

UCLV
Universidad Central
"Marta Abreu" de Las Villas



FIE
Facultad de
Ingeniería Eléctrica

TRABAJO DE DIPLOMA

Departamento de Automática y Sistemas Computacionales

Título Evaluación de una propuesta de software como herramienta de apoyo a la docencia de robótica en la carrera Ingeniería Automática de la UCLV.

Autores Alexander Rodríguez Conte

Tutores Dr. Luis Hernández Santana

Santa Clara, julio 2018
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

**Evaluación de una propuesta de software como
herramienta de apoyo a la docencia de robótica en
la carrera Ingeniería Automática de la UCLV.**

Escriba aquí el Título de la Tesis

Autor: Alexander Rodríguez Conte

Tutor: Dr., Luis Hernández Santana

Santa Clara

2018

"Año 60 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

**Evaluación de una propuesta de software como
herramienta de apoyo a la docencia de robótica en
la carrera Ingeniería Automática de la UCLV.**

Autor: Alexander Rodríguez Conte

arconte@uclv.cu

Tutor: Dr., Luis Hernández Santana

Profesor titular, Departamento de Automática y Sistemas
Computacionales, Facultad de Ingeniería Eléctrica,

luishs@uclv.edu.cu

Santa Clara

2018

"Año 60 de la Revolución"

PENSAMIENTO

Es pues la fe, certeza de lo que se espera, convicción de lo que no se ve.

Hebreos 11.1, La Biblia

DEDICATORIA

A mis padres, que tanto lucharon por esto. A mi hermana que siempre me ayudo. A mi esposa, por su total comprensión y amor. A mis amigos, porque ellos también son parte de este logro.

AGRADECIMIENTOS

A Dios primeramente por su sustento. A mis padres y mi hermana porque ellos siempre esperaron que ocurriera esto. A mi esposa por ser mi ayuda idónea. A los profesores a lo largo de mi vida porque ellos me instruyeron.

RESUMEN

El uso de simuladores para la enseñanza de la robótica ha tenido un aumento debido al creciente interés de usar mejor esa tecnología en favor del desarrollo humano. En la carrera Ingeniería Automática de la Universidad Central “Marta Abreu” de Las Villas, se imparte la asignatura Robótica que actualmente carece de un simulador de robótica industrial que permita apoyar la docencia. En esta investigación se analizan diferentes simuladores que puedan usarse en dicha asignatura con énfasis en los que son gratuitos, sencillos en cuanto a su uso y en la programación y sobre todo que cumplan con los objetivos de la asignatura. Entre ellos ARTE destacó por su baja complejidad para usarse, ser gratuito y la posibilidad de abordar todos los contenidos que forman parte de la asignatura Robótica, incluyendo algunos que actualmente no se pueden impartir. Este proyecto muestra una solución viable ante la carencia de un simulador para la docencia y la investigación de la robótica industrial en la UCLV.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
RESUMEN	iv
INTRODUCCIÓN	1
Organización del informe	5
CAPÍTULO 1. LA ROBÓICA INDUSTRIAL. EJEMPLOS DE SIMULADORES	6
1.1 Conceptos relacionados con la robótica	7
1.1.1 Robótica	7
1.1.2 Representacion espacial.	8
1.1.3 Cinemática	10
1.1.4 Trayectoria	12
1.1.5 Dinámica	12
1.2 Simuladores de Robótica industrial (manipulación)	13
1.2.1 RoboAnalyzer (RA)	14
1.2.2 RoboDK	16
1.2.3 Cosumir Educational.....	17
1.2.4 ROBOLAB (RobUAlab)	18

1.2.5	Virtual Robot Experimentation Platform (VREP).....	19
1.2.6	Robotics Toolbox de Peter Corke	21
1.2.7	A Robotic Toolbox for Education (ARTE)	22
1.3	Consideraciones finales del capítulo.	24
CAPÍTULO 2. ANÁLISIS DE ARTE PARA LA DOCENCIA.		25
2.1	Instalación y elementos básicos	25
2.1.1	Instalacion	25
2.1.2	Cargar un robot y su representación CAD	27
2.1.3	Representación espacial	29
2.2	Cinemática.....	30
2.2.1	Cinemática directa	31
2.2.2	Cinemática inversa.....	33
2.3	Generación de trayectorias	35
2.4	Dinámica	36
2.4.1	Dinámica directa	37
2.4.2	Dinámica inversa	38
2.5	Programación (RAPID).....	38
2.5.1	Pequeño ejemplo.....	40
2.6	Simulación de controladores	40
2.7	Consideraciones finales del capítulo	42
CAPÍTULO 3. IMPLEMENTACIÓN DE LA ASIGNATURA ROBÓTICA CON ARTE.		43
3.1	Contenidos de cinemática	43
3.1.1	Representación espacial.....	43
3.1.2	Cinemática directa	46

