



## Enseñanza y Aprendizaje de Robótica Industrial desde la Virtualidad

### Teaching and Learning of Industrial Robotics from Virtuality

Wilmer Sanz-Fernández<sup>1</sup>



EDICIÓN: CIVTAC

Recibido: 24/junio/2021  
Aceptado: 18/septiembre/2021  
Publicado: 24/septiembre/2021

País  
<sup>1</sup>Venezuela

Institución  
<sup>1</sup>Universidad José Antonio Páez

Correo Electrónico  
<sup>1</sup>wsanzf@gmail.com

ORCID  
<sup>1</sup><https://orcid.org/0000-0001-7847-2372>

Citar así: APA / IEEE

Sanz, W. (2021). Enseñanza y Aprendizaje de Robótica Industrial desde la Virtualidad. *Revista Tecnológica-Educativa Docentes 2.0*, 11(2), 19-27. <https://doi.org/10.37843/rted.v11i2.245>

W. Sanz, "Enseñanza y Aprendizaje de Robótica Industrial desde la Virtualidad", *RTED*, vol. 11, n.º 2, pp. 19-27, sep. 2021.

#### Resumen

Junto al primer robot manipulador, el UNIMATE de Devol y Engelberger, nació la preocupación de investigadores e ingenieros por demostrar el potencial de autómatas similares en aplicaciones industriales. En consecuencia muchas prestigiosas universidades percibieron cuán importante era incluir asignaturas del área en planes de estudios de carreras como Ingeniería Eléctrica, Electrónica, Mecánica o Mecatrónica. La presente investigación fue bajo el método constructivista y el análisis de resultados se apoya en la respuesta observada en un simulador mediante el uso combinado de una Toolbox para modelación cinemática de robots (basada en scripts ejecutables en un entorno como GNU Octave o similar), además de un software para simulación 3D. La ejemplificación del uso de las herramientas y el reto al estudiante de programar RI virtuales en escenarios típicos de manufactura, empaquetado o paletizado, ha sido probado a lo largo de cuatro períodos lectivos, obteniéndose una realimentación positiva de los aprobados y egresados. Los resultados alcanzados al simular demuestran la exactitud de los modelos cinemáticos obtenidos mediante formulaciones matemáticas, lo cual evidencia la utilidad de las herramientas descritas para el aprendizaje a distancia basado en la práctica.

**Palabras clave:** Robot Industrial, simulación, modelación, robótica, virtualidad.

#### Abstract

Together with the first manipulator robot, the UNIMATE from Devol and Engelberger, the concern of researchers and engineers was born to demonstrate the potential of similar robots in industrial applications. Consequently, many prestigious universities perceived how important it was to include subjects in curricula of careers such as Electrical Engineering, Electronics, Mechanics, or Mechatronics. The present investigation was under a mixed methodology. The analysis of results is supported by the response observed in a simulator through the combined use of a Toolbox for kinematic modeling of robots (based on executable scripts in an environment such as GNU Octave or similar). Of software for 3D simulation. The modeling of the use of the tools and the challenge to the student of programming virtual IRs in typical manufacturing, packaging, or palletizing scenarios, has been tested over four academic periods, obtaining positive feedback from those approved and graduates. The results achieved when simulating demonstrate the accuracy of the kinematic models obtained through mathematical formulations, which shows the usefulness of the tools described for distance learning based on practice.

**Keywords:** Industrial robot, simulation, modeling, robotics, virtuality.



## Introducción

Junto al primer robot manipulador, el UNIMATE de George Devol y Joseph Engelberger, nació la preocupación de investigadores e ingenieros por demostrar el potencial de autómatas similares en aplicaciones industriales. Robots históricos como el *Stanford Arm*, PUMA, Famulus y SCARA, entre otros, se desarrollaron para reemplazar la mano de obra en procesos riesgosos, mejorar la calidad y aumentar la productividad. Apuntando a estas metas, el estudio de Robótica Industrial en universidades o postgrados incluye dos temas fundamentales: modelación cinemática y programación de robots para la ejecución de tareas.

El primero de estos temas se basa en el uso de transformaciones matriciales en un espacio vectorial de 4 dimensiones junto a la obtención de ecuaciones que describan el movimiento de los robots, de acuerdo con sus particulares configuraciones. Por su parte, programar Robots Industriales (RI) para la ejecución de tareas, depende del manejo de lenguajes desarrollados por cada fabricante, siendo irremplazable, en lo respectivo al aprendizaje o capacitación de los estudiantes, la significación generada por el trabajo práctico llevado a cabo en un laboratorio dotado con un robot físico.

¿Es posible lograr una buena aproximación al trabajo real con los RI, mediante simulación en ambiente 3D? ¿Cómo simplificar la complejidad que supone obtener modelos matemáticos para describir el comportamiento cinemático de los autómatas usados como manipuladores en la industria? ¿Puede aprovecharse el empleo de herramientas de software libre para facilitar un curso de Robótica Industrial a distancia?

Es la finalidad de este estudio fue describir la aplicación de algunos comandos desarrollados como parte de la *Toolbox* e Interfaz Gráfica *PGIBOTMat*, un producto académico hecho para modelar cambios en la localización de objetos manipulados por Robots Industriales, programar representaciones virtuales de los mismos y simular la ejecución de tareas en un ambiente 3D.

