

El algoritmo de Denavit-Hartenberg es un procedimiento que describe la estructura cinemática de una cadena articulada, que ayuda a establecer un sistema de referencia relativo para cada eslabón.

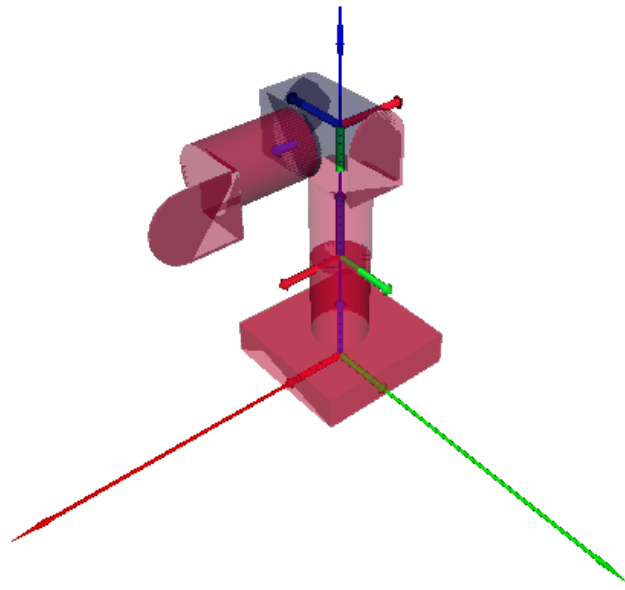


Figura 1. Robot LWA 4D a 3R

Paso 0.

Primeramente, se determinó el número de eslabones con la relación de las articulaciones del ensamble, en este caso 3 eslabones. Si bien guarda la relación $n = \text{número de articulaciones}$ y $n+1 = \text{número de eslabones}$; por la fijación de la base, sin considerarla un eslabone se tienen, 3 articulaciones y 3 eslabones.

Posteriormente se enumeran los eslabones considerando a la base como eslabón 0, así hasta el eslabón final (efector final).

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

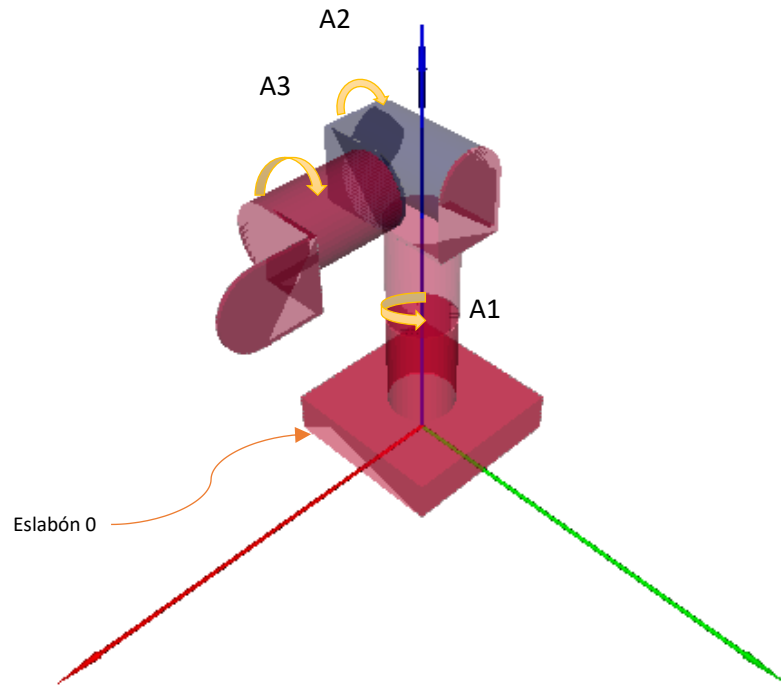


Figura 2. Enumeración y tipo de movimiento (revoluta los 3 casos)

Caso especial en la orientación del eslabón 0

El sistema del eslabón cero $\{\hat{i}_0, \hat{j}_0, \hat{k}_0\}$ se alinea con el sistema global $\{\hat{i}, \hat{j}, \hat{k}\}$, esto debido a que al ser la base ésta permanecerá inmóvil ante los demás eslabones al igual que el sistema global de referencia.

Paso 1.

En este paso se establece que el sistema de referencias que adapte cada eslabón con respecto al siguiente eslabón debe cumplir con el sistema de la mano derecha. Lo que es lo mismo a $\hat{i} \times \hat{j} = \hat{k}$, $\hat{j} \times \hat{k} = \hat{i}$ & $\hat{k} \times \hat{i} = \hat{j}$; correspondientes una rotación positiva en el sistema en dichas direcciones.

Para determinar la referencia del sistema se toma como primera relación la base del robot (alineada en sus ejes con el sistema global).

El método de Denavit-Hartenberg dice que la orientación inicial de los ejes se dará acomodando la orientación en los ejes " \hat{k} " para cada eslabón, considerando el tipo de movimiento que efectuará el eslabón siguiente en la cadena.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

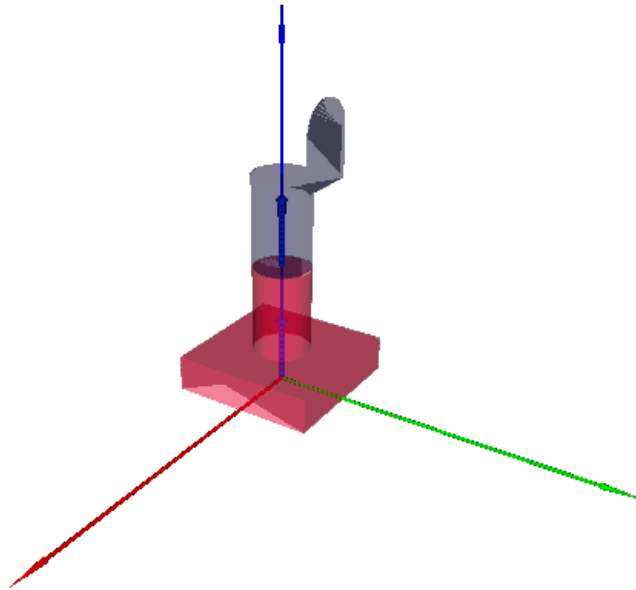


Figura 3. Representación del eje \hat{k}_0 colineal con el eje \hat{k} del eslabón 1.

El eje \hat{k}_0 actuara al ser colineal con el eje \hat{k} del eslabón 1 hasta el eslabón 2 tomando en consideración una rotación en el eje.

