

## DESARROLLO DE UN SISTEMA DE SIMULACIÓN Y PROGRAMACION PARA EL ROBOT EXPERIMENTAL *SALVIATI*

*Rafael Fematt M. Eliseo Juárez R. Alfonso Pámanes G. Guillermo Franco Q.*

### Instituto Tecnológico de la Laguna

Grupo de Robótica del Departamento de Ingeniería Eléctrica y Electrónica  
Blvd. Revolución y Cuauhtémoc; Torreón, Coah. 27000; MÉXICO  
Dir. electrónica: [apamanes@halcon.laguna.ual.mx](mailto:apamanes@halcon.laguna.ual.mx) Fax: +52 (17) 13.09.70

### RESUMEN

*En este artículo se describe el sistema SISPROS, de simulación y programación de tareas del robot manipulador SALVIATI, modelo TL2, desarrollado en el Instituto Tecnológico de la Laguna para fines experimentales y didácticos. Gracias a su simulador gráfico, el SISPROS facilita la programación del robot fuera de línea, lo cual permite planificar un desempeño óptimo de éste para la ejecución de sus tareas. Asimismo es posible realizar la programación del robot en línea. El sistema opera en una computadora personal (PC) en ambiente Windows, y presenta al usuario una interfase gráfica amigable, provista de diversas funciones que simplifican la programación del robot.*

### 1. INTRODUCCIÓN.

Una de las características más interesantes de los robots manipuladores desde el punto de vista industrial es su flexibilidad operacional. En efecto, estas máquinas son capaces de desarrollar múltiples tareas, y cada una de ellas la pueden realizar de diferentes maneras, de tal suerte que se adaptan en poco tiempo y a un costo reducido a las necesidades del usuario.

El proceso de adaptación de un robot a una nueva tarea consiste básicamente en la programación de la secuencia de los nuevos movimientos que deberá ejecutar. Típicamente, la programación de un robot manipulador se realiza bajo dos enfoques: *en línea* y *fuera de línea*. El primer enfoque implica la *enseñanza* de la tarea al robot por parte del usuario, conduciéndolo de tal manera que el órgano terminal se sitúe en un cierto número de puntos correspondientes a la trayectoria a seguir. La programación fuera de línea, en cambio, utiliza un simulador gráfico basado en el modelo matemático del robot y de su ambiente con el propósito de planificar los movimientos del mismo.

Al realizar la programación en línea se requiere la utilización del robot, por lo que éste se vuelve improductivo durante el tiempo necesario para la programación. Ese periodo puede ser bastante prolongado dependiendo de la complejidad de la tarea. No obstante, a favor de este enfoque se tiene el hecho de que el método es relativamente fácil de aprender, lo cual implica una capacitación reducida del programador.

En el método fuera de línea el tiempo de programación se reduce sustancialmente, y la participación del robot es requerida durante un periodo más corto. Otra ventaja importante de

este enfoque es que permite planificar las tareas del robot de tal manera que su desempeño sea óptimo durante la ejecución de éstas (Pámanes y Zeghloul, 1991). El procedimiento fuera de línea, sin embargo, implica un costo adicional respecto a la programación en línea debido a la necesidad del uso de herramientas computacionales (*hardware* y *software*) y de instrumentos de calibración.

En la literatura técnica se ha tratado ampliamente el tema de la programación de robots bajo los enfoques aquí mencionados; v.gr. Derby (1983), Zeghloul et al. (1987), Gini (1987), Buchal et al. (1989).

En el desarrollo del robot manipulador experimental *SALVIATI*, modelo *TL-2*, se diseñó y realizó un sistema computacional que permite efectuar la programación de tareas bajo cualquiera de los dos enfoques citados en los párrafos precedentes. Este sistema, llamado *SISPROS* (Sistema de Simulación y Programación del ROBot Salviati), que opera en una computadora personal (PC), está orientado hacia la enseñanza y el entrenamiento para la programación de robots a nivel elemental. La estructura y las funciones básicas del *SISPROS* se describen en el presente artículo.