## Metodología

El constructivismo es el método que fundamentó la presente investigación, la cual la permitió la implementación de las estrategias para la enseñanza. Si bien las demostraciones sobre modelación y el manejo de software especializado se encuentran presentes, el aprendizaje se centra en el quehacer del estudiante, quien debe replicar lo ilustrado y aplicarlo creativamente para resolver problemas reales, aunque en un ambiente virtual.

*Top-Down Design* es la metodología empleada para desarrollar el estudio. Partiendo desde un principio de modularización se conciben bloques o unidades genéricas con capacidad para producir las respuestas deseadas del sistema, luego se dividen en bloques especializados, según sea necesario (Pardo, 1997) y finalmente se definen los componentes reales que permiten hacer la implementación. En Ingeniería, equivale a una estrategia inspirada en la antigua máxima “dividir para vencer”, donde los grandes problemas se abordan como conjuntos de otros menores, logrando la solución completa mediante la unión de soluciones parciales (Arango, 2006). Su complemento, metodología *Bottom-Up*, se usa como estrategia de realimentación para redefinir las funciones de los bloques originales, una vez que se tienen suficientemente claras las características del sistema finalmente implementado.

Resulta natural la aplicación del presente estudio bajo una modalidad *virtual* o *e-Learning* ante la idea de seleccionar y adaptar herramientas tecnológicas para su utilización en un curso de Robótica bajo modalidad no presencial. Efectivamente, el acceso al ciberespacio y sus distintos escenarios (chats, páginas Web, Campus Virtuales, entre otros.) “acentúa el carácter transterritorial, temporo-espacial y transfronterizo de la modalidad de educación a distancia” (Salazar, 2015), todo lo cual se precisa cuando una comunidad estudiantil se encuentra dispersa y no dispone del recurso de la conectividad en forma permanente.

Los medios para lograr el intercambio de información se alcanzó de manera sincrónica o asincrónica, las cuales incluyeron el multimedia, hipermedia, simuladores y recursos que aprovechan las continuas actualizaciones

tecnológicas; siendo particularmente útiles en el aislamiento demandado por una pandemia. Tal es la dinámica impuesta en los actuales momentos, impulsada además por una nueva Revolución Industrial (Industria 4.0), la cual inevitablemente permea en el ámbito educativo.

Finalmente, el análisis de resultados se apoya en la respuesta observada en un simulador. Afirma González, ante la disyuntiva de abordar un estudio mediante modelos analíticos o mediante la observación real: “se hace imprescindible la utilización de una nueva metodología que ayude, por un lado a la resolución de los problemas planteados en la modelización de estos sistemas complejos y por otro, a la creación de estos modelos adaptados a los grandes problemas con los que nos podemos enfrentar en la actualidad; y esta nueva forma de afrontar estos temas es la Simulación” (2014, p.1). Tal como el mismo autor manifiesta la simulación es una herramienta útil para tomar decisiones y analizar cómo funcionan los sistemas. Aplicada a los Robots Industriales, permite comprobar la exactitud y eficacia de los programas que pueden transferirse posteriormente a controladores en autómatas reales.

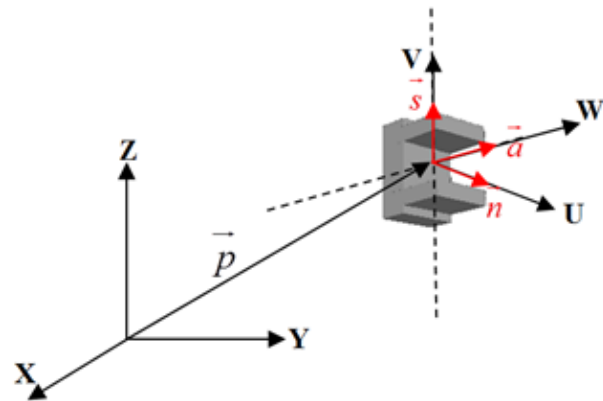
### Transformaciones Homogéneas

Antes de abordar cualquier otro aspecto, fundamental es invocar una definición formal de RI. La Organización Internacional de Estándares en su estándar 8373:2012 (ISO, 2016) establece el concepto de Robot Industrial: “un manipulador multipropósito, reprogramable, controlado automáticamente, programable en tres o más ejes, que puede ser fijo o móvil para su uso en aplicaciones de automatización industrial”. Tal definición es aceptada por la Federación Internacional de Robótica (IFR) y establece cuál es el tipo de autómatas referidos en este artículo.

Ya entrando de lleno en lo referente a modelación cinemática, se parte del uso de Transformaciones homogéneas (propias de un espacio vectorial con cuatro dimensiones) para describir los cambios en posición y orientación de objetos manipulados, las cuales son cantidades imprescindibles en Robótica (Ollero, 2007),

determinadas mediante los vectores  $\vec{p}$ ,  $\vec{n}$ ,  $\vec{s}$  y  $\vec{a}$  (Figura 1).