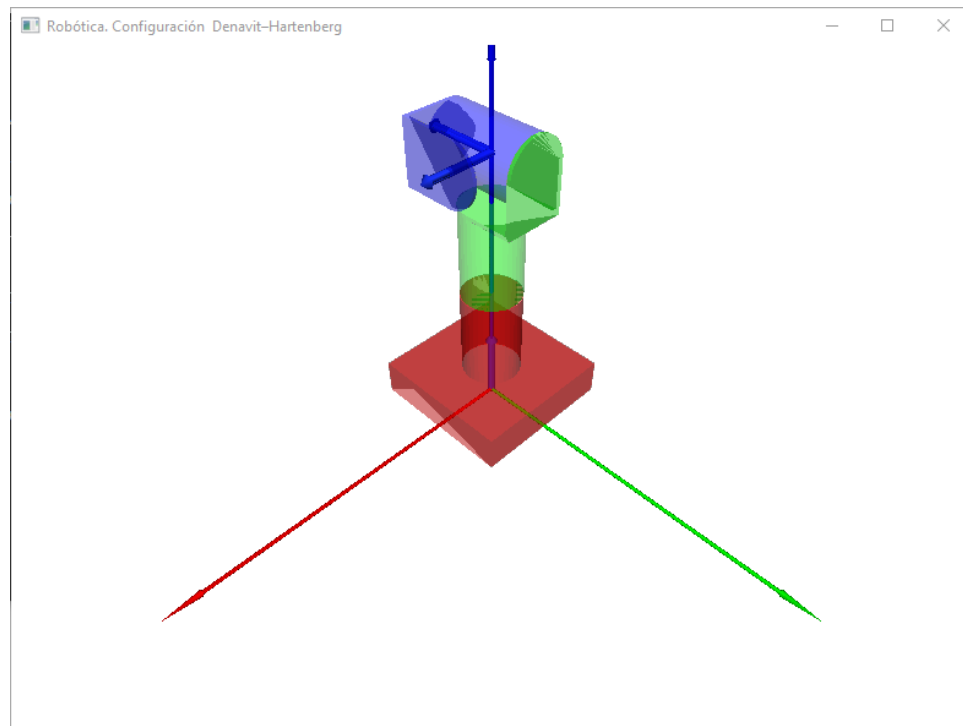


Figura 4. Representación del eje \hat{k}_1 referenciado al movimiento para el eslabón 3.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

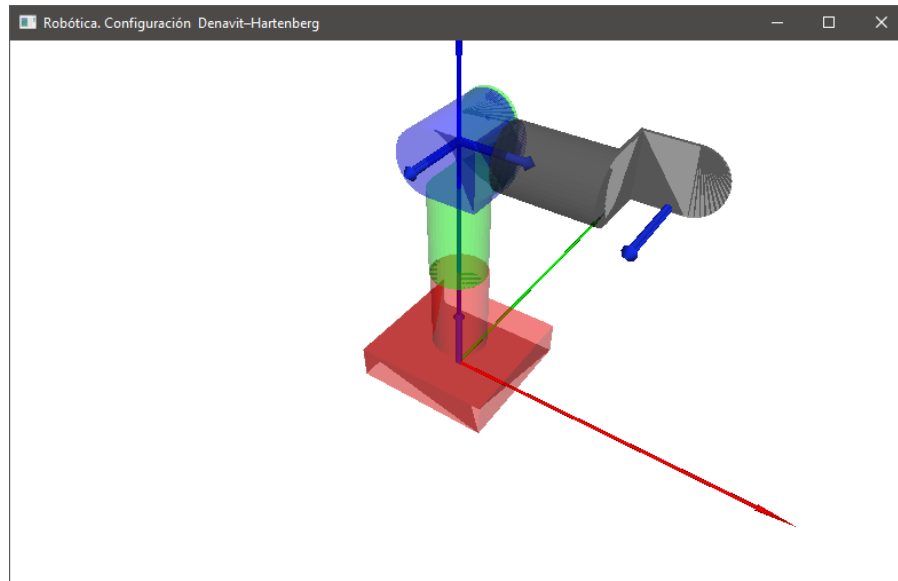


Figura 5. Representación del eje \hat{k}_2 referenciado para el movimiento del siguiente eslabón.

Este procedimiento es necesario a seguir para todos los eslabones del robot los cuales ocupen un reajuste en su orientación.

Paso 2.

En este paso se asignan los ejes \hat{l}_n para cada articulación.

Caso 1

En este caso se hace referencia a la intersección de dos o más ejes \hat{k} , donde ocurre dicho caso se debe contemplar que en ese punto donde se encuentran los ejes será el origen de la base de la articulación siguiente en la cadena.

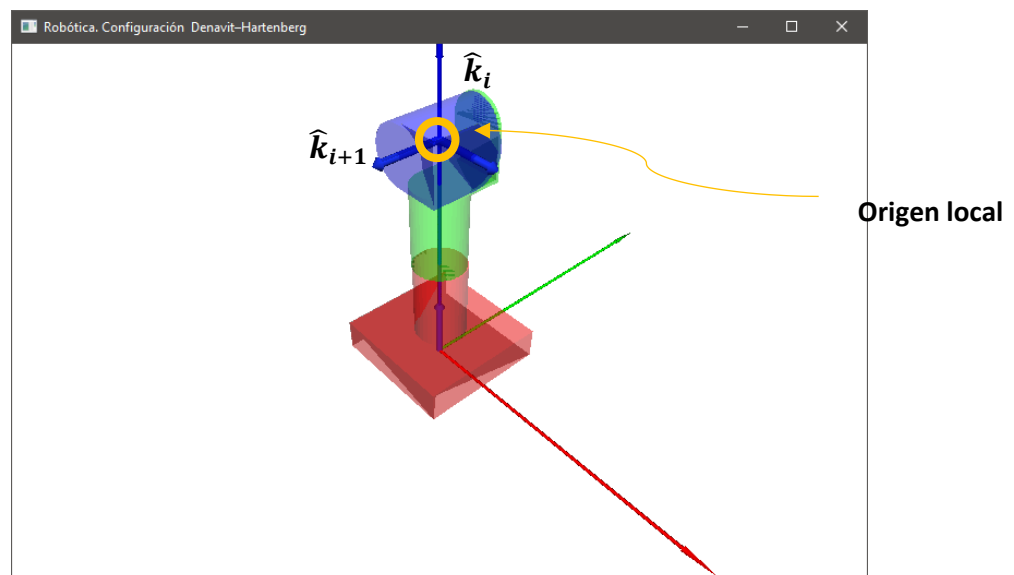


Figura 6. Intersección de ejes \hat{k} . Creación de origen local para la articulación.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

Para encontrar el eje \hat{l} se toma en consideración $\hat{k}_{i+1} \times \hat{k}_i = \hat{l}_{i+1}$. Así para todos los ejes que cumplan con dicha condición.

