



Representación tridimensional de robots



Importación de archivos .stl

La principal característica de un archivo STL ("Standard Tessellation Language") es que describe la superficie de un objeto tridimensional mediante una serie de triángulos. Gracias a esto, es posible importar la matriz de vértices y polígonos (en este caso triángulos) de cualquier sólido en formato .stl para luego dibujarlo usando la función `path()` como se vé en figura 1 (en el caso de MATLAB).



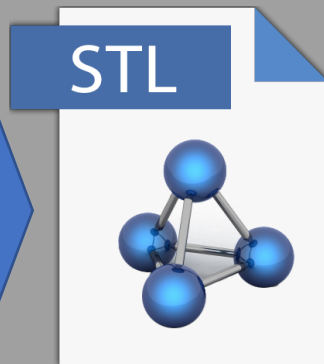
Inventor



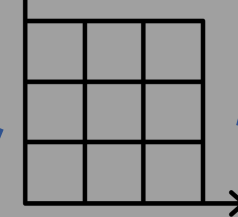
AUTODESK®
AUTOCAD®



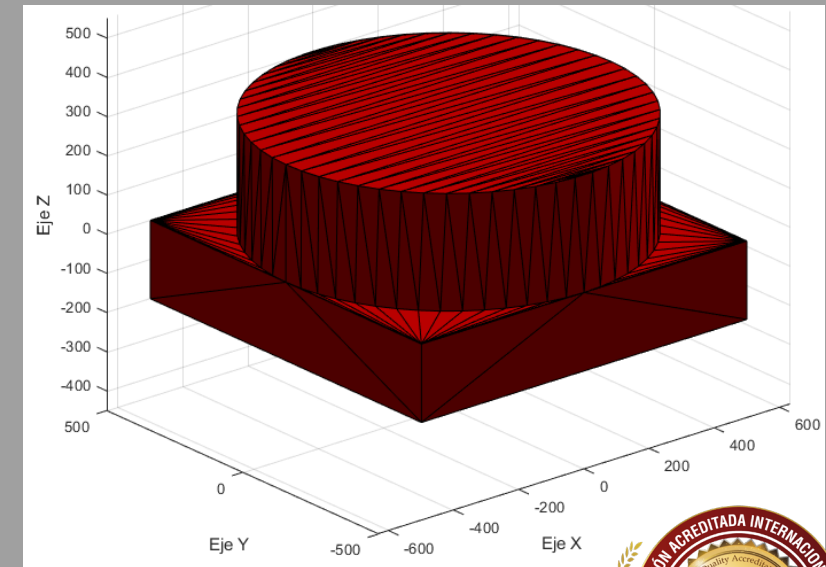
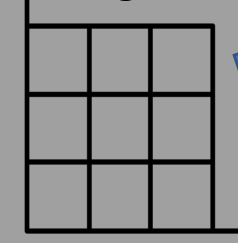
AUTODESK
FUSION 360™



Vértices



Polígonos



Ejemplo de importación .stl

Código 1

```
clear;
% Lee el archivo STL
[V,~,~,~] = stlread('Eslabon_0.stl');
vertices = V.Points;
faces = V.ConnectivityList;
% Crear una figura
figure(1);
clf;
hold on;
grid on;
axis equal; % Configurar los ejes para que tengan la misma escala
light;
view(3);
xlabel('Eje X');
ylabel('Eje Y');
zlabel('Eje Z');
title('Modelo STL en R3');
% Dibujar el modelo
patch('Faces', faces, 'Vertices', vertices, 'FaceColor', 'r', 'EdgeColor', 'k');
```