**Figura 1**  
Vectores de posición y orientación



*Nota.* El vector de posición  $\vec{p}$  indica las coordenadas del Punto Central de la Herramienta (TCP) y los vectores unitarios  $\vec{n}$ ,  $\vec{s}$  y  $\vec{a}$  definen las direcciones de un sistema de referencia móvil, sólidamente unido a los objetos manipulados, elaborado por Sanz (2016).

Para calcular cambios de localización (posición más orientación) se expresa un punto dado en el espacio homogéneo y se les aplica una Transformación correspondiente a movimientos de giro, traslación, giro seguido de traslación o traslación seguida de giro. Nueve (9) expresiones matriciales como la mostrada (Ecuación 1: rotación un ángulo alfa alrededor de eje X, seguida de una traslación determinada por el vector de componentes  $px, py, pz$ ), definen estas transformaciones; y para su aplicación se han elaborado igual número de *scripts* ejecutables en la ventana de comandos de GNU Octave (Eaton et al., 2021). Los comandos puestos a disposición para los estudiantes se muestran en la Tabla 1.

$$T((x, \alpha), p) = \begin{bmatrix} 1 & 0 & 0 & px \\ 0 & C\alpha & -S\alpha & py \\ 0 & S\alpha & C\alpha & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

**Tabla 1**

*Funciones para el uso de Transformaciones Homogéneas (TH) en GNU Octave*

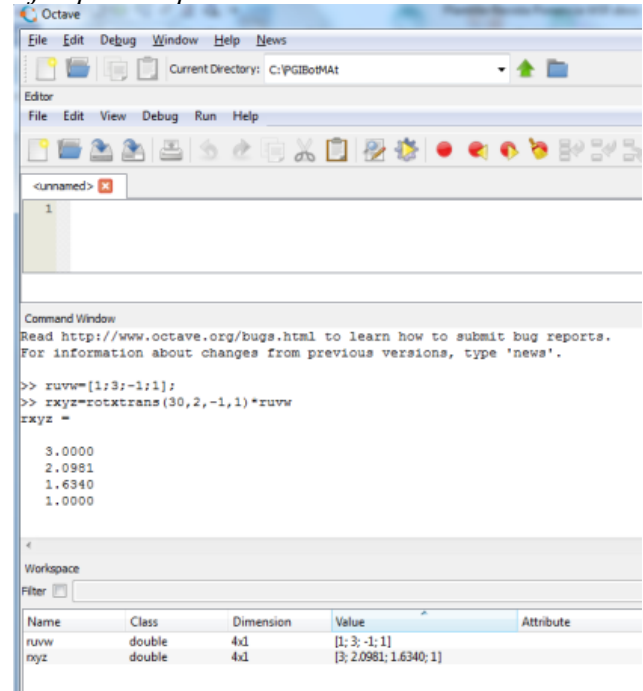
Comando	Descripción
<b>Rotx (alpha)</b>	Retorna la Transformación Homogénea de rotación en el eje X, un ángulo alpha.
<b>Roty (phi)</b>	Retorna la Transformación Homogénea de rotación en el eje Y, un ángulo phi.
<b>Rotz (theta)</b>	Retorna la Transformación Homogénea de rotación en el eje Z, un ángulo theta.
<b>Rotxtrans (alpha,px,py,pz)</b>	Retorna la Transformación Homogénea de rotación en el eje X, un ángulo alpha, seguida por una traslación con un vector p.
<b>Rotytrans (phi,px,py,pz)</b>	Retorna la Transformación Homogénea de rotación en el eje Y, un ángulo phi, seguida por una traslación con un vector p.
<b>Rotztrans (theta,px,py,pz)</b>	Retorna la Transformación Homogénea de rotación en el eje Z, un ángulo theta, seguida por una traslación con un vector p.
<b>Transrotx (px,py,pz,alpha)</b>	Retorna la Transformación Homogénea de traslación con un vector p, seguida por una rotación alrededor del eje X, un ángulo alpha.
<b>Transroty (px,py,pz,phi)</b>	Retorna la Transformación Homogénea de traslación con un vector p, seguida por una rotación alrededor del eje Y, un ángulo phi.
<b>Transrotz (px,py,pz,theta)</b>	Retorna la Transformación Homogénea de traslación con un vector p, seguida por una rotación alrededor del eje Z, un ángulo theta.

*Nota.* Sintaxis y descripción de Transformaciones Homogéneas usadas para cálculos de cambios en posiciones y orientaciones, elaborado por Sanz (2016).

Por ejemplo, si desea determinar el resultado de aplicar una rotación de 30°, alrededor del eje X, seguido de una traslación con 2 unidades en “X”, -1 en “Y”, 1 en “Z”, sobre un punto ruvw = (1,3,-1); el procedimiento a seguir en GNU Octave puede observarse en la Figura 2. El uso del comando exige expresar el punto dado como vector columna, e incluir el factor de escalado “1”, a modo de cuarta componente. El resultado es un vector en el espacio homogéneo, cuyo equivalente en el espacio cartesiano es rxyz = [3, 2.0981, 1.634].