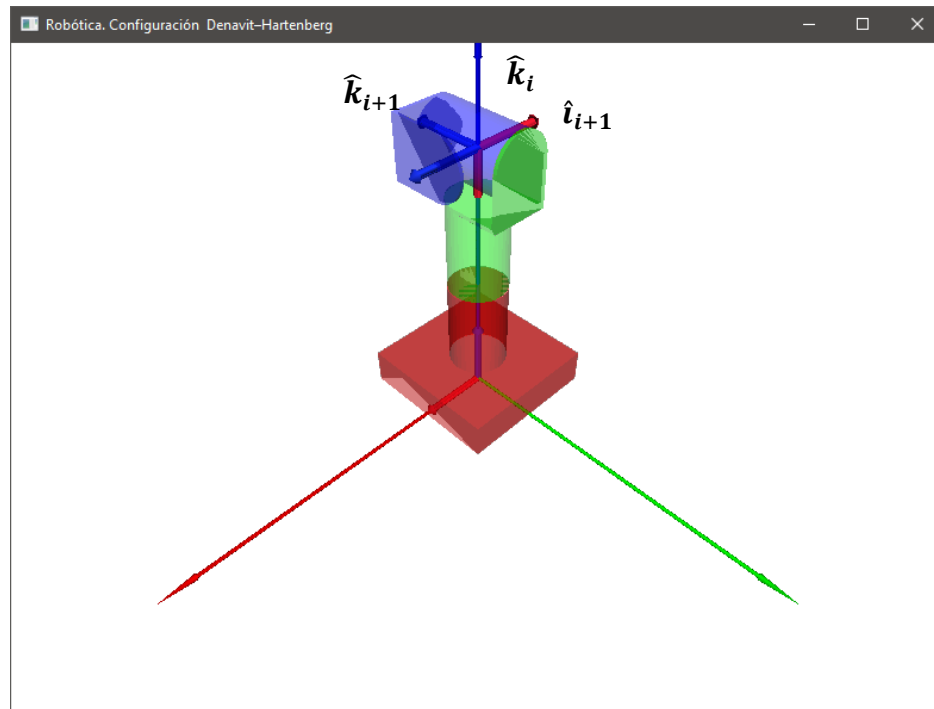


Figura 7. Resultante $\hat{k}_{i+1} \times \hat{k}_i = \hat{l}_{i+1}$.

Caso 2.

Si los ejes \hat{k}_{i+1} y \hat{k}_i son paralelos, hay un número infinito de normales comunes, se puede escoger cualquiera, la dirección del eje "x i" está dada por la dirección de la normal común que se eligió. Este robot no cuenta con dicho caso.

Caso 3.

Si los ejes y no son paralelos ni se intersectan, la dirección del eje está dada por la dirección de la normal común entre dichos ejes. Este caso tampoco es considerado en este estudio puesto que el robot no presenta ninguna relación parecida.

Paso 3.

Este consiste en realizar la regla de la mano derecha para terminar la referenciación de los sistemas \hat{l} , \hat{j} y \hat{k} .

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

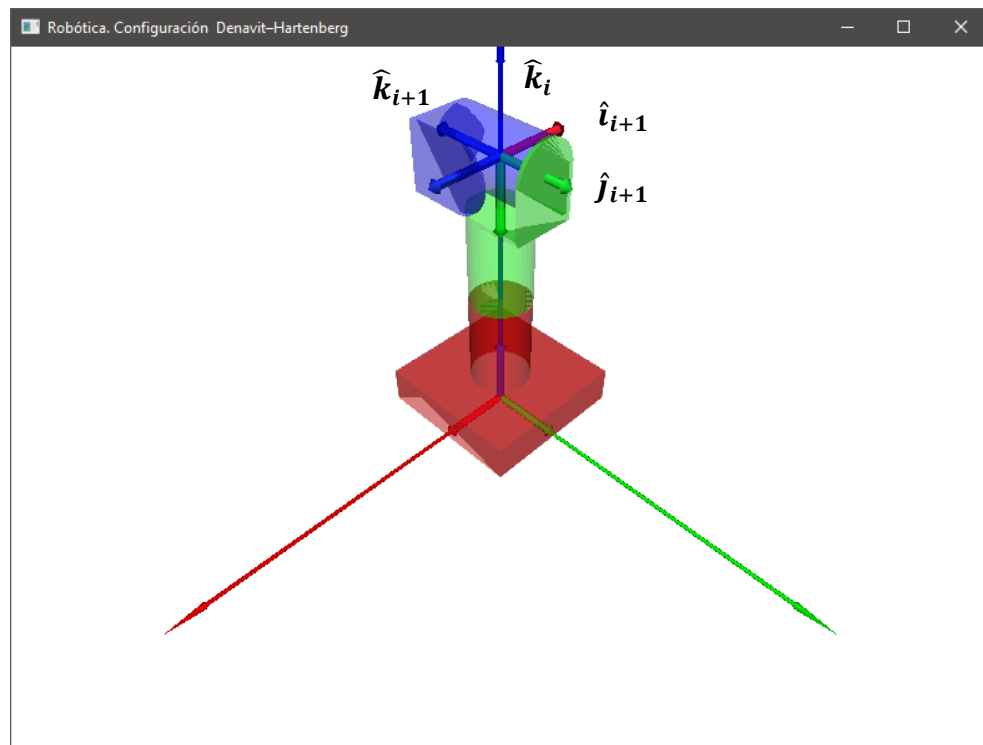


Figura 8. Representación $\{\hat{i}_{i+1}, \hat{j}_{i+1}, \hat{k}_{i+1}\}$.

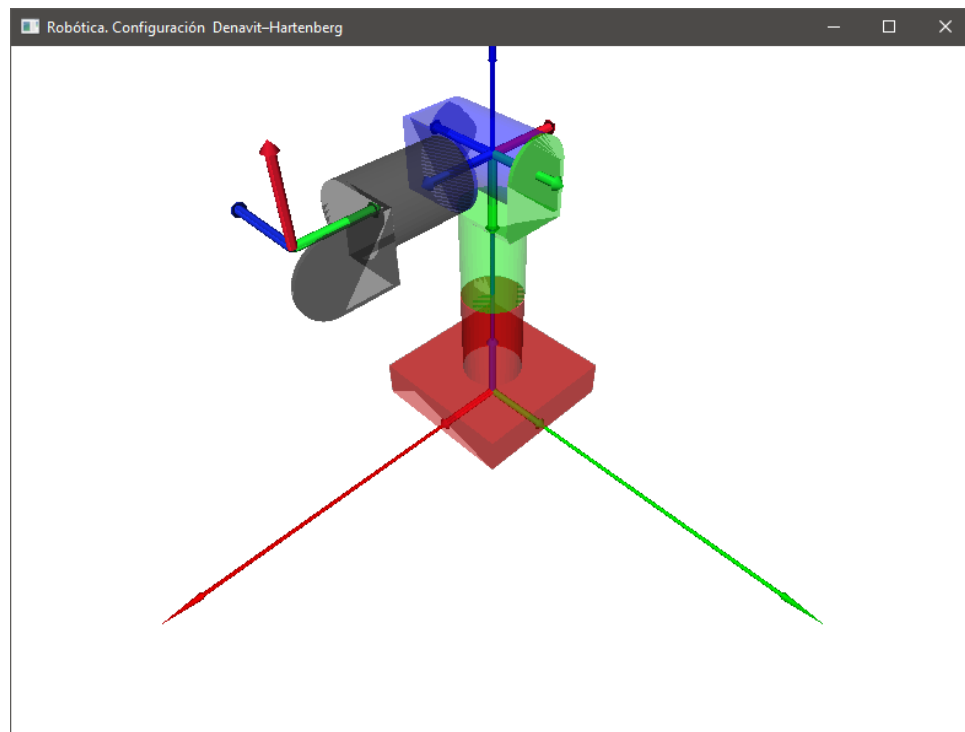


Figura 9. Ensamble 3R completo.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

En este caso de estudio se toman las piezas para su ensamble en formato .stl, dichos archivos deben poseer adecuadamente el sistema de referencia de la pieza para poder realizar adecuadamente las rotaciones.