Ejemplo de importación .stl

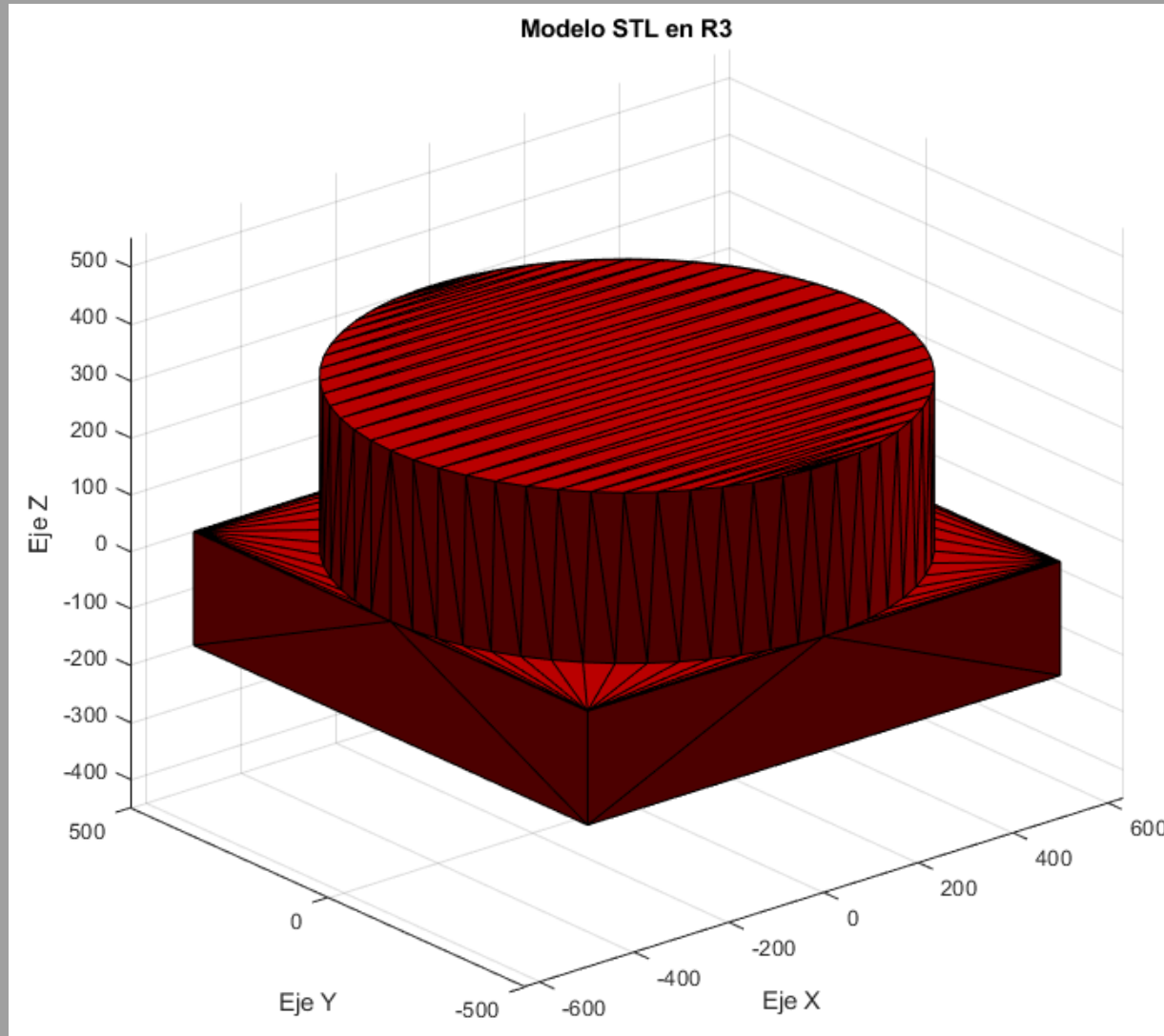


Figura 1. Sólido de archivo .stl dibujado en MATLAB



Ejemplo de dibujar robot 3D

Código 2

```
clear;  
figure(1);  
clf;  
hold on;  
grid on;  
axis equal;  
view(3);  
light;  
dibujarRobot(0,300,250,0)
```