**Figura 2**

*Ejemplo de aplicación de una TH en GNU Octave*



```

>> ruvw=[1;3;-1;1];
>> rxyz=rotxtrans(30,2,-1,1)*ruvw
rxyz =
    3.0000
    2.0981
    1.6340
    1.0000

Workspace
Filter
Name      Class      Dimension  Value
ruvw      double    4x1        [1; 3; -1; 1]
rxyz      double    4x1        [3; 2.0981; 1.6340; 1]
    
```

*Nota.* Se muestran dos transformaciones consecutivas (rotación de 30°, seguida una traslación conforme al vector 2,-1,1) resumidas en un solo comando: *rotxtrans*, elaborado por Sanz (2016).

### Parámetros de Denavit-Hartenberg

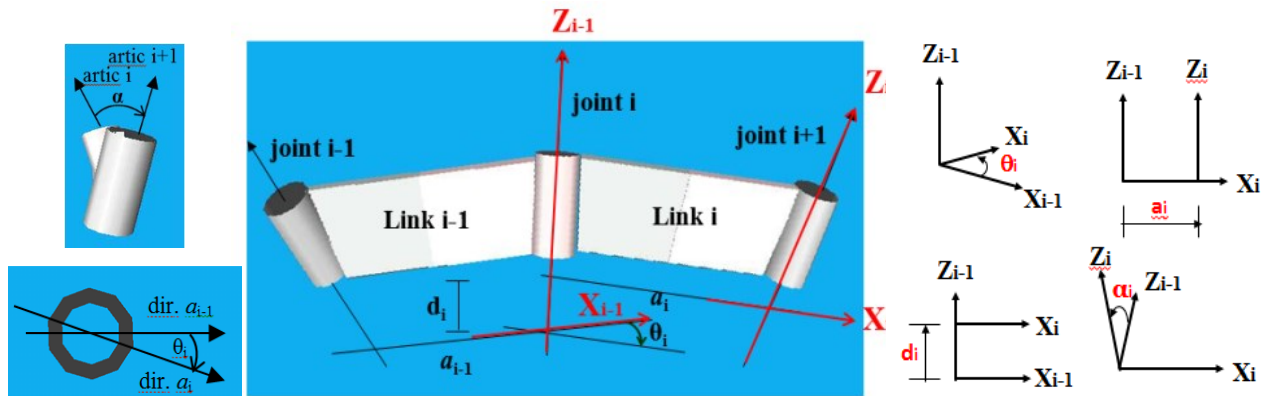
Un problema clásico en Robótica Industrial es modelar su cinemática directa. Solucionarlo consiste en obtener una matriz, llamada Matriz de Transformación T del robot, que permita describir la posición y orientación adoptada por un objeto manipulado, luego de variar los parámetros de sus articulaciones. Según Craig J. (2006) “cualquier robot puede describirse en forma cinemática proporcionando los valores de cuatro cantidades para cada vínculo” (p. 67); sin embargo, esa simple afirmación no es tan inocente como parece, pues los parámetros o cantidades referidos tienen definiciones específicas demandantes de cuidadosamente interpretación. Todo se resume en usar las convenciones establecidas por Denavit-Hartenberg (Denavit, Hartenberg, 1955) y aplicar el algoritmo del mismo nombre (Fu, et al., 1988) aunque debe tenerse en cuenta la existencia de métodos alternativos (Chennakaseva, 2014).

Tal algoritmo exige identificar, en cada articulación de una cadena cinemática a los cuatro parámetros mostrados en la Figura 3 (llamados

parámetros DH):  $\theta$ , el ángulo de giro en articulaciones de rotación (tipo R),  $d$ , desplazamiento en las articulaciones prismáticas (tipo P),  $a$ , distancia de la recta normal o

simultáneamente perpendicular a ambos ejes que comprenden a un eslabón; y  $\alpha$ , ángulo de torsión del eslabón.

**Figura 3**  
Parámetros DH



Nota. Representación de los 4 parámetros DH en elementos consecutivos de una cadena cinemática. Se añaden representaciones gráficas de sus definiciones, elaborado por Sanz (2016).

El movimiento relativo entre eslabones consecutivos se calcula mediante la llamada Matriz de Transición o Matriz de Paso Homogénea (Ecuación 2). El producto (postmultiplicaciones) de todas las matrices de transición correspondientes a enlaces consecutivos, permite determinar la Matriz de Transformación del robot (Ecuación 3), única en su forma algebraica para cada modelo, debido a su geometría y configuración.

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & aC\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & aS\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2)

$$T = A_1^0 * A_2^1 *** A_n^{n-1}$$

(3)

La nomenclatura utilizada (Ecuación 2) alude a las funciones trigonométricas Seno y Coseno, mediante los literales S y C, respectivamente. Ahora bien, entre el conjunto de comandos ejecutables en GNU Octave utilizar *mpash.m* ahorra un trabajo significativo. Este

comando requiere introducir los valores numéricos de los parámetros DH para cada articulación arrojando como resultado la Matriz de Transición correspondiente (Ecuación 4).