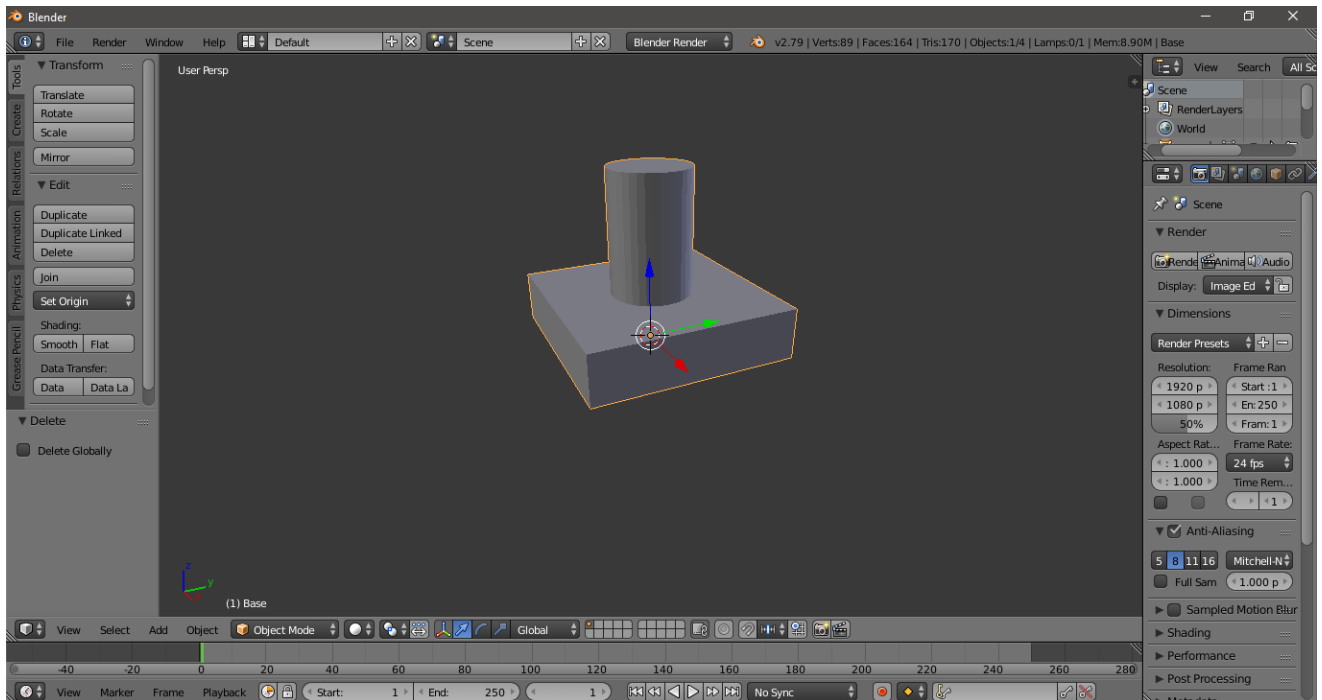


Figura 10. Pieza Base.