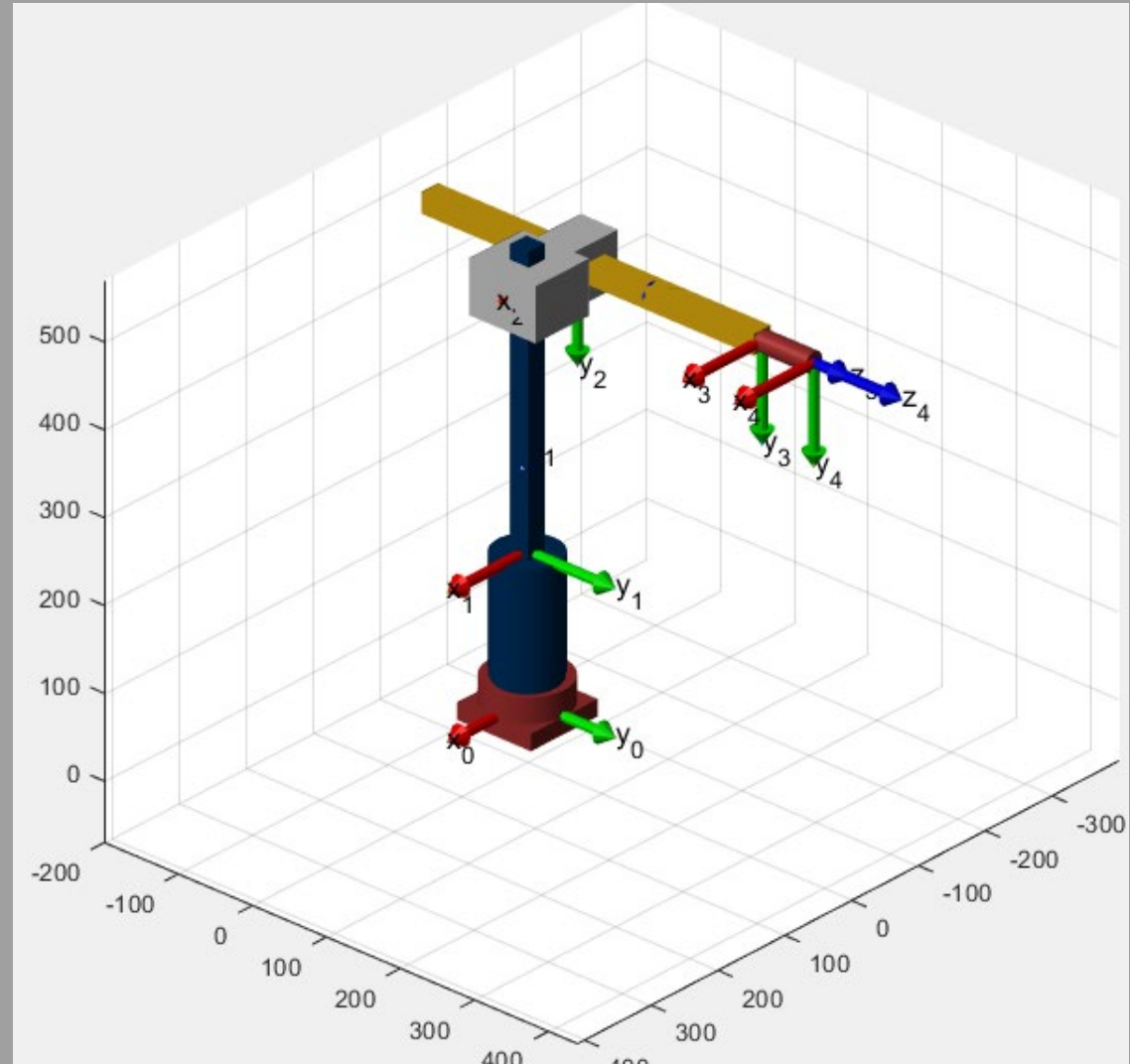


Figura 2. Robot de 4 GDL 3D en MATLAB



Taller

1) Obtener los parámetros Denavit-Hartenberg y las matrices de transformación homogénea correspondientes a cada uno de los ejes coordenados. Recomendación: crear funciones que calculen las matrices homogéneas a partir de los parámetros de cada una de las articulaciones. Crear una función que dibuje los ejes coordenados y represente los eslabones del robot por medio de líneas, teniendo como parámetros de entrada un vector con las coordenadas articulares del robot (Cinemática directa del robot).

2) Realizar una función que calcule la cinemática inversa del robot de la figura 3 y pinte el robot en la posición indicada.



