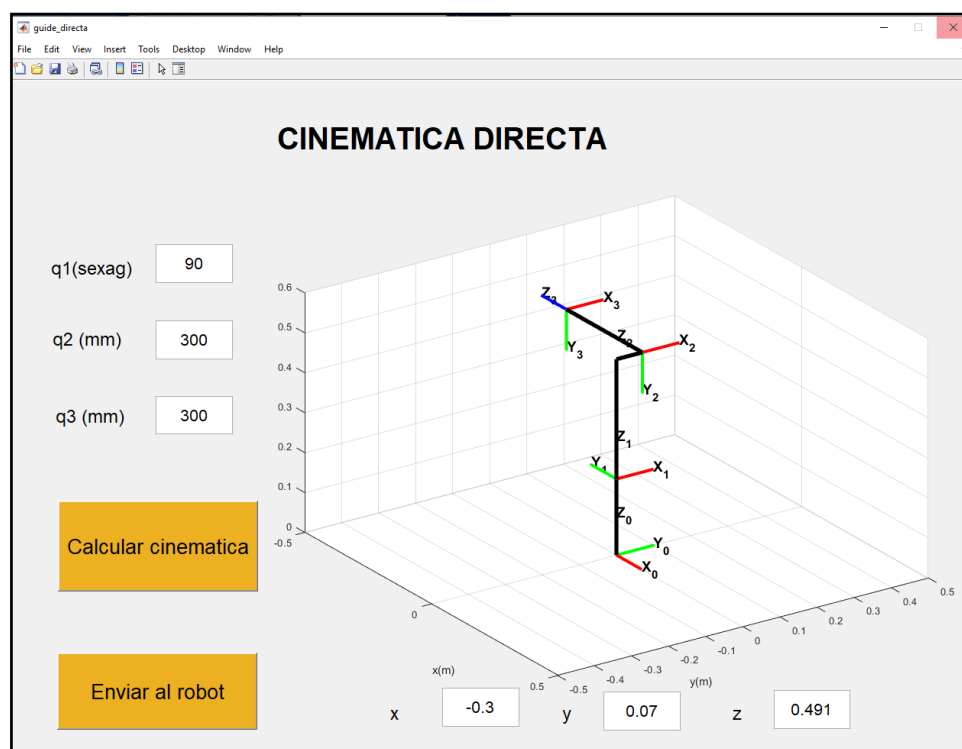
```

Simulación:

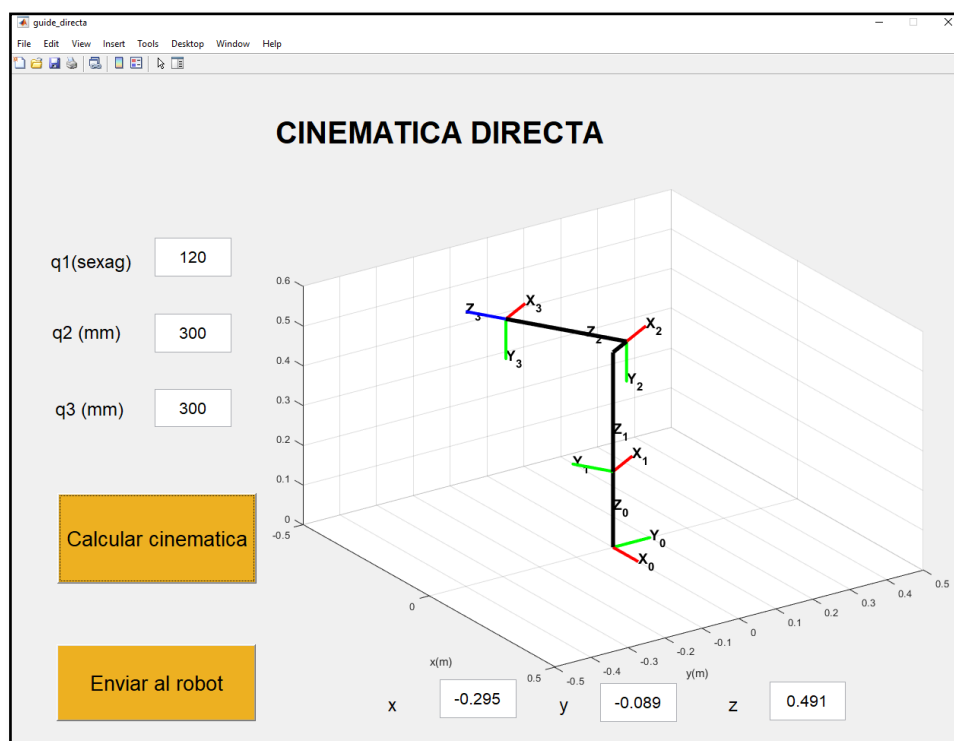
Valores iniciales del robot (home position):



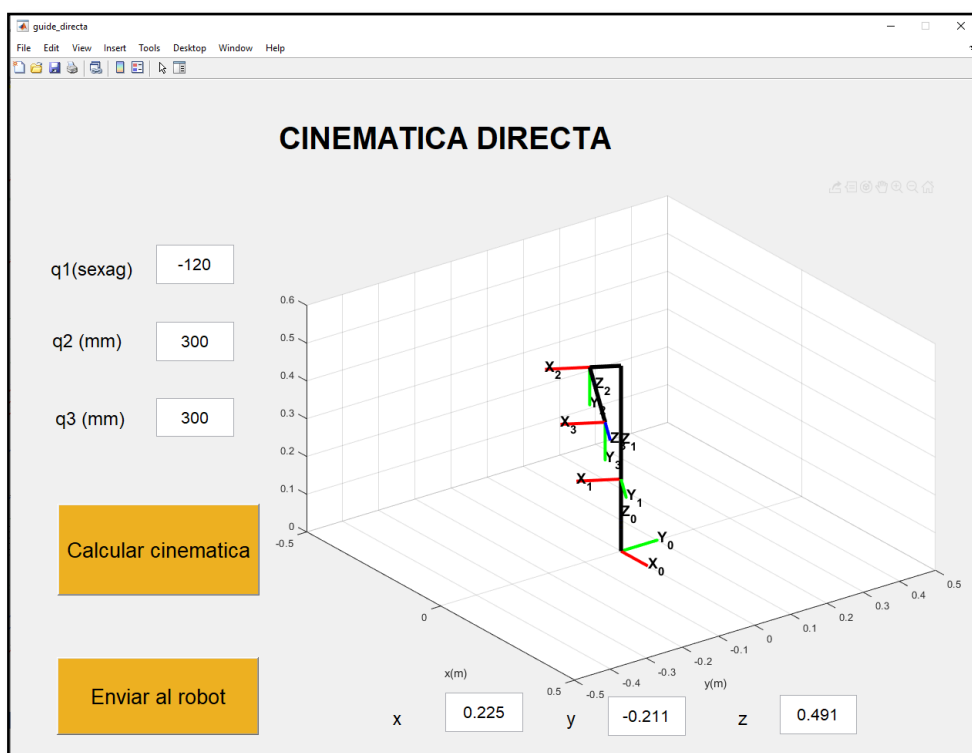
Posición: $q1=90^\circ$, $q2=300\text{mm}$, $q3=300\text{mm}$



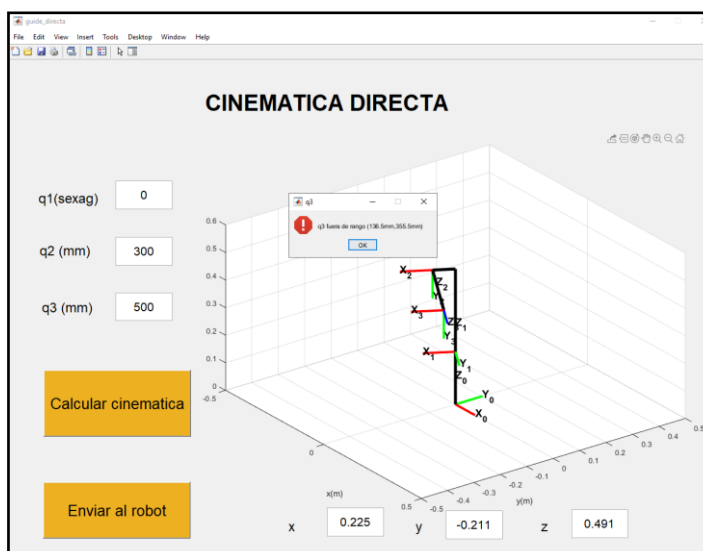
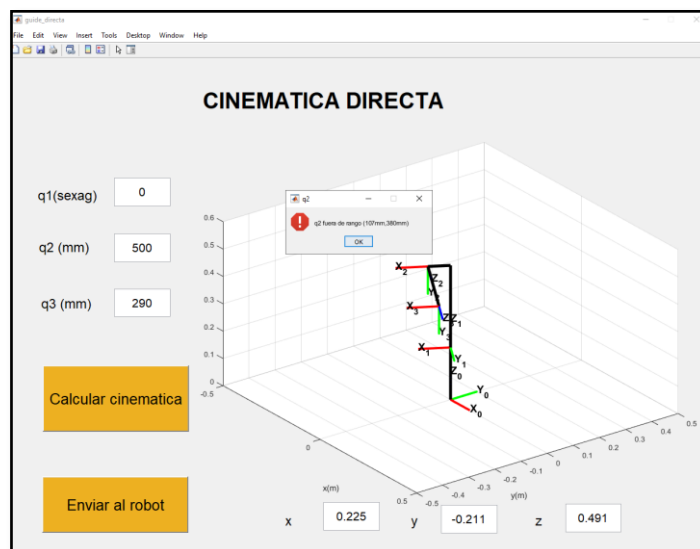
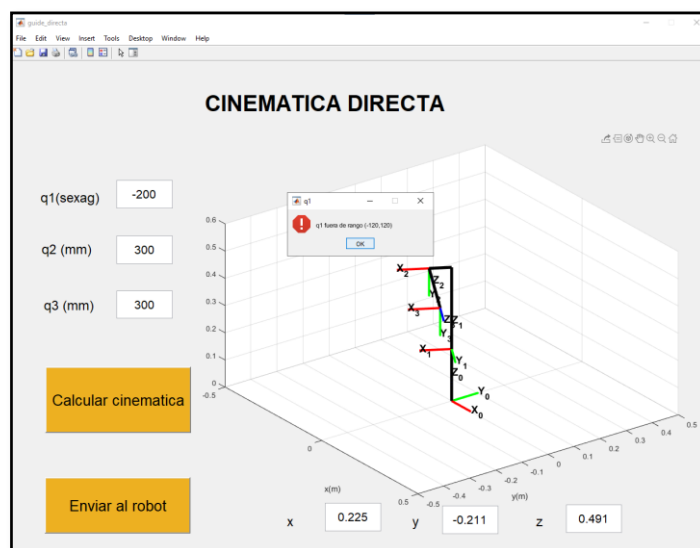
Posición: $q_1=120^\circ$, $q_2=300\text{mm}$, $q_3=300\text{mm}$



Posición: $q_1 = -120^\circ$, $q_2=300\text{mm}$, $q_3=300\text{mm}$



Valores fuera de rango para q_1 , q_2 , q_3 :



7.2. Cinemática Inversa:

```
function [q1,q2,q3]=cinematica_inversa(x,y,z, DATA)
% De nuestro manipulador cilindrico usando Waldron Paul
% En la estructura DATA estan los parametros constantes
% del robot

% 3. CINEMATICA INVERSA
%-----%
%-----%
% 3.1. EMPEZAMOS CON "q2=d2"
q2 = z-0.1908;
q2 = 1000*q2;
fprintf('q2:%2.4f (mm)\n', q2)
% 3.2. SEGUIMOS CON "q3=d3"
q3 = sqrt(x^2+y^2-(0.07)^2);
d3 = q3;
q3 = 1000*q3;
fprintf('q3:%2.4f (mm)\n', q3)
% 3.3. SEGUIMOS CON "q1=theta1"
theta1=atan2(-x,y)+atan2(0.07,sqrt(x^2+y^2-(0.07)^2));
q1 = theta1*180/pi;
fprintf('q1:%2.4f, %2.4f(sexag)\n', theta1, q1)
```

EN LA PROGRAMACIÓN DE LA GUIDE

Programación de la inicialización de la simulación:

```
%-----%
% B. VALORES INICIALES DEL ROBOT (HOME POSITION)
%-----%
% B.1. VALORES INICIALES DE LAS COORDENADAS
x = 70;
y = 246;
z = 430.8;
set(handles.x, 'String',round(x*1000)/1000);
set(handles.y, 'String',round(y*1000)/1000);
set(handles.z, 'String',round(z*1000)/1000);

% B.2. ESTRUCTURA PARA ALMACENAR LAS CONSTANTES DEL ROBOT
DATA.d1 = 190.8/1000; % En metros
DATA.a1 = 0;
DATA.a2 = 70/1000;
DATA.a3 = 0;
handles.DATA = DATA; % Lo guardamos para poder usarlo en