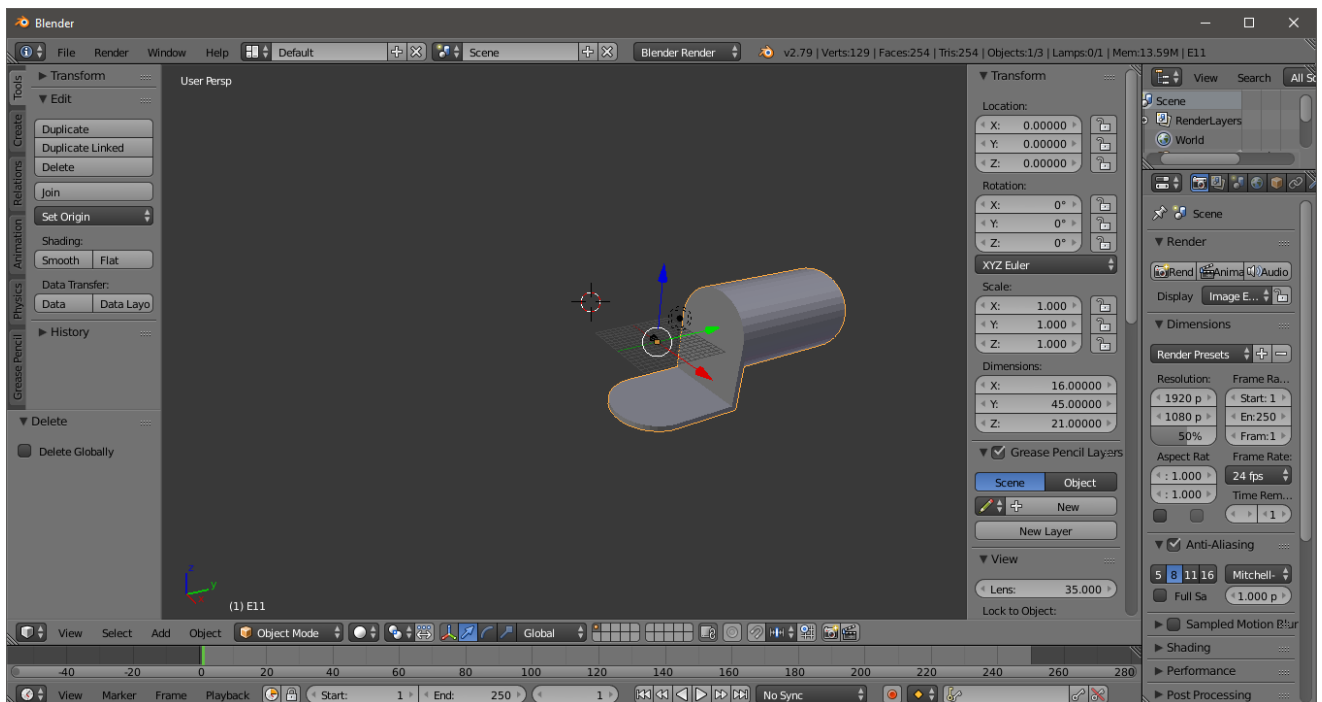


Figura 11. Eslabón 1.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

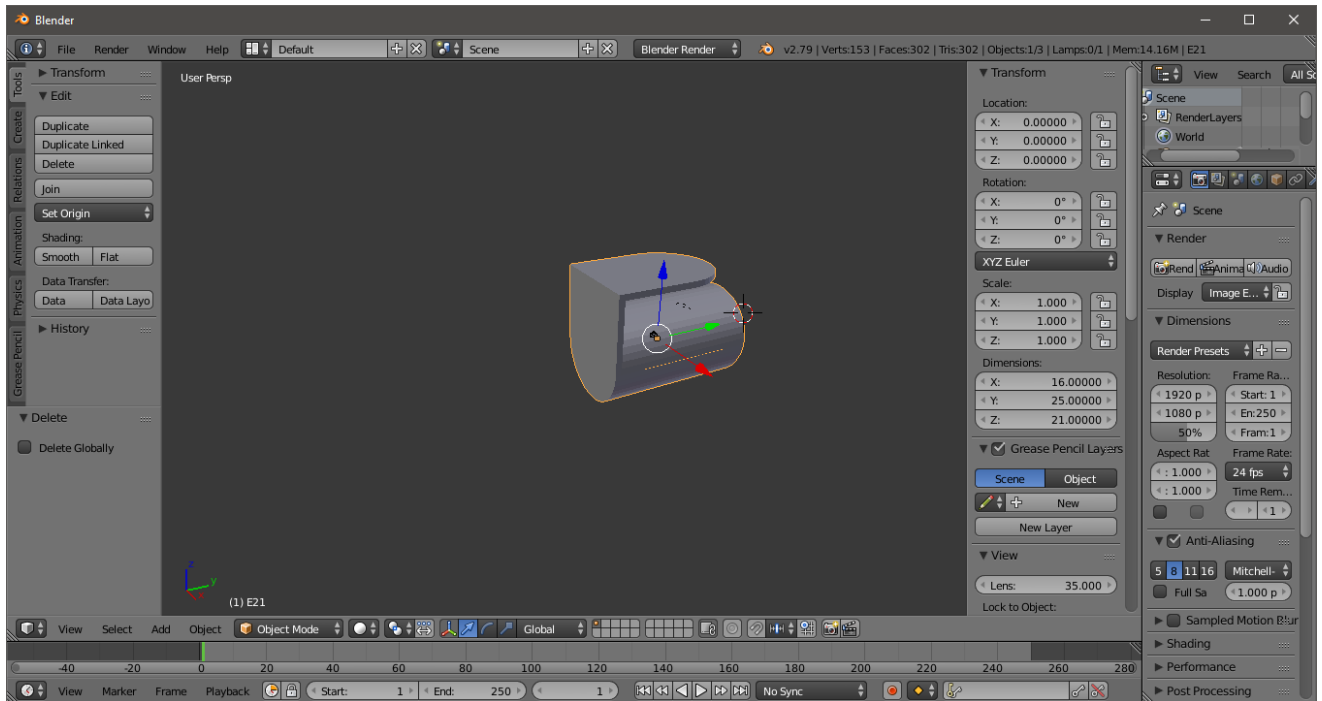


Figura 12. Eslabón 2.

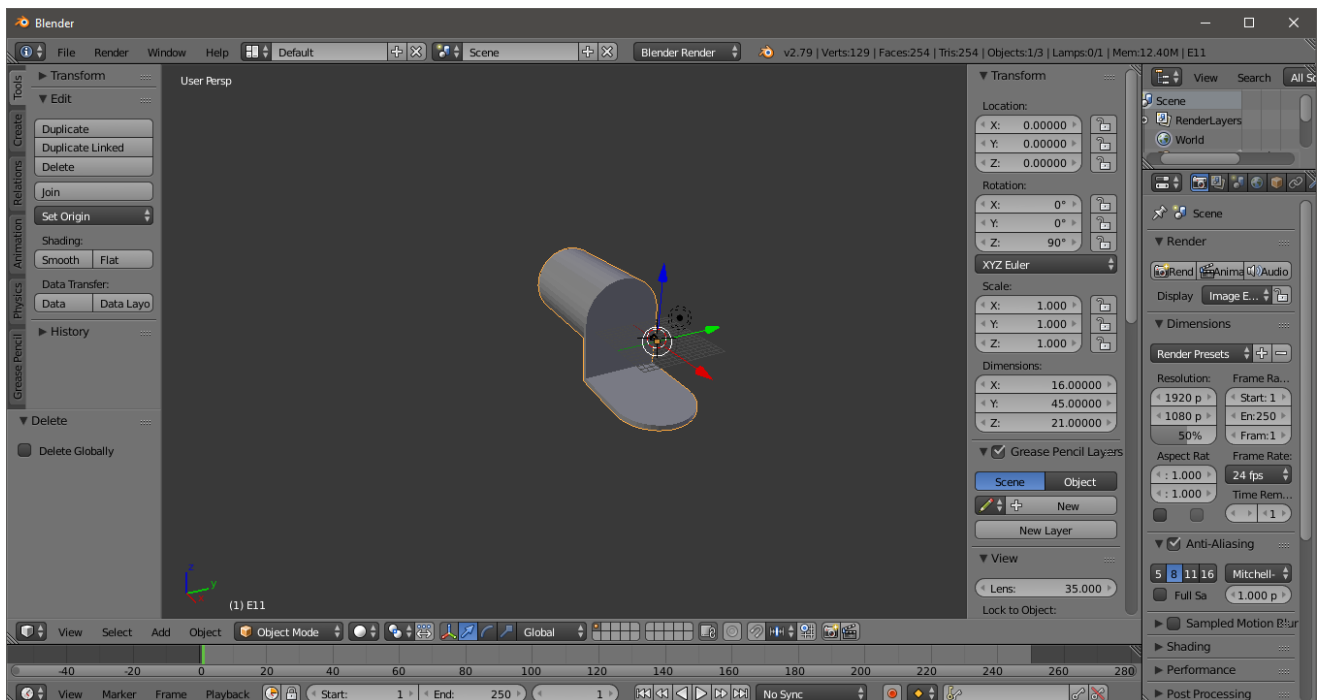


Figura 13. Eslabón 3.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

Visualización de los eslabones del robot a partir del sistema local alineado al sistema del global se muestran en la figura 13.

