

# Modelo y Simulación de un Brazo Robótico tipo Scara Utilizando Procesamiento de Imágenes y Splines para el Trazado de una Ruta

Altamirano Soria José Edgar, Ramírez Ángeles Iván Joel, Román Rivera Luis Rogelio  
y Vargas Soto José Emilio

Universidad Autónoma de Querétaro, Facultad de Ingeniería.

## Resumen

*Un brazo robótico tipo SCARA (Selective Complican Assembly Robot Arm) permite trazar una ruta sobre la cual se pueden realizar diferentes aplicaciones como cortes, gravados e impresiones por mencionar algunas. El presente trabajo es una guía para modelar un brazo robótico tipo SCARA compuesto de tres grados de libertad, que permite simular la trayectoria que sigue el brazo para anticipar las capacidades y límites del diseño al trazar diferentes rutas. Se ejemplifica el modelo mediante el trazo de una ruta de una imagen previamente procesada usando splines cúbicas. El demo presentado se realiza en matlab sin embargo el modelo es de aplicación general. Se realiza un análisis cinemático inverso, para obtener los valores de los ángulos correspondientes a las articulaciones rotacionales, de acuerdo al punto (x,y) del plano donde se desee posicionar a la terminal del robot.*

**Palabras clave:** scara, spline, procesamiento de imágenes, trayectoria.

## 1. Introducción

Un brazo robótico se compone de cierto número de eslabones unidos a través de articulaciones, que pueden ser prismáticas o rotacionales, como se muestra en la Figura 1, en forma de cadena cinemática abierta, donde los puntos inicial y final del brazo no se encuentran físicamente unidos.

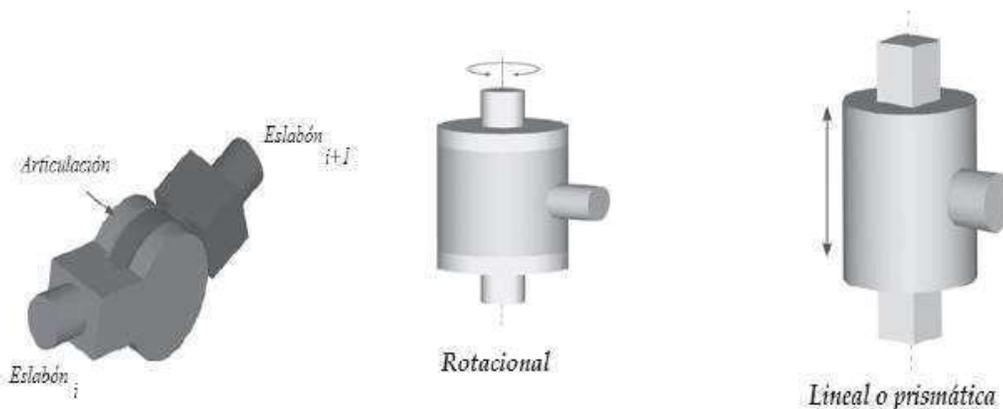


Figura 1. Tipos de articulaciones.

De acuerdo con el tipo de articulaciones presentes en el diseño en cinemática abierta del brazo, estos se pueden clasificar de acuerdo con la siguiente tabla:



**Tabla 1. Clasificación de brazo robóticos.**

Robot	Características
Antropomórfico (RRR)	3 articulaciones rotacionales
SCARA (RRP)	2 articulaciones rotacionales y 1 prismática
Esférico (RRP)	2 articulaciones rotacionales y 1 prismática
Cilíndrico (RPP)	1 articulación rotacional y 2 prismáticas
Cartesiano (PPP)	3 articulaciones prismáticas

En este trabajo se implementa la modelación y simulación de un brazo robótico tipo SCARA como se muestra en la Figura 2, que cuenta con dos articulaciones rotacionales y una prismática, por lo que el movimiento descrito por los segmentos del brazo, a los que corresponden los ángulos rotacionales, se puede representar en planos bidimensionales dado que la tercera articulación es prismática y solo presenta un movimiento lineal a lo largo del eje z.



**Figura 2. Robot SCARA**

En Robótica, la Cinemática inversa (I.K por sus siglas en ingles) es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta. El problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot  $q = [q_1, q_2, \dots, q_n]$  según la posición que se desea que tenga el extremo del brazo.

La idea principal del trabajo es poder presentar un modelo de un robot tipo SCARA que permita trazar la ruta exterior de una imagen umbralizada, esto gracias a el área de procesamiento de imágenes. Aunque el objetivo del trabajo es presentar el funcionamiento, nos limitamos a presentar el análisis de la cinemática inversa y el splines y no profundizamos en el tema de procesamiento de imágenes debido al amplio tema que abarca.

## 2. Antecedentes

Existen varias implementaciones de Robots tipo Scara que siguen cierta trayectoria definida por una función determinada. En algunos casos se tiene un análisis de 5 o más grados de libertad en



donde se considera el comportamiento de los actuadores que se usaran en la implementación física, como en el caso de Kern, que simula y modela un robot tipo SCARA con 5 grados de libertad en donde incluye el modelo dinámico de los actuadores y valida su sistema mediante el seguimiento de una trayectoria dada por una función continua [2].

Otras implementaciones buscan mejorar el tiempo de computo requerido para el cálculo de las posiciones de las articulaciones (modelo cinemático inverso), este es el caso de Noriega, que presenta una formulación para la resolución de la cinemática inversa para un robot tipo Scara, en donde toma en cuenta la posición, velocidad y aceleración para el desarrollo del modelo. El robot sigue una trayectoria definida por una función [3].

Diversas aplicaciones fueron dadas a los modelos hechos del robot Scara. En el 2016, Villarreal, presenta el diseño de una arquitectura de control abierta con fines didácticos. El robot Scara es capaz de seguir una trayectoria y su principal objetivo es que los estudiantes puedan hacer pruebas con el robot [4]. Otra aplicación fue desarrollada por Pillajo, en la que la trayectoria que sigue el robot tipo Scara es controlado por medio de los datos obtenidos por un sensor Kinect. El Kinect interpreta los movimientos del brazo del usuario para ajustar la posición del actuador final del robot [5].

### 3. Análisis cinemático inverso

Los ángulos y distancias que se toman en cuenta en el análisis cinemático inverso del robot SCARA a simular, se muestran en la siguiente figura:

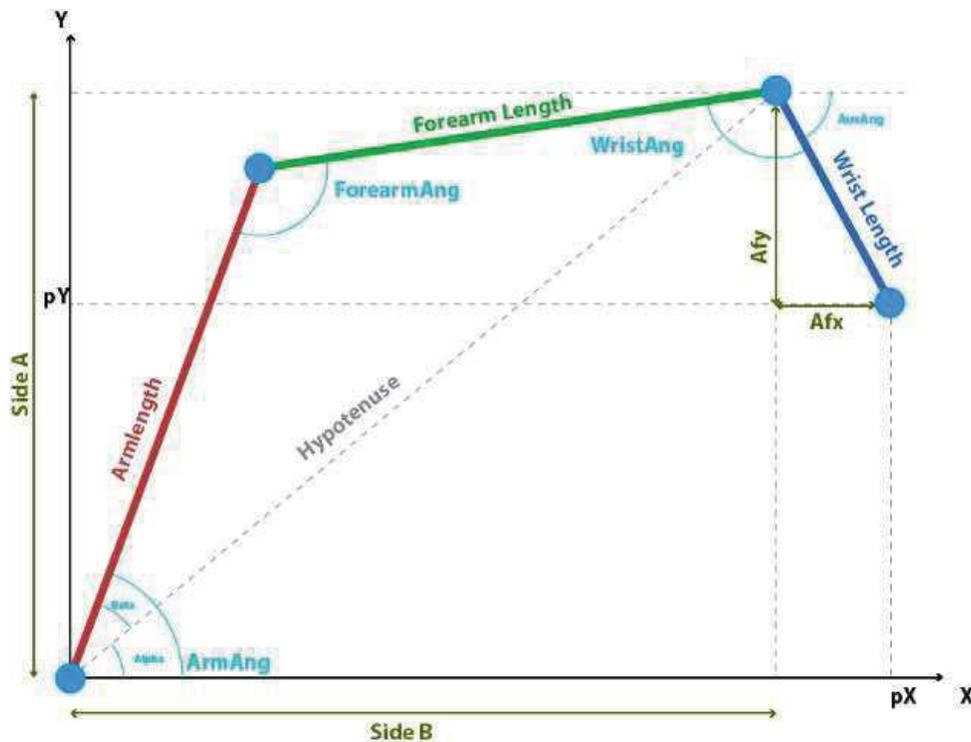


Figura 3. Ángulos y dimensiones del modelo

Las variables X e Y, representan las coordenadas en las que se debe posicionar la terminal del brazo, por lo que sus valores son previamente conocidos al análisis cinemático inverso, dado que los ángulos de las articulaciones dependerán de estos valores. La longitud del brazo (ArmLength), longitud del antebrazo (ForearmLength) y la longitud de la muñeca (WristLength), son distancias constantes conocidas, es decir, su valor no varía durante la simulación. Así mismo, el ángulo de cabeceo



(AuxAng), es otro valor constante conocido y tiene la finalidad de orientar el terminal, es decir, posicionar el objeto que se encuentra en el punto final con un ángulo determinado.

A continuación, se desarrolla el análisis matemático para la obtención de los ángulos de las articulaciones (ArmAng, ForearmAng, WristAng), en función de las aristas que las unen (Armlength, ForearmLength, WristLength).

$$Afx = \cos(AuxAng) * WristLength \quad (1)$$

$$SideB = X - Afx \quad (2)$$

$$Afy = \sin(AuxAng) * WristLength \quad (3)$$

$$SideA = Y + Afy \quad (4)$$

Aplicando el teorema de Pitágoras:

$$Hypotenuse = \sqrt{SideA^2 + SideB^2} \quad (5)$$

$$Alpha = \tan^{-1}\left(\frac{SideA}{SideB}\right) \quad (6)$$

Aplicando la ley de cosenos

$$ForearmLength^2 = Armlength^2 + hypotenuse^2 - 2 * Armlength * hypotenuse * \cos(Beta)$$

$$Beta = \cos^{-1}\left(\frac{ForearmLength^2 - Armlength^2 - Hypotenuse^2}{-2 * Armlength * Hypotenuse}\right) \quad (7)$$

$$ArmAng = Alpha + Beta \quad (8)$$

Aplicando ley de cosenos

$$ForearmAng = \cos^{-1}\left(\frac{ForearmLength^2 + ArmLength^2 - Hypotenuse^2}{2 * ArmLength * ForearmLength}\right)$$

$$WristAng = AuxAng - ArmAng - ForearmAng \quad (9)$$

## 4. Procesamiento de Imagen

Inicialmente contamos con una imagen que se desea procesar. Es importante, que se cuente con un fondo que contraste con la imagen, para que el proceso de umbralización sea mucho más eficiente. La imagen que utilizaremos se muestra en la Figura 4.

El rango del umbral es definido como una variable al iniciar el programa, en un futuro desarrollo este valor será definido automáticamente para evitar la intervención humana.

Una vez que el proceso de umbralización haya sido satisfactorio, se procede a utilizar un algoritmo propio para la detección de bordes de la imagen. En la Figura 5 se muestra el resultado de umbralización y detección de bordes.



Figura 4. Imagen del contorno de una mano

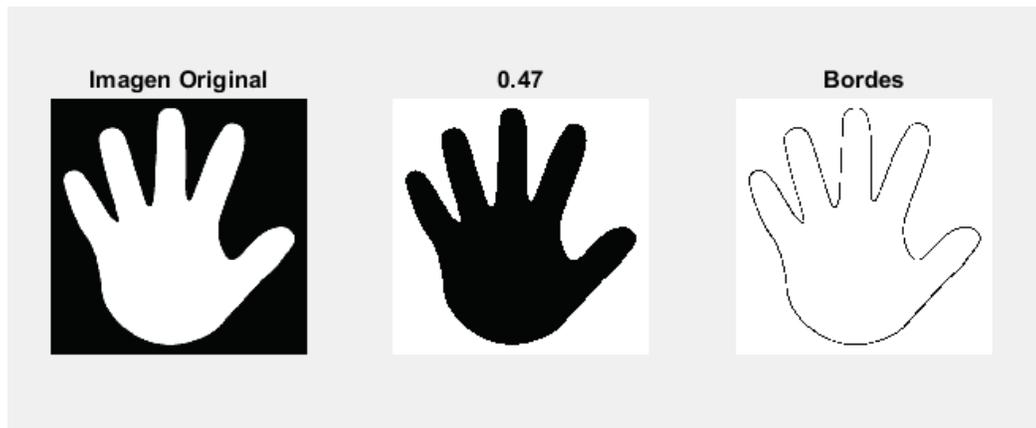
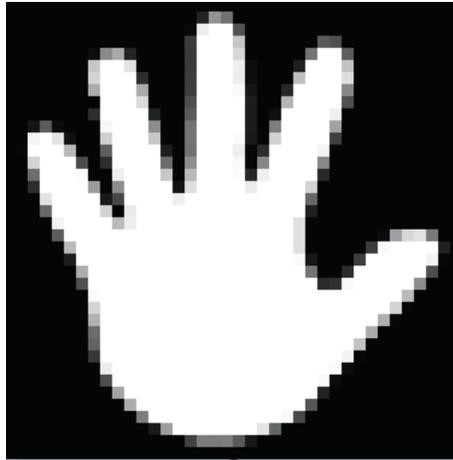


Figura 5. Resultados de la umbralización.

Para el ordenamiento de los puntos ordenados, se trabajo un algoritmo propio para obtener los puntos ordenados en sentido de las manecillas del reloj.

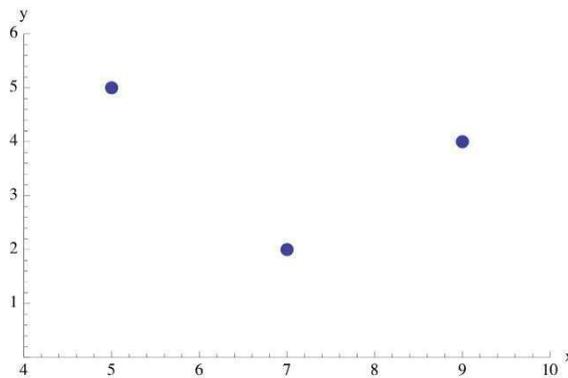
## 5. Análisis de SPLINE

Una de las razones para implementar el spline, es ahorrar memoria y ofrecer un mejor aproximamiento a la imagen en caso de que cambie de tamaño y evitar el efecto de pixelado como se muestra en la Figura 6.



**Figura 6. Imagen con efecto pixelado.**

La ventaja de los splines es que no presentan este efecto. A continuación, se explica mediante un ejemplo el funcionamiento de la spline cúbica. Dados 3 puntos  $\{(5,5), (7,2), (9,4)\}$  se requiere una curva que pase por todos los puntos.



**Figura 7. Representación gráfica de los puntos.**

Para lo cual se requieren dos ecuaciones cúbicas, una por intervalo entre 2 puntos.

$$S_0(X) = a_0 + b_0(x - 5) + c_0(x - 5)^2 + d_0(x - 5)^3 \quad (12)$$

$$S_1(X) = a_1 + b_1(x - 7) + c_1(x - 7)^2 + d_1(x - 7)^3 \quad (13)$$

Debido a que hay 8 coeficientes, se deben de resolver 8 ecuaciones. Los splines deben de coincidir con la función (coordenada "y") en los puntos (coordenada "x").

$$5 = S_0(5) = a_0 \quad (14)$$

$$2 = S_0(7) = a_0 + 2b_0 + 4c_0 + 8d_0 \quad (15)$$

$$2 = S_1(7) = a_1 \quad (16)$$

$$4 = S_1(9) = a_1 + 2b_1 + 4c_1 + 8d_1$$

La primera y segunda derivada de las ecuaciones cubicas se representan en las fórmulas 17 y 18.



$$S'_0(7) = b_0 + 4c_0 + 12d_0 = b_1 = S'_1(7) \quad (17)$$

$$S''_0(7) = 2c_0 + 12d_0 = 2c_1 = S''_1(7) \quad (18)$$

Las dos ecuaciones finales provienen de las dos condiciones de los límites o extremos, y se representan en las fórmulas 19 y 20.

$$S''_0(5) = 0 = 2c_0 \quad (19)$$

$$S''_1(9) = 0 = 2c_1 + 12d_1 \quad (20)$$

Las 8 ecuaciones juntas forman el siguiente sistema:

$$5 = a_0 \quad (21)$$

$$2 = a_0 + 2b_0 + 4c_0 + 8d_0 \quad (22)$$

$$2 = a_1 \quad (23)$$

$$4 = a_1 + 2b_1 + 4c_1 + 8d_1 \quad (24)$$

$$0 = b_0 + 4c_0 + 12d_0 - b_1 \quad (25)$$

$$0 = 2c_0 + 12d_0 - 2c_1 \quad (26)$$

$$0 = 2c_0 \quad (28)$$

$$0 = 2c_1 + 12d_1 \quad (29)$$

La solución al sistema se muestra en la tabla 2.

**Tabla 2. Solución al sistema de ecuaciones**

$i$	$ai$	$bi$	$ci$	$di$
0	5	-17/8	0	5/32
1	2	-1/4	15/16	-5/32

La curva spline natural puede ser expresada como:

$$S(x) = \begin{cases} 5 - \frac{17}{8}(x-5) + \frac{5}{32}(x-5)^3, & 5 \leq x \leq 7 \\ 2 - \frac{1}{4}(x-7) + \frac{15}{16}(x-7)^2 - \frac{5}{32}(x-7)^3, & 7 \leq x \leq 9 \end{cases} \quad (30)$$

La Figura 8 muestra los puntos de la figura 7, unidos por las ecuaciones obtenidas de la spline cubica.

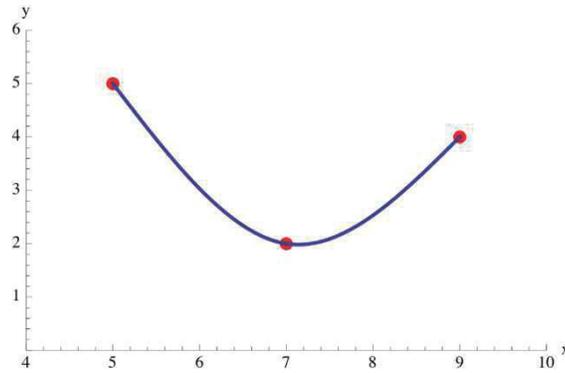


Figura 8. Representación gráfica de la interpolación.

## 6. Simulación

El diagrama de flujo presentado en la Figura 9, presenta la metodología seguida para la resolución del modelo del brazo robótico SCARA.

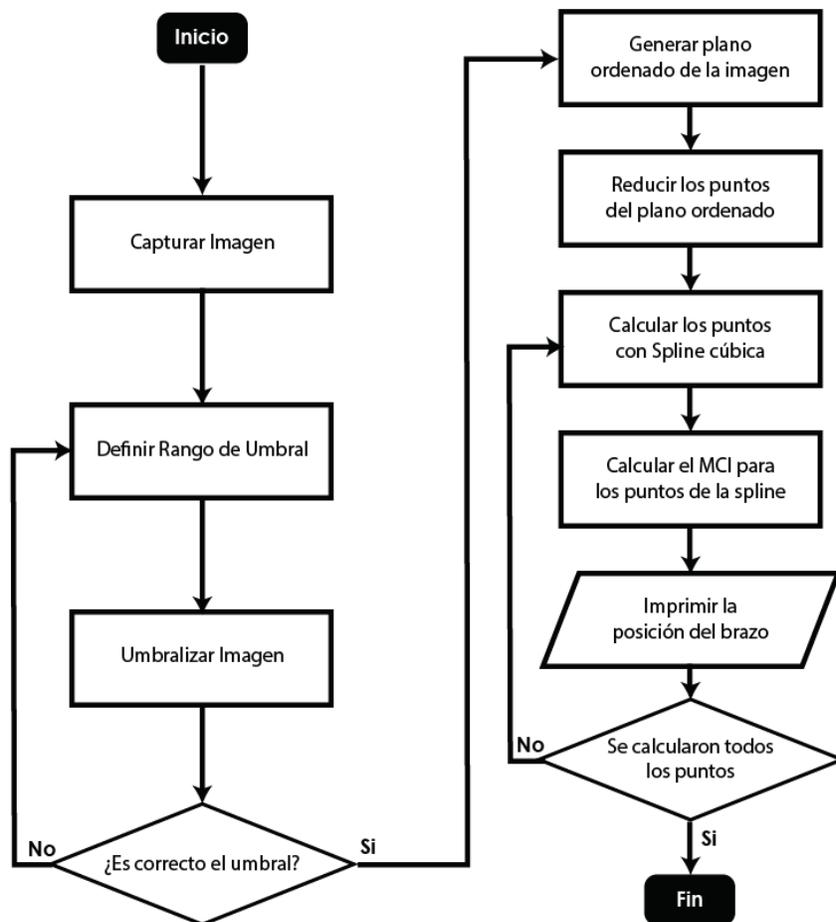


Figura 9. Diagrama de Flujo del Proceso



Inicialmente se tienen que definir la longitud de los segmentos del brazo robotico: brazo, antebrazo y muñeca. Posteriormente se carga la imagen como una matriz de puntos para poder realizar el calculo del spline para cada segmento de la imagen. Finalmente se calcula el modelo cinematico inverso para cada punto que haya regresado el spline cubico.

## 7. Resultados

El modelo resultante se aplico a la imagen de la mano (figura 10) pero para apreciar mejor el uso de las splines se presenta la figura 11, que representan 6 puntos en el espacio.

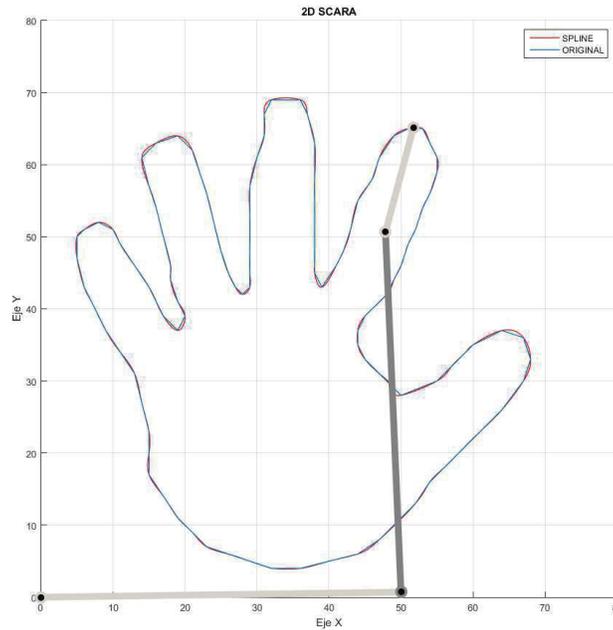


Figura 10. Resultados de la modelación con la imagen inicial.

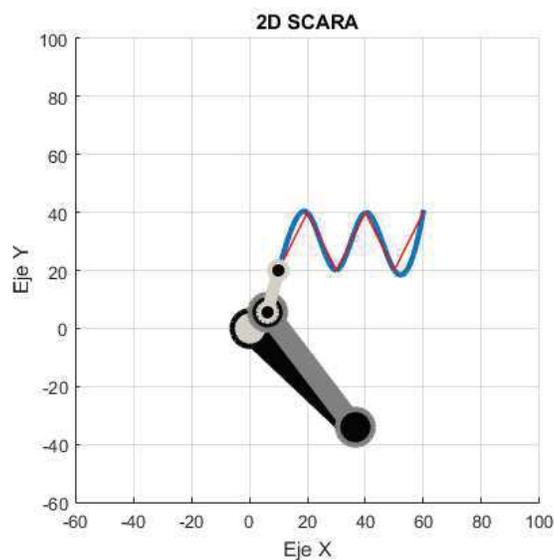


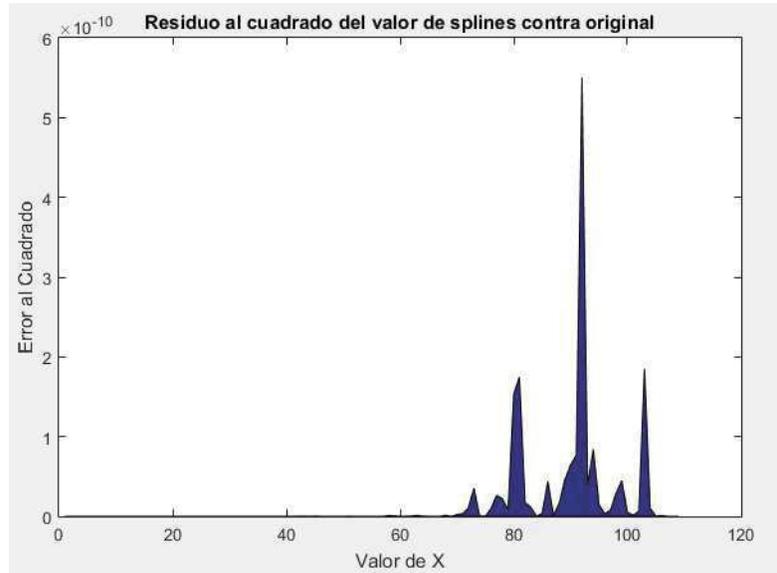
Figura 11. Comparacion de los puntos distribuidos con una SPLINE



Para calcular el error de la spline contra los puntos originales se utilizó la suma de error al cuadrado (RSS), y el valor de la media cuadrática (RMS). En la figura 12 se muestra el área de error y en la tabla 3 los valores obtenidos para cada técnica.

**Tabla 3. Valores de la suma del error al cuadrado y media cuadrática**

<i>RSS</i>	<i>RMS</i>
1.7174478151215e-09	6.2108759834023e-11



**Figura 12. Representación visual de la suma del error al cuadrado**

Es importante aclarar que este procedimiento funciona solamente para los bordes exteriores de las imágenes, si se desea procesar alguna imagen con algún borde interior, esta debe ser tratada como un nuevo objeto.

La ventaja de poder utilizar splines, es que el procesamiento de la imagen es menor al tener inicialmente menos puntos que calcular.

## 8. Conclusiones

En este trabajo se ha descrito el modelo y simulación de un brazo robótico tipo SCARA en 2 dimensiones utilizando SPLINE cúbica como método de mejora para el suavizado de la imagen trazada. El modelo realiza un recorrido de la imagen siguiendo la trayectoria y calculando cada posición del brazo mediante cinemática inversa. El presente trabajo propone una alternativa para la impresión de imágenes físicamente realizando operaciones que no requieran de un gran poder de computo. Actualmente el uso de esta técnica puede ser llevada desde máquinas de suajes, impresión o técnicas de uso médico.

## 9. Trabajo Futuro

Como trabajo futuro se puede implementar una 3ra dimensión en la posición del brazo, permitiendo realizar rutas en 3 dimensiones.



## Referencias

- [1] Reyes F., *MATLAB aplicado a Robotica y Mecatronica*. 2012.
- [2] J. Kern and C. Urrea, "Modelamiento y simulación de un robot redundante de tipo manipulador SCARA," *Am. J. Phys. Educ*, vol. 5, no. 4, 2011.
- [3] A. Noriega González, A. García Martínez, and M. Muñiz Calvente, "Resolución del problema cinemático inverso en un robot SCARA mediante grupos de Assur."
- [4] M. G. Villarreal-cervantes, D. De-la-cruz-muciño, J. S. Pantoja-garcía, and G. Sepúlveda-cervantes, "Arquitectura de control abierta por medio de una PC para sistemas mecatrónicos," vol. 54, pp. 17–24, 2016.
- [5] C. Pillajo and J. E. Sierra, "Human Machine Interface HMI using Kinect sensor to control a SCARA robot," *2013 IEEE Colomb. Conf. Commun. Comput. COLCOM 2013 - Conf. Proc.*, pp. 1–5, 2013.
- [6] Buchanan, J. Robert. "Cubic Spline Interpolation." *MATH 375, Numer Anal* (2010).