

Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines

Trajectory generation for a manipulator robot using image processing and splines

J. A. Soto Cajiga¹, J. E. Vargas Soto², J. C. Pedraza Ortega³,

¹Estudiante de Posgrado, Centro de Ingeniería y Desarrollo Industrial. ²Director Facultad de Ingeniería, Universidad Anahuac México Sur. ³Investigador, Centro de Ingeniería y Desarrollo Industrial.

RESUMEN. En este trabajo se presenta una propuesta para generar trayectorias para un robot manipulador, específicamente el robot manipulador RV-2AJ de Mitsubishi. A partir de una imagen específica que contenga un objeto, se le aplica un procesamiento de imagen para obtener el contorno del objeto. Con ayuda de éste contorno y aplicando una interpolación por splines, se realizan los cálculos de la cinemática inversa para generar la trayectoria del contorno del objeto que debe seguir el manipulador. Se desarrolla un software de simulación en 3D para las pruebas del algoritmo y la visualización de los movimientos del manipulador. Finalmente, se realizan las pruebas reales con el robot manipulador.

Palabras clave: Splines, Cinemática, Trayectoria, OpenGL, Robot manipulador.

ABSTRACT. The purpose of this work is to generate trajectories for a manipulator robot, specifically the manipulator RV-2AJ of Mitsubishi. Beginning with a specific image that contains an object and applying image processing, the object outline is obtained. With the help of this outline and applying an interpolation by splines, the inverse kinematics is computed to generate the trajectory of the object outline that the manipulator must follow. Also, 3D simulation software is developed for the test algorithm and visualization of the manipulator movements. Finally, tests with the RV-2AJ manipulator robot are made.

Key words: Splines, Kinematics, Trajectory, OpenGL, Manipulator robot.

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, complicadas y posibles.

La idea principal de este trabajo es generar una aplicación práctica, utilizando un robot manipulador de 5 grados de libertad, procesamiento digital de imágenes y una herramienta matemática como lo son las splines. Dentro de las posibles aplicaciones se tienen por ejemplo; la digitalización de patrones en 2D para cortar, soldar, dibujar etc. 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 realizarle el procesamiento digital.
- Generar la trayectoria del objeto en curvas paramétricas, utilizando interpolación por splines.

- Con ayuda de las curvas paramétricas, calcular la cinemática inversa para generar la trayectoria en el robot manipulador.
- Realizar la simulación del seguimiento de la trayectoria, utilizando librerías de OpenGL.

2. PROCESAMIENTO DE IMAGEN

El procesamiento de imágenes es básico pero no menos importante en este trabajo. El proceso de captura de imagen y su procesamiento cuentan con un gran numero 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 optimo y funcional.

2.1 Obtención de la imagen.

Un proceso clave para que el procesamiento de imágenes no sea tan complicado, es la obtención de una imagen de buena calidad, para los fines que convenga a la aplicación.

En la obtención de la imagen para ésta aplicación, son necesarias dos cosas, que el objeto del cual se requiere digitalizar el contorno sea el único en la imagen y que el contraste entre el objeto y el fondo se muy bueno. Como se muestra en las imágenes de la figura 1.

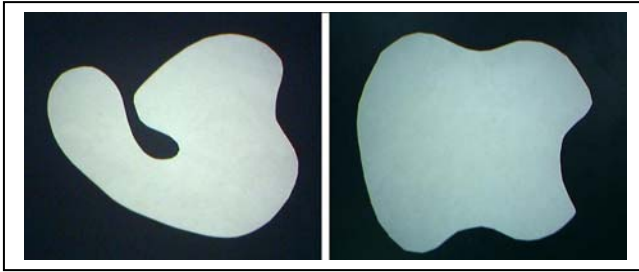


Figura 1. Imágenes de objetos a digitalizar

La obtención de éstas imágenes de la figura 1, se realizó con una cámara digital estándar, que puede ser cualquier tipo de cámara digital fotográfica. Entre mejores características fotográficas tenga la cámara y mejor sistema de iluminación se tenga, se obtiene mejor calidad en las imágenes y por consiguiente el procesamiento se simplifica.

Cabe señalar que en éste trabajo no se profundiza en el tema de visión, sólo en el procesamiento, ya que hablar de visión implica el conocimientos de diferentes equipos y técnicas tales como la óptica y la iluminación, que son campos de estudio muy amplios.

2.2 Generación de contorno

El proceso para la obtención de contorno del objeto cuenta con dos fases que a continuación se describen.

- En la primera fase, se umbraliza la imagen para resaltar aún más las características del objeto sobre el fondo, al umbralizar se obtiene una imagen binaria representada por ceros y unos que corresponden a los colores negro y blanco respectivamente [4]. En la figura 2, se observa el resultado de la umbralización.

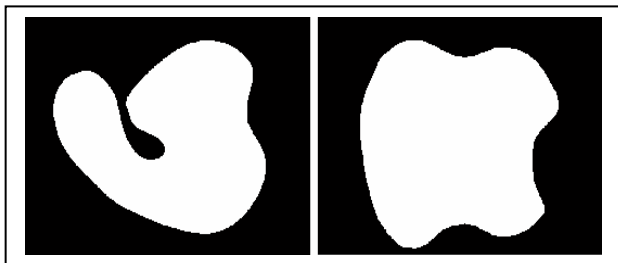


Figura 2. Imágenes de objetos binarizados

- En la segunda fase, se desarrolló un algoritmo propio para generar el seguimiento del contorno, que consiste en asignar la dirección de búsqueda (sentido horario o antihorario), hacer un barrido

para 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 del contorno y son necesarios para obtener las splines. En la figura 3 se muestran los resultados de este proceso. En la misma figura para el contorno del objeto de la izquierda se generaron 1555 pares ordenados (x,y) y para el de la derecha 1339 pares ordenados.

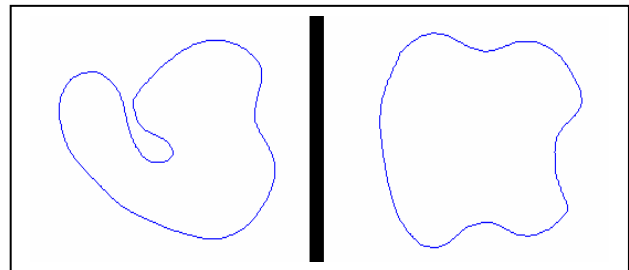


Figura 3. Imágenes de contorno de objetos

3. INTERPOLACIÓN POR SPLINES

La mayoría de los robots manipuladores realizan trayectorias curvas por la interpolación de tres puntos, esto ahorra memoria, trabajo y tiempo, ya que no es lo mismo almacenar y programar 1555 puntos que 40 puntos.

La interpolación por splines no es del todo necesaria para la programación del robot manipulador, ya que el controlador del robot cuenta con sus propios algoritmos de interpolación, sin embargo, si se desea diseñar y/o desarrollar una maquina o un robot, las splines pueden ser la solución para las interpolaciones [3], además que ayudan a suavizar el contorno del objeto y a tener una representación matemática del mismo.

Existen dos características por las que se utiliza la interpolación por splines y no otro tipo de interpolación. La primera es porque las splines que se manejan, son polinomios cúbicos, los cuales son fáciles de derivar. La segunda y más importante es porque la derivada del punto final de un polinomio cúbico es la misma que la del punto inicial del siguiente polinomio cúbico, por lo que el resultado obtenido es una serie de funciones cúbicas en un intervalo determinado, que juntas forman una sola función continua para todo el intervalo.

El proceso para obtener las splines es el siguiente:

- De los puntos de pares ordenados del contorno del objeto, se seleccionan una cantidad específica para realizar la interpolación por splines, a mayor

cantidad mejor es la aproximación al contorno real. Por ejemplo para la imagen de la figura 4 se seleccionaron 40 puntos de 1555 que se tenían.

- Con los puntos considerados se realiza la interpolación por splines parametrizando para x e y en función de t , siendo t un vector de las mismas dimensiones de x e y con incrementos constantes. Por cada punto que se considero, se obtienen 4 valores que son los coeficientes que componen un polinomio cúbico, así se obtienen tantos polinomios cúbicos como puntos fueron considerados para cada vector x e y .

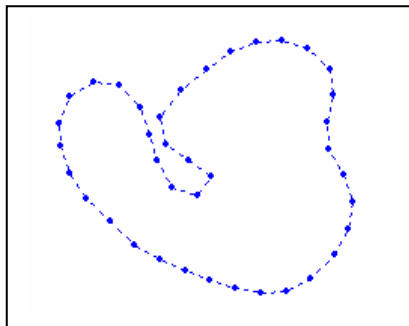


Figura 4. Puntos considerados

- Una vez que se tienen los coeficientes, se forman los polinomios y se evalúan para el intervalo $t_0 < t < t_n$ para obtener las curvas paramétricas. Ver figura 5 spline paramétrica $x(t)$ e $y(t)$.

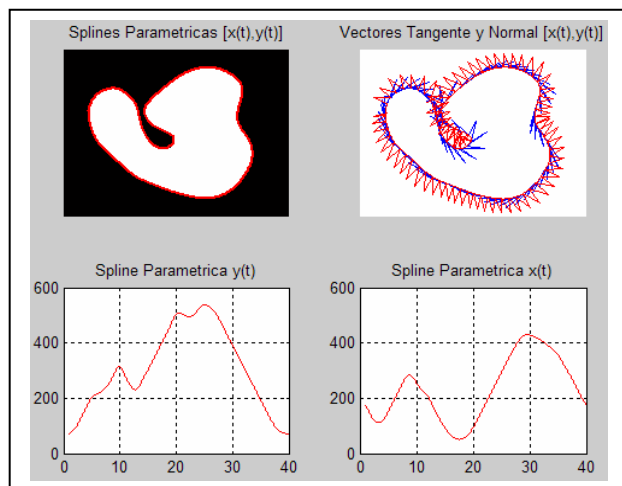


Figura 5. Resultados de la interpolación

- Finalmente se derivan los polinomios para obtener los vectores tangentes y normales a la

trayectoria del contorno, los cuales serán útiles para resolver la cinemática inversa del robot manipulador. En la parte superior derecha de la figura 5 se representan algunos de estos vectores.

En la parte superior izquierda de la figura 5, se muestra la comparación de las curva paramétricas y la figura real binarizada y en la figura 6 se muestran algunas ampliaciones de ésta imagen en puntos determinados para visualizar la aproximación.

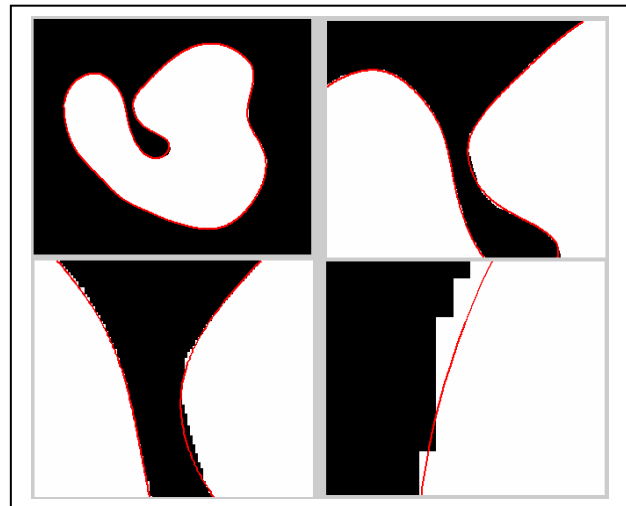


Figura 6. Resultados de la interpolación ampliados

De la figura 6 se puede ver cómo la trayectoria es suavizada y la aproximación es muy buena. Que podría ser mejor si se consideraran más puntos para la interpolación.

4. CINEMÁTICA DEL ROBOT MANIPULADOR

La cinemática del manipulador es necesaria, ya que es la base para desarrollar la simulación, poder observar cómo el robot manipulador sigue la trayectoria y evaluar si los resultados de los algoritmos son correctos, para poder pasar éstos resultados al robot manipulador real.

No es de interés en éste artículo el desarrollo completo de la cinemática del manipulador, pero si de dar a entender el procedimiento que se siguió para el cálculo de la cinemática inversa con base a los resultados de la interpolación por las splines. Por lo que a continuación se presenta éste procedimiento.

4.1 Cinemática directa

Para desarrollar la cinemática del manipulador se utilizan matrices de transformación homogénea [1], en la figura 7

se muestra una imagen del robot. El robot esta dibujado en C++ con librerías OpenGL [5][6] el cual es una replica del robot Melfa RV-2AJ de Mitsubishi.

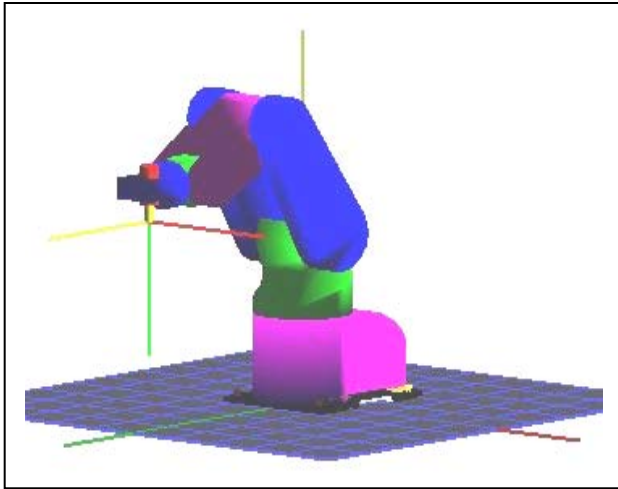


Figura 7. Configuración del Robot

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 tabla 1.

Tabla 1. Restricciones del Manipulador

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

Las matrices de transformación homogénea para la cinemática directa son 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. Los términos q_i representan el ángulo en radianes de cada eslabón, partiendo desde la base y hasta el último eslabón y los L_i representan la longitud de cada eslabón, de igual forma desde la base y hasta el último eslabón, considerando la herramienta.

$$A_0^1 = \begin{bmatrix} \cos(q_1) & 0 & \text{sen}(q_1) & 0 \\ \text{sen}(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_1^2 = \begin{bmatrix} \cos(q_2) & -\text{sen}(q_2) & 0 & L_2 \cos(q_2) \\ \text{sen}(q_2) & \cos(q_2) & 0 & L_2 \text{sen}(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_2^3 = \begin{bmatrix} \cos(q_3) & -\text{sen}(q_3) & 0 & L_3 \cos(q_3) \\ \text{sen}(q_3) & \cos(q_3) & 0 & L_3 \text{sen}(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_3^4 = \begin{bmatrix} \cos(q_4) & 0 & \text{sen}(q_4) & 0 \\ \text{sen}(q_4) & 0 & -\cos(q_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_4^5 = \begin{bmatrix} \cos(q_5) & -\text{sen}(q_5) & 0 & 0 \\ \text{sen}(q_5) & \cos(q_5) & 0 & L_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A_5^6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & L_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

La cinemática directa es el resultado de multiplicar todas las matrices de transformación homogénea, para tener una matriz que va del eje de coordenadas 0 al eje de coordenadas 6 representada como A_0^6 [1], como muestra la formula 7.

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

4.2 Cinemática inversa

Para los cálculos de la cinemática inversa, primero hay que conocer la trayectoria que debe seguir el robot, 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 $q1, q2, q3, q4$ y $q5$ [1].

La orientación de la herramienta en el espacio, esta determinada por la trayectoria que se obtuvo de las curvas paramétricas de las splines y de la ubicación de éstas en el espacio de trabajo del manipulador. El espacio de trabajo para este manipulador se muestra en la figura 8.

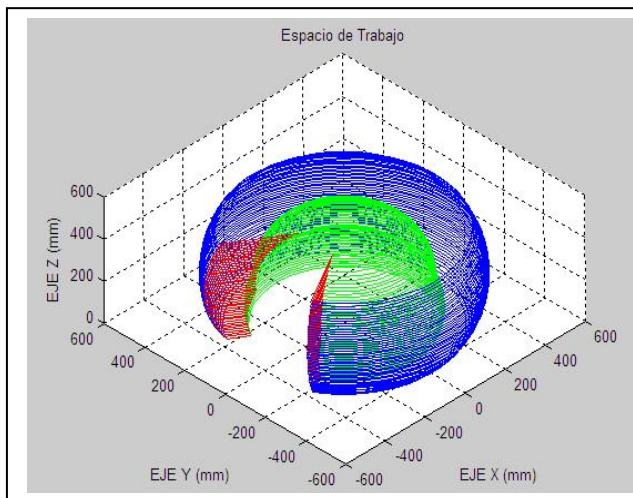


Figura 8. Espacio de trabajo del manipulador

Para ejemplificar los resultados de la cinemática inversa, se muestran dos casos.

Para el caso A, la orientación de la herramienta es constante apuntando en dirección del eje z negativo, como se muestra en la figura 9. El robot sigue la trayectoria manteniendo la herramienta en la misma dirección, por lo que los vectores n , o y a de la matriz de transformación

deseada varían muy poco, simplificando el calculo de la cinemática inversa.

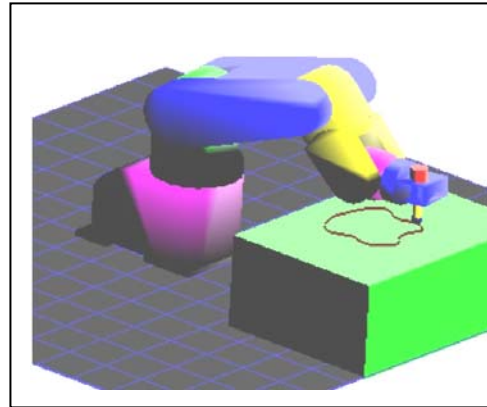


Figura 9. Trayectoria, caso A

Para el caso B, la orientación de la herramienta es en dirección normal a la trayectoria, por lo que en la matriz de transformación deseada T_d los vectores n o y a están determinados por los vectores normales de la trayectoria a seguir, que son los vectores normales que se obtuvieron con ayuda de las splines. En la figura 10 se muestra un instante en el que el manipulador está siguiendo la trayectoria con la herramienta siempre en dirección normal a la trayectoria.

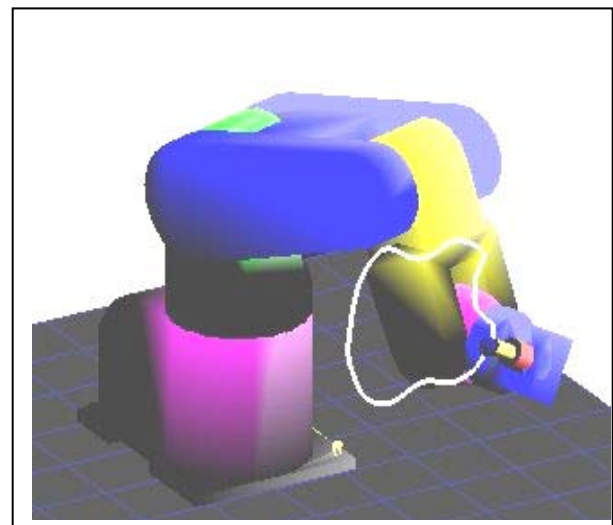


Figura 10. Trayectoria, caso B

Una vez que se resuelve la cinemática inversa, hay que considerar las restricciones del manipulador y en función de las restricciones llevar al robot de su posición inicial al punto inicial de la trayectoria, hacer que la siga y regrese a

su posición inicial, cuidando las colisiones. En la figura 11 se muestran las graficas de posiciones y velocidades para las articulaciones q_1 , q_2 , q_3 y q_4 de la trayectoria para el caso A, donde la posición de la articulación q_5 es constante por lo que no se muestra en éstas graficas.