3.1.3	Cinemática inversa.....	47
3.1.4	Generación de trayectoria	48
3.2	Contenidos de dinámica	50
3.2.1	Dinámica directa	50
3.2.2	Dinámica inversa	51
3.3	Programación de robots y simulación de controladores	52
3.3.1	Programación de robots	52
3.3.2	Simulación de controladores.....	53
3.4	Análisis económico medioambiental	54
3.5	Consideraciones finales del capítulo.....	54
CONCLUSIONES Y RECOMENDACIONES		56
Conclusiones.....		56
Recomendaciones		56
REFERENCIAS BIBLIOGRÁFICAS		58
ANEXOS		60
Anexo I	Parámetros de la variable robot	60
Anexo II	Contenido de robots presentes en el simulador.....	61
Anexo III	Programa completo de manipulación de objetos.	62

INTRODUCCIÓN

Desde el pasado siglo la robótica se ha convertido en uno de los pilares del desarrollo económico y social en el mundo. Según el Instituto Norteamericano de Robótica, robot es “un manipulador multifuncional y reprogramable diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados y variables que permiten llevar a cabo diversas tareas”.

La robótica se divide en dos áreas fundamentales según su aplicación: robótica industrial y de servicios. La Federación Internacional de Robótica (IFR, siglas en inglés) usa el término robótica industrial dado por la Organización Internacional de Estandarización (ISO, siglas en inglés) como: “un manipulador controlado automáticamente, reprogramable, multipropósito, en tres o más ejes” (ISO 8373). Estos robots pueden ser clasificados según su estructura mecánica en articulados, cilíndricos, lineales, paralelos o robots SCARA. Por otro lado, para la robótica de servicios, la IFR usa el término definido por la ISO: “un robot que realiza útiles tareas humanas o de equipos excluyendo las aplicaciones industriales”, (ISO 8373). Según la norma, los robots requieren de un grado de automatización, referido a la habilidad de realizar tareas basadas en el estado corriente sin intervención humana. Los robots de servicio son clasificados según el uso: personal o profesional.

La presencia de los robots en la sociedad aumenta cada año. Según la IFR en el 2016 los robots industriales tuvieron un crecimiento de 16% en las ventas de los mismos respecto al año 2015, lo que representa que se comercializaron 294,312 unidades, fundamentalmente en la industria de la electrónica y eléctrica, en países asiáticos como China, Japón y Corea del Sur (IFR, 2017). En el caso de la robótica de servicios, según la misma fuente, se vendieron 59,706 unidades, un 24% mayor a lo vendido en 2015, fundamentalmente en

aplicaciones como los sistemas de defensa, relaciones públicas, médicas, entre otras (Corke, 2017).

Entre las principales aplicaciones de la robótica, se encuentran las militares, las médicas, las industriales (manipulación, soldadura, pintura, ensamblaje, carga y descarga, entre otras), las científicas, las de vigilancia, seguridad, labores de rescate y las civiles. La presencia de los robots en la sociedad actual es cada vez más creciente por lo que muchos planes de estudio han incluido la enseñanza de la robótica como parte de sus currículos en primaria, secundaria y universidad, involucrando áreas del conocimiento como las matemáticas, la física, la electrónica, la mecánica y la informática. Algunos de estos planes no buscan formar un especialista en robótica, sino aprovechar la interdisciplinariedad de la misma para elevar el aprendizaje colaborativo. A su vez, otros planes sí profundizan en aspectos importantes del estudio de la robótica como son la cinemática y la dinámica, la programación *online* y *offline* y el diseño y control de robots en diferentes ambientes, con el objetivo de formar a personas capacitadas para el correcto uso y explotación de la robótica, que posibilite el desarrollo económico y social de su país. Estos contenidos por su complejidad resultan difíciles de enseñar sin el empleo de simuladores, de ahí la importancia de estas herramientas informáticas para la enseñanza de la robótica (Sánchez and Guzmán, 2012).

Diferentes universidades donde se estudian carreras vinculadas con la robótica emplean diversas estrategias para la docencia e investigación. Softwares como ROBOLAB, Robot Operating System (ROS), RoboAnalyzer (RA), RobotScene, RoboDK, VREP y A Robotic Toolbox for Education (ARTE) son una muestra de la gran variedad de simuladores destinados a la enseñanza de la robótica en el ámbito universitario en países como España, Estados Unidos y la India.

En nuestro país, a pesar del discreto impacto de la robótica en los flujos productivos, existen experiencias que demuestran nuestra inevitable inserción en la explotación de estas novedosas tecnologías. Ejemplo de ello son los proyectos que desarrollan centros de altos estudios como la Universidad de la Habana (UH), el Instituto Superior Politécnico José Antonio Echevarría (ISPJAE) y la Universidad de Ciencias Informáticas (UCI), en investigaciones sobre la navegación en la robótica móvil, la creación de un robot

explorador en zonas de derrumbe controlado mediante dispositivos móviles y el Sistema robótico ZEUS para operaciones quirúrgicas (Aguilera et al., 2014). También destacan los resultados obtenidos por la Universidad Central “Marta Abreu” de Las Villas (UCLV) en el estudio de la robótica serie y paralela y la robótica móvil, manifestado en trabajos de control visual en robots industriales (Hernández et al., 2008, González-Rodríguez et al., 2011), el modelado y control de plataformas de dos y tres Grados de Libertad (GDL) basadas en robótica paralela (Izaguirre et al., 2011, Izaguirre Castellanos, 2012), aplicaciones desarrolladas en robótica móvil (Garcia Garcia et al., 2015), entre otros trabajos. Además resaltan las investigaciones de la Universidad de Oriente (UO) en el estudio de la manipulación y control de robots (Mazaira-Morales and Chang-Mumáñ, 2010).