%-----%
% C. HALLAMOS LA CINEMATICA INVERSA PARA LA CONF. INICIAL
%-----%
x_metros = x/1000; % A metros
y_metros = y/1000; % A metros
z_metros = z/1000; % A metros
[q1,q2,q3] = cinematica_inversa(x_metros,y_metros,z_metros, DATA);
set(handles.q1, 'String',round((q1*pi/180)*1000)/1000);
set(handles.q10, 'String',round(q1*10)/10);
set(handles.q2, 'String',round(q2*1000)/1000);
set(handles.q3, 'String',round(q3*1000)/1000);
```

```

% B.3. VARIABLES PARA ALMACENAR EL ULTIMO VALOR DE LAS JUNTAS
handles.q1_old = q1;
handles.q2_old = q2;
handles.q3_old = q3;
%-----%
% C. HALLAMOS LA CINEMATICA DIRECTA PARA LA CONF. INICIAL
%-----%
q1_rad = q1*pi/180;      % A radianes
q2_metros = q2/1000;     % A metros
q3_metros = q3/1000;
[T01,T02,T03] = cinematica_directa(q1_rad,q2_metros,q3_metros, DATA);
%-----%
% CONFIGURACION DE LA FIGURA PARA EL ROBOT
%-----%
view(3)
grid on
axis([-0.5  0.5  -0.5  0.5  0  0.6])
xlabel('x (m)')
ylabel('y (m)')
% SISTEMA INERCIAL - TIERRA
T0 = eye(4);
plot_frame(T0, 'frame', '0', 'length', 0.1);
%-----%
% PLOTEAMOS LOS SISTEMAS COORDENADOS
h1 = plot_frame(T01, 'frame', '1', 'length', 0.1);
h2 = plot_frame(T02, 'frame', '2', 'length', 0.1);
h3 = plot_frame(T03, 'frame', '3', 'length', 0.1);
handles.h1 = h1;
handles.h2 = h2;
handles.h3 = h3;
%-----%
% PLOTEAMOS LOS ESLABONES
% ESLABON 1
h_link1 = plot_link(T01,T0,DATA.a1, 'k');
% ESLABON 2
h_link2 = plot_link_DH(T02,T01,'k');
% ESLABON 3
h_link3 = plot_link(T03,T02,DATA.a3, 'k');
% GUARDAMOS LOS HANDLES
handles.h_link1 = h_link1;
handles.h_link2 = h_link2;
handles.h_link3 = h_link3;

```

Cinemática:

```

% 1. LEEMOS EL VALOR DE LOS ANGULOS (caja de texto)
%-----%
x = get(handles.x, 'String');      % esta en sexag
x = str2double(x);
y = get(handles.y, 'String');      % esta en mm
y = str2double(y);
z = get(handles.z, 'String');      % esta en mm
z = str2double(z);
disp('---')
fprintf('x (mm):%2.4f\n', x)
fprintf('y (mm):%2.4f\n', y)
fprintf('z (mm):%2.4f\n', z)

x_metros = x/1000;      % A metros

```

```

y_metros = y/1000;      % A metros
z_metros = z/1000;      % A metros
[q1,q2,q3] = cinematica_inversa(x_metros,y_metros,z_metros, handles.DATA);
set(handles.q1, 'String',round((q1*pi/180)*1000)/1000);
set(handles.q10, 'String',round(q1*10)/10);
set(handles.q2, 'String',round(q2*1000)/1000);
set(handles.q3, 'String',round(q3*1000)/1000);

%-----%
% 2. VERIFICAMOS LIMITACIONES
%-----%
if(q1<-120 || q1>120)
    errordlg('q1 fuera de rango (-120,120)', 'q1')
    return
end
if(q2<107 || q2>380)
    errordlg('q2 fuera de rango (107mm,380mm)', 'q2')
    return
end
if(q3<136.5 || q3>355.5) % Prismatico
    errordlg('q3 fuera de rango (136.5mm,355.5mm)', 'q3')
    return
end

%-----%
% 3. ACTUALIZAMOS LA CINEMATICA DIRECTA
%-----%
q1_rad = q1*pi/180;
q2_metros = q2/1000;
q3_metros = q3/1000;
[T01,T02,T03] = cinematica_directa(q1_rad,q2_metros,q3_metros, handles.DATA);

%-----%
% PLOTEAMOS LOS SISTEMAS COORDENADOS
%-----%
%axes(handles.axes_robot)
plot_frame(handles.h1, T01);
plot_frame(handles.h2, T02);
plot_frame(handles.h3, T03);
% PLOTEAMOS LOS ESLABONES
T0 = eye(4);
plot_link(T01,T0,handles.DATA.a1, handles.h_link1);
% eslabon 2
plot_link_DH(T02,T01,handles.h_link2);
% eslabon 3
plot_link(T03,T02,handles.DATA.a3, handles.h_link3);

```

Comunicacion con Arduino:

```

% 1. LEEMOS EL VALOR DE LOS ANGULOS (caja de texto)
%-----%
x = get(handles.x, 'String');      % esta en sexag
x = str2double(x);
y = get(handles.y, 'String');      % esta en mm
y = str2double(y);
z = get(handles.z, 'String');      % esta en mm
z = str2double(z);
disp('---')
fprintf('x (mm) :%2.4f\n', x)

```

```

fprintf('y (mm):%2.4f\n', y)
fprintf('z (mm):%2.4f\n', z)

x_metros = x/1000;      % A metros
y_metros = y/1000;      % A metros
z_metros = z/1000;      % A metros
[q1,q2,q3] = cinematica_inversa(x_metros,y_metros,z_metros, handles.DATA);
set(handles.q1, 'String',round((q1*pi/180)*1000)/1000);
set(handles.q10, 'String',round(q1*10)/10);
set(handles.q2, 'String',round(q2*1000)/1000);
set(handles.q3, 'String',round(q3*1000)/1000);

%-----%
% 2. VERIFICAMOS LIMITACIONES
%-----%
if(q1<-120 || q1>120)
    errordlg('q1 fuera de rango (-120,120)', 'q1')
    return
end
if(q2<107 || q2>380)
    errordlg('q2 fuera de rango (107mm,380mm)', 'q2')
    return
end
if(q3<136.5 || q3>355.5) % Prismatico
    errordlg('q3 fuera de rango (136.5mm,355.5mm)', 'q3')
    return
end