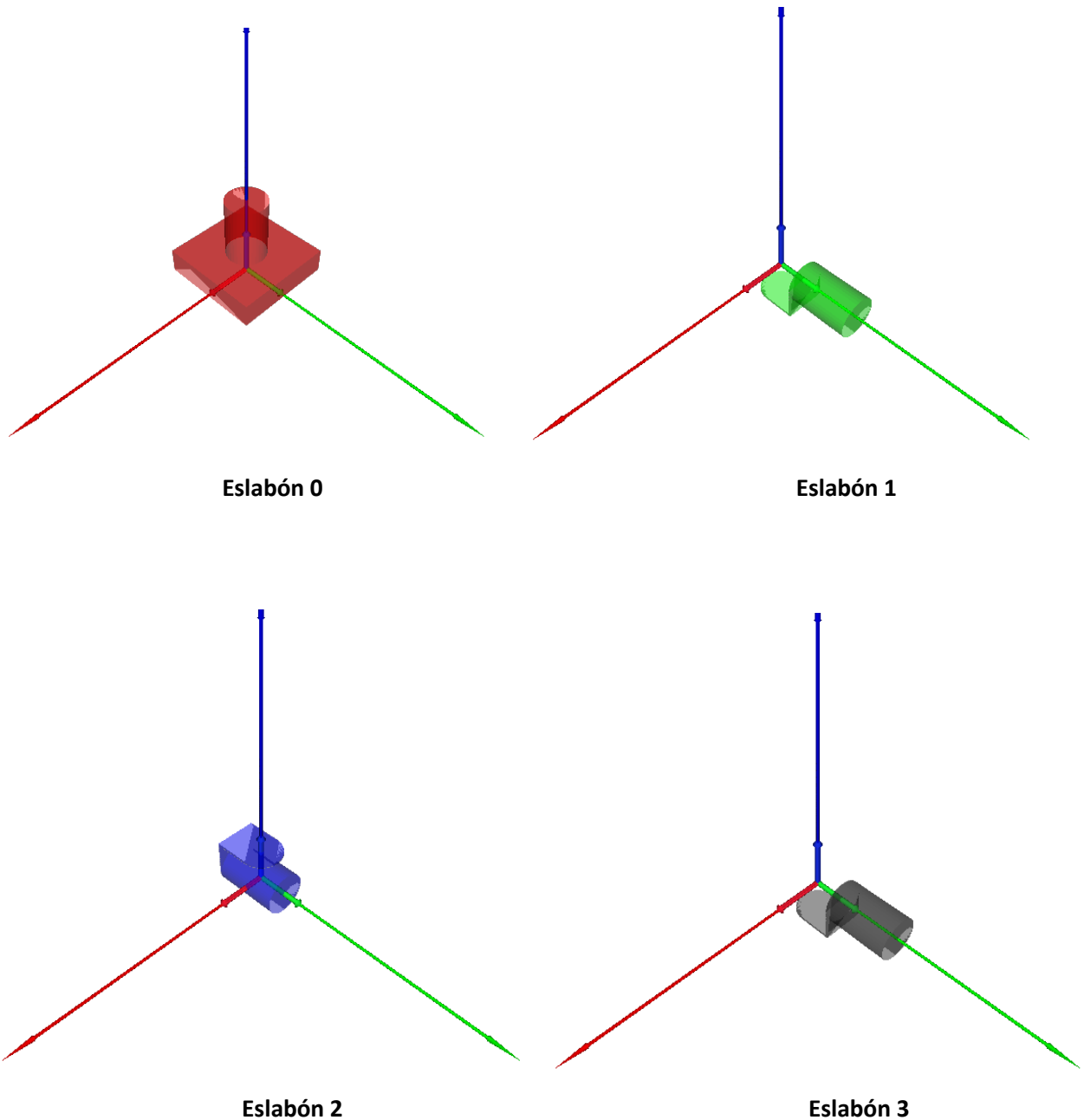


Figura 14. Eslabones orientados según sus sistemas locales respecto al sistema global.

Para el adecuado montaje de los eslabones se deben considerar sus rotaciones, esta como producto de dos matrices. Que son las rotaciones en z y x.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

2. DESCRIPCIÓN DE LAS TRANSFORMACIONES

Durante la colocación de los ejes para cada eslabón existieron configuraciones que harán que el ensamble no encaje adecuadamente entre eslabones. Para esto el algoritmo propone realizar la multiplicación de dos matrices que representan las rotaciones de las piezas, que según con los criterios antes mencionados son necesarias y suficientes para adecuar la posición de los eslabones de manera correcta.

Para realizar las rotaciones correctas es necesario establecer las dimensiones de cada eslabón del ensamble, así como las traslaciones entre sistemas de referencia. Para ello se recomienda tener una referencia visual del ensamble final y un croquis correspondiente a el mismo.

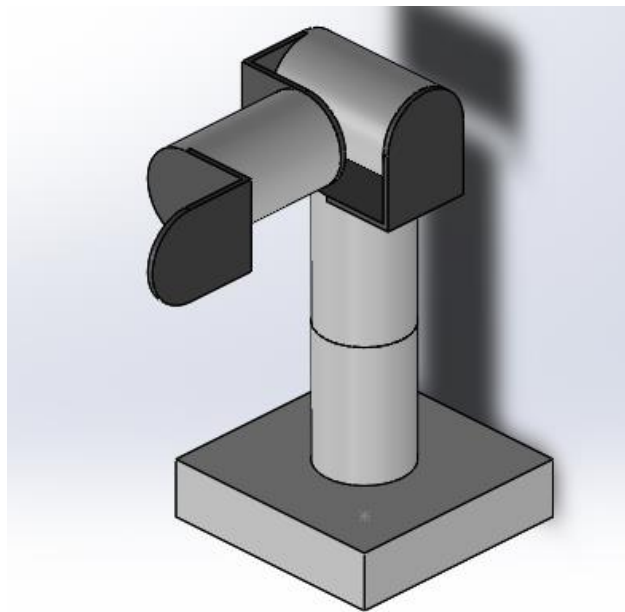


Figura 15. Ensamble realizado en SolidWorks.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

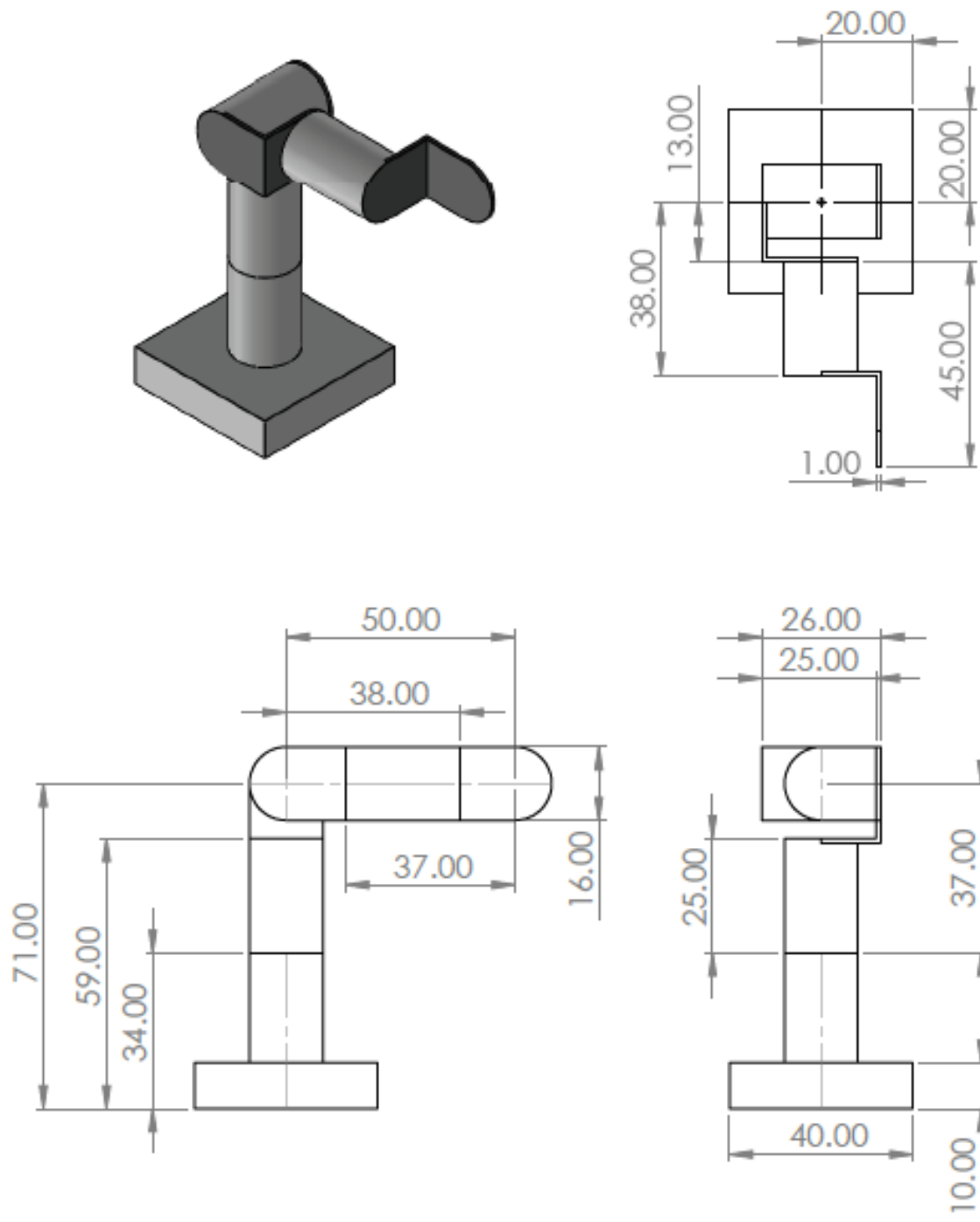


Figura 16. Croquis del ensamble.