En las graficas se pueden observar las posiciones, y las velocidades que desarrolla el manipulador durante su recorrido y con ayuda de la tabla 1, se pueden comparar las graficas con las restricciones del manipulador y así poder aumentar o disminuir las velocidades según se requiera, sin exceder los limites. También de las graficas se obtiene el tiempo aproximado que tardaría el robot en recorrer ésta trayectoria que está en función de las velocidades máximas de cada articulación.

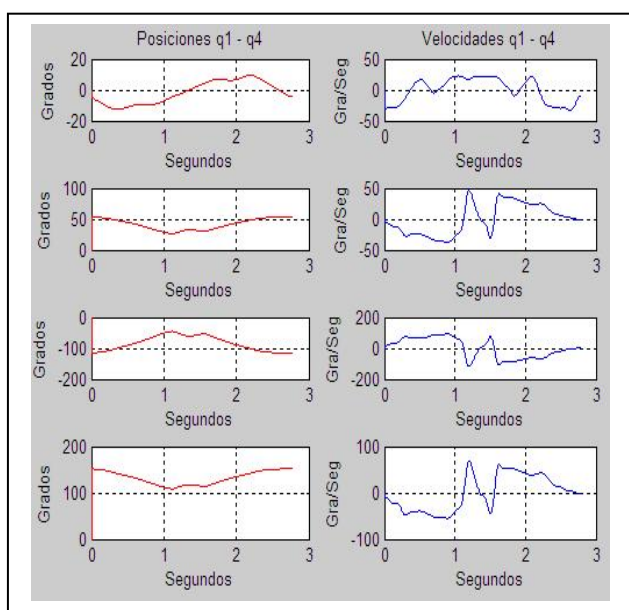


Figura 11. Graficas para la trayectoria, caso A

5. SIMULACIÓN

La simulación fue desarrollada a la par con el modelo matemático, para ir comprobando los resultados obtenidos. Fue desarrollada en dos ambientes Matlab y C++ con librerías de OpenGL, en Matlab se realizó el procesamiento de la imagen, la interpolación de las splines, los cálculos de la cinemática y la simulación grafica, éstos resultados se exportaron a C++ donde se desarrollo la simulación en 3D del sistema.

Se trabajo en dos ambientes, Matlab y C++, ya que Matlab es muy útil para resolver sistemas donde se involucra el manejo de matrices, tal como el procesamiento de imágenes o el calculo de la cinemática, por otro lado en C++ con ayuda de las librerías de OpenGL [5][6], que son librerías libres, la simulación en

3D se facilita, además que actualmente existen diferentes técnicas para realizar interfases entre diferentes ambientes de programación incluyendo C++ y Matlab [2].

El robot es una replica de un robot manipulador industrial, RV-2AJ de Mitsubishi, fue dibujado en Autocad, donde se obtuvieron las primitivas de cada eslabón, se pasaron a C++ en formato de vértices y con ayuda de las librerías de OpenGL se dibujo el manipulador y se le dio movimiento.

En la figura 12 se muestra un diagrama a bloques, donde se representan los pasos que se realizaron para llegar a la simulación.

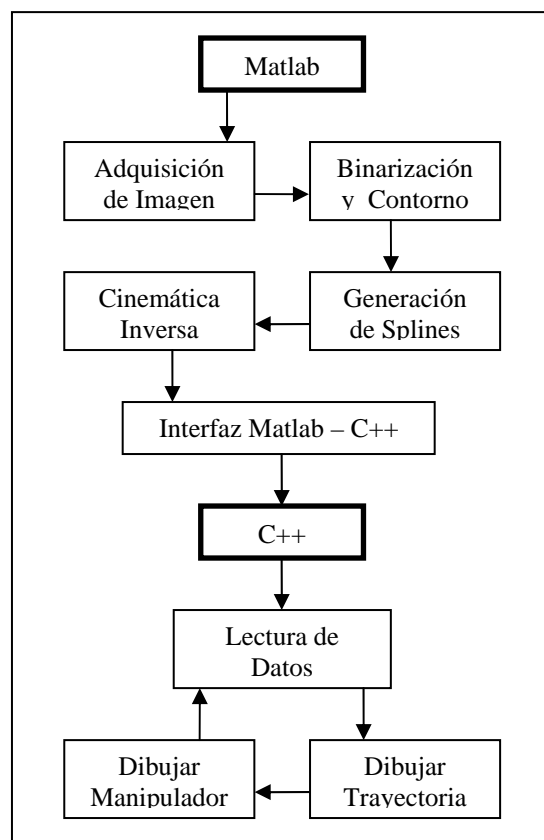


Figura 12. Diagrama a bloques para la simulación

6. CONCLUSIONES

En el transcurso del trabajo se pudo observar cómo 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 está en función de las splines obtenidas del procesamiento de la imagen y dado que las splines son polinomios cúbicos y pueden ser tratados sin mucho problema, se pudieron obtener la

derivada y el vector normal de la trayectoria de manera sencilla, los cuales simplificaron los cálculos de la cinemática inversa y así fue posible realizar una simulación muy parecida a la realidad.

En la figura 13, se muestra una foto del robot real cuando esta siguiendo una trayectoria. Cabe señalar que para implementar la trayectoria en el robot sólo se cargan los puntos considerados del contorno y el robot interpola para éstos puntos con su propio algoritmo, por lo que los algoritmos de interpolación por spline solo sirven para el calculo de la cinemática inversa, la simulación y por supuesto para cuestiones de diseño.

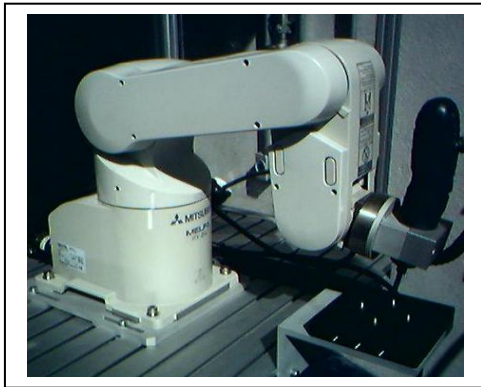


Figura 13. Foto del robot siguiendo una trayectoria

En este trabajo la única limitante fue el robot manipulador, ya que es un robot industrial básico y son limitadas las aplicaciones que se pueden desarrollar, tal como modificar la posición en tiempo real.

7. TRABAJO FUTURO

En un futuro se pretende trabajar en la predicción de la trayectoria para un robot manipulador, en el cual se pueda modificar la trayectoria en tiempo real con algún sistema de retroalimentación, ya sea visión o algún sensor especial.

Por otro lado el desarrollo de la simulación en 3D ha sido de gran interés, debido a que para poder programar un robot industrial normalmente es necesario el sistema físico y el software de programación del mismo, que normalmente son de un costo no muy accesible, especialmente para instituciones de educación pública media y superior. Por lo que se pretende trabajar en la simulación en 3D de diferentes configuraciones de robots industriales, utilizando comandos de programación lo más parecidos a los reales, de ésta forma los estudiantes podrán tener un entrenamiento y acercamiento a la programación de robots industriales, sin necesidad del robot físico.

8. REFERENCIAS

- [1] Barrientos A. y Balaguer C., 1997, "*Fundamentos de Robótica*", Mc Graw Hill, Primera Edición, España.
- [2] Bay Y., 2003, "*Applications Interface Programming Using Multiple Languages*", Prentice Hall, USA.
- [3] Burden R. y Faires J., 2002, "*Análisis Numérico*", Thomson Learning, Séptima Edición, USA.
- [4] Myler H. y Weeks A., 1993, "*Computer Imaging Recipes in C*", Prentice-Hall, USA.
- [5] Neider J. y Davis T., 1997, "*ReedBook*", Addison Wesley, Segunda Edición, USA.
- [6] Wright R. y Sweet M., 1999, "*OpenGL Super Bible*", Waite Group Press, Segunda Edición, USA.