%-----%
% 3. HALLAMOS LOS DELTAS DE CADA JUNTA
%-----%
% 3.1. PARA LA JUNTA 1 - MOTOR 1
q1_delta = q1 - handles.q1_old;      % Esta en sexag
if(q1_delta>=0)
    %q1_delta = q1_delta;
    q1_dir = 1;
else
    q1_delta = -q1_delta;
    q1_dir = 0;
end
% 3.2. PARA LA JUNTA 2 - MOTOR 2
q2_delta = q2 - handles.q2_old;      % En mm
if(q2_delta>=0)
    %J2_delta = q2_delta;
    q2_dir = 1;
else
    q2_delta = -q2_delta;
    q2_dir = 0;
end
% 3.3. PARA LA JUNTA 3 - MOTOR 3
q3_delta = q3 - handles.q3_old;
if(q3_delta>=0)
    %q3_delta = q3_delta;
    q3_dir = 1;
else
    q3_delta = -q3_delta;
    q3_dir = 0;
end
fprintf('q1_delta(sexag)=%2.4f, q1_dir:%d\n', q1_delta, q1_dir)

```



```

fprintf('q2_delta(mm)=%2.4f, q2_dir:%d\n', q2_delta, q2_dir)
fprintf('q3_delta(mm)=%2.4f, q3_dir:%d\n', q3_delta, q3_dir)

%-----%
% 4. APLICAMOS LA TRANSMISION PARA HALLAR EL GIRO
%     QUE DEBE REALIZAR LOS MOTORES
%-----%
% FACTOR DE TRANSMISION DE LAS JUNTAS
k1 = 4.5;
k2 = 45.7;
k3 = 9;
% DELTA EN EL EJE DEL ROTOR
delta_motor1 = k1*q1_delta;
delta_motor2 = k2*q2_delta;
delta_motor3 = k3*q3_delta;

%-----%
% 5. HALLAMOS LOS PASOS Y DIRECCION PARA CADA MOTOR
%-----%
% 5.1. MOTOR 1
N1 = 200;    % Pasos por vuelta de mi motor
m1_pasos = N1*delta_motor1/360;
m1_pasos = round(m1_pasos); % Convertimos a entero
m1_dir = q1_dir;
fprintf('m1_pasos:%d, m1_dir:%d\n', m1_pasos,m1_dir)
% 5.2. MOTOR 2
N2 = 200;    % Pasos por vuelta de mi motor
m2_pasos = N2*delta_motor2/360;
m2_pasos = round(m2_pasos); % Convertimos a entero
m2_dir = q2_dir;
fprintf('m2_pasos:%d, m2_dir:%d\n', m2_pasos,m2_dir)
% 5.3. MOTOR 3
N3 = 200;    % Pasos por vuelta de mi motor
m3_pasos = N3*delta_motor3/360;
m3_pasos = round(m3_pasos); % Convertimos a entero
m3_dir = q3_dir;
fprintf('m3_pasos:%d, m3_dir:%d\n', m3_pasos,m3_dir)

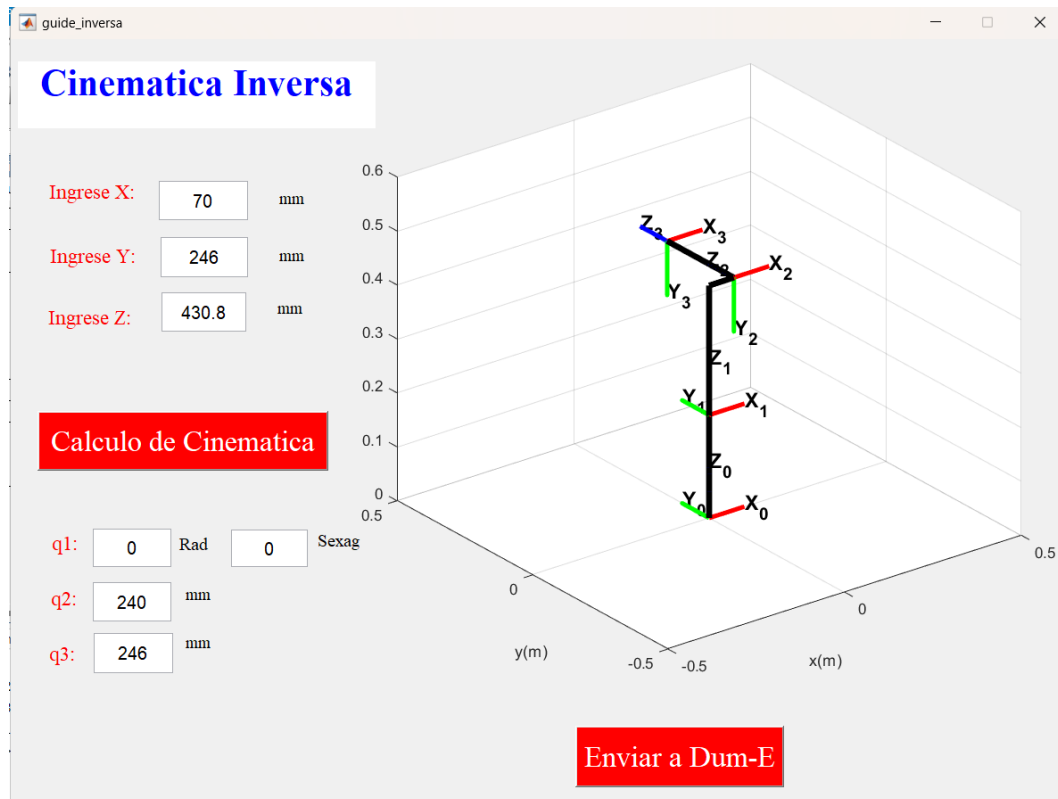
%-----%
% 6. ENVIAMOS LOS ANGULOS AL ARDUINO
%-----%
% LO CONVERTIMOS EN UNA CADENA
cad = strcat(num2str(m1_pasos), ',', num2str(m1_dir), ',',
num2str(m2_pasos), ',', num2str(m2_dir), ',', num2str(m3_pasos), ',',
num2str(m3_dir));
fprintf('cadena a enviar: %s\n', cad)
% ENVIAMOS AL ARDUINO
fprintf(handles.puerto, '%s', cad);

%-----%
% ACTUALIZAMOS VARIABLES
%-----%
handles.q1_old = q1;
handles.q2_old = q2;
handles.q3_old = q3;