Las transformaciones se definen para cada eslabón montado.

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

Configuración HOME

Articulación	Z_i [mm]	X_i [mm]	α_i [rad]	θ_i [rad]
0	0	0	0	0
1	71	0	$-\pi/2$	π
2	0	0	$-\pi/2$	$\pi/2$
3	50	0	$-\pi/2$	π

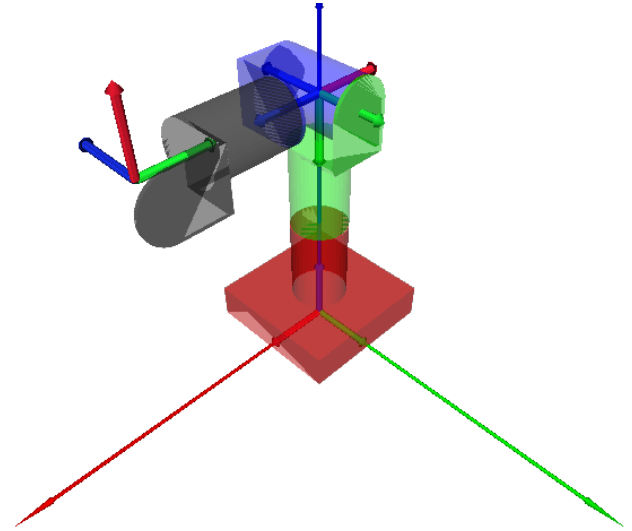


Figura 17. Ensamble realizado con archivos .stl

3. CINEMÁTICA DIRECTA

Las transformaciones para posicionar el robot en home se utilizan entre otras cosas para poder definir la cinemática directa y posteriormente la inversa del robot, así que es preciso definir con denavit las transformadas correspondientes, la cuales en este caso de estudio quedan como:

$$\begin{aligned}
 {}^{S_0}_{S_1}T &= T(z_1, \theta_1) \cdot T(x_1, \alpha_1) = \begin{pmatrix} \cos(180) & -\sin(180) & 0 & 0 \\ \sin(180) & \cos(180) & 0 & 0 \\ 0 & 0 & 1 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90) & -\sin(-90) & 0 \\ 0 & \sin(-90) & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

$$\begin{aligned}
 {}^{S_1}_{S_2}T &= T(z_2, \theta_2) \cdot T(x_2, \alpha_2) = \begin{pmatrix} \cos(90) & -\sin(90) & 0 & 0 \\ \sin(90) & \cos(90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90) & -\sin(-90) & 0 \\ 0 & \sin(-90) & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

$$\begin{aligned} {}^{S_2}_{S_3}T &= T(z_3, \theta_3) \cdot T(x_3, \alpha_3) = \begin{pmatrix} \cos(180) & -\sin(180) & 0 & 0 \\ \sin(180) & \cos(180) & 0 & 0 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90) & -\sin(-90) & 0 \\ 0 & \sin(-90) & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

$${}^{S_0}_{S_2}T = {}^{S_0}_{S_1}T * {}^{S_1}_{S_2}T = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{S_0}_{S_3}T = {}^{S_0}_{S_2}T * {}^{S_2}_{S_3}T = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 & 50 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 71 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Implementación de las rotaciones en el código:

```
void Robot::configurarTH() {
    x1= 0; z1 = 71; theta1=radian(180); alpha1=radian(-90);
    x2= 0; z2 = 0 ; theta2=radian(90) ; alpha2=radian(-90);
    x3= 0; z3 = 50; theta3=radian(180); alpha3=radian(-90);
    x4= 0; z4 = 0 ; theta4=radian(0) ; alpha4=radian(0);
    x5= 0; z5 = 0 ; theta5=radian(0) ; alpha5=radian(0);
    x6= 0; z6 = 0 ; theta6=radian(0) ; alpha6=radian(0);
    x7= 0; z7 = 0 ; theta7=radian(0) ; alpha7=radian(0);
    // Base
    DefinirTHz(0, vector3d(0,0,0));
    THList.push_back(THz);
    DefinirTHx(0, vector3d(0,0,0));
    THList.push_back(THx);
    //El1
    DefinirTHz(theta1, vector3d(0,0,z1));
    THList.push_back(THz);
    DefinirTHx(alpha1, vector3d(x1,0,0));
    THList.push_back(THx);
}
```

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

```
//E21
DefinirTHz(theta2,vector3d(0,0,z2));
THList.push_back(THz);
DefinirTHx(alpha2,vector3d(x2,0,0));
THList.push_back(THx);
//E31
DefinirTHz(theta3,vector3d(0,0,z3));
THList.push_back(THz);
DefinirTHx(alpha3,vector3d(x3,0,0));
THList.push_back(THx);

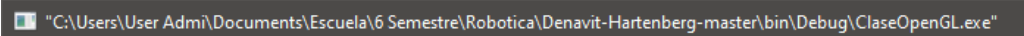
/// Multiplicación de rotaciones
Matrix T(4,4);
T.identity(4);
cout << "La transformacion T01 es: " << endl;
T = T*THList[2]*THList[3];
T.mostrar();

cout << "La transformacion T02 es: " << endl;
T = T*THList[4]*THList[5];
T.mostrar();

cout << "La transformacion T03 es: " << endl;
T = T*THList[6]*THList[7];
T.mostrar();

}
```

Resultados de la implementación en código.



```
"C:\Users\User Admi\Documents\Escuela\6 Semestre\Robotica\Denavit-Hartenberg-master\bin\Debug\ClaseOpenGL.exe"
La transformacion T01 es:

-1 -3.82137e-015 8.74228e-008 0
-8.74228e-008 4.37114e-008 -1 0
0 -1 -4.37114e-008 71
0 0 0 1

La transformacion T02 es:

4.37114e-008 -1.31134e-007 1 0
4.37114e-008 1 1.31134e-007 0
-1 4.37114e-008 4.37114e-008 71
0 0 0 1

La transformacion T03 es:

-4.37114e-008 -1 8.74228e-008 50
-1.31134e-007 -8.74228e-008 -1 6.55671e-006
1 -4.37114e-008 -1.31134e-007 71
0 0 0 1
```

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

4. COORDENADAS GENERALIZADAS DE UN ROBOT

Para saber las coordenadas generalizadas del robot para n articulaciones, en este caso con n=3;

Por cuestiones de procesamiento en C++ de valores evaluados en senos y cosenos para π ; se debe realizar un cambio de variable en las matrices generadas, al final acudiendo a una matriz descriptiva de todo el sistema la cual corresponderá a 3_0T , esto se logra cambiando $N*\pi$, para N = número real.