Taller

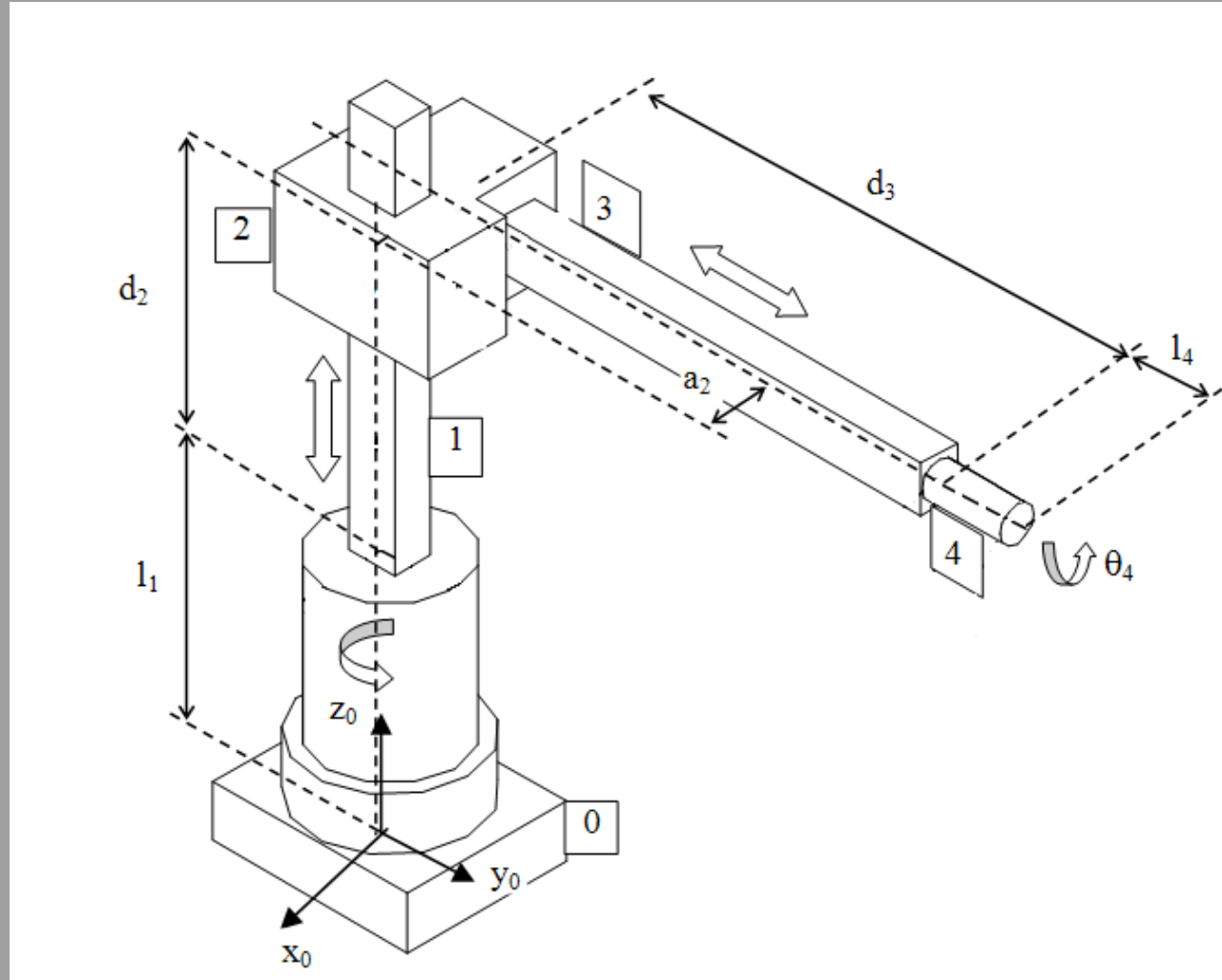


Figura 3. Robot de 4 GDL



MARCO TEORICO – FORMULAS UTILES

Rotaciones:

$$R(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, R(y, \phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}, R(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Par de rotación:

$$Rot(k, \theta)p = p \cos \theta + (k \times p) \sin \theta + k(k \cdot p)(1 - \cos \theta)$$



MARCO TEORICO – FORMULAS UTILES

Algoritmo Denavit – Hartenberg:

- D-H 1.- Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- D-H 2.- Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n
- D-H 3.- Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- D-H 4.- Para i de 0 a n-1 situar el eje z_i sobre el eje de la articulación i+1.
- D-H 5.- Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0



MARCO TEORICO – FORMULAS UTILES

Algoritmo Denavit – Hartenberg:

- D-H 6.- Para i de 1 a $n-1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i+1$
- D-H 7.- Situar x_i en la línea normal común a z_{i-1} y z_i
- D-H 8.- Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .
- D-H 9.- Situar el sistema $\{S_i\}$ en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
- D-H 10.- Obtener q_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
- D-H 11.- Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.



MARCO TEORICO – FORMULAS UTILES

Algoritmo Denavit – Hartenberg:

- DH 12.- Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
- DH 13.- Obtener α_i como el ángulo que habría que girar entorno a x_i (que ahora coincidiría con x_{i-1}), para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
- DH 14.- Obtener las matrices de transformación ${}^{i-1}A_i$
- DH 15.- Obtener la matriz de transformación entre la base y el extremo del robot $T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n$.
- DH 16.- La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares



MARCO TEORICO – FORMULAS UTILES

Algoritmo Denavit – Hartenberg:

$${}^{i-1}A_i = \text{Rotz}(\theta_i)T(0,0,d_i)T(a_i,0,0)\text{Rotx}(\alpha_i)$$

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Matriz D-H})$$





Formando **líderes** para la
construcción de un nuevo
país en paz