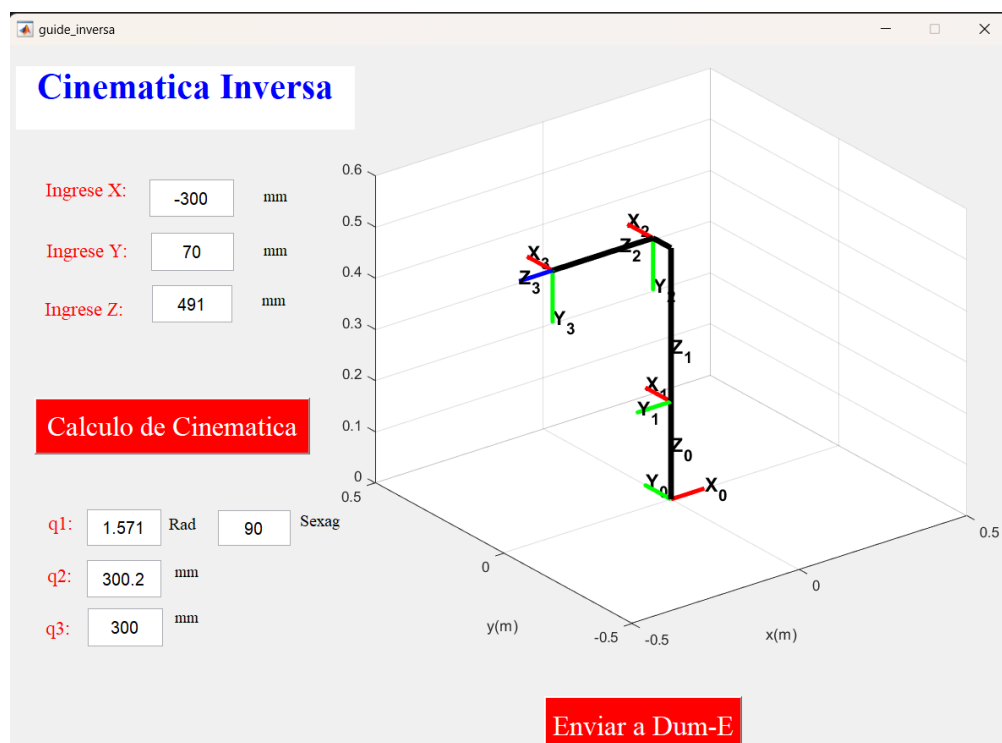
```

Simulación:

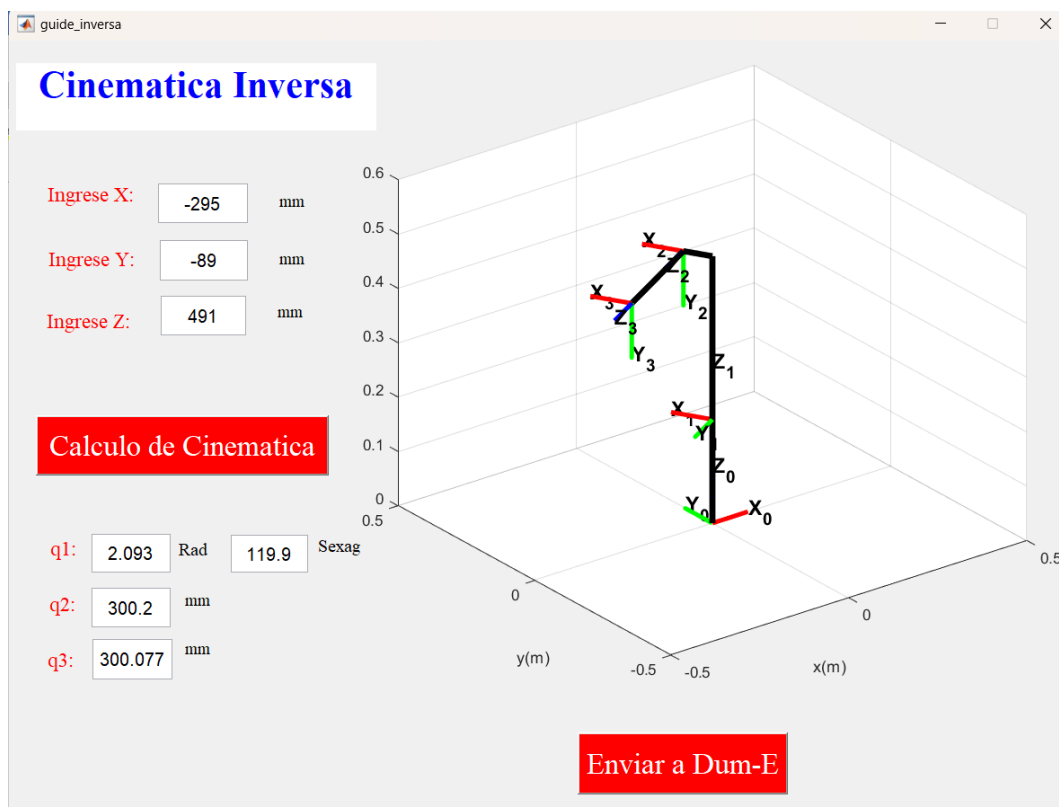
Valores iniciales del robot (home position):



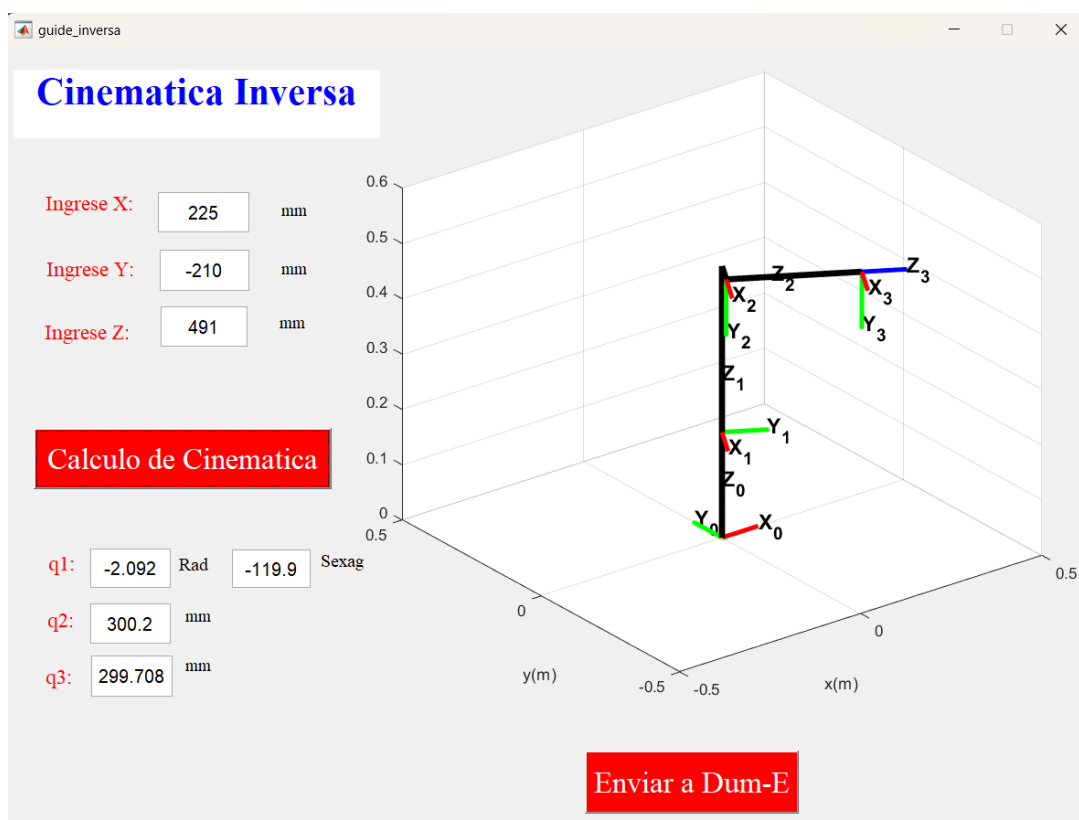
Posición: $x = -300$ mm, $y = 70$ mm, $z = 491$ mm



Posición: $x = -295$ mm, $y = -89$ mm, $z = 491$ mm



Posición: $x = -295$ mm, $y = -89$ mm, $z = 491$ mm

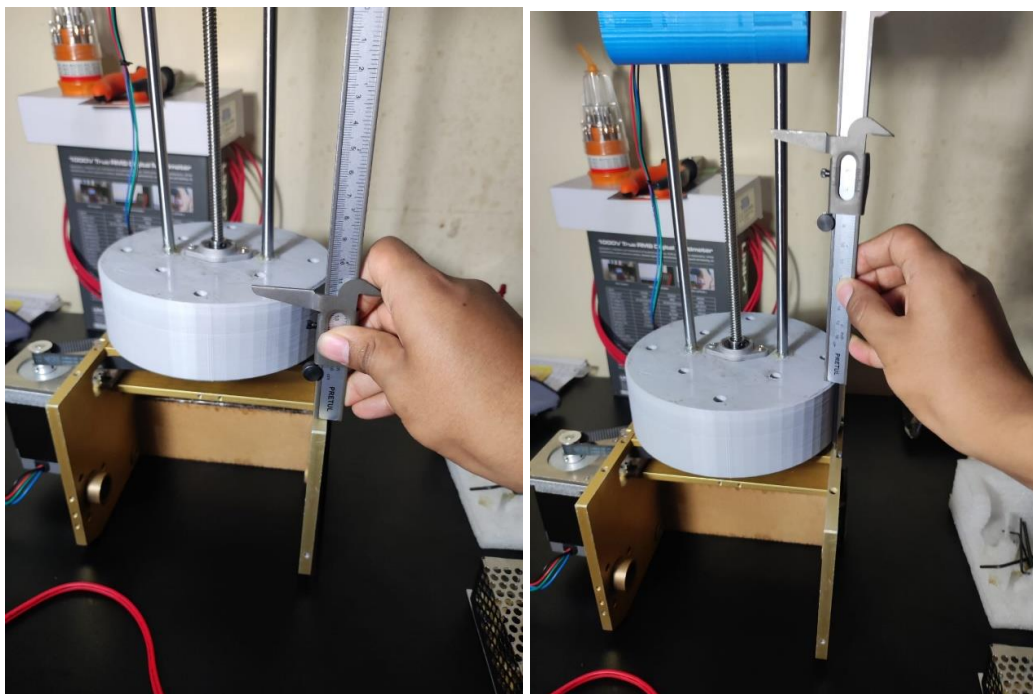


[illegible]

- B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. Robotics, Modelling, Planning and Control. Springer-Verlag, 2009.
- B. Siciliano, and O. Khatib. Springer Handbook of Robotics, 2nd Edition. Springer-Verlag, 2016.
- Código de la cinemática directa para el robot cilíndrico extraído del Aula virtual el 5 de diciembre del 2022.
- M. Spong, S. Hutchinson, and M. Vidyasagar. Robot Modeling and Control, 2nd Edition. Jhon Wiley & Sons, 2020.
- R. Murray, Z. Li, and S. Shankar. A Mathematical Introduction to Robotic Manipulation. CRC Press 1994.

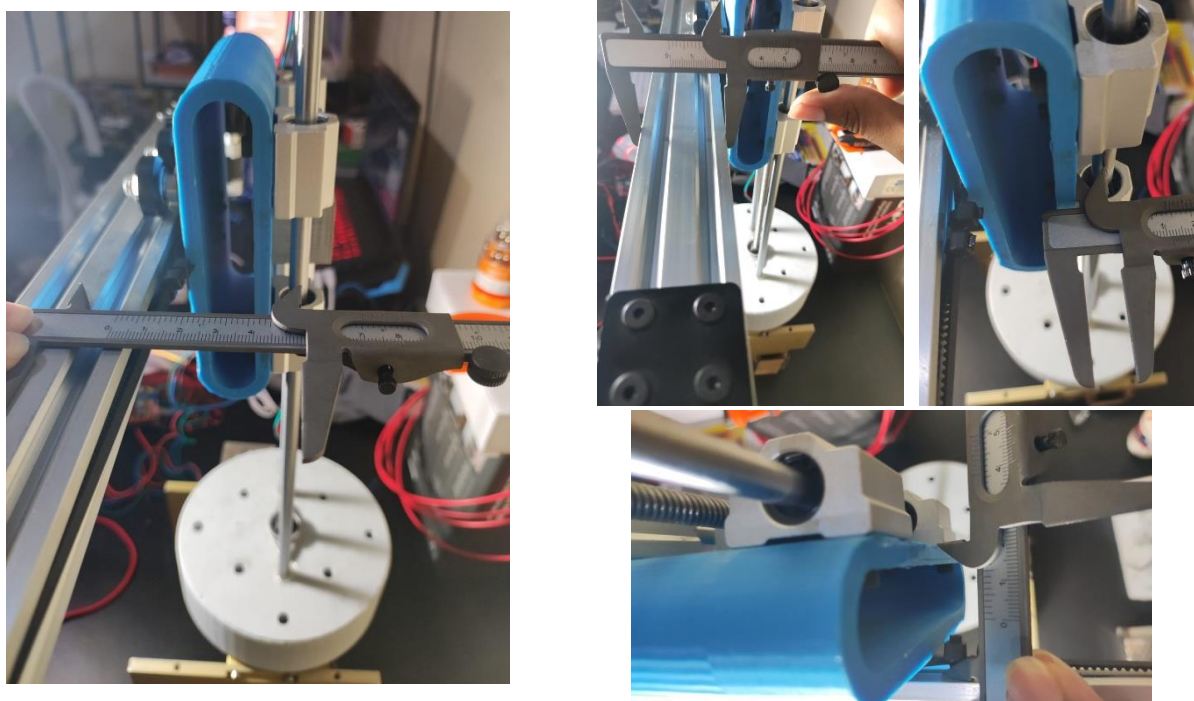
10.1. Mediciones de los parámetros constantes

Medición de parámetro d_1 :



$$d_1 = 133 + 57.8 = 190.8 \text{ mm}$$

Medición de parámetro L :



$$L = 40/2 + 39 + 11 = 70 \text{ mm}$$