La manera para conseguir dicho cambio se consigue agregando un desfaseamiento a Θ_i de q_i . Puesto que los términos de α_i se utilizaron para ensamblar el robot no son requeridos para la orientación de los eslabones, por ende, se pueden agregar como constantes a la multiplicación de las matrices.

Las matrices desarrolladas en Scientific WorkPlace acudiendo a variables simbólicas, omitiendo desplazamientos en x_i , puesto que son igual a cero; quedan como:

$${}^1_0T = \begin{bmatrix} \cos(\pi + q_1) & -\sin(\pi + q_1) & 0 & 0 & 1 & 0 & 0 & 0 \\ \sin(\pi + q_1) & \cos(\pi + q_1) & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & z_1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\cos q_1 & 0 & \sin q_1 & 0 \\ -\sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & -1 & 0 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_1T = \begin{bmatrix} \cos((\pi/2) + q_2) & -\sin((\pi/2) + q_2) & 0 & 0 & 1 & 0 & 0 & 0 \\ \sin((\pi/2) + q_2) & \cos((\pi/2) + q_2) & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & z_2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\frac{1}{2}\pi + q_2) & 0 & -\sin(\frac{1}{2}\pi + q_2) & 0 \\ \sin(\frac{1}{2}\pi + q_2) & 0 & \cos(\frac{1}{2}\pi + q_2) & 0 \\ 0 & -1 & 0 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} :$$

$$= \begin{bmatrix} -\sin q_2 & 0 & -\cos q_2 & 0 \\ \cos q_2 & 0 & -\sin q_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_2T = \begin{bmatrix} \cos(\pi + q_3) & -\sin(\pi + q_3) & 0 & 0 & 1 & 0 & 0 & 0 \\ \sin(\pi + q_3) & \cos(\pi + q_3) & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & z_3 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\cos q_3 & 0 & \sin q_3 & 0 \\ -\sin q_3 & 0 & -\cos q_3 & 0 \\ 0 & -1 & 0 & z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Caso de estudio del robot LWA 4D

por Jesús Ismael Pineda Hernández

$${}^3_0T = \begin{bmatrix} -\cos q_1 & 0 & \sin q_1 & 0 \\ -\sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & -1 & 0 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\sin q_2 & 0 & -\cos q_2 & 0 \\ \cos q_2 & 0 & -\sin q_2 & 0 \\ 0 & -1 & 0 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\cos q_3 & 0 & \sin q_3 & 0 \\ -\sin q_3 & 0 & -\cos q_3 & 0 \\ 0 & -1 & 0 & z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \sin q_1 \sin q_3 - \cos q_1 \cos q_3 \sin q_2 & -\cos q_1 \cos q_2 & \cos q_3 \sin q_1 + \cos q_1 \sin q_2 \sin q_3 & 50 \cos q_1 \cos q_2 \\ -\cos q_1 \sin q_3 - \cos q_3 \sin q_1 \sin q_2 & -\cos q_2 \sin q_1 & \sin q_1 \sin q_2 \sin q_3 - \cos q_1 \cos q_3 & 50 \cos q_2 \sin q_1 \\ \cos q_2 \cos q_3 & -\sin q_2 & -\cos q_2 \sin q_3 & 50 \sin q_2 + 71 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Con los q_i se consigue que el sistema empiece en $q_i = 0$, para todo $i = 1, 2, 3, \dots, n$; eslabones según sea el caso del robot.

Por último, se toma de referencia a 3_0T , para encontrar la posición de un punto P4 donde posteriormente se ensamblará el gripper del robot.

Las coordenadas del punto 4 en base a b3 son:

$$P_4 = \begin{bmatrix} 0 \\ 0 \\ z_4 \\ 1 \end{bmatrix}$$

Con $z_4 = 0$; se tiene el vector:

$$r_{O_4} = \begin{bmatrix} \sin q_1 \sin q_3 - \cos q_1 \cos q_3 \sin q_2 & -\cos q_1 \cos q_2 & \cos q_3 \sin q_1 + \cos q_1 \sin q_2 \sin q_3 & 50 \cos q_1 \cos q_2 & 0 \\ -\cos q_1 \sin q_3 - \cos q_3 \sin q_1 \sin q_2 & -\cos q_2 \sin q_1 & \sin q_1 \sin q_2 \sin q_3 - \cos q_1 \cos q_3 & 50 \cos q_2 \sin q_1 & 0 \\ \cos q_2 \cos q_3 & -\sin q_2 & -\cos q_2 \sin q_3 & 50 \sin q_2 + 71 & z_4 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 50 \cos q_1 \cos q_2 \\ 50 \cos q_2 \sin q_1 \\ 50 \sin q_2 + 71 \\ 1 \end{bmatrix}$$

5. CINEMÁTICA INVERSA

En la cinemática directa, como se puede ver en la implementación del código, orienta los eslabones del robot conociendo sus ángulos θ_1 , θ_2 y θ_3 ; o bien sus desfases, como lo son q_1 , q_2 y q_3 .

Si nosotros no conociéramos los ángulos correspondientes, pero si las coordenadas del punto r_{o_4} , se podría asumir por medio de la sustitución y despejes de variables la equivalencia para cualquier ángulo de las articulaciones.

Aludiendo al desarrollo anterior se puede observar que x_1 , x_2 y x_3 , así como, z_1 , z_2 y z_3 , ya han sido sustituidas por sus constantes correspondientes en las matrices. En caso de no tener asignado un valor numérico a dichas variables se seguirá operando con los valores simbólicos correspondientes. Para el caso del estudio de este robot se seguirán conservando los valores de x_1 , x_2 y x_3 , iguales a cero, mientras que a z_1 , z_2 y z_3 , se les considera como apoyo para observar su generalización.

Código simplificado en Matlab:

$$R_{O4} = \begin{vmatrix} z4*(\cos(q3)*\sin(q1) + \cos(q1)*\sin(q2)*\sin(q3)) + z2*\sin(q1) + z3*\cos(q1)*\cos(q2) & | \\ z3*\cos(q2)*\sin(q1) - z2*\cos(q1) - z4*(\cos(q1)*\cos(q3) - \sin(q1)*\sin(q2)*\sin(q3)) & | \\ z1 + z3*\sin(q2) - z4*\cos(q2)*\sin(q3) & | \\ 1 & | \end{vmatrix}$$

Si se conoce una posición en el punto $R_{O4} =$

$$\begin{pmatrix} x4 \\ y4 \\ z4 \\ 1 \end{pmatrix}$$

Se pueden encontrar los valores correspondientes a q_1 , q_2 y q_3 , para este caso. Esto igualando los valores.

$$\begin{pmatrix} x4 \\ y4 \\ z4 \\ 1 \end{pmatrix} = \begin{vmatrix} 50 \cos q_1 \cos q_2 \\ 50 \cos q_2 \sin q_1 \\ 50 \sin q_2 + 71 \\ 1 \end{vmatrix}$$