

CONSTRUCCIÓN DE UN GENERADOR DE TRAYECTORIAS PARA SU UTILIZACIÓN EN LA SIMULACIÓN DINÁMICA DE UN ROBOT FLEXIBLE DE SEIS GRADOS DE LIBERTAD

A.García, V.Feliu, J.A.Somolinos, P.Fraile
E.T.S.I.I.-U.C.L.M.
Avda. Camilo J. Cela, 3
13071 CIUDAD REAL
aghiguer@ind-cr.uclm.es

Resumen

Los Robots Industriales cuentan con unidades de programación muy elaboradas que permiten la programación avanzada del robot y facilitan el diseño de trayectorias suaves. Los robots con estructura flexibles cuyo comportamiento se encuentra en fase de estudio por numerosos investigadores se encuentran con el problema añadido de no contar con unidades de programación diseñadas teniendo en cuenta sus especiales características. De esta forma la realización de cualquier operación sencilla con un robot flexible requiere de unos engorrosos cálculos previos.

Por otra parte, y debido a lo novedoso de las técnicas de control que se pretenden ensayar en estos robots, es necesario en muchos casos la realización de ensayos previos en herramientas simuladoras. Estas herramientas presentan el mismo problema de la carencia de utilidades destinadas a facilitar el trabajo de la generación de trayectorias que tengan en cuenta las particularidades del robot.

Este trabajo se centra en la realización de una herramienta que va a permitir solventar todos estos problemas acelerando el avance de las investigaciones sobre el control de robots flexibles.

Palabras Clave: Brazos flexibles, control de robots, generación de trayectorias.

1 INTRODUCCIÓN

A medida que la robótica se va haciendo más habitual en la industria, aparecen nuevas aplicaciones en las que resulta cada vez más necesaria la interacción física del robot con su entorno. El intercambio de esfuerzos entre el robot y su entorno conduce a un replanteamiento de los métodos de control [9]. Los tradicionales controles de posición, en robots que normalmente se componen de

eslabones pesados, ocasionan grandes esfuerzos en el instante de la colisión debido a una energía de impacto que, frecuentemente, deberá absorber el entorno.

Los eslabones que componen el robot son pesados porque deben presentar una rigidez que garantice la precisión en la posición del extremo. Recientemente están apareciendo nuevas técnicas de control que posibilitan la utilización de eslabones de robot más ligeros y flexibles ([1] [3] [4] [5] [8] [10] [11]).

Se trata de técnicas complejas que deben ser ensayadas en potentes entornos de simulación que permiten establecer el comportamiento dinámico de la estructura del control ante diferentes condiciones de funcionamiento. Estos entornos de simulación dinámica permiten la introducción de diferentes estructuras construidas con materiales cuyas características mecánicas pueden ser también especificadas, además es posible establecer articulaciones y las fuerzas y pares que actúan sobre la estructura así como los bucles de control y las funciones de transferencia de los reguladores que se pretende analizar.

Sin embargo estos entornos de simulación dinámica no están preparados para el análisis de trayectorias complejas como las que puede describir el extremo de un robot y eso limita en gran medida el análisis de los nuevos sistemas de control de robots flexibles.

Un posible método de trabajo para el análisis de estos sistemas es el compuesto de las siguientes fases:

1. Generación de la estructura con sus características mecánicas (flexibilidad, distribución de masas...) en el entorno de análisis por elementos finitos *Ansys*[®].
2. Importación de esta estructura en el entorno de simulación dinámica *Adams*[®] donde se introducen las articulaciones y el efecto de los motores y fuerzas externas (gravedad, posibles colisiones con el entorno...).

3. Desarrollo de un método que permita la generación de trayectorias complejas proporcionando un fichero de puntos intermedios que puede ser introducido en *Adams*[®] o en el controlador del prototipo.
4. Realización de simulaciones en *Adams*[®].
5. Ensayos sobre un prototipo real (figura 1).

El presente artículo se centra en el tercero de estos puntos pudiéndose encontrar los resultados obtenidos para todos los demás en las referencias [4][7][10].



Figura 1: Prototipo de robot flexible

2 ESPECIFICACIONES DE LAS TRAYECTORIAS

Es necesario que el robot siga una trayectoria determinada que permita evaluar la precisión con que funcionan los bucles de control de la posición, tanto en régimen transitorio como en régimen permanente o estacionario. Esta evaluación se realiza midiendo la desviación existente entre la señal de referencia dada a los motores y el giro producido en cada uno de ellos. Cuanto menor sea esta desviación, más se ajustará el movimiento del extremo del robot al que ha sido definido por la trayectoria de referencia.

Las trayectorias deben satisfacer una serie de requisitos:

- Deben hacer pasar al extremo por los puntos definidos por el usuario.

- Deben ser funciones temporales, al menos derivables una vez para que el movimiento sea suave en todo momento.
- Deben satisfacer los requisitos cinemáticos y dinámicos que imponen los motores, como son: no sobrepasar la velocidad o aceleración máximas, el par admisible, etc.

Satisfacer todas estas condiciones a la hora de diseñar una trayectoria analíticamente implica, en el mejor de los casos, resolver pesadas ecuaciones para satisfacer las condiciones de contorno de continuidad y derivabilidad para cada una de las articulaciones. Si además se desea que la trayectoria sea cartesiana, entonces el problema se torna irresoluble por métodos manuales, ya que es necesario resolver la transformada cinemática inversa en cada uno de los instantes que componen la discretización. Por esta razón se hace necesario desarrollar una aplicación informática que genere las trayectorias, reduciendo el trabajo a introducir los puntos de control y algunas otras opciones adicionales. Además esta aplicación permite introducir modificaciones sin más que cambiar los parámetros que se desee.

La ventaja adicional de utilizar una herramienta específica es la automatización que supone en la transmisión de información hacia el entorno *Adams*[®], ya que éste utiliza un formato de ficheros muy específico, como ya se verá a lo largo de este artículo.

En el siguiente apartado se realiza una clasificación de los tipos de trayectorias empleados y sus principales características. Una mayor información puede encontrarse en [6].

3 DESCRIPCIÓN DEL SISTEMA DE GENERACIÓN DE TRAYECTORIAS

El programa ha sido desarrollado sobre un entorno de ventanas y utilizando la herramienta de cálculo matemático *Matlab*[®]. Por lo tanto la plataforma desde la que se ejecute debe satisfacer estos dos requisitos: funcionamiento bajo un sistema operativo basado en ventanas (*Windows*[®] 95/98, *Unix* o cualquier otro) y la inclusión en el sistema de una versión de *Matlab*[®] suficientemente actualizada. El *software* ha sido probado en la versión *Matlab*[®] 5.3, por lo que no se puede garantizar un correcto funcionamiento en versiones anteriores. En concreto el problema puede surgir en el formato de salida de los ficheros creados para el intercambio de información, como se explicará más adelante.

En cuanto a los requisitos de *hardware*, no existe ninguna restricción, y el programa funcionará correctamente en cualquier equipo que soporte el *software* antes mencionado.

3.1 FUNCIONAMIENTO DEL SISTEMA

En este apartado se va a analizar el funcionamiento interno del programa. A partir de los datos introducidos por el usuario, la aplicación genera un fichero apto para su lectura desde *Adams*[®]. Sea una trayectoria que debe hacer pasar al extremo por N puntos en el espacio; los datos relativos a la posición y duración del movimiento que se deben introducir son:

- Posición y orientación del extremo (seis coordenadas) en cada uno de los puntos (en milímetros y radianes, respectivamente). Puesto que el archivo de salida puede ser utilizado tanto por el robot de tres grados de libertad como por el de seis, la información relativa al giro sólo será necesaria en este último caso.
- Duración del recorrido entre un punto y el siguiente (en segundos).
- Discretización temporal de cada uno de los intervalos.
- Pausa que realiza el robot al llegar al punto i antes de comenzar a desplazarse hacia el punto $i+1$. El objeto de esta pausa es comprobar el comportamiento del robot en régimen estacionario. Así se pueden medir las vibraciones que se producen en la estructura cuando debería estar en reposo.

Todas estas variables se definen en cada posición y pueden variar de un intervalo al siguiente. En cuanto a cómo se realiza el movimiento entre los puntos, los datos a especificar son los siguientes:

- Tipo de trayectoria. Aquí es necesario distinguir entre dos clasificaciones distintas, atendiendo a la ruta a seguir:
 - Cartesiana. Se discretiza el camino que debe recorrer el robot en línea recta entre ambos puntos, y se halla la orientación que debe adquirir cada una de las articulaciones en todos los puntos de la discretización, mediante la transformada cinemática inversa. Utilizando esta técnica cabe la posibilidad de que se haga pasar al robot por un punto singular.
 - Articular. El movimiento entre un punto y el siguiente se realiza hallando el incremento para cada una de las articulaciones, y generando un movimiento independiente para cada una de ellas. La ventaja de este movimiento frente al anterior es que se evitan siempre los puntos singulares; como

contrapartida se pierde el control sobre la posición en los puntos intermedios.

- Atendiendo a la cinemática del movimiento entre cada uno de los puntos:

- Cúbica. La trayectoria entre el punto i y el $i+1$ se realiza siguiendo un polinomio de grado tres dependiente del tiempo para cada articulación. Se han elegido estas trayectorias porque, pese a no requerir cálculos complicados, proporciona movimientos de robot lo suficientemente suaves (puesto que presenta continuidad en la posición, en la velocidad y en la aceleración). Las ecuaciones que rigen este movimiento son las siguientes:

$$q(t) = q_i + \frac{3}{\Delta t_i^2}(q_{i+1} - q_i)t^2 - \frac{2}{\Delta t_i^3}(q_{i+1} - q_i)t^3 \quad (1)$$

$$t_i < t < t_{i+1}, \quad i = 1 \dots N - 1$$

donde q es el giro de una coordenada articular genérica, y $\Delta t_i = t_{i+1} - t_i$.

- Aceleración máxima. El recorrido se divide en dos mitades, una de aceleración máxima y otra de deceleración o aceleración negativa máxima. De esta forma el resultado es un movimiento compuesto por dos parábolas cuyo punto de unión es continuo y derivable. Las ecuaciones para cada tramo de la trayectoria son las siguientes:

$$q(t) = \begin{cases} a(t-t_i)^2 + q_i, & t_i < t < t_i + \frac{\Delta t_i}{2} \\ -a(t-t_{i+1})^2 + q_{i+1}, & t_i + \frac{\Delta t_i}{2} < t < t_{i+1} \end{cases} \quad (2)$$

$$i = 1 \dots N - 1$$

donde, para satisfacer la continuidad en el punto medio del intervalo, se debe cumplir que

$$a = 2 \frac{(q_i + q_{i+1})}{(t_i + t_{i+1})^2} \quad (3)$$

Este valor debe ser menor en cada articulación que la aceleración máxima que imponen como restricción las especificaciones de los motores. Si no fuera así, la trayectoria no sería viable y habría que volver a definir la duración del movimiento, o hacer el camino más corto.

- Velocidad constante. La mayor parte del recorrido se realiza a velocidad constante. Ésta viene dada por el intervalo de tiempo empleado en el movimiento y por la aceleración máxima permitida (la máxima que da el motor de la articulación). El camino entre la posición i y la $i+1$

se divide para cada articulación en tres tramos: un primer tramo de aceleración máxima hasta alcanzar la velocidad requerida; un segundo a velocidad constante y un tercero de deceleración máxima hasta llegar a la posición final. Las ecuaciones que rigen el movimiento son:

$$q(t) = \begin{cases} a(t-t_i)^2 + q_i, & t_i < t < t_b + t_i \\ v \left(t - (t_i + \frac{t_b}{2}) \right) + q_i, & t_b < t < t_{i+1} - t_b \\ q_i + q_{i+1} - a(t - (t_i + t_{i+1}))^2, & t_{i+1} - t_b < t < t_{i+1} \end{cases} \quad (4)$$

$i = 1..N-1$

donde v , velocidad constante y t_b , tiempo de aceleración, se definen respectivamente como

$$v = \frac{q_{i+1} - 2q_b}{t_{i+1} - 2t_b} \quad (5)$$

$$t_b = \frac{a\Delta t_i - \sqrt{(a\Delta t_i)^2 - 2a(q_{i+1} - q_i)}}{2a} \quad (6)$$

Estas dos clasificaciones son independientes entre sí, es decir, una trayectoria puede ser articular o cartesiana, independientemente de que el movimiento se haga a velocidad constante, a aceleración máxima o siguiendo una cúbica. En cuanto a estas últimas, hay que decir que, si bien son C^2 en todo el intervalo (al contrario que los otros dos tipos de movimiento), lo cual supone una ventaja, su empleo está poco extendido debido a que no es fácil localizar los máximos de velocidad y aceleración.

Todas las trayectorias generadas son isocronas, es decir, todas las articulaciones llegan a la posición final en el mismo instante.

La notación empleada es la más genérica posible, y admite en cada posición inicial i un tiempo y un giro distintos de cero, para representar el hecho de que la posición dada puede ser la resultante de un movimiento anterior desde otra posición $i-1$.

Estos últimos parámetros se definen **para toda la trayectoria**, y por lo tanto en un mismo archivo no pueden alternar distintos tipos de movimiento. Los valores que el programa considera son los activos en el momento de generar la trayectoria, por lo que los cambios efectuados en instantes anteriores no tendrán efecto. En versiones posteriores de la aplicación se puede contemplar la posibilidad de alternar distintos tipos de trayectoria, para mejorar la eficiencia del robot (por ejemplo, aproximación del extremo desde el origen hasta un punto de interés mediante trayectoria articular; seguimiento de un objetivo

según una trayectoria cartesiana y regreso a la posición de inicio con una trayectoria articular).

3.2 DESCRIPCIÓN DEL ENTORNO DE PROGRAMACIÓN

La aplicación se basa en un entorno de ventanas generadas a partir de un código de *Matlab*[®]. Esto presenta diversos inconvenientes respecto a la utilización de lenguajes de programación “visuales”, entre los que destaca la lentitud con la que se generan los objetos y se lee la información. Esto es debido a que el código no está compilado, sino que se interpreta en el momento de la ejecución. Como ventajas del uso de la herramienta *Matlab*[®] están el aprovechamiento de la potencia de sus rutinas de cálculo numérico, la facilidad de modificación del código, la sencillez a la hora de obtener representaciones gráficas o lo portable del entorno, al no depender del sistema operativo empleado.

El programa está centrado en una ventana principal, desde la que se modifican las diversas opciones de las trayectorias y se accede a los resultados. Esta ventana principal se muestra en la figura 2.

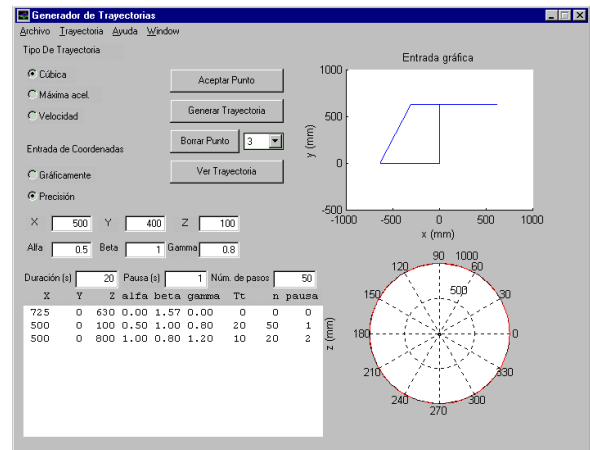


Figura 2: Aspecto de la ventana principal del generador de trayectorias

En ella se observan los botones y cuadros de texto relativos a las distintas opciones comentadas en el apartado anterior, excepto la elección entre trayectoria articular o cartesiana, que se realiza desde el menú *Trayectoria*. También se pueden observar dos figuras en la parte derecha de la ventana que representan una vista en alzado y planta del robot. El fin de este esquema se explica a continuación. A la hora de introducir un nuevo punto en la trayectoria, el programa permite elegir entre una entrada de precisión o una entrada gráfica de dicho punto. Si la entrada es de precisión, las cajas de texto correspondientes a las coordenadas X,Y,Z están habilitadas y su valor será almacenado cuando se pulse el botón *Aceptar Punto*. Si se ha optado por una entrada gráfica, entonces las cajas se deshabilitan

(color gris) y el usuario debe introducir las coordenadas X,Y en la figura de arriba y la coordenada Z en la de abajo al pulsar el botón *Aceptar Punto*.

En cualquier caso se habrá de introducir la orientación deseada del extremo en el punto indicado. Para ello habrán de utilizarse las cajas de texto provistas al efecto, tal como se puede apreciar en la figura 2.

3.3 DESCRIPCIÓN DEL FUNCIONAMIENTO INTERNO

Una vez introducidos los datos definitivos, después de haber realizado las modificaciones necesarias, se debe generar la trayectoria. Para ello se pulsa el botón *Generar Trayectoria*, lo cual ejecuta la subrutina *fintray.m*. En ella se leen las variables que indican las distintas opciones elegidas, y de acuerdo con ellas se decide el tipo de trayectoria a generar. En primer lugar el programa genera un vector de tiempo auxiliar, compuesto por $n(i)$ divisiones en el tramo i (desde la posición $i-1$ a la i) de duración T_i segundos, seguido cada tramo por una división adicional de dimensión $pausa(i)$. Este vector se puede representar matemáticamente como se muestra a continuación. Sea

$$h_i = \frac{1}{n_i}, \quad 1 \leq i \leq N-1 \quad (7)$$

el paso de tiempo para el intervalo i , definido como arriba; entonces el vector se crea de la siguiente forma:

$$t = 0, k_i T_i h_i, P_i \quad \begin{matrix} k_i = 1, \dots, n_i \\ i = 1, \dots, N \end{matrix} \quad (8)$$

donde T_i es la duración y P_i es el tiempo de pausa del intervalo i . De esta forma se crea un solo vector que contiene toda la información temporal introducida por el usuario, para la totalidad de la duración del movimiento programado. Cada intervalo tiene las divisiones suficientes y no es preciso introducir más de las necesarias (los tiempo de pausa se representan con un solo elemento del vector), ahorrando tiempo de cálculo. Es necesario observar que el vector resultante es equiespaciado en cada tramo.

Posteriormente se reconoce la cinemática del movimiento (velocidad constante, aceleración máxima o siguiendo una cúbica). Las funciones descritas en el apartado 3.1 no se aplican directamente a las coordenadas de las articulaciones, lo cual implicaría la necesidad primero de resolver el problema y luego transformar las coordenadas resultantes. En lugar de hacer esto, la función que

describe la trayectoria se aplica al espaciado fijo del vector auxiliar de tiempo, para dar lugar a un nuevo vector de tiempo de espaciado variable. Para ello se sustituye la variable articular $q(t)$ por una nueva variable temporal, $\tau(t)$. Las ecuaciones no se modifican de ninguna manera, salvo por la consideración de que en los tiempos de pausa no hay transformación, es decir

$$\tau(t) = t, \quad t \in P_i, \quad i = 1, \dots, N-1 \quad (9)$$

De esta forma, velocidades mayores corresponden con incrementos más cortos, y aceleraciones (o deceleraciones), con pasos progresivamente decrecientes (o crecientes). Con este método las coordenadas articulares se calculan siempre con la misma rutina, en la que únicamente varían los incrementos de tiempo.

Por último se distingue entre trayectoria articular o cartesiana. En el caso de la primera, se obtienen las coordenadas articulares realizando la transformación cinemática inversa únicamente en las posiciones introducidas por el usuario. El resto de posiciones intermedias se calculan por interpolación. Si la trayectoria es cartesiana, primero hay que obtener la línea recta por la que debe pasar el extremo (la función de tiempo ya ha sido asignada con anterioridad) y luego resolver la cinemática inversa en cada posición.

Para formar la matriz de transformación homogénea en cada posición introducida (y en las intermedias en el caso de trayectorias cartesianas) a partir de la posición y orientación del extremo, se emplea la función *transf.m* que utiliza la composición de rotaciones *Roll, Pitch, Yaw*.

El programa, durante su ejecución, genera gran cantidad de variables, la mayoría de las cuales no tienen ninguna utilidad para el usuario. Para que no supongan un estorbo, la aplicación salva el espacio de trabajo existente antes de ejecutarse en el archivo *tmp.mat*; cuando se decide cerrar la aplicación, se borran todas las variables, se carga el espacio de trabajo *tmp.mat* y se elimina dicho archivo del directorio. De esta forma, la información existente en el equipo al finalizar la sesión es exactamente la misma que la que había antes de iniciarse.

4 UTILIZACIÓN DEL SISTEMA

En este apartado se dan las referencias básicas para el correcto uso del programa.

- **Introducción de datos.** Las coordenadas del extremo se introducen en milímetros, y los ángulos en radianes. Los datos introducidos serán aceptados cuando se pulse el botón *Aceptar Punto*. Mientras tanto, se pueden modificar como se desee. Si se opta por realizar la entrada *gráficamente*, en el momento de aceptar el cursor se transforma en un puntero para que se introduzca la posición del extremo pinchando en las gráficas del robot. Primero se han de seleccionar las coordenadas X,Y en la figura de arriba (alzado) y luego la Z en la de abajo (planta). Si se hace al revés, aparecerá un mensaje de aviso y se solicitarán de nuevo las coordenadas. Los ángulos de orientación del extremo habrán de ser introducidos en las cajas de texto correspondientes. Una vez que el punto ha sido introducido, aparece en el cuadro de texto inferior izquierdo. Los datos se almacenan hasta el momento en que se genere la trayectoria.

- **Borrado de datos.** Si se pulsa el botón *Borrar Punto* se borrará el punto que aparece en el menú adyacente, y el resto de puntos serán numerados consecutivamente. El punto número uno u origen no se puede borrar, ya que cualquier trayectoria debe partir necesariamente de esa posición. Si se desea crear una trayectoria que parta de una posición distinta, se puede introducir un camino extra desde el origen establecido hasta el origen que se desea.

- **Tipos de trayectoria.** La elección de la cinemática se hace sobre la propia ventana, en los botones superiores izquierdos. La elección del camino se hace desde el menú trayectoria. Cualquier cambio en estas opciones únicamente se hará efectivo cuando se genere la trayectoria. No es posible generar trayectorias que incluyan varias de estas opciones en distintos tramos.

- **Generar trayectorias.** Los resultados se calculan al pulsar el botón *Generar Trayectoria*, a partir de los datos introducidos hasta ese momento. Cualquier modificación posterior se hará efectiva si se pulsa de nuevo el botón. Los datos y las opciones se pueden modificar tantas veces como se desee, y se incluirán en una nueva generación de trayectoria. Únicamente se almacena en memoria una trayectoria.

- **Visualizar los resultados.** Si se desean ver las coordenadas articulares obtenidas, o la posición del extremo, o ambas cosas, se puede pulsar el botón *Visualizar Trayectoria* (por supuesto también se puede hacer manualmente en la ventana de comandos de *Matlab*[®]). Aparecerá la última trayectoria generada.

- **Grabar datos.** Cuando se haya obtenido una trayectoria definitiva, se puede grabar en un

formato que el *Adams*[®] puede interpretar. Para ello se debe elegir la opción *guardar trayectoria* dentro del menú *archivo*. Es importante no modificar la extensión del archivo (*.req). Un ejemplo del formato del archivo de intercambio se muestra en el próximo apartado.

- **Salir.** Si el usuario ha modificado manualmente el espacio de trabajo durante la ejecución del generador de trayectorias, se recomienda guardar aquel en un archivo antes de cerrar la ventana principal ya que, al cerrarse, el programa elimina todas las variables residentes en memoria para restablecer el espacio de trabajo existente antes de la ejecución. La razón de esta operación es eliminar la gran cantidad de variables de control que quedan almacenadas en memoria y que pueden confundir al usuario, e incluso borrar algún dato que pudiera existir en alguna variable del mismo nombre.

4.1 EJEMPLO DE UTILIZACIÓN DEL GENERADOR

En este apartado se va a desarrollar un ejemplo de utilización del programa, para ilustrar mejor su funcionamiento. Se quiere generar una trayectoria que pase por los puntos mostrados en la figura 3, en los tiempos dados y con las opciones especificadas. El tipo de trayectoria elegido es articular.

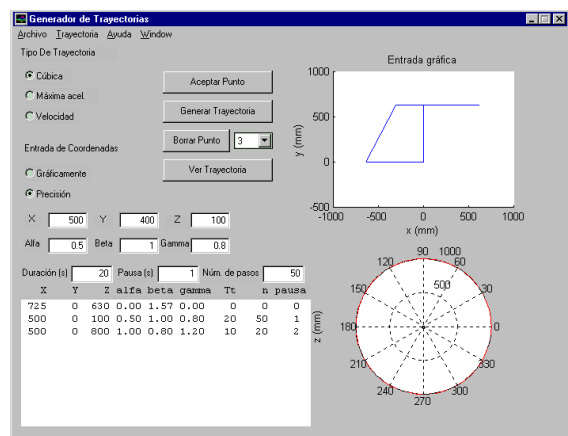


Figura 3: Datos introducidos para el ejemplo

De acuerdo con estos datos, se quiere que entre cada una de las posiciones fijadas como condiciones de contorno, la trayectoria seguida por cada articulación siga un polinomio cúbico. Como ya se ha explicado, esto se consigue modificando el intervalo de tiempo entre dos posiciones consecutivas dentro de la trayectoria. Para ilustrar mejor el ejemplo, en la figura 4 se ha representado las tres posibles transformaciones del vector equiespaciado, para generar cada uno de los tres posibles movimientos.

Las coordenadas obtenidas como resultado de la ejecución del programa son las que se muestran en la

figura 5. En ella se observan las trayectorias cúbicas que describen cada una de las articulaciones, como consecuencia de la transformación realizada sobre el vector de tiempo. También se observan los intervalos de pausa después de cada movimiento, en los que la posición del robot no varía.

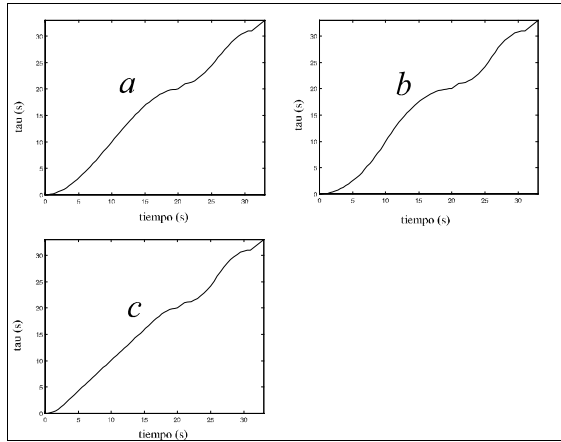


Figura 4: Transformaciones para obtener un vector tiempo variable; a) polinomio cúbico; b) máxima aceleración (parabólica); c) velocidad constante

Por último resta crear el archivo con los resultados, que posteriormente será exportado al entorno *Adams*[®] para su utilización en la generación de señales de referencia para los motores. El sistema es, por tanto, capaz de generar el estricto formato numérico de escritura necesario para esta operación.

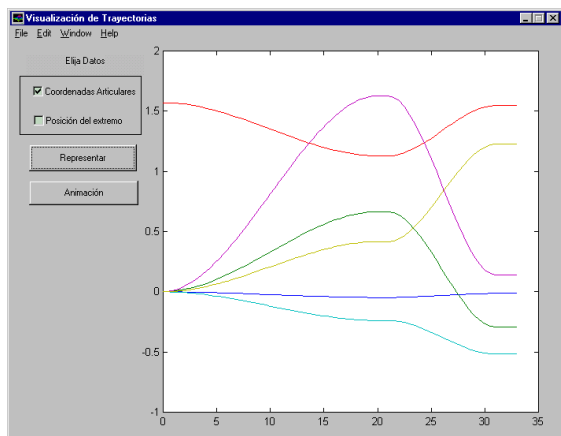


Figura 5: Coordenadas articulares obtenidas mediante el generador de trayectorias (ejemplo)

4.2 EJEMPLO DE SIMULACIÓN EN ADAMS

En este apartado se va a realizar una simulación referida al prototipo de la figura 1, que únicamente tiene instalados los tres primeros grados de libertad. El aspecto gráfico que presenta el modelo realizado en *Adams*[®] es el que se muestra en la figura 6. En él se observan las diferentes barras que componen la estructura, cada una de ellas compuesta por tres

elementos (cuatro nodos) en su parte flexible. También se representan los tres motores situados en la base, y la masa del extremo, que tiene un valor de 1.815 kg simulando el eventual peso de la “muñeca”.

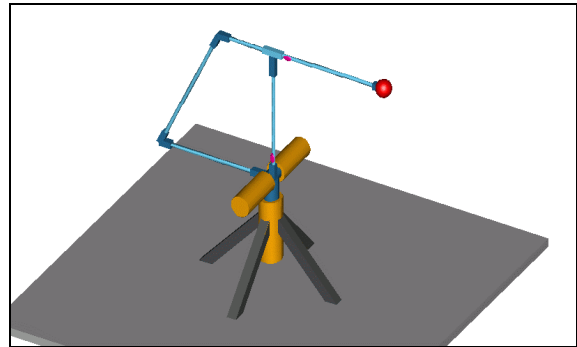


Figura 6: modelo del robot de tres grados de libertad

El ejemplo que se va a desarrollar comprende el movimiento del robot a través de una trayectoria cartesiana rectilínea cuyos extremos están definidos en la Tabla 1, y que pasa por los puntos que se facilitan en la Tabla 2. A partir de estos once puntos se generan las trayectorias articulares continuas que se observan en la figura 7. Se ha generado una trayectoria de pocos puntos para que se pueda apreciar la interpolación realizada entre ellos.

	X (m)	Y(m)	Z(m)
Posición inicial	0.64	0.63	0
Posición final	0	0.40	0.50

Tabla 1: puntos extremos de la trayectoria ejemplo.

θ_{m1} (rad/s)	θ_{m2} (rad/s)	θ_{m3} (rad/s)
0.00000E+00	-4.58367E-05	4.58367E-05
2.30412E-02	-9.32955E-03	-1.81468E-02
9.25916E-02	-3.39871E-02	-6.49734E-02
2.16940E-01	-6.83640E-02	-1.26678E-01
4.09947E-01	-1.02658E-01	-1.88817E-01
6.74741E-01	-1.18375E-01	-2.38788E-01
9.74295E-01	-9.51045E-02	-2.68382E-01
1.23913E+00	-3.35539E-02	-2.76228E-01
1.42671E+00	3.97199E-02	2.68427E-01
1.53480E+00	9.59502E-02	-2.55920E-01
1.57080E+00	1.17273E-01	-2.49883E-01

Tabla 2: trayectoria ejemplo.

Las referencias de los motores dos y tres se representan en la misma gráfica, para observar que la composición de movimientos del pantógrafo puede producir en el extremo una trayectoria lineal, aunque aquellas no lo sean.

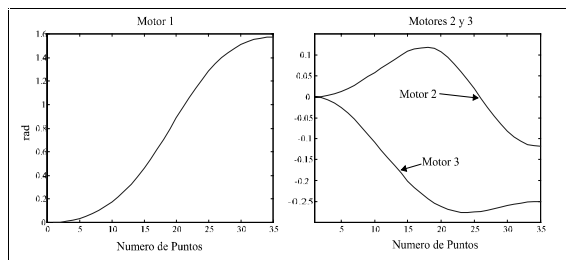


Figura 7: trayectorias articulares generadas a partir de los datos.

En la primera gráfica se observa como no existe oscilación al llegar al régimen estacionario. En el eje de abscisas se ha representado el número de puntos, en lugar del tiempo. De esta forma la misma señal de referencia se puede aplicar a diferentes escalas temporales, observando la distinta respuesta del sistema en cada caso. El movimiento que estos giros en los motores producirían en el extremo del manipulador sería una línea recta exacta, suponiendo que el robot es perfectamente rígido, y que su dinámica no tiene efectos en el comportamiento global. Obviamente esto no ocurre, porque la dinámica del robot influye de manera importante, al ser ésta fuertemente oscilante. Por ello, el movimiento del extremo se alejará de la línea recta ideal, y mayor será esta desviación cuanto más rápido sea el movimiento. Este es el efecto que se va a estudiar a continuación.

Se ha repetido el ensayo varias veces, disminuyendo la velocidad de los motores progresivamente, lo que se ha conseguido escalando la trayectoria en el tiempo, para observar los efectos de la vibración en el movimiento del extremo, y en el funcionamiento de los bucles de control de la posición. La duración de las simulaciones se muestra en la Tabla 3. En ella se muestra únicamente el tiempo que tarda la referencia en alcanzar la posición final, sin contar el tiempo que tarda el extremo en alcanzar el régimen permanente, para lo cual se han prolongado los ensayos un 40% de la duración indicada.

Ensayo	Duración
1	5
2	10
3	25
4	50

Tabla 3: duración de las simulaciones hasta alcanzar la posición final.

4.2.1 Respuesta articular

Para observar la influencia de la dinámica de la estructura en su comportamiento al realizar las simulaciones, se pueden comparar los resultados obtenidos en cada uno de los ensayos realizados, superponiendo las respuestas en una escala independiente de la duración del ensayo. Esto es lo que se muestra en la figura 8. En cada una de las gráficas se muestra la evolución de la respuesta de un motor frente a una escala de tiempo normalizada, de tal forma que permita comparar ensayos de duraciones distintas.

En la figura 8 se puede observar el funcionamiento del bucle de regulación de posición del extremo. A medida que la velocidad del movimiento aumenta, la posición de los motores se aleja más de la referencia, para corregir la flexión de las barras de la estructura, de modo que el extremo siga la trayectoria lo más fielmente posible. Este efecto se puede apreciar en la gráfica del ángulo del motor tres, θ_{m3} . Cuando el motor comienza a frenarse (aproximadamente en el paso 40 de la simulación), la deflexión de la barra invierte su signo, de modo que el extremo se adelanta a la posición del motor. Para compensarlo, el bucle de control hace que el ángulo del motor se retrase, de acuerdo con las medidas de flexión obtenidas por las galgas situadas sobre los eslabones del robot. El proceso inverso ocurre cuando el motor vuelve a acelerar hasta alcanzar el valor final de giro.

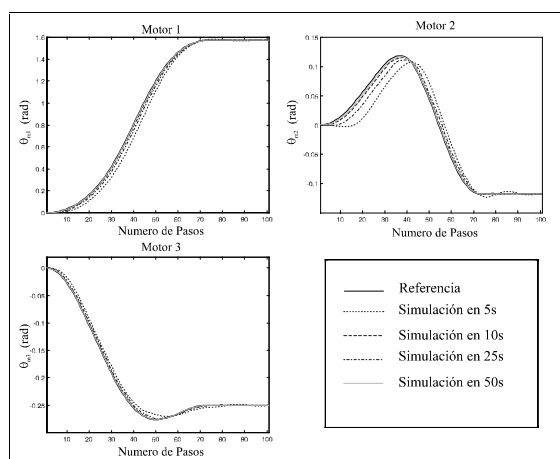


Figura 8: seguimiento de la referencia en cada motor para distintas duraciones del ensayo.

En los ensayos más largos, en los que las velocidades de los motores son muy lentas, este efecto es inapreciable ya que apenas se produce flexión en las barras, por lo que no es necesario corregirla. Por lo tanto, las desviaciones producidas en estos ensayos se deben más a errores de cálculo que al correcto funcionamiento del motor.

5 CONCLUSIONES

El objetivo del programa es obtener de manera cómoda trayectorias de distintas características para realizar simulaciones del robot en el entorno *Adams*[®]. No obstante, y debido a la generalidad de los resultados, puede extenderse su utilización a la generación de trayectorias de referencia para el robot real, con unas ligeras modificaciones en los parámetros mecánicos. Si se desean funciones continuas, se puede emplear el algoritmo de interpolación utilizado, de fácil implementación en código. Con ello se reduce el volumen de cálculo necesario cuando el periodo de muestreo es muy pequeño, si se trata de obtener trayectorias en tiempo real. El polinomio de interpolación es de orden cúbico entre cada par de puntos, y el error cometido en la aproximación es de orden $O(h^3)$, suficiente precisión en la mayoría de los casos.

La versatilidad de esta herramienta ha permitido la realización de complejas simulaciones de un robot flexible de seis grados de libertad. Así mismo ha sido posible utilizar las mismas trayectorias, una vez probadas en la simulación, sobre un prototipo real. Esto ha convertido a nuestro generador de trayectorias en una herramienta imprescindible que nos ha permitido avanzar en el desarrollo de nuevas técnicas de control de robots flexibles.

Agradecimientos

Este trabajo ha sido financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT) bajo el proyecto número TAP-96-1028-C02-01.

Referencias

- [1] Bayo E., (1987) "A Finite Element Approach to Control the End Point Motion of a Single-Link Flexible Robot", *Journal of Robotic Systems*, Vol 4, No 1.
- [2] Feliu V., Rattan K.S., Brown H.B., (1993) "Control of Flexible Arms with Friction in the Joints", *IEEE Transactions on Robotic and Automation*, Vol 9, No 4.
- [3] Feliu V., Rattan K.S., Brown H.B., (1995) "Control of a Two-Degree-of-Freedom Lightweight Flexible Arm with Friction in the Joints", *Journal of Robotics Systems*, Vol 12, No 1.
- [4] Feliu V., García A., Somolinos J.A., Sánchez L., (2000) "*Hacia un Robot Flexible de Especificaciones Industriales*", Conferencia Internacional sobre el Control de Sistemas Industriales, Convención Informática 2000, La Habana, Cuba.
- [5] Fraser A., Daniel R., (1991) "Perturbation techniques for flexible manipulators", Kluwer Academic Publishers.
- [6] Fu K.S., González R.C., Lee C.S.G., (1990) "Robótica: Control, Detección Visión e Inteligencia", Mc Graw Hill, Inc.
- [7] García A., Feliu V., (2000) "*Force Control of a Single-Link Flexible Robot Based on a Collision Detection Mechanism*", *IEE Proceedings on Control Theory and Applications*.
- [8] Gawronski W., Ih C.-H.C., Wang S.J., (1995) "On Dynamics and Control of Multi-Link Flexible Manipulators", *ASME Journal of Dynamics Systems, Measurement, and Control*, Vol. 117.
- [9] Nevins J.L., Whitney D.E., (1973) "The Force Vector Assembler Concept", *Proceedings 1st CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*. ASME, Udine, Italy.
- [10] Somolinos J.A., Feliu V., Sánchez L., Cerrada J.A. (1999) "Modeling and Control of a New Three - degree - of - freedom Flexible Arm with Simplified Dynamics", *IEEE Proceedings International Conference on Robotics and Automation*.
- [11] Torfs D.E., Vuerinckx R., Swevers J., Schoukens J., (1998) "Comparison of Two Feedforward Design Methods Aiming at Accurate Trajectory Tracking of the End Point of a Flexible robot Arm", *Transactions on Control Systems Technology*, Vol. 6, No 1, IEEE.