10.2. Rango de movimiento de los mecanismos (Restricciones Reales)

Junta de revolución – Junta 1

Es evidente que la junta de revolución se puede identificar con la variable θ_1 , a la cual tiene un rango de 240° para prevenir enredos de los cables. Entonces, se puede decir que:

$$-120^\circ < \theta_1 < 120^\circ$$



Límite inferior

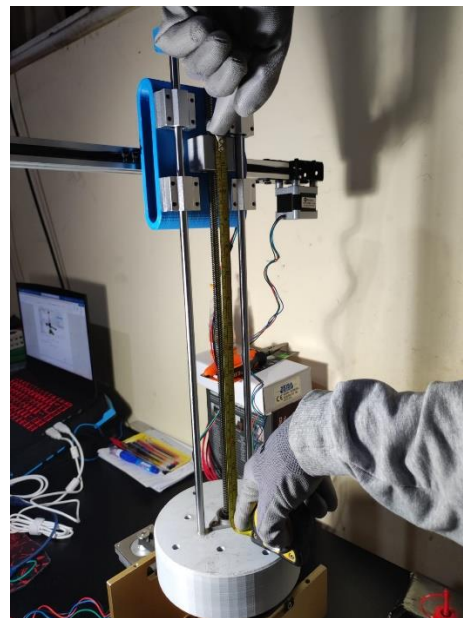


Límite superior

Junta prismática – Junta 2



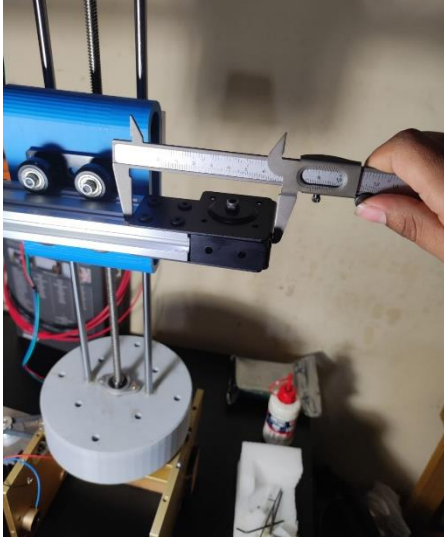
Límite inferior



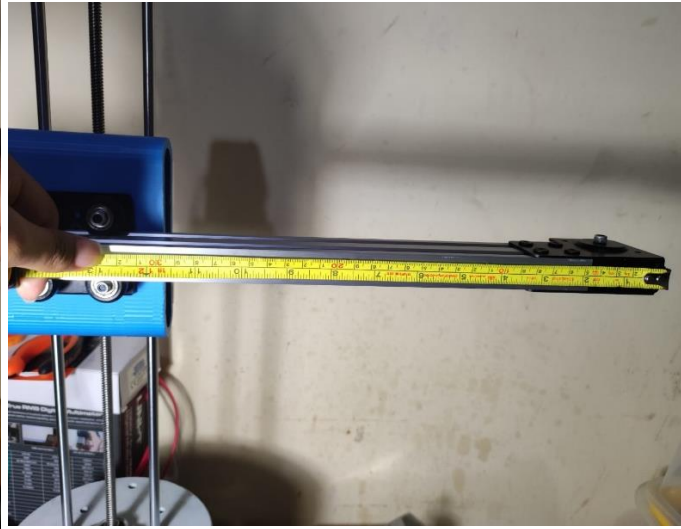
Límite superior

$$107 \text{ mm } (122 - 15) < d_2 < 380 \text{ mm } (395 - 15)$$

Junta prismática – Junta 3



Límite inferior



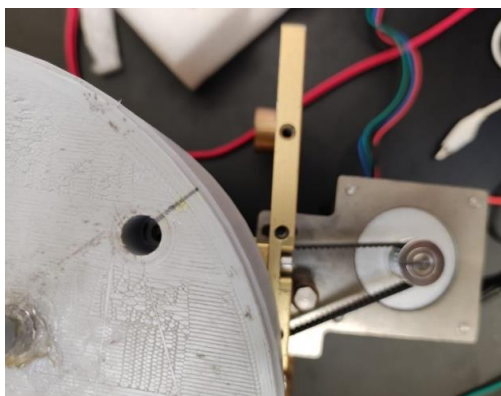
Límite superior

$$136.5 \text{ mm } (50.5 + 86) < d_3 < 355.5 \text{ mm } (50.5 + 305)$$

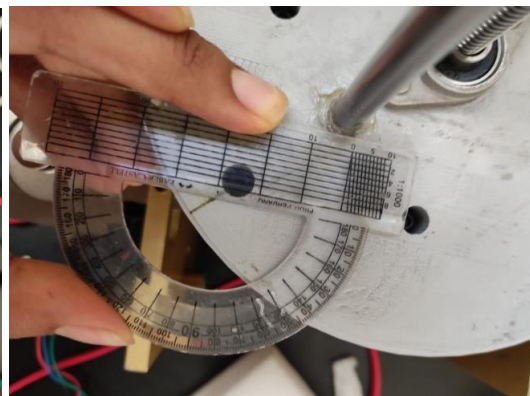
10.3. Cálculo de Factores de Transmisión de las juntas

Junta de revolución – Junta 1 (q_1)

Se tomo múltiples medidas del desplazamiento angular de la variable θ_1 que sucede al enviar cierto número de pasos al motor que controla dichas juntas.



Posición inicial



Posición final

Instrucciones enviadas al motor y respuesta:

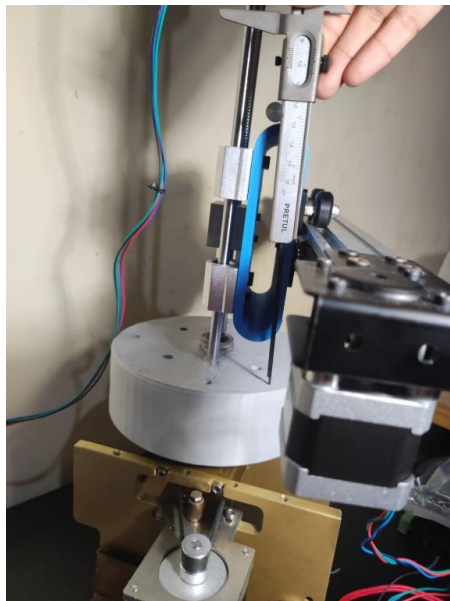
if dir == 1
Giro antihorario

if dir == 0