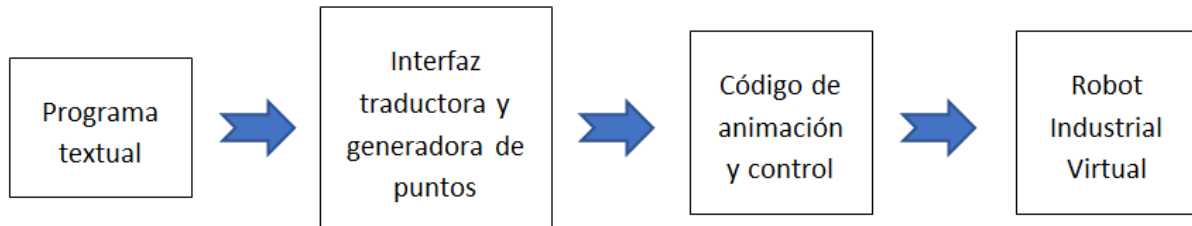
$$A_i^{i-1} = mpash(\theta_i, d_i, a_i, \alpha_i)$$

(4)

## Resultados

El apoyo en los recursos tecnológicos para facilitar el aprendizaje es un factor importante en la educación actual. Al respecto, Mujica asevera: “Aquí la tecnología es un medio que se pone al alcance tanto de los estudiantes como de los docentes. Su uso enriquece y facilita el entendimiento de áreas que por su complejidad son más inteligibles con la ayuda del material multimedia” (2019, p. 7). Sobre tales ideas se procedió a diseñar un sistema cuya función es proveer a sus usuarios con las herramientas para aprender y practicar los fundamentos de la programación de Robots Industriales, sin tener la interacción con autómatas físicos (Figura 4).

**Figura 4**  
*Sistema para programación de robots virtuales*

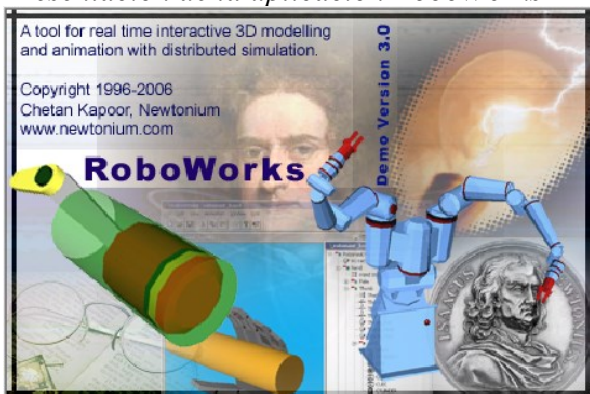


*Nota:* Cada elemento del diagrama es objeto de una implementación descrita posteriormente, elaborado por Sanz (2016).

La simulación es “una metodología de análisis de sistemas basada en la construcción de un modelo típicamente implementado en un ordenador, que describe el comportamiento del sistema y que permite generar observaciones dadas ciertas entradas” (Ríos, Ríos & Martín, 2000, p. 9). Por su parte, los RI comerciales por lo general cuentan con un entorno de simulación 3D para elaborar y probar programas en robots virtuales (Koehler, 2020). El uso de tal herramienta implica ventajas significativas en los procesos de aprendizaje, con una importancia obvia si se trata de actividades prácticas, resultando además en una alternativa idónea cuando el estudiante no tiene acceso a un laboratorio e interactúa a distancia con su profesor.

El simulador elegido es RoboWorks Demo 3.0 (Kappor, 2006), una aplicación libre (la versión completa es paga) con total funcionalidad para trabajar con modelos virtuales, escenarios y archivos editables al reproducir animaciones (Figura 5). Su capacidad de simular depende del ingreso de archivos editables o datos mediante comunicación por puerto serial (estándar RS-232 compatible), pero solo se usa la primera opción.

**Figura 5**  
*Presentación de la aplicación RoboWorks*



*Nota:* Captura de pantalla de la ventana correspondiente al menú textual, elaborado por Kapoor (2006).

Para ejecutar animaciones en RoboWorks es necesario usar como entrada un archivo con un formato específico (extensión *dat*). Tal archivo se genera a partir de otro más simple (extensión *try*) mediante una Interfaz Gráfica de Usuario (GUI) PGIBOTMat, la cual se ejecuta en GNU Octave. La relación de los elementos nombrados con los ficheros de entrada y salida se muestra en la Figura 6.

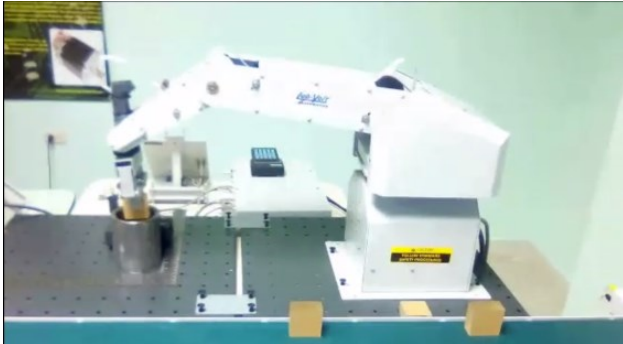
**Figura 6**  
*Herramientas para la programación de Robots Virtuales en simulador 3D*



*Nota:* La Infografía representa a la Interfaz Gráfica de Usuario PGIBOTMat 3.0 como traductor de un archivo con comandos típicos de programación de un RI a la versión requiere el software RoboWorks Demo 3.0, para simular la ejecución de una tarea, elaborado por Sanz (2016).