## 2. EL ROBOT MANIPULADOR *SALVIATI TL-2*.

El modelo *TL-2* del *SALVIATI* se construyó en el año de 1994, y constituye la segunda versión de este robot experimental y didáctico del Instituto Tecnológico de la Laguna (la primera versión -el modelo *TL-1*- se realizó en 1987). Como se indica en el esquema de la Figura 1, el robot está integrado por los siguientes sistemas: mecánico, de control, de comunicación, y de percepción.

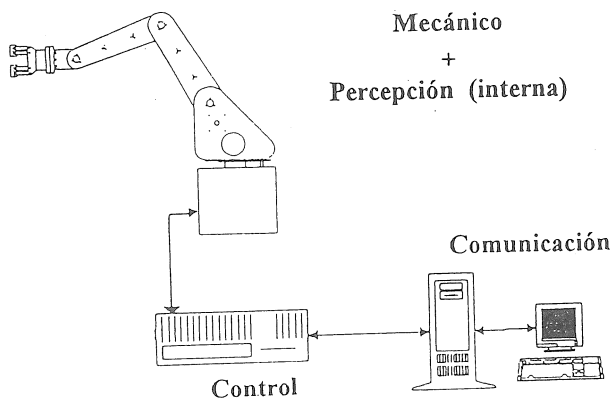


Figura 1. Sistemas del robot manipulador *SALVIATI*.

El sistema mecánico consiste en un manipulador de cuatro grados de libertad, los actuadores y las transmisiones de potencia. El manipulador posee una arquitectura de coordenadas angulares, como se muestra en la Figura 2, en donde también se representa la sección transversal del espacio de trabajo. Los motores son eléctricos de corriente directa, a partir de los cuales se conduce la potencia hacia los eslabones del manipulador mediante transmisiones de engranes, cadenas y bandas dentadas (Pámanes et al, 1995).

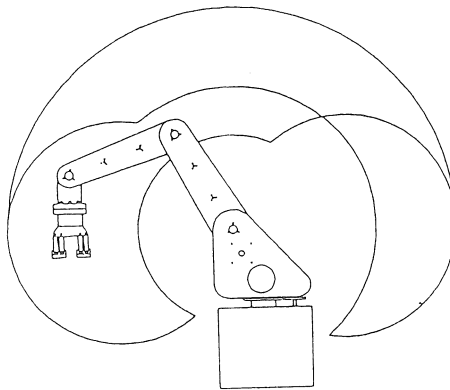
El sistema de percepción está constituido por cuatro captadores analógicos, instalados en las articulaciones del manipulador, los cuales miden las coordenadas articulares.

La tarjeta principal del sistema de control del *SALVIATI* se basa en el uso de microcontroladores HCTL-1000 de *Hewlett Packard*. Esta tarjeta es la misma que se diseñó para el modelo TL-1, y fue desarrollada en cooperación con el Instituto de Automática Industrial (IAI), de Madrid, España, en el marco del programa CYTED (Santibáñez et al, 1990). Además de esta tarjeta, en el sistema de control se incluye una interfase de potencia, que amplifica las señales de control para alimentar a los actuadores.

Finalmente, el sistema de comunicación incluye una computadora PC 486 DX2, y al paquete SISPROS para la programación y simulación del robot.

Las características de operación del *SALVIATI TL2* son las siguientes:

- Capacidad de carga: **9.81 N**
- Alcance máximo: **0.90 m**
- Velocidad máxima: **0.50 m/s**



$$A = 1.097 \text{ m}^2$$

Figura 2. Sección transversal del espacio de trabajo del robot *SALVIATI*

### 3. DESCRIPCIÓN GENERAL DEL *SISPROS*.

La misión del programa *SISPROS* consiste en establecer la comunicación entre el usuario y los otros sistemas del robot. Para ello el programa posee funciones que ofrecen al usuario las siguientes posibilidades:

- Informarse acerca de la situación del manipulador en términos de sus coordenadas operacionales y sus coordenadas articulares.
- Asignarle al manipulador los programas de movimientos necesarios para la realización de sus tareas.
- Modificar los valores de las ganancias para las leyes de control de los motores.
- Utilizar el recurso de la simulación gráfica para la planificación y evaluación de los movimientos del manipulador antes de su ejecución en tiempo real.