Giro horario

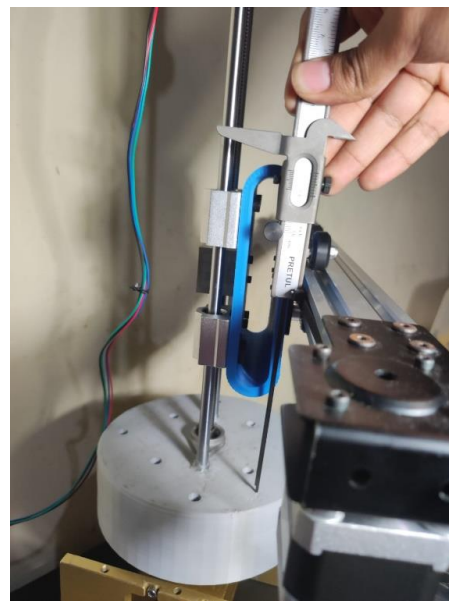
Para el movimiento giratorio: junta q1			
Pasos	Δ Giro Motor($^{\circ}$)	Δ Giro Base($^{\circ}$)	K1 ($^{\circ}/^{\circ}$)
200	360	75	4.8000
-	420	90	4.6667
400	720	150	4.8000
100	180	40	4.5000
200	360	78	4.6154
50	90	20	4.5000
300	540	120	4.5000
900	1620	360	4.5000
-	400	90	4.4444
Promedio			4.5100

Junta de revolución – Junta 2 (q_2)

Se tomo múltiples medidas del desplazamiento lineal de la variable d_2 que sucede al enviar cierto número de pasos al motor que controla dicha junta.



Posición inicial



Posición final

Instrucciones enviadas al motor y respuesta:

if dir == 1

Sube

if dir == 0

Baja

Para el movimiento vertical: junta q2			
Pasos	Dirección	Posición Inicial (mm)	Posición Final(mm)
1000	Subida	92.8	129.4
1000	Bajada	129.4	90.2
1000	Subida	90.2	130.9
1000	Bajada	130.9	93
1000	Subida	93	123
1000	Bajada	123	82.5
2000	Subida	82.5	163.5
2000	Bajada	163.5	82
1000	Subida	87	126.7
500	Bajada	126.7	107
1500	Subida	107	166

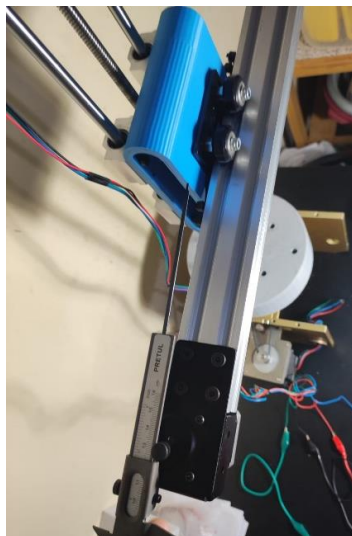
Pruebas más estables			
1000	Subida	105	144.5
1000	Bajada	144.5	104.8
1000	Subida	88	125.4
1000	Bajada	125.4	85.6
2000	Subida	85.6	165
1500	Bajada	165	107.5
1000	Subida	107.5	140