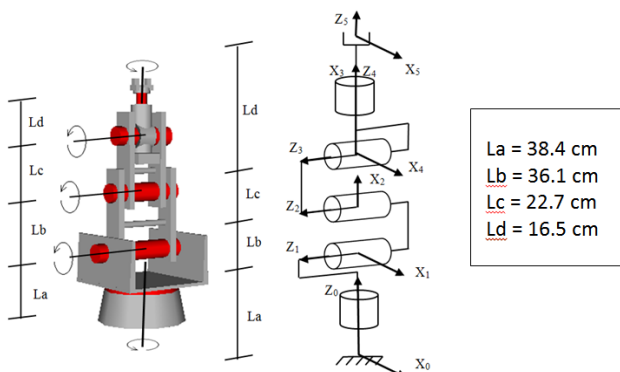
Para mostrar el uso de las herramientas descritas se toma como caso de estudio al robot antropomórfico Lab-Volt 5150 (Festo, 2021) del Laboratorio de Robótica de la Universidad José Antonio Páez (Figura 7). Se inicia el estudio a partir de una representación simplificada (Figura 8), se determina la Tabla de parámetros DH (Tabla II) y se calcula su Matriz de Transformación T mediante un sencillo *script* como el indicado ( $\theta_1=0^\circ$ ,  $\theta_2=90^\circ$ ,  $\theta_3=0^\circ$ ,  $\theta_4=-90^\circ$ ,  $\theta_5=0^\circ$ ).

**Figura 7**  
*Robot Lab-Volt 5150*



*Nota:* Imagen capturada en el Laboratorio de Robótica Industrial de la Universidad José Antonio Páez, durante una práctica de Toma y Colocación de objetos. Presentada aquí con el permiso de la institución, elaborado por Sanz (2016).

**Figura 8**  
*Representación funcional del robot 5R Lab-Volt 5150*



*Nota:* Se muestran los ejes de sistemas de coordenadas generalizadas definidos conforme a las convenciones de Denavit Hartenberg. Las dimensiones indicadas se obtuvieron de mediciones directas hechas sobre el robot, elaborado por Sanz (2016).

**Tabla 2**  
*Parámetros DH del Robot Lab-Volt 5150*

Articulación	$\theta$	$d$	$a$	$\alpha$
1	q	38.4	0	90
2	q	0	36.1	0
3	q	0	22.7	0
4	q	0	0	-90
5	q	16.5	0	0

*Nota:* Los parámetros tabulados se han obtenido a partir de las definiciones mostradas en la Figura 3, aplicándolas al robot 5R mostrado en la Figura 8, elaborado por Sanz (2016).

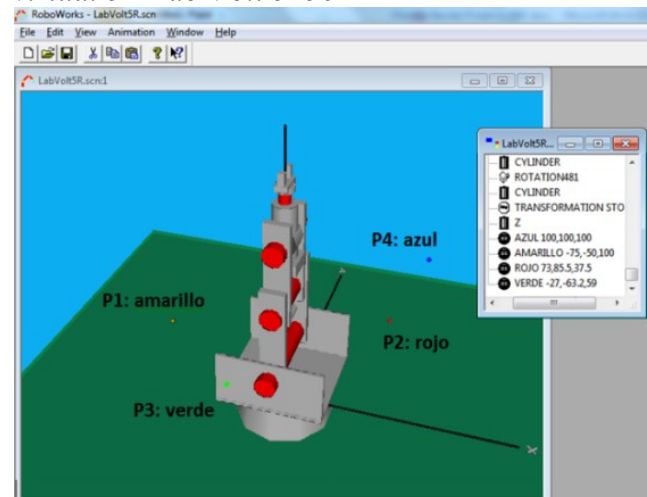
```
>> A01=mpash(0,38.4,0,90);
>> A12=mpash(90,0,36.1,0);
>> A23=mpash(0,0,22.7,0);
>> A34=mpash(-90,0,0,-90);
>> A45=mpash(0,16.5,0,0);
>> T=A01*A12*A23*A34*A45
```

$$T = \begin{bmatrix} 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 1.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 1.00000 & 113.70000 \\ 0.00000 & 0.00000 & 0.00000 & 1.00000 \end{bmatrix}$$

Resultados como el obtenido incluyen en la última columna de la Matriz T las componentes de la nueva posición alcanzada (0 en X, 0 en Y, 113.7 cm en Z), pudiendo verificarse la exactitud del modelo al introducir los ángulos de entrada en la interfaz de control del robot o, en el caso de interés, a través de un simulador adecuado.

La programación de tareas consiste en la enseñanza de puntos al robot virtual Lab-Volt5R dispuesto en un escenario de RoboWorks. Una tarea se asocia a trayectorias y acciones necesarias dentro de un proceso, las cuales pueden o no requerir un orden específico de ejecución. En este caso se pretende alcanzar una serie de puntos de colores (Figura 9), prestando atención a una secuencia específica: amarillo, rojo, verde y azul.