Por tanto, la importancia del estudio de la robótica en nuestro país es evidente. Al contar con una herramienta informática que permita a los investigadores nacionales desarrollar sus aplicaciones, al menos en un ambiente de simulación, garantizará un aumento de la eficiencia y eficacia en la correcta explotación de la robótica para el desarrollo económico y social del país.

En ese sentido, la carrera Ingeniería Automática de la UCLV, imparte un curso optativo de robótica, donde se abordan aspectos como la robótica industrial y móvil, su impacto en la sociedad, las leyes físicas que se cumplen (cinemática y dinámica) y las estrategias de control que pueden ser implementadas, todo en un ambiente de simulación por no contar con robots para llevar a cabo experimentos reales. El simulador usado hasta el momento es el software *Virtual Robot Simulator*, desarrollado por el grupo de investigación en robótica (<http://robotica.isa.upv.es>) del Departamento de Ingeniería de Sistemas y Automática (DISA) de la Universidad Politécnica de Valencia (UPV).

Este software es una aplicación de modelado gráfico que permite la simulación de robots en diferentes ambientes. Presenta una herramienta para el control de robots, simulando un *Teach Pendant* con el que se realizan movimientos del robot de forma articular y cartesiana y una herramienta para la manipulación de piezas en el entorno del robot llamada *Part Handling*. Con esas herramientas es posible realizar además, un análisis de la cinemática directa e inversa respectivamente (Arteche, 2004).

La principal dificultad que tiene el uso del software *Virtual Robot Simulator* es que la actualización del mismo no se realiza desde el año 2004, lo que dificulta su uso en cursos recientes por su dificultad de instalación en las últimas versiones de los sistemas operativos.

Por tanto, en estos momentos no se cuenta con un simulador de robots con el que se pueda impartir las clases de la asignatura Robótica de la carrera Ingeniería Automática de la UCLV, aspecto que constituye la base del problema científico que se propone resolver con esta investigación.

Objetivo general:

Evaluar una propuesta de software de simulación para ser usado como herramienta en la enseñanza de la robótica en la UCLV.

Objetivos específicos:

- Establecer las bases teóricas que permitan seleccionar un simulador para la enseñanza de la robótica.
- Seleccionar un software para impartir las clases de robótica en la carrera Ingeniería Automática de la UCLV.
- Analizar el software seleccionado en torno a los contenidos impartidos en la asignatura Robótica.
- Implementar los contenidos de las diferentes unidades de la asignatura para evaluar su aplicación en la impartición de la teoría y la práctica.

Hipótesis

Existiendo la posibilidad de seleccionar un software que pueda ser usado como herramienta de apoyo en las clases de robótica en la carrera Ingeniería Automática de la UCLV, se podrán cumplir con los objetivos de la investigación.

El resultado de esta tesis dotará a la carrera Ingeniería Automática de la UCLV de un simulador para ser usado tanto en la docencia como en la investigación, lo que permitirá la explotación de la robótica de una forma más científica y eficiente para el desarrollo económico social de nuestro país.

Organización del informe

La organización del informe de esta investigación tendrá la siguiente estructura

- A. Introducción
- B. Capítulos
 - a. CAPITULO I: En este capítulo se caracterizarán algunos de los softwares usados para impartir clases de robótica en diferentes universidades en la actualidad, finalizando con una propuesta de simulador para las clases de robótica de la UCLV.
 - b. CAPITULO II: En este capítulo se expondrá de forma sintetizada la manera de usar el software seleccionado como herramienta para la docencia en la asignatura Robótica de la UCLV.
 - c. CAPAITULO III: Este capítulo se dedicará a exponer una guía para el uso del software seleccionado en las clases de robótica de la UCLV, demostrando así su aplicabilidad para la docencia en dicha Universidad.
- C. Conclusiones
- D. Recomendaciones
- E. Referencias Bibliográficas y/o Bibliografía
- F. Anexos

CAPÍTULO 1. LA ROBÓTICA INDUSTRIAL. EJEMPLOS DE SIMULADORES

Existen un gran número de simuladores para ser usados en la enseñanza de la robótica. Estos pueden ser de uso propietario, desarrollados para robots específicos y otros tienen la posibilidad de ser explotados de forma gratuita. Asimismo, la principal característica que estos simuladores presentan para la enseñanza es la de permitir el análisis y entendimiento de complejos conceptos presentes en el estudio de la robótica.

La asignatura Robótica impartida en la UCLