



**I CAIM 2008**  
**Primer Congreso Argentino**  
**de Ingeniería Mecánica**  
**Octubre 2008**

## **CONSTRUCCIÓN DE SIMULADORES PARA LA COMPRENSIÓN DE LA ROBÓTICA**

O. Daniel Morán, José A. Cuello, Germán F. Künning

*Departamento de Ingeniería, Universidad Nacional de San Luis*  
*Av. 25 de Mayo 384. (5730) Villa Mercedes (S.L). e-mail: [dmoran@fices.unsl.edu.ar](mailto:dmoran@fices.unsl.edu.ar)*

### **RESUMEN**

El presente trabajo surge de la necesidad de investigar la efectividad en el aula de un proceso didáctico implementado en el dictado de la asignatura robótica, en el marco de la propuesta pedagógica-didáctica de la “enseñanza para la comprensión”. Se decidió implementar un proceso didáctico innovador a partir de ciertos inconvenientes encontrados en los procesos de enseñanza y de aprendizaje de la robótica. Estos inconvenientes que son comunes a la enseñanza de ciertas asignaturas, están relacionados con la enseñanza del funcionamiento de máquinas o dispositivos que presentan ciertas partes que se mueven. Enseñar dispositivos que se mueven siguiendo leyes, sin la posibilidad de contar con estos, es un problema. Las tecnologías de la información y la comunicación (TIC's) constituyen valiosos recursos para desarrollar estrategias que apunten a mejorar los procesos de enseñanza y de aprendizaje de materias con estos contenidos o similares. Para el dictado de la asignatura se implementó una estrategia basada en la construcción de simuladores como actividades de comprensión para ayudar a los estudiantes a transferir conceptos matemáticos, informáticos, científicos y tecnológicos en un proceso didáctico que culmina con la construcción del simulador de un robot. Es decir se realiza la imitación de un modelo real por medio de dispositivos artificiales. Esto se logra mediante la utilización de recursos informáticos de programación y software para el manejo y presentación de la información.

En el presente trabajo se determina que los programas realizados por los estudiantes para la construcción del simulador, constituyen actividades de comprensión de tópicos que implican movimientos controlados numéricamente y se determina hasta que punto estas actividades son una aproximación para el caso de tener que controlar un robot real.

**Palabras Claves:** Robótica - Simulación - Enseñanza - Comprensión.

## 1. INTRODUCCIÓN

En la Asignatura Robótica, de 5to año de las carreras de Ingeniería Electromecánica y Electrónica, de la Facultad de Ingeniería y Ciencias Económico - Sociales, de la Universidad Nacional de San Luis, se enseña el funcionamiento de robots antropomorfos, principalmente enfocado al estudio del sistema de control de los mismos. Esto requiere la compleja tarea de enseñar como se mueven los ejes de los motores de las articulaciones, controlados numéricamente. Lograr que los alumnos se apropien de este conocimiento, que lo puedan usar con flexibilidad y más allá de lo que saben, es complicado.

Considerando que para comprender ideas complejas y formas de investigación, los estudiantes deben aprender haciendo, se diseñó una programación didáctica basada en el marco teórico de la enseñanza para la comprensión [1-3]. La misma incluye actividades de comprensión que deben realizar los estudiantes, que consisten en la construcción de simuladores de robots con animaciones gráficas. Parcialmente el alumno construye el simulador del problema cinemático directo, el simulador del problema cinemático inverso, para culminar con un trabajo integrador, que consiste en construir un simulador con animación gráfica del control cinemático de un robot antropomorfo de tres grados de libertad.

El programa del simulador construido es en cierta medida, un acercamiento a lo que hace un experto en la realidad, cuando realiza el programa de control de un robot real. Surge la pregunta, ¿Cuáles son las similitudes y las diferencias entre el programa de control desarrollado por los estudiantes para el simulador del robot y el programa de control desarrollado por un experto para el caso de un robot real?

El objetivo del presente trabajo es realizar una comparación entre el programa de control desarrollado por los estudiantes para la construcción del simulador del robot y el programa de control realizado por un experto para el control de un robot real. Para esta comparación se cuenta con un robot diseñado y construido en el Laboratorio de Mecatrónica de la Facultad donde se desarrollan estas actividades de docencia e investigación (Figura 1).



Figura 1: Robot construido en la FICES

## 2. REALIZACIÓN DE LA COMPARACIÓN

Se realiza el análisis comparativo entre el programa realizado por los estudiantes para construir el simulador y el que debe realizar un experto para controlar un robot real. En este caso la comparación se hace para el caso de una articulación o eje para facilitar la misma (Figura 2), pero las conclusiones pueden ser directamente generalizadas al caso de más ejes.

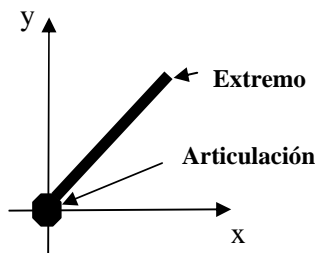


Figura 2: Brazo utilizado para la comparación

### 2.1 Condiciones y criterios para la comparación

Para realizar la comparación se utilizarán dos programas. Por un lado se utiliza el programa realizado para construir el simulador desarrollado en Matlab. Se usa un programa tipo simplificado para realizar el control de un solo eje, aunque en la asignatura se lo realiza para tres ejes como mínimo. Por otra parte se utiliza el programa realizado en C++ desarrollado para el control del Robot Didáctico CXN-I (Figura 1).

Se debe considerar que los alumnos conocen en profundidad la solución del Control Cinemático de un robot desarrollado en el dictado de la asignatura [4,5].

Una síntesis de los trabajos prácticos desarrollados en la asignatura como actividades de comprensión, a modo de ilustrar los conocimientos previos de los estudiantes hasta llegar al trabajo final integrador, es la siguiente (Tabla1):

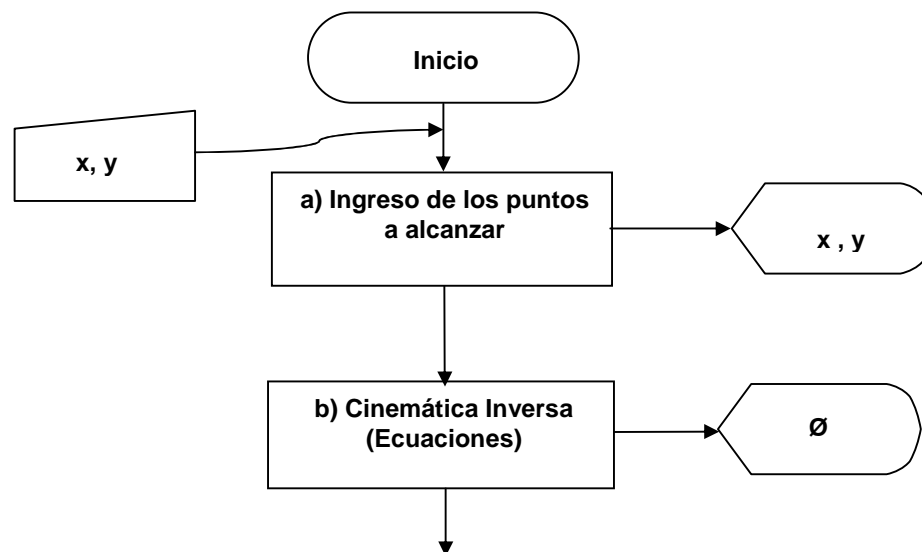
Tabla 1: Síntesis de los trabajos prácticos

Práctico N°	ACTIVIDAD DE COMPRENSIÓN	METAS DE COMPRENSIÓN
1	Realizar operaciones de transformaciones espaciales aplicando matrices de rotación 3x3 utilizando Matlab	- Dominar las transformaciones espaciales de rotación. - Obtener destreza en el uso de Matlab para resolver este tipo de operaciones.
2	-Realizar operaciones de transformaciones espaciales de traslación y rotación combinadas -Graficar vectores y mover líneas usando Matlab	-Dominar las transformaciones espaciales de traslación y de rotación combinadas. -Animar objetos gráficos con Matlab. -Obtener destreza en el uso de Matlab.
3	Obtención de modelos para distintas	Obtener modelos de robots, en forma

	configuraciones de robots aplicando la metodología sistemática de Denavit-Hartenberg.	metódica y sistemática.
4	-Obtención de la cinemática directa para distintas configuraciones de robots. -Construcción de un simulador con animación gráfica, con ingreso de datos por teclado, para cada una de las configuraciones de robots propuestas	Comprender la solución de la cinemática directa de un robot.
5	- Obtención de la cinemática inversa para distintas configuraciones de robot. - Construcción de un simulador con animación gráfica, con ingreso de datos por teclado, para cada una de las configuraciones de robots propuestas.	Comprender la solución de la cinemática inversa de un robot.
Trabajo final Integrador	Construcción de un simulador con animación gráfica de un robot antropomorfo de tres grado de libertad.	- Comprender la solución del control cinemático de un robot. - Integrar todos los conocimientos vistos hasta este momento.

## 2.2 Diagrama de flujo para el control de un eje del robot

El siguiente es el diagrama de flujo utilizado tanto para el simulador como para el robot real (Figura 3). A partir de la etapa d, se bifurca en dos partes, una es la que muestra el camino seguido para construir el simulador y la otra para el robot.



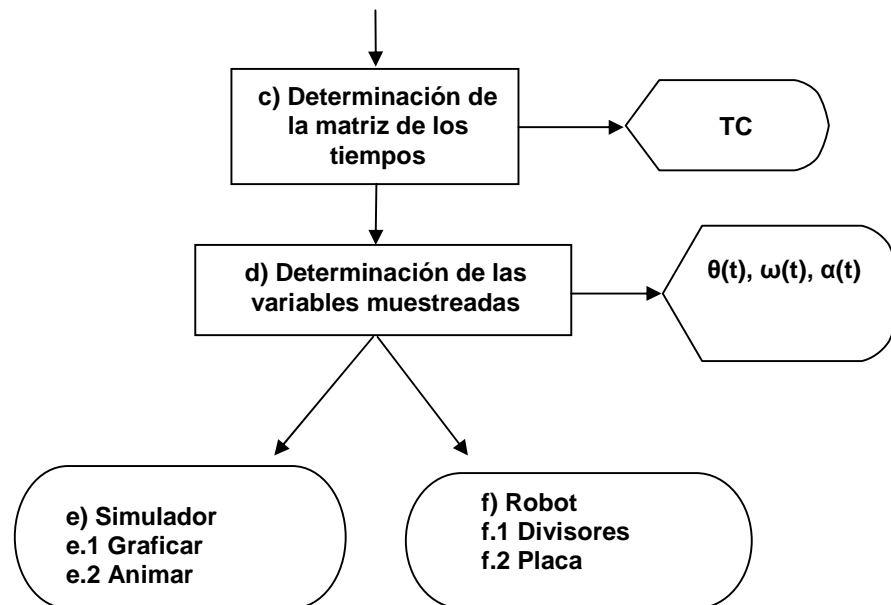


Figura 3: Diagrama de flujo

El diagrama de flujo tanto para el caso del simulador realizado en Matlab, como para el caso del programa del robot realizado en C++, son iguales hasta el cuadro d. Observando el diagrama de flujo se encuentran las siguientes partes:

a) En esta instancia se obtienen los puntos del espacio cartesiano, que debe alcanzar el robot. Los mismos pueden provenir de un usuario que los ingresa por teclado o estar previamente definidos en una matriz de puntos. También en esta parte del programa, se ingresan las constantes de la consigna:

Tiempo final: Es el tiempo que se le da al motor para ir de un punto a otro.

Tiempo muerto: Es el tiempo que se detiene el motor entre dos puntos consecutivos.

Aceleración máxima: Es la máxima aceleración que puede desarrollar el motor.

b) En esta parte del programa por medio de las ecuaciones de la inversa se obtiene el valor del ángulo que debe girar el eje (espacio articular) para que el extremo del robot se mueva de un punto a otro (espacio cartesiano).

c) En esta parte el programa debe calcular los tiempos  $T_b$  correspondiente a las transiciones parabólicas que se generan en una incursión del eje para ir de un punto a otro, es decir de un ángulo inicial a un ángulo final. Hay 5 (cinco) tiempos fundamentales para desarrollar el movimiento articular entre un ángulo inicial y un ángulo final:

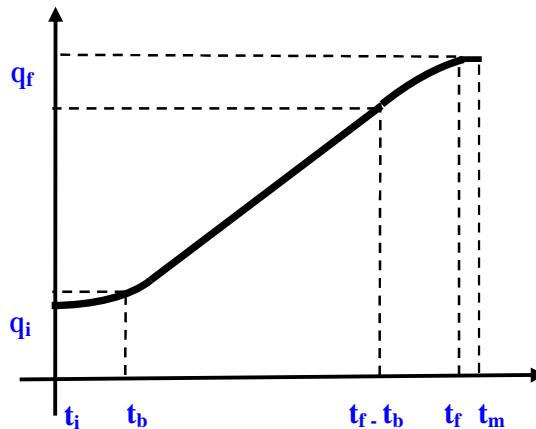


Figura 4: Posición articular

Tiempo  $T_i$ : Es el tiempo donde inicia cada movimiento.

Tiempo  $T_b$ : El tiempo de la primera transición parabólica.

Tiempo  $T_f - T_b$ : Es el tiempo correspondiente al segmento lineal.

Tiempo  $T_f$ : Es el tiempo final

Tiempo  $T_f + T_m$ : Es el tiempo final mas el tiempo muerto.

d) En este punto del programa se deben calcular los valores de posición, velocidad y aceleración para cada incursión, pero muestreados en función del tiempo. Estos datos son fundamentales para realizar las gráficas de estas magnitudes, lo cual es de mucho valor para el estudio y análisis del control, pero que no es necesario realizarlo en la etapa de control del Robot, por lo que no se realiza en este caso, por lo tanto la comparación pierde interés.

e y f) En esta etapa los estudiantes se encargan de realizar la simulación que consta de dos partes, la representación gráfica de las variables en función del tiempo ( $\theta(t)$ ,  $\omega(t)$ ,  $\alpha(t)$ ) y la animación gráfica del simulador. Por otra parte el experto se encarga de adaptar los valores obtenidos a las reducciones de la articulación y de "sacar" estos datos mediante el hardware disponible. Aunque, incluso estas actividades tienen similitudes no se comparan debido a que el control real de los motores sería objeto de estudio de otra **asignatura**.

### 2.3 Cuadros comparativos entre los dos programas

Seguidamente se comparan solo las 2 (dos) partes principales de los programas, el realizado en Matlab para el simulador y el realizado en C++ para el robot didáctico real (Tabla 2). Se puede observar la similitud entre ambos programas, aún considerando que están escritos utilizando herramientas y lenguajes diferentes. Esta parte del programa denominada principal, es la que permite ir llamando a las diferentes funciones que van a ir realizando las tareas necesarias para lograr el control del simulador o del robot según corresponda. Por cuestiones de espacio no se realizan otras comparaciones. La similitud entre los programas se mantiene

en las funciones: Puntosxy. Función Inversa. Función Tiempo TC . Determinación de las variables muestreadas.

Tabla 2: Comparación ente los dos programas

SIMULADOR en MATLAB	ROBOT en C++	COMENTARIOS
<p><b>% PROGRAMA: PRINCIPAL</b></p> <p>%Este programa se usa para orientar al alumno en los pasos necesarios para comprender el control de un robot. Realiza el control de una sola articulación%</p> <pre>clc clear close all</pre> <p>% Variables y parámetros</p> <pre>global AcelMax, • iempote, TiempoMuerto AcelMax = 80; % Amax [grados/seg<sup>2</sup>] • iempote = 2; % T entre puntos [seg] TiempoMuerto = 0.2; % Parada en [seg]</pre> <p><b>% 1- Ingreso puntos de la trayectoria</b></p> <pre>PuntosXY = PuntosXY pause(2)</pre> <p><b>% 2- Cinemática inversa</b> Los ángulos se determinan por medio de la función CinematicalInversa</p> <pre>Angulos = CinematicalInversa(PuntosXY ) pause(2)</pre> <p><b>%3- Matriz de los tiempos TC</b> % La función TiempoTC, retorna una matriz de</p>	<p><b>% PROGRAMA: Principal</b></p> <pre>/* Inclusión de los archivos de cabecera y archivos de funciones */ #include &lt;iostream&gt; #include "TareaDelRobot.h" #include "CinematicalInversa.h" #include "TiempoTC.h"  using namespace std;  int main() { /* Definición de variables a utilizar */  double AcelMax = 80; // [grados/seg<sup>2</sup>] double • iempote = 2; // [seg] double TiempoMuerto = .2; // [seg]  double **PuntosXY; int nPuntosXY; double *Angulos; double *TC; double *pos; double *vel; double *acel; double *t;</pre> <p><b>/* 1- Ingreso puntos trayectoria */</b></p> <pre>PuntosXY(&amp;PuntosXY, &amp;nPuntosXY);</pre> <p><b>/* 2- Cinemática inversa */</b> Los ángulos se determinan por medio de la función CinematicalInversa</p> <pre>CinematicalInversa(PuntosXY, nPuntosXY, &amp;Angulos);</pre> <p><b>/* 3- Matriz de los tiempos TC*/</b></p>	<p>Los programas realizados en Matlab como en C++, cuentan de seis partes.</p> <p>El primer archivo o función, corresponde al primer cuadro (inicio) del diagrama de flujo de la Fig 3. En ambos programas esta parte es para fijar parámetros, declarar variables y realizar las llamadas a otras funciones para ir resolviendo las distintas partes en que se divide.</p> <p>En esta parte se fijan los valores de aceleración máxima, tiempo entre puntos y tiempo de parada, para ir de un punto a otro. Se declaran las variables de acuerdo con las características de cada entorno.</p> <p>Se llama a la función PuntosXY, que es donde se ingresan los puntos que debe recorrer el extremo del Robot en el espacio cartesiano. Se llama a la función CinematicalInversa para calcular cada ángulo que debe alcanzar la articulación, para que el extremo del brazo se posicione en los PuntosXY.</p> <p>Se llama a la función TiempoTC, para calcular los tiempos de las transiciones</p>

<p>tiempos, que permiten desarrollar el interpolador por tramos.</p> <p>TC = TiempoTC(Angulos) pause(2)</p> <p><b>% 4- Determinación de las variables muestreadas %</b></p> <p>[pos,vel,acel,t]=VariablesMuestreadas(1,TC,0.01,Angulos);</p> <p><b>% 5- Gráfica de las variables muestreadas %</b></p> <p>Grafico(t,pos,vel,acel);</p> <p><b>% 6- Animación Grafica %</b></p> <p><b>Fin del programa principal</b></p>	<p>TiempoTC(Angulos, nPuntosXY, AcelMax, • iempote, TiempoMuerto, &amp;TC);</p> <p><b>/* 4- Determinación de las variables muestreadas */</b></p> <p>VariablesMuestreadas(1, TC, 0.01, Ang, pos, vel, acel, t);</p> <p><b>/* 5- Determinación de los divisores de frecuencia del timer */</b></p> <p><b>/* 6- Envío de datos a la placa */</b></p> <p><b>Fin del programa principal</b></p>	<p>Tb, (Figura 4) y retorna la matriz de tiempos.</p> <p>Se llama a la función variables, para determinar en forma muestreada la pos, vel y acel para cada incursión que deba realizar el robot.</p> <p>A partir de este punto las tareas a realizar para el simulador o para el robot son diferentes y aunque desde lo conceptual “estar moviendo un simulador o un motor” tiene alguna similitud, la diferencia entre los lenguajes del Matlab y C++ no permite esta comparación.</p>
--	---	---

### 3. CONCLUSIONES

Queda demostrado que la construcción del simulador permite alcanzar uno de los objetivos del proceso didáctico de la asignatura, de realizar actividades prácticas de comprensión que tengan similitud con el trabajo profesional y permitan “aprender haciendo”.

Se ha realizado la comparación entre los dos programas y se han encontrado las similitudes y diferencias entre ellos. Por otra parte es importante destacar la alta probabilidad de transferencia de estos conocimientos a otras aplicaciones que implican procesos industriales Automatizados. El resultado alcanzado es muy importante e interesante ya que estos desempeños están orientados a la obtención del modelo matemático de un sistema físico y al control de ese sistema mediante la construcción de un simulador, realizando actividades similares a las de un profesional y en un claro compromiso con la realización de actividades prácticas y la metacognición.

### 4. REFERENCIAS

- [1] Wiske, M. S. *Enseñar para la Comprensión: Con Nuevas Tecnologías*. Paidós. Bs. As. 2006.
- [2] Blythe, T. *La Enseñanza para la Comprensión: guía para el docente*. Paidós. Bs. As. 1999.
- [3] Pogré, P.; Lombardi, G. *Escuelas que Enseñan a Pensar: Enseñanza para la Comprensión un Marco Teórico para la Acción*. Educación Papers Editores. Buenos Aires. 2004.
- [4] Barrientos, A.; Peñin, L.; Balaguer, C.; Aracil, R. *Fundamentos de Robótica*. McGraw-Hill. Buenos Aires. 1997.
- [5] Kelly, R.; Santibáñez, V. *Control de Movimientos de Robots Manipuladores*. Prentice Hall. Buenos Aires. 2003.