1000	Bajada	140	100
1000	Subida	100	138.3
1000	Bajada	138.3	98.7
1000	Subida	98.7	137.5

Equivalencias Δ Giro Motor-Desplazamiento		
Δ Giro Motor(°)	Δ Desplazamiento	K2 (°/mm)
1800	36.6	49.1803
1800	39.2	45.9184
1800	40.7	44.2260
1800	37.9	47.4934

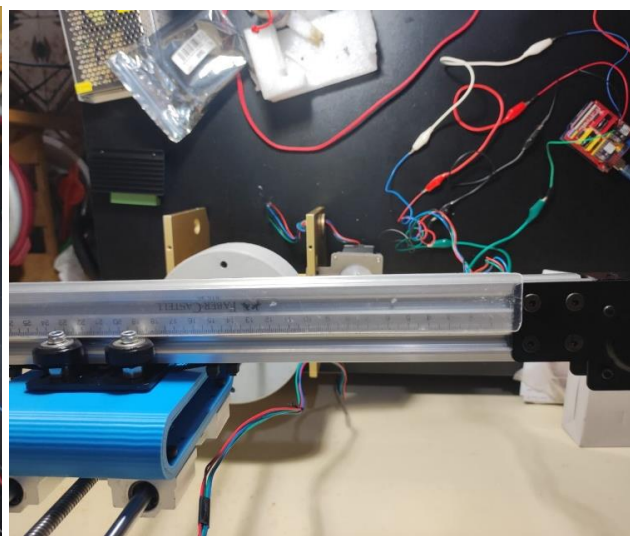
1800	40.5	44.4444
3600	81	44.4444
3600	81.5	44.1718
1800	39.7	45.3401
900	19.7	45.6853
2700	59	45.7627
1800	39.5	45.5696
1800	39.5	45.5696
1800	37.4	48.1283
1800	39.8	45.2261
3600	79.4	45.3401
2700	57.5	46.9565
1800	40	45.0000
1800	38.3	46.9974
1800	39.6	45.4545
1800	38.8	46.3918
Promedio		45.6906

Junta de revolución – Junta 3 (q_3)

Se tomo múltiples medidas del desplazamiento lineal de la variable d_3 que sucede al enviar cierto número de pasos al motor que controla dicha junta.



Posición inicial



Posición final

Instrucciones enviadas al motor y respuesta:

if dir == 1
 q_3 crece (d_3 avanza)

if dir == 0
 q_3 decrece (d_3 retrocede)

Para el movimiento horizontal: junta q3			
Pasos	Δ Giro Motor($^{\circ}$)	Δ Desplazamiento	K3 ($^{\circ}/\text{mm}$)
500	900	98.3	9.1556
500	900	98.5	9.1371
800	1440	160	9.0000
600	1080	119	9.0756
400	720	80	9.0000
700	1260	139	9.0647
200	360	40	9.0000
Promedio			9.0619