**Figura 9**  
*Escenario para tarea a ejecutar por el robot virtual 5R Lab-Volt 5150*

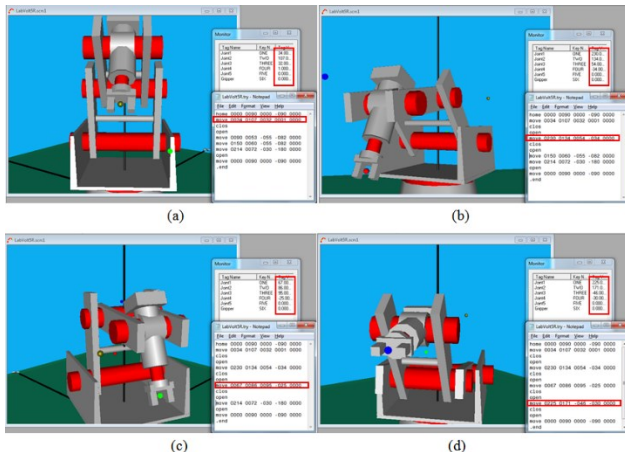


*Nota:* Los valores indicados en la ventana pequeña dentro de la imagen son las coordenadas de cada punto en relación con el sistema XYZ con origen en la base del Robot. Escenario elaborado en RoboWorks Demo 3.0, elaborado por Sanz (2016).

Para enseñar puntos se hace del mismo modo aplicado a cualquier RI físico: interactuando en una Interfaz Gráfica (o a través de un *Teach pendant*) para almacenar en un archivo las combinaciones de las coordenadas articulares se alcanzan puntos clave con la herramienta de agarre (TCP). El fichero de

extensión *try* se edita con líneas de comandos comunes a casi cualquier lenguaje de programación de RI comerciales, tomando nota de las coordenadas visibles en la ventana *Monitor* de RoboWorks (Figura 10) en un editor de texto.

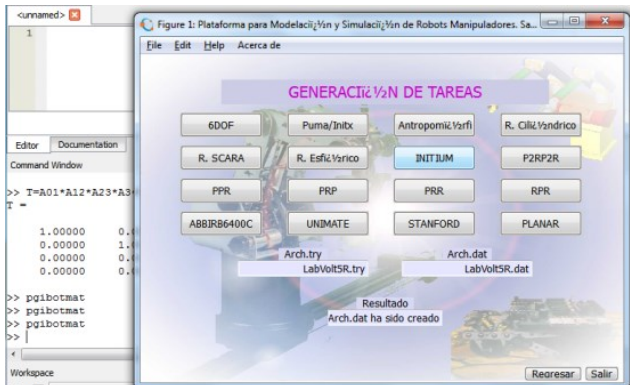
**Figura 10**  
Secuencia de programación para el robot virtual 5R Lab-Volt 5150



*Nota:* a) Toma del 1er punto. b) Toma del 2do punto. c) Toma del 3er punto d) Toma del 4to punto. El proceso mostrado emula el trabajo que se realiza en un robot físico, mediante la interfaz conocida como *Teach Pendant*, elaborado por Sanz (2016).

Una vez guardado el archivo *LabVoltR.try*, se ingresa a la Interfaz Gráfica de Usuario PGIBOTMat (dentro de Octave GNU) y se presiona el botón *Gen. Tar.* Emerge una ventana donde se introduce el nombre del archivo *try* y se nombra al archivo de salida (*dat*). La operación se completa presionando el botón *Initium*, correspondiente a un robot 5R similar al Lab-Volt 5150 (Figura 11).

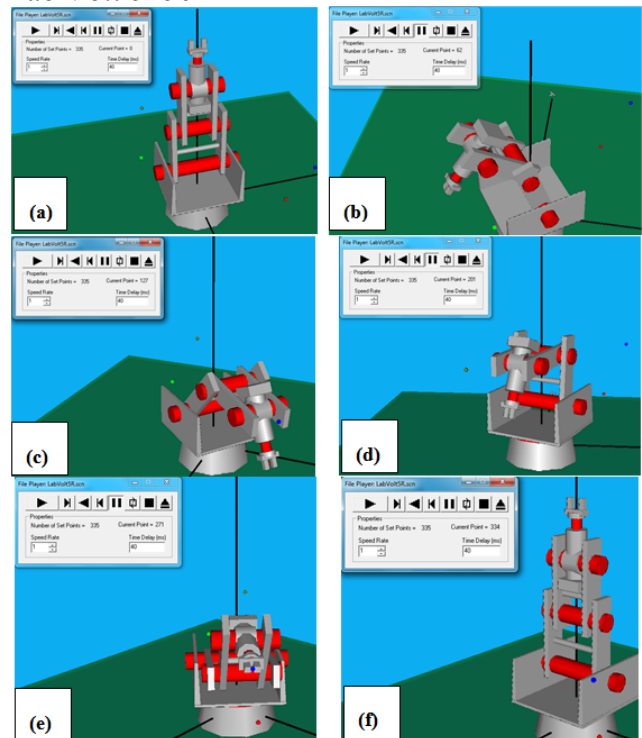
**Figura 11**  
Generación del archivo de salida *LabVolt5R.dat*



*Nota:* Se muestra la ventana “Generación de Tareas” de la GUI PGIBOTMat, dentro del entorno GNU Octave, elaborado por Sanz (2016).

Finalmente, el archivo generado (*LabVolt5R.dat*) se introduce mediante un menú de animación y se ejecuta en el simulador RoboWorks Demo. Se puede comprobar que el robot sigue la secuencia de trabajo esperada, sin embargo como no es posible insertar aquí un vídeo, en la Figura 12 se captura una sucesión representativa de los 335 cuadros generados.

**Figura 12**  
Simulación de la tarea programada para el robot Lab-Volt 5150



*Nota:* a) Home. b) Move al 1er punto. c) Move al 2do punto. d) Move al 3er punto. e) Move al 4to punto. f) Move al Home, elaborado por Sanz (2016).

## Conclusiones

La modelación cinemática de Robots Industriales, basada en el algoritmo de Denavit-Hartenberg, implica el manejo de un álgebra matricial susceptible de simplificarse sustancialmente mediante el uso de las herramientas contenidas en la plataforma PGIBOTMat. Tal Toolbox funciona también como Interfaz Gráfica de Usuario para traducir programas creados mediante la técnica de enseñanza de puntos a un formato aceptado por el ambiente de simulación RoboWorks; requiriendo únicamente cuidar una sintaxis y usar comandos similares a los de la mayoría de los lenguajes de



programación de Manipuladores Industriales comerciales.

Los resultados logrados al simular demuestran la exactitud de los modelos cinemáticos obtenidos mediante formulaciones matemáticas, lo cual evidencia la utilidad de las herramientas descritas para el aprendizaje a distancia basado en la práctica. En adición, es remarcable su carácter gratuito, dejando al alcance de cualquier estudiante con la formación básica, computadoras y acceso a internet, la posibilidad de utilizarlas en distintos escenarios virtuales, según se dispongan.

Recomendación pertinente a las instituciones educativas que imparten cursos de Robótica Industrial enmarcados en las modalidades distante, presencial o mixta, mediante plataformas como Moodle, edX Google Classroom o similares, es que procuren incorporar el uso de Robots de última generación (COBOTS o Sensitivos). Gracias al aporte de fabricantes y desarrolladores (Universal Robots, por ejemplo) que ofrecen gratuitamente nuevos entornos de simulación, para cualquier entidad o persona dispuesta a descargarlos desde su sitio web, es posible contar con un laboratorio de Robótica en casa (virtual, claro está). Si bien no se facilita el trabajo de la modelación matemática, se garantiza la completa transferibilidad de los programas generados a los autómatas físicos.

## Referencias

- Arango, C. (2006). *Exploración de la metodología Top-Down Design en sistemas CAD para el desarrollo de un modelo de ingeniería*. [Trabajo de Grado, Universidad EAFIT]. <https://repository.eafit.edu.co/handle/10784/276>
- Chennakaseva, A. (2014). *Difference between Denavit-Hartenberg (D-H) classical and modified conventions for Forward Kinematics of robot with case study*. International Conference on Advanced Materials and manufacturing (AMMT). JNTUH College of Engineering Hyderabad.
- Craig, J. (2006). *Robótica* (3era ed.). Prentice Hall
- Denavit, J. & Hartenberg, R. (1955). A kinematic notation for Lower-pair Mechanisms based on matrices. *Transactions ASME Journal of Applied Mechanics*, pp. 215–221.
- Eaton, J., Bateman, D., Hauberg, S. & Wehbring, R. (2021). GNU Octave (6.2.0) [Software]. <https://www.gnu.org/software/octave/download>
- Fu, K., González, R. & Lee, C. (1988). *Robótica: Control, Detección, Visión e Inteligencia*. McGraw-Hill.
- González, E. (2014). *La Simulación como herramienta de análisis de procesos*. <https://www.milenio.com/opinion/varios-autores/universidad-politecnica-de-tulancingo/la-simulacion-como-herramienta-de-analisis-de-procesos>
- Koehler, M., Usevitch, N.S. & Okamura, A.M. (2020). Model-Based Design of a Soft 3-D Haptic Shape Display. *IEEE Transactions on Robotics* 36(3), pp.613-628
- Mujica, R. (2019). La tecnología en la educación. *Revista Tecnológica-Educativa Docentes* 2.0, 4(4), 4–7. <https://ojs.docentes20.com/index.php/revista-docentes20/article/view/57>
- Ollero, A. (2007). *ROBÓTICA. Manipuladores y robots móviles*. Alfaomega - Marcombo.
- Pardo, F. (1997). *VHDL. Lenguaje para descripción y modelación de circuitos*. Universidad de Valencia
- Ríos, I., Ríos, S. & Martín, J. (2000). *SIMULACIÓN. Métodos aplicaciones*. Alfaomega.
- Salazar, R. (2015). *Desarrollo de las metodologías en la modalidad de educación a distancia*. <https://aprenderenlasociedaddigital.blogspot.com/>
- Sanz, W. (2016). *Cinemática de Robots Industriales: Un enfoque pedagógico para el aprendizaje de la Robótica Industrial*. (3era ed.). Editorial Académica Española.