La *ventana* inicial del programa que se le presenta al usuario está organizada en cuatro secciones (ver la Figura 3). La *sección de animación* corresponde al área de la pantalla reservada para la simulación gráfica del manipulador. En la *sección de comando articular* se presenta una *botonadura* para la generación de desplazamientos articulares, controlados por el usuario. La *sección de programación* contiene un conjunto de *iconos* asociados a las funciones necesarias para el establecimiento de un programa de movimientos. Finalmente la *sección de coordenadas articulares* presenta cuatro pequeñas *ventanas* que exhiben los valores de dichas coordenadas. En la Tabla 1 se presenta un resumen de las funciones disponibles en cada una de estas secciones.

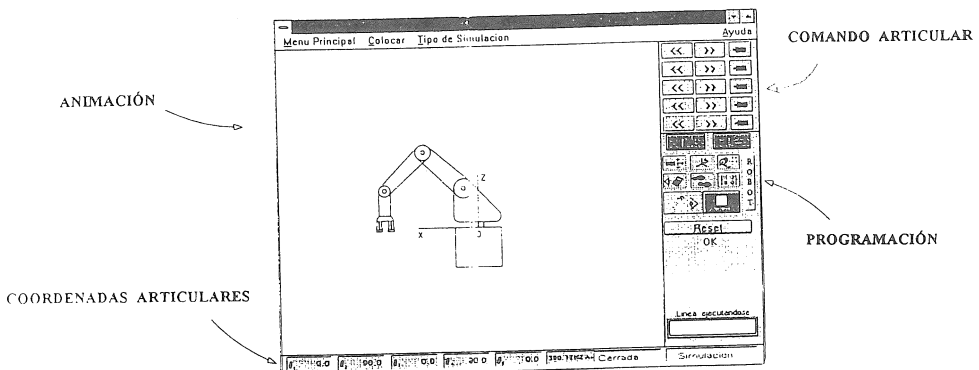













Figura 3. Ventana principal del *SISPROS*.

Tabla 1. Resumen de funciones del paquete *SISPROS*.

FUNCIONES DE CADA SECCIÓN	ÍCONO O BOTÓN	OPCIONES
<u>SECCION ANIMACIÓN</u>		
* Simulación.		
* Menú de entrada:		<ul style="list-style-type: none"> <li>- Gestión de archivos, salir del sistema.</li> <li>- Restablecimiento de la configuración del manipulador.</li> <li>- Modo de operación (<i>en línea y fuera de línea</i>).</li> </ul>
<u>SECCION COMANDO ARTICULAR</u>		
* Desplazamiento articular:		- Selección de la articulación (1, 2, 3, 4).
* Modificación de parámetros de ley de control:		- Selección del motor (1, 2, 3, 4).
<u>SECCION PROGRAMACIÓN</u>		
* Comando de la pinza:		- Apertura y cierre.
* Desplazamiento ortogonal de la pinza:		- Dirección de desplazamiento (X, Y, Z).
* Punto de observación de la escena simulada:		- Selección de cualquier punto de observación.
* Grabación de configuraciones:		<ul style="list-style-type: none"> <li>- Grabar una configuración presentada en pantalla.</li> <li>- Suprimir una configuración existente en archivo.</li> <li>- Ejecutar una configuración existente en archivo.</li> </ul>
* Exhibición de coordenadas operacionales:		<ul style="list-style-type: none"> <li>- Coordenadas cartesianas de la pinza.</li> <li>- Ángulos de Euler.</li> <li>- Matriz homogénea de la pinza.</li> </ul>
* Modificación del <i>paso</i> :		<ul style="list-style-type: none"> <li>- Paso de las rotaciones articulares.</li> <li>- Paso de desplazamiento lineal de la pinza.</li> </ul>
* Consigna de coordenadas operacionales:		- Especificar la matriz homogénea deseada de la pinza.
* Observación de movimientos en línea:		- Abrir y cerrar la ventana de observación.
* Edición de programas:		<ul style="list-style-type: none"> <li>- Edición programas de movimientos.</li> <li>- Gestión de archivos de programas.</li> <li>- Ejecución y paro de programas.</li> </ul>
<u>SECCION COORDENADAS ARTICULARES</u>		
* Exhibición de las coordenadas articulares.		

#### 4. PROCESO DE PROGRAMACIÓN.

Para facilitar la enseñanza, el SISPROS cuenta con una botonadura gráfica equivalente a un *teach pendant* industrial que le permite al usuario, con la ayuda del *ratón*, controlar los desplazamientos de la pinza en las direcciones de los ejes de un marco ortogonal fijo. La implantación de esta función se basa en el uso de los modelos cinemáticos de posición del manipulador, como se ilustra en el esquema de la Figura 4. En el procedimiento ahí mostrado, la situación de la pinza es especificada implícitamente por el usuario al demandar la ejecución de un desplazamiento en cualquier dirección, considerando el punto de partida y el valor vigente del paso. Estos desplazamientos se pueden realizar *en línea* o *fuera de línea*. En ambos casos se simulan los movimientos del manipulador en la sección de animación. Durante los desplazamientos de la pinza su orientación se conserva. La orientación puede ser establecida por el usuario activando los botones para la rotación de la cuarta articulación (sección de comando articular).

Una vez que la pinza se encuentra en la situación deseada, la configuración del manipulador se registra en un archivo (Figura 5). A cada configuración registrada se le asigna un número consecutivo, con el que se identificará posteriormente durante la edición del programa de movimientos. En la ventana de edición se prescribe la secuencia deseada de ejecución de configuraciones previamente registradas, y se asignan instrucciones suplementarias tales como *pausa*, *saltar a...*, y *abrir / cerrar pinza*. Adicionalmente, desde esta ventana se ordena la ejecución / paro del programa, y la suspensión temporal.

El procedimiento descrito en los párrafos precedentes se puede seguir también en la programación *fuera de línea*. En este caso no es necesaria la participación del manipulador. Los movimientos de éste se simulan gráficamente para guiar al usuario en la definición de las configuraciones requeridas por la tarea. Para definir el modo de programación es necesario hacer la selección pertinente en el menú de la sección de animación.

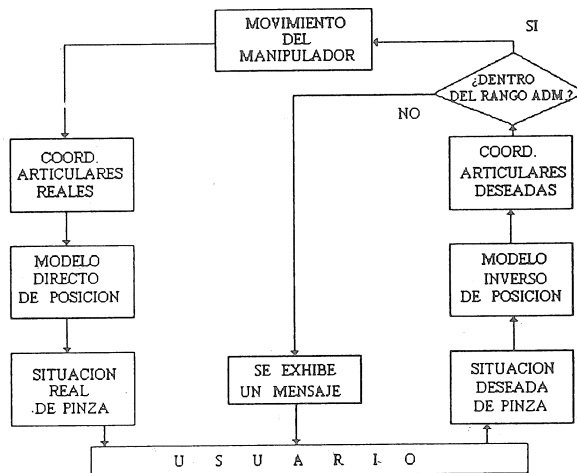


Figura 4. Estrategia para los desplazamientos ortogonales de la pinza

	Añadir	Eliminar	Posición	Cerrar
0	0.000	1.571	0.000	1.571
1	0.000	2.007	0.000	1.571
2	0.000	2.007	0.000	1.658
3	0.000	2.007	0.000	1.658
4	0.436	2.007	0.436	1.658

Figura 5. Ventana para la grabación de configuraciones.

En la Figura 6 se muestra la secuencia de configuraciones que se observa en el monitor de la computadora durante la simulación de la tarea definida por las tres posiciones especificadas en la Tabla 2. La orientación de la pinza es tal que debe mantenerse en dirección vertical, con los dedos apuntando hacia abajo, en los tres puntos. Esta tarea fue programada en el modo *fuera de línea*, desplazando la pinza de manera interactiva con la ayuda de la función de *desplazamiento ortogonal*. La orientación fue establecida utilizando los botones de la cuarta articulación, de la sección de comando articular. Las coordenadas articulares que se dan en la Tabla 2 son obtenidas automáticamente en el SISPROS al colocar la pinza en las sucesivas posiciones de la tarea.

## 5. CONCLUSIÓN.

En el presente trabajo se describieron las características básicas del paquete computacional SISPROS, utilizado para la simulación y programación del robot manipulador *SALVIATI TL-2*, desarrollado en el Insitituto Tecnológico de la Laguna. El SISPROS está orientado hacia la enseñanza y el entrenamiento para la programación de robots. Su utilización es muy simple gracias al uso de una interfase gráfica en ambiente *Windows*, y a la simulación gráfica de los movimientos del manipulador. Esta última característica permite realizar la programación del robot *fuera de línea*.

Actualmente, el sistema de control del robot solo cuenta con un lazo de control de posición para cada actuador, y no dispone de la etapa de velocidad. Por ello, en la versión actual del SISPROS no se incluye la opción de programación de velocidad. En el desarrollo futuro del robot se prevé la implantación del control y la programación de velocidad. En el SISPROS, adicionalmente, se contempla la incorporación de diversas funciones orientadas al desarrollo de la parte de programación *fuera de línea*. Actualmente se trabaja en la implantación de un procedimiento interactivo para la determinación del emplazamiento óptimo del manipulador en relación a su tarea. Otro trabajo en curso se refiere a la incorporación de un algoritmo para la generación automática de trayectorias teniendo en cuenta la prevención de colisiones.

Tabla 2. Posición de la pinza y coordenadas articulares del manipulador.  
Tarea de la Figura 6.

Punto	X (cm)	Y (cm)	Z (cm)	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )
P1	25	0	18	0	60.3	-41.0	-109.3
P2	25	0	8	0	59.6	-80.6	-69.0
P3	25	0	0	0	42.5	-87.6	-44.9

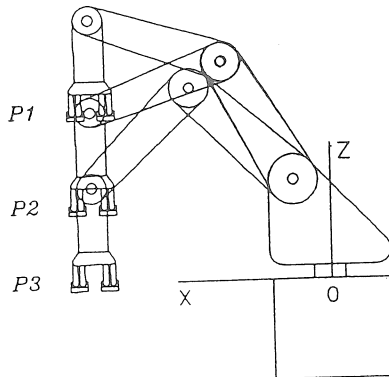


Figura 6. Simulación de una tarea en el SISPROS.

**REFERENCIAS BIBLIOGRÁFICAS.**

**Buchal R.O., Chercas D.B., Sassani F., Duncan J.P.;** *Simulated off-line programming of welding robots*; International Journal of Robotics Research, Vol. 8, No. 3, pp. 31-43 (1989).

**Derby S.;** *Simulating motion elements of general purpose robot arms*; International Journal of Robotics Research, Vol. 2, No. 1, pp. 3-12 (1983).

**Fuentes O., Santibáñez V., Fematt A., Pámanes A.;** *Mejoras a los sistemas mecánico, de control y de comunicación de un prototipo de robot manipulador*; Informe Técnico del Proyecto 918.88 del Centro de Graduados e Investigación del Instituto Tecnológico de la Laguna.

**Gini M.;** *The future of robot programming*; Robotica, Vol. 5, pp.235-246 (1987).

**Pámanes A., Zegloul S.;** *Optimal placement of robotic manipulators using multiple kinematic criteria*; Proceedings of the IEEE International Conference on Robotics and Automation, Vol.1, pp.933-938 (1991).

**Pámanes A., Fematt A., Franco G, Juárez E., Guajardo E.;** *Mejoras en el diseño del robot manipulador SALVIATI*. Memorias del I Congreso de la Sociedad Mexicana de Ingeniería Mecánica (1995). pp. 157-162

**Santibáñez D.V., Armada R.M., González de S.P.;** *Red de adelanto para control digital de motores*; Informe Técnico IAI-UEIASS-001/90; Instituto de Automática Industrial; Madrid, España (1990).

**Zegloul S., Murguet D., Lallemand J.P.;** *Outils de simulation et de programmation hors ligne d'environnements multi-robots*; Proceedings of the 7-th IFToMM Congress on Theory of Machines and Mechanisms, pp. 1089-1092 (1987).