

Generación de Trayectorias por Visión para un Robot Manipulador de 5 Grados de Libertad

Soto Cajiga Jorge Alberto¹, Vargas Soto José Emilio²,
Pedraza Ortega Jesús Carlos¹.

¹Centro de Ingeniería y Desarrollo Industrial, Unidad de Investigación y Postgrado

²Universidad Anáhuac – México Sur. Facultad de Ingeniería

Resumen

En este trabajo se presenta una propuesta para generar trayectorias cerradas para un robot manipulador. A partir de una imagen específica, que contenga un objeto, se le aplica un procesamiento de imagen para obtener el contorno del objeto, con este contorno y aplicando cinemática inversa se genera la trayectoria que debe seguir el manipulador, para finalmente simular y generar la trayectoria en un manipulador de 5 grados de libertad.

Palabras clave: Imagen, Cámara, Splines, Cinemática, Manipulador, Trayectoria, OpenGL.

1. Introducción

La robótica, la visión artificial y sus aplicaciones, son hoy en día un campo de gran interés. Investigaciones en estos campos prometen desarrollos avanzados y novedades en muchos aspectos. Aplicaciones de proyectos que combinan la robótica con la visión artificial son encontradas cada vez más en nuestros días y cada día son más interesantes y complicadas.

La idea principal de este trabajo es generar una aplicación práctica, utilizando un robot manipulador de 5 grados de libertad y una cámara digital. Dentro de las posibles aplicaciones se tienen por ejemplo; cortar, soldar y/o dibujar. La generación de la trayectoria puede ser extrapolada a cualquier otro robot o maquina que pueda seguir la trayectoria.

Los pasos a seguir en el desarrollo de este trabajo son los siguientes:

- Obtener la imagen y realizar el preprocesamiento.

- Generar la trayectoria del objeto en curvas paramétricas utilizando trazadores cúbicos (splines) para suavizar el contorno.
- Con la cinemática inversa y las curvas paramétricas, generar la trayectoria para el robot manipulador.
- Realizar la simulación del seguimiento de la trayectoria, utilizando librerías de OpenGL.

2. Generación de Trayectoria

Existen dos formas básicas de programar la trayectoria en un robot manipulador, la programación Explícita y la programación Implícita.

La programación Explícita a su vez se divide en programación Gestual y programación Textual. En la programación Gestual se le enseña al robot, llevándolo a la posición de interés, guardándola y definiendo la velocidad y aceleración, a este tipo de programación también se le conoce como programación en línea, muchas veces es necesario detener el proceso en el que actuara el manipulador para poder programarlo. En la programación Textual, se realizan los cálculos cuidadosamente de la cinemática inversa, para generar la trayectoria, considerando la posición, la velocidad, la aceleración y el espacio de trabajo, este tipo de programación también es conocido como programación fuera de línea y no es necesario detener el proceso en el que actuara el manipulador.

La programación Implícita, es cuando el robot cuenta con tareas predefinidas y con ayuda de un sistema de sensores y programación adecuados, hace parecer inteligencia en el robot. Esta programación cae dentro de la inteligencia artificial y es lo más avanzado en cuanto a programación de manipuladores de refiere.

2.1 Obtención de imagen y preprocesamiento

El procesamiento de imágenes es básico pero no menos importante en este trabajo. El proceso de captura de imagen y su procesamiento cuenta con un gran número de metodologías y algoritmos muy estudiados. El conocimiento de la aplicación y modo de empleo de estas herramientas son la clave para tener un resultado óptimo y funcional.

Primero se obtiene la imagen con cualquier cámara, procurando tener una buena iluminación y un buen contraste entre el fondo y el objeto deseado (ver figura 1, imagen real). La imagen se pasa a una escala de grises, si es necesario. Posteriormente se binariza y finalmente se genera el contorno, guardando los datos en dos vectores (x,y), los cuales representan la trayectoria real de la imagen. En la figura 1 se muestran los resultados de este proceso.

El algoritmo para seguir el contorno, es una aportación propia, consiste en asignar la dirección de búsqueda (sentido horario o antihorario), buscar un punto inicial, seguir todos y cada uno de los puntos del contorno y guardarlos en un par de vectores (x,y), los cuales nos representan la ubicación en el plano y son necesarios para obtener las splines.

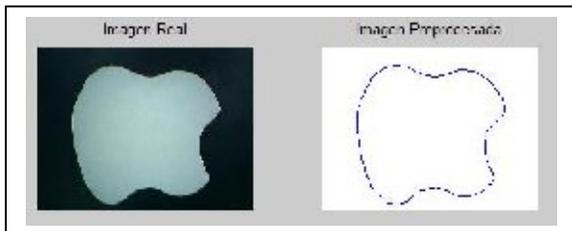


Fig. 1. Resultados del Procesamiento.

2.2 Suavización del contorno con Splines

Una vez que ya tenemos el contorno de la imagen en los vectores (x,y), puede ser que el contorno presente irregularidades no deseadas, debido a defectos en la obtención de la imagen o en el preprocesamiento, las cuales se pueden ver como ruido, es por esto que en ocasiones es necesario suavizar el contorno de la imagen.

Una opción para suavizar la imagen, es utilizar trazadores cúbicos (Splines), que no son más que una serie de polinomios cúbicos, los cuales representan la trayectoria. El proceso para suavizar la trayectoria, consiste en conocer la complejidad de la misma, seleccionar los puntos que representen de mejor

manera el contorno deseado y aplicar los trazadores cúbicos.

En la figura 2, se representan los puntos seleccionados para suavizar el contorno. Para este ejemplo la longitud del vector inicial es de 1339 puntos, de los cuales se seleccionan 35 y como podemos observar en la misma figura 2, los resultados son interesantes.

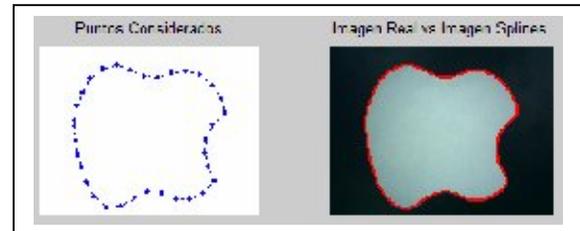


Fig. 2. Resultados de suavizar.

Las splines no solo nos ayudan a suavizar el contorno, sino que también, se reduce el número de memoria utilizado y lo más interesante, dado que son polinomios cúbicos, podemos conocer la primera y la segunda derivada para cada uno de los puntos, lo cual nos facilita el trabajo para obtener la cinemática inversa para el robot.

2.3 Ubicación en el espacio de trabajo

A partir de la trayectoria obtenida de la imagen, el siguiente paso es ubicarla en el espacio de trabajo del robot, para este caso la trayectoria es en 2D por lo que se ubica en el plano xy a una altura constante $z=300$ mm, (escalando los píxeles de la imagen a milímetros). En la figura 3 se muestra la trayectoria ubicada en el espacio de trabajo del manipulador.

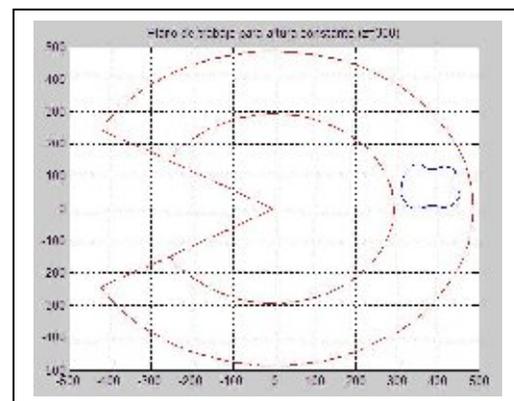


Fig. 3. Trayectoria en el espacio de trabajo.

2.4 Cinemática del Manipulador

Para desarrollar la cinemática del manipulador se utilizan matrices de transformación homogénea, en la figura 4 se muestra una imagen del robot. El robot esta dibujado en OpenGL y se tomo como base el robot Melfa RV-2AJ de Mitsubishi.

El robot cuenta con 5 grados de libertad, los 5 son rotacionales y cada uno tiene sus restricciones de rotación, estas restricciones y las de velocidad se muestran en la figura 5.

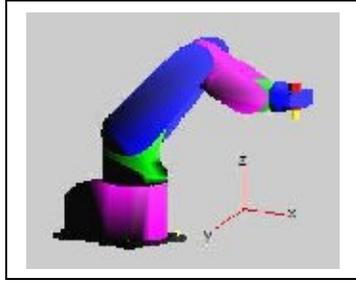


Fig. 4. Configuración del Robot.

Articulación	Limite de Rotación Deg.	Velocidad Máxima Deg./s
q1	-150 a +150	180
q2	-60 a +120	90
q3	-110 a +120	135
q4	-90 a +90	180
q5	-200 a +200	210

Fig. 5. Tabla de Restricciones del Manipulador.

La cinemática inversa, para este ejemplo, se simplifica, dado que la orientación de la herramienta en el plano es constante en dirección del eje z negativo y por consiguiente la orientación del eslabón 4 será siempre ortogonal al eje z, simplificándose la cinemática de manera considerable. Pero no siempre es así, por lo que hay que considerar todas las matrices de transformación. A continuación se presentan las matrices de transformación para la cinemática directa, representadas por la letra A, con un subíndice que representa el eje de coordenadas inicial y un superíndice que representa el eje de coordenadas final.

$$A_0^1 = \begin{bmatrix} \cos(q1) & 0 & \text{sen}(q1) & 0 \\ \text{sen}(q1) & 0 & -\cos(q1) & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_1^2 = \begin{bmatrix} \cos(q2) & -\text{sen}(q2) & 0 & L2\cos(q2) \\ \text{sen}(q2) & \cos(q2) & 0 & L2\text{sen}(q2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_2^3 = \begin{bmatrix} \cos(q3) & -\text{sen}(q3) & 0 & L3\cos(q3) \\ \text{sen}(q3) & \cos(q3) & 0 & L3\text{sen}(q3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_3^4 = \begin{bmatrix} \cos(q4) & 0 & \text{sen}(q4) & 0 \\ \text{sen}(q4) & 0 & -\cos(q4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_4^5 = \begin{bmatrix} \cos(q5) & -\text{sen}(q5) & 0 & 0 \\ \text{sen}(q5) & \cos(q5) & 0 & L4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A_5^6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & L5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

La cinemática directa sale de multiplicar cada una de las matrices de transformación, para tener una matriz que va del eje de coordenadas 0 al eje de coordenadas 6 representada como A_0^6 , como muestra la formula 7. La cinemática directa es necesaria para comprobar los cálculos de la cinemática inversa.

$$A_0^6 = T = A_0^1 A_1^2 A_2^3 A_3^4 A_4^5 A_5^6 \quad (7)$$

Para los cálculos de la cinemática inversa, primero hay que conocer la trayectoria que debe seguir el robot, la cual conocemos. En función de la trayectoria se encuentra la orientación del eje de coordenadas 6 (la herramienta) con respecto al eje de coordenadas 0 (la base), esta orientación junto con cada punto de la posición de la trayectoria (px,py,pz), forman la matriz de transformación deseada T_d que se muestra en la formula 8.

$$T_d = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Donde los vectores n , o y a representan la orientación del eje de coordenadas 6 con respecto al eje 0 y el vector p la posición en el espacio igualmente referenciado al eje 0. Para resolver la cinemática inversa basta igualar la matriz de transformación deseada con la matriz de transformación original, $T_d=T$, y resolver para los ángulos q_1 , q_2 , q_3 , q_4 y q_5 . Como ya se había mencionado, esto se simplifica, ya que para el seguimiento de esta trayectoria, la orientación de la herramienta es constante.

Una vez se conocen los ángulos, hay que considerar la velocidad máximas de cada articulación y en función de estas velocidades, llevar el robot de su posición inicial al punto inicial de la trayectoria, cuidando las colisiones. Y de igual forma la velocidad máxima de seguimiento de la trayectoria están en función de las velocidades de las articulaciones.

En la figura 5 se muestra la tabla de las velocidades máximas del robot para cada articulación y en la figura 6 se muestra un diagrama a bloques, donde se representan los pasos que se realizaron para llegar a la simulación.

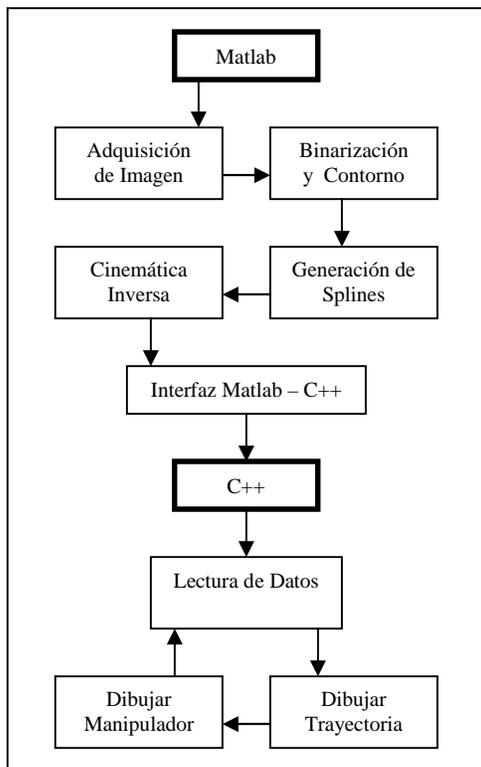


Fig. 6. Diagrama a bloques del sistema.

3 Análisis de resultados

Para algunos diseños de ingeniería y cálculos, normalmente es necesario manejar matrices. En esta propuesta se manejan matrices tanto para el procesamiento de imágenes, como para la cinemática del manipulador. Matlab cuenta con una amplia gama de funciones para la operación de matrices, además existen diferentes maneras para realizar interfaces entre Matlab y diferentes lenguajes, entre ellos C++. Utilizando estas herramientas se desarrollo la simulación en C++. Primero se realizan todos los cálculos necesarios en Matlab, hacemos la interfase con C++, para finalmente simular con las librerías de OpenGL.

Los resultados, al realizar la simulación son muy interesantes. En la figura 6 se muestra un instante en el que el robot esta siguiendo esta trayectoria, la simulación se realizo en C++ con librerías de OpenGL.

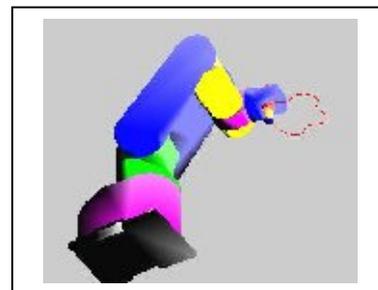


Fig. 7. Seguimiento de Trayectoria.

En la figura 7, se realiza el mismo análisis para una figura diferente, un poco más complicada y al igual que en la anterior, no hubo ningún problema para seguir la trayectoria.

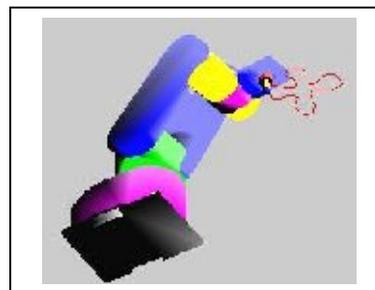


Fig. 8. Seguimiento de Trayectoria.

De esta forma se puede seguir prácticamente cualquier tipo de trayectoria en 2D, siempre y cuando

se encuentre dentro del espacio de trabajo del manipulador.

4 Conclusiones

Los avances de este trabajo, hasta el día en que se escribió el artículo, son prometedores y se espera que en la práctica, ya con el robot real, estos resultados se corroboren.

En el transcurso del artículo se pudo observar como un procesamiento de imágenes simple complementado con una herramienta matemática como las splines, fue un resultado poderoso para poder generar trayectorias suaves para un robot manipulador.

Dado que la cinemática del manipulador esta en función de las splines obtenidas del procesamiento, y dado que las splines son polinomios cúbicos y pueden ser tratados sin mucho problema, el siguiente punto que se pretende desarrollar en este trabajo, es el seguimiento de la misma trayectoria, pero esta vez orientando la herramienta en dirección normal a la tangente de cada punto de la trayectoria, lo cual modifica y hace que se complique la cinemática del manipulador.

Finalmente se desea trabajar en la intersección, captura y manipulación de objetos en movimiento, inicialmente sobre una banda transportadora (a una altura constante) y posteriormente escalarlo a cualquier punto en el espacio de trabajo del manipulador.

Referencias

- [1] Myler H, Weeks A, "*Computer Imaging Recipes in C*", Prentice-Hall, USA, 1993.
- [2] Barrientos A, Balaguer C, "*Fundamentos de Robótica*", Mc Graw Hill, Primera Edición, España, 1997.
- [3] Neider J, Davis T, "*ReedBook*", Addison Wesley, Segunda Edición, USA, 1997.
- [4] Wright R, Sweet M, "*OpenGL Super Bible*", Waite Group Press, Segunda Edición, USA, 1999.
- [5] Bay Y, "*Applications Interface Programming Using Multiple Languages*", Prentice Hall, USA, 2003.