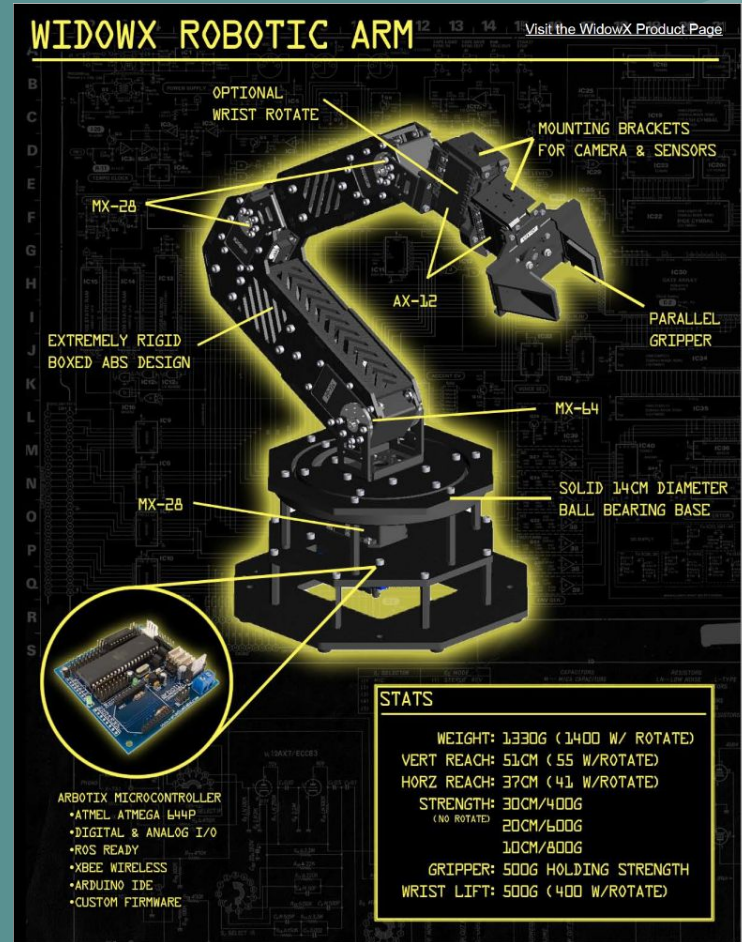


# Automación Industrial - Caracterización y Simulación de Control del TROSSEN – WidowX MK-II

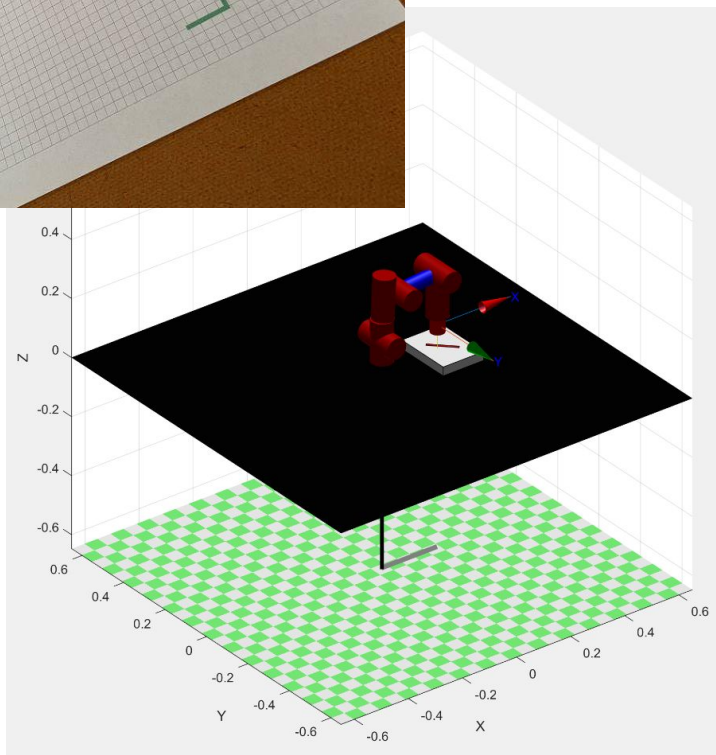
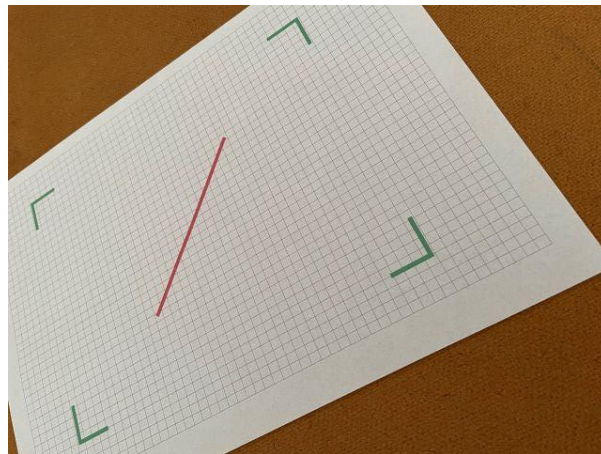
Integrantes:

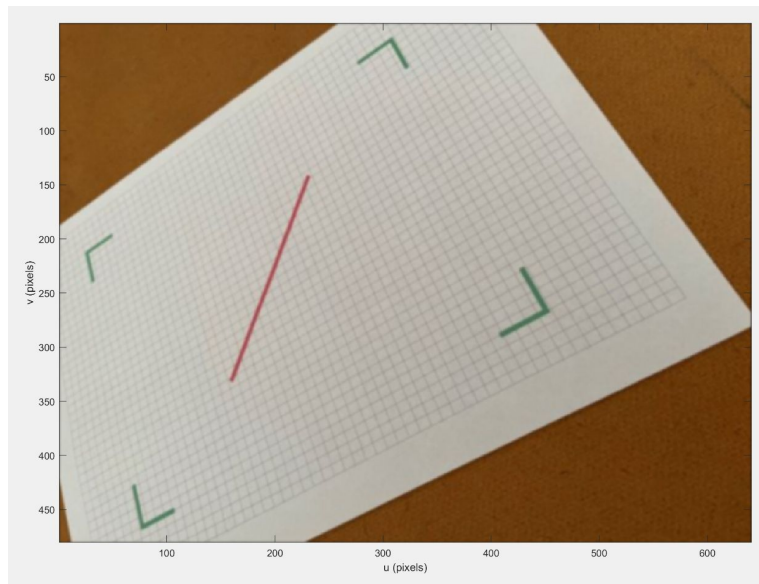
Tobias Scala 55391  
Juan Laguinge 57430  
Martín Blas Trillo 55014



# Objetivo

Se tendrá una imagen inicial en el cual se mostrará una línea contenida en una imagen. Mediante procesamiento de imágenes, se obtendrán los puntos inicial y final de la misma en coordenadas cartesianas. El brazo robótico deberá luego dibujarla con las mismas características (posición, longitud, ángulo, etc) sobre un pizarrón que se encuentra al frente.

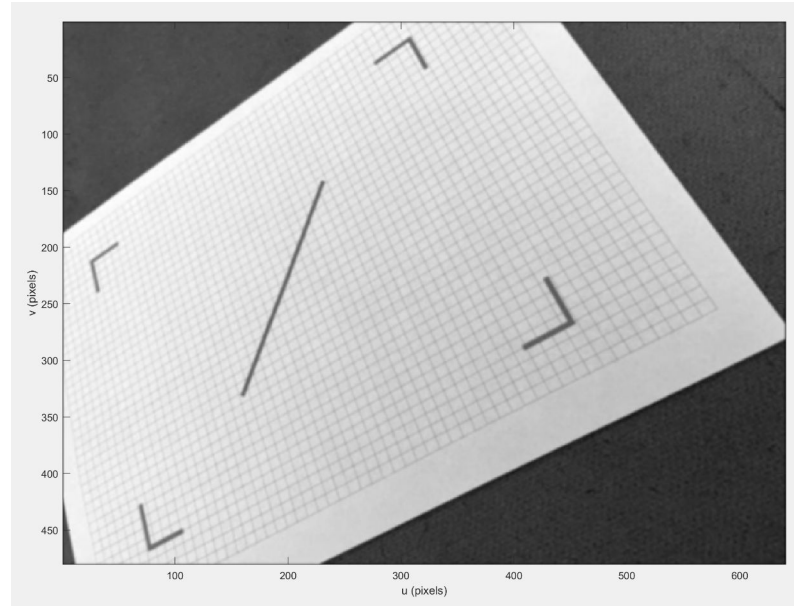






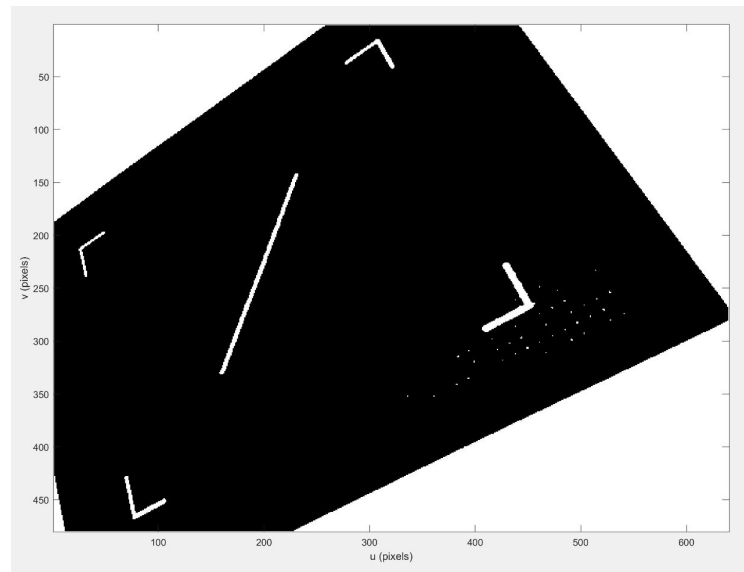
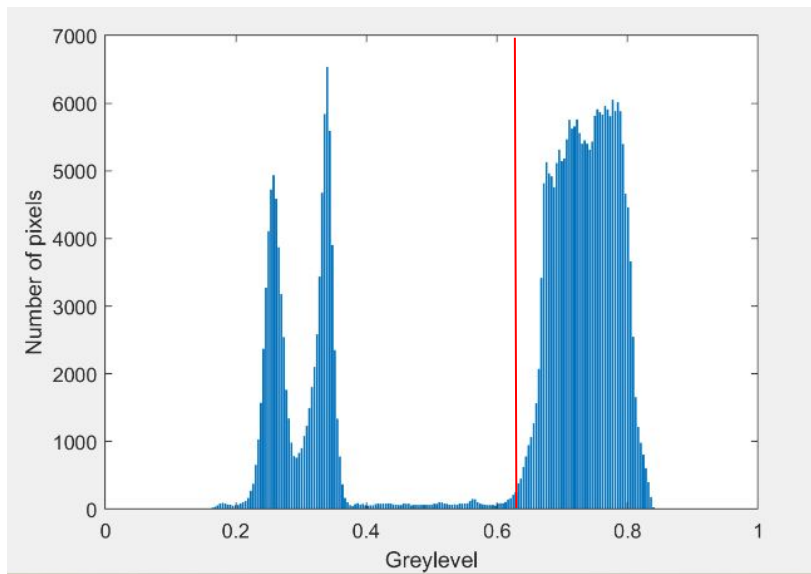
# Procesamiento de Imágenes

Como primer paso, se convierte la imagen principal en escala de grises para poder contener toda la información en un único vector. De esta manera se podrá manipular la información de una forma más compacta.



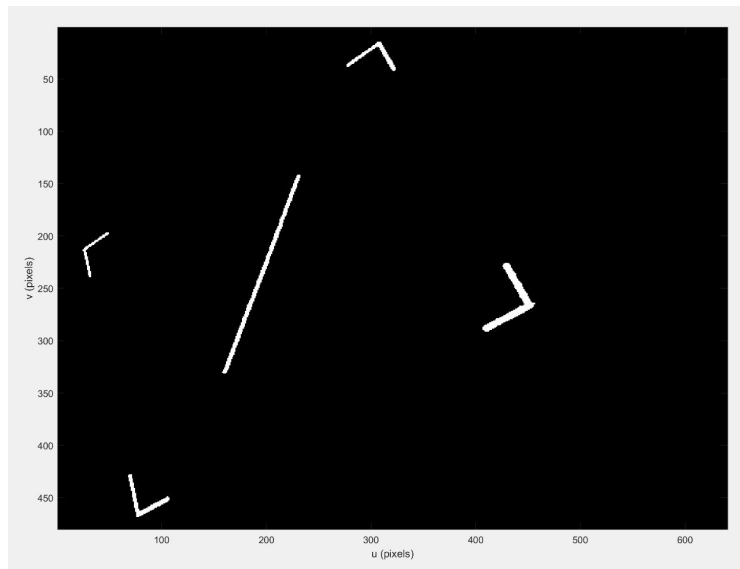
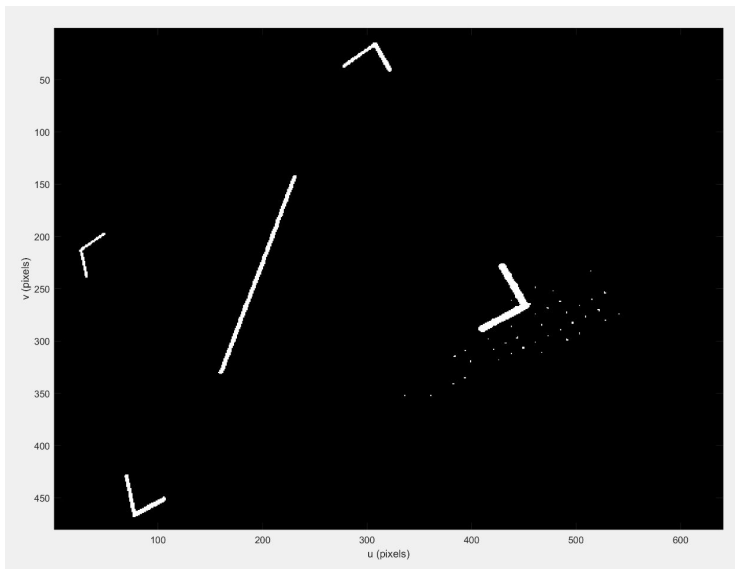
# Procesamiento de Imágenes

Dado que la hoja es lo más claro en la imagen, hay una gran concentración de píxeles “blancos” en el espectro. Por lo tanto, mediante la función **seekBright()**, se busca el inicio del lóbulo que contiene dichos píxeles. Luego, se procede a realizar un threshold convirtiendo los píxeles correspondientes a la hoja a negro.



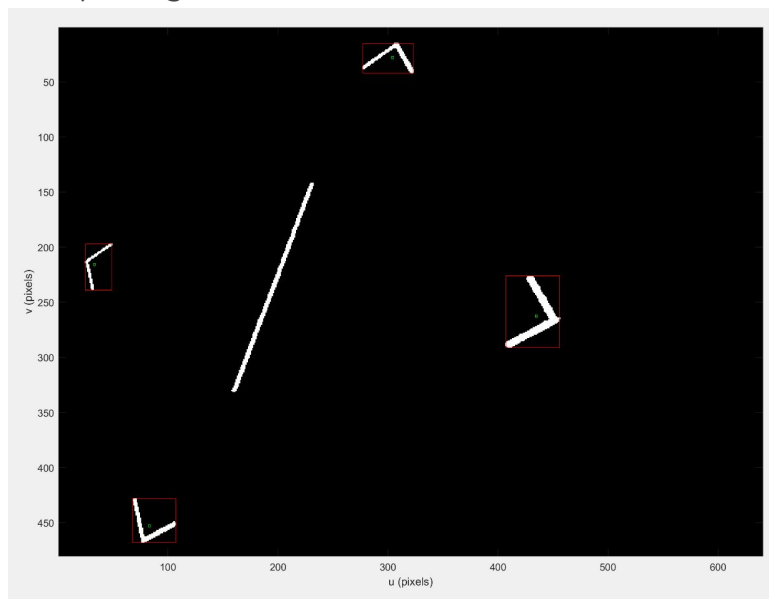
# Procesamiento de Imágenes

Ahora, mediante la función **deleteBackground()**, se procede a retirar todos los píxeles blancos que se encuentran entre los extremos (de la imagen) y la hoja. Luego, se debe eliminar el ruido remanente (puntos o islas blancas) con la función **deleteIslands()**. El cual utiliza un kernel cuadrado que elimina todos los píxeles blancos dentro del mismo.



# Procesamiento de Imágenes

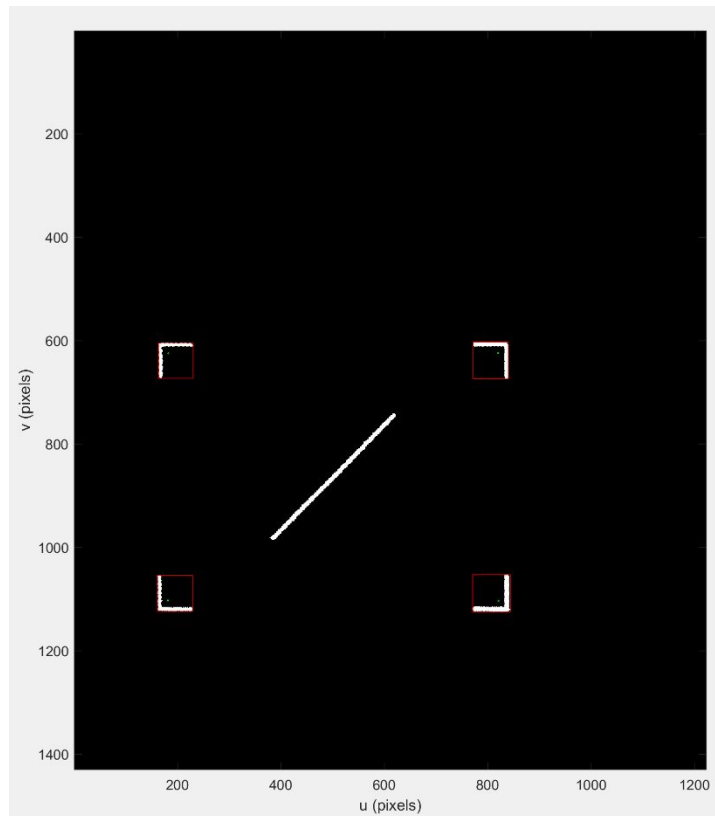
Dado que ya se tienen los objetos de interés (la línea y los corners) en blanco, lo próximo es identificar los corners. Para esto, se utiliza la función propia del toolbox **blobs()** y la función **seekCorners()** el cual identifica los blobs extremos (ubicado más arriba, más abajo, más a la derecha y más a la izquierda). Luego se obtiene el centro de masa de cada corner.



# Procesamiento de Imágenes

A continuación, se utiliza la función **sortPoints()** el cual ordena los puntos de los centros de masa de cada corner para colocarlos en la función propia del toolbox **homography()** y, así, calcular la matriz de homografía.

Luego, con la función propia del toolbox **homwarp()** se cambia la perspectiva de la imagen como si la hoja fuera vista de frente. El cambio de perspectiva se realiza tomando los centros de masa de cada corner como puntos iniciales y el tamaño de la imagen  $((0,0),(640,0),(640,480),(0,480))$  como puntos finales.

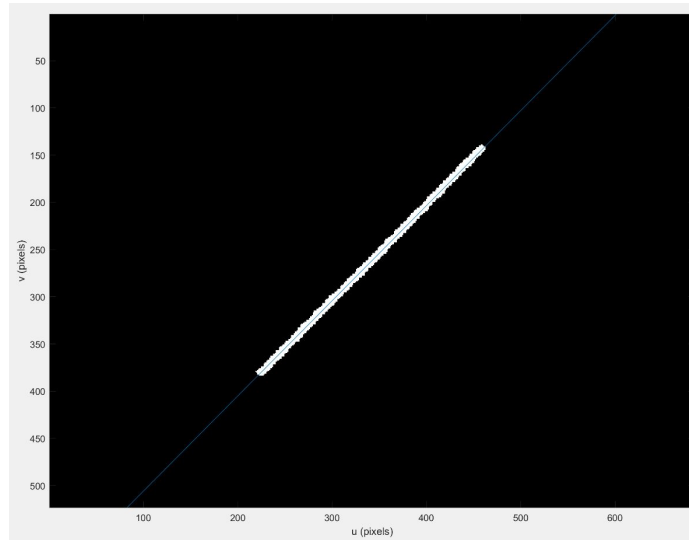






# Procesamiento de Imágenes

Para conocer los puntos iniciales y finales de la recta, es necesario eliminar los corners. Para eliminarlos se utilizan la función propia del toolbox **blob()**, y las funciones **seekCorners()** y **deleteCorners()**. Para obtener los puntos se utiliza la función propia del toolbox **Hough()** y la función **seekLine()** el cual recorre la línea de Hough hasta encontrar 2 cambios de color guardando la posición de cada punto y ordenando el punto más cercano al manipulador como inicial y el más lejano como final.

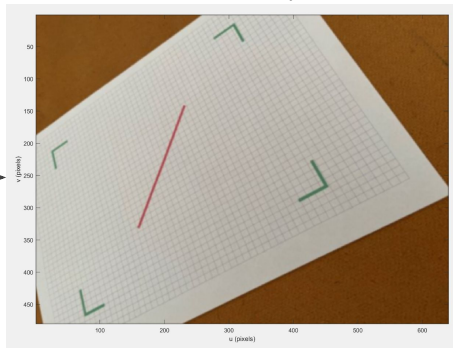




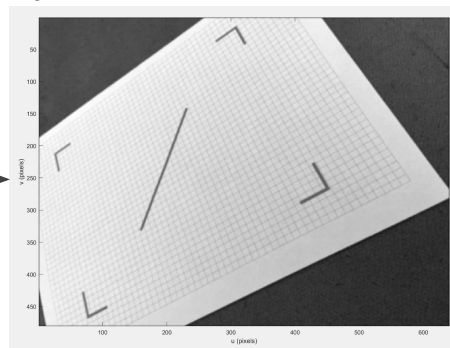
# Procesamiento de Imágenes: FlowChart

Resumen de las funciones que conforman el proyecto y su flujo.

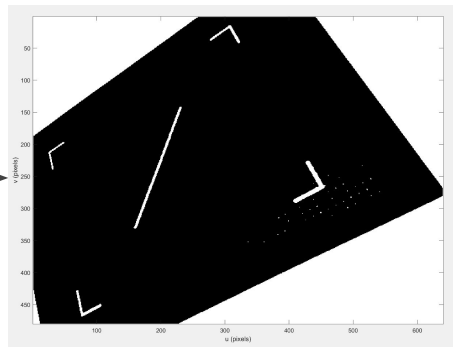
`iread()`



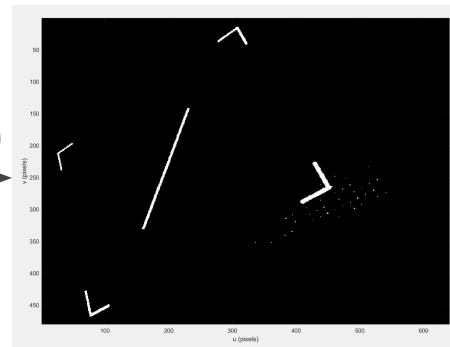
`imono()`



`seekBright()`



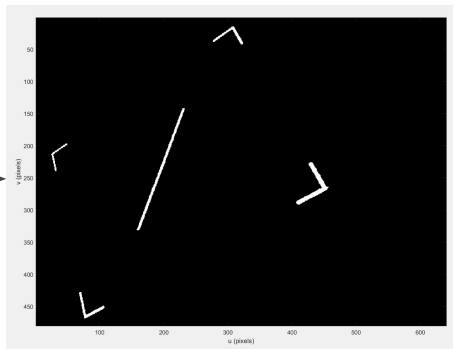
`deleteBackground()`



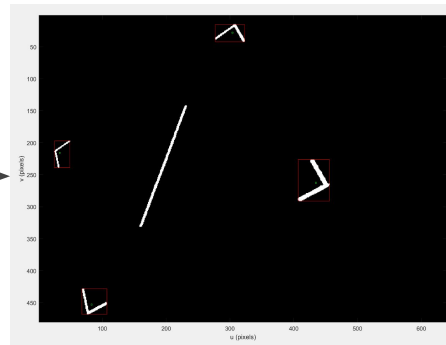


# Diagrama de Flujo

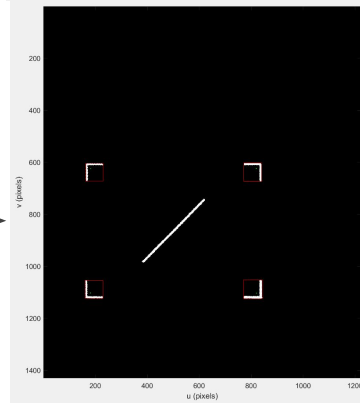
`deleteIslands()`



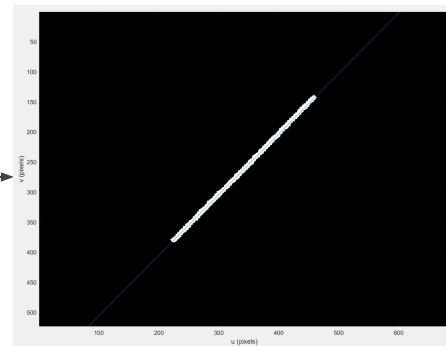
`blobs()`  
`seekCorners()`



`sortPoints()`  
`homography()`  
`homwarp()`  
`blobs()`  
`seekCorners()`

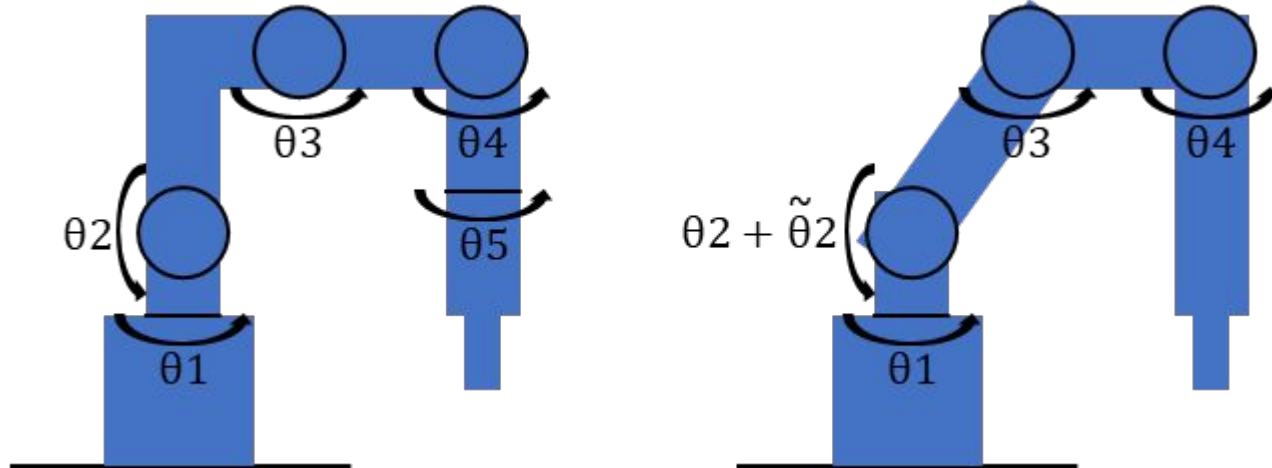


`deleteCorners()`  
`Hough()`  
`seekLine()`



# Modelo Matemático del Brazo Robótico

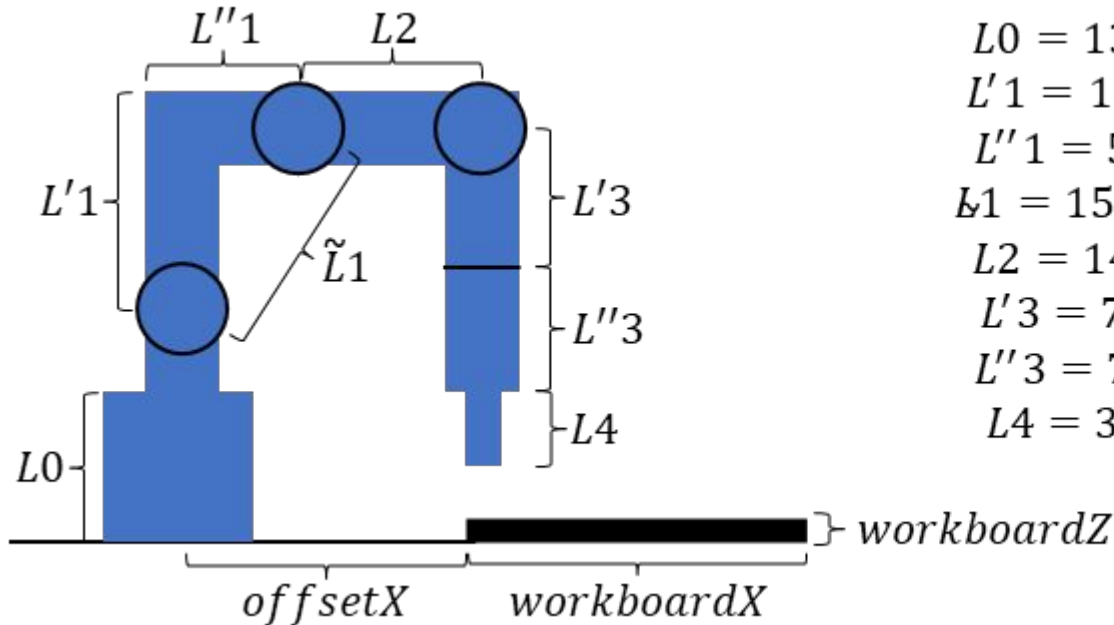
A pesar de que el robot sea de 5 ejes, se puede considerar un modelo reducido de 4 ejes para esta aplicación. Esto es debido a que se requiere que la herramienta esté siempre perpendicular al plano del workspace, entonces el ángulo  $\theta_4$  dependerá de  $\theta_2$  y de  $\theta_3$  y, además, el ángulo  $\theta_5$  no es de importancia en esta aplicación y se podrá ignorar. Además, el link en forma de 'L' puede ser reemplazado por un link recto pero rotado.





# Modelo Matemático del Brazo Robótico

Las dimensiones del sistema completo son las siguientes.



$$L_0 = 130mm$$

$$L'_1 = 144mm$$

$$L''_1 = 56mm$$

$$L_1 = 154.5mm$$

$$L_2 = 144mm$$

$$L'_3 = 72mm$$

$$L''_3 = 72mm$$

$$L_4 = 30mm$$

$$offsetX = 120mm$$

$$workboardX = 150mm$$

$$workboardY = 200mm$$

$$workboardZ = 30mm$$



# Modelo Matemático del Brazo Robótico

A continuación se muestran los parámetros DH del modelo completo y del modelo simplificado.

i	$a_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	0	0	$\theta_1$	$L_0$
2	$90^\circ$	0	$\theta_2$	0
*	$-90^\circ$	0	0	$L'_1$
3	$90^\circ$	$L''_1$	$\theta_3$	0
4	0	$L_2$	$\theta_4$	0
5	$90^\circ$	0	$\theta_5$	$L'_3$
6	0	0	0	$L''_3+L_4$

i	$a_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	0	0	$\theta_1$	$L_0$
2	$90^\circ$	0	$\theta_2+\text{atan}(L''_1/L'_1)$	0
3	0	$\text{sqrt}(L'_1^2+L''_1^2)$	$\theta_3-\text{atan}(L''_1/L'_1)$	0
4	0	$L_2$	$\theta_4$	0
5	$90^\circ$	0	0	$L'_3+L''_3+L_4$



# Modelo Matemático del Brazo Robótico

Las ecuaciones de cinemática inversa son las siguientes.

$$\theta_1 = \text{atan}\left(\frac{P_y}{P_x}\right) \quad \text{if } P_x \neq 0$$

$$\theta_3 = \text{asin}\left(\frac{P_x^2 + P_y^2 + P_z^2 - \tilde{L}_1^2 - L_2^2}{2\tilde{L}_1 L_2}\right)$$

$$\tilde{\theta}_2 = \text{atan}\left(\frac{L_1''}{L_1'}\right)$$

$$\theta_2 = \text{asin}\left(\frac{P_z L_2 C\theta_3 - (\tilde{L}_1 + L_2 S\theta_3) \sqrt{P_x^2 + P_y^2}}{\tilde{L}_1^2 + 2\tilde{L}_1 L_2 S\theta_3 + L_2^2}\right) + \tilde{\theta}_2 \quad \text{if } \theta_3 \neq 90$$

$$\theta_4 = -\theta_2 - \theta_3$$



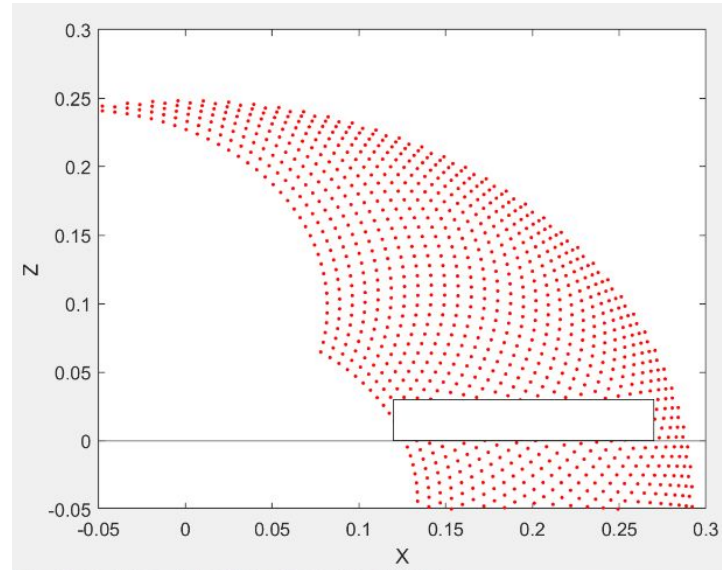
# Modelo Matemático del Brazo Robótico

A continuación, se muestra un corte transversal del workspace del sistema. Los ángulos de cada joint están acotados por los siguientes valores.

$$\begin{aligned}\theta_1 &= (-\text{atan}(\text{workboardY}/(2*\text{offsetX})), \text{atan}(\text{workboardY}/(2*\text{offsetX}))) = \\ &= (-0.7, 0.7)\text{rad} = (-40.1, 40.1)\text{deg} \\ \theta_2 &= (-\pi/3, \pi/4)\text{rad} = (-60, 45)\text{deg} \\ \theta_3 &= (-\pi/3, \pi/4)\text{rad} = (-60, 45)\text{deg}\end{aligned}$$

De esta forma, la superficie del pizarrón se encuentra dentro del workspace.

Pero, este workspace no es seguro. El end effector puede adquirir mucha altura y, también, puede “perforar” el pizarrón.



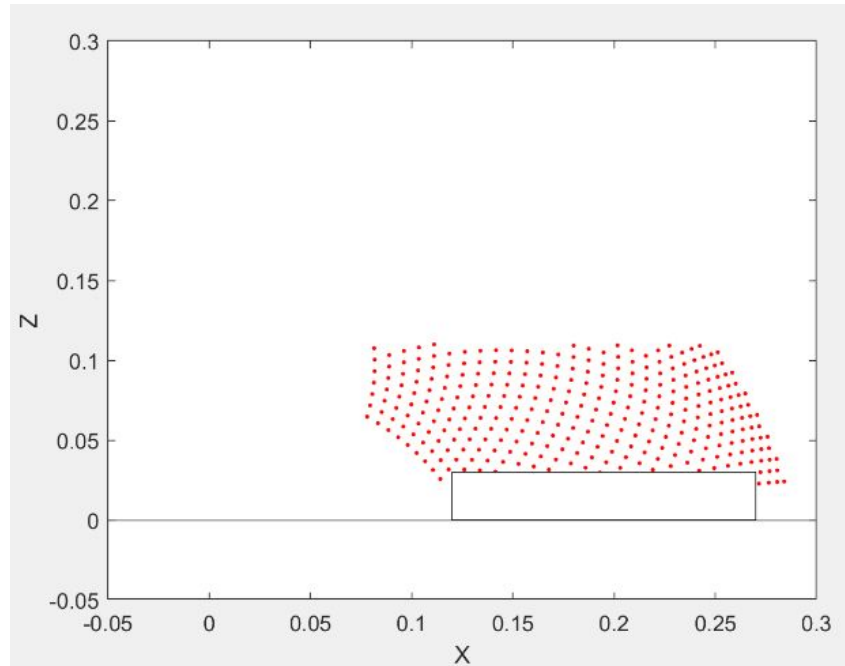




# Modelo Matemático del Brazo Robótico

Para definir un workspace más seguro, se limita también la altura del end effector a los siguientes valores.

$$Y6=(0.02, 0.11)$$



# ¡Muchas gracias!

Profesores:

Rodolfo Arias

Federico Sofio Avogadro

Mariano Tomás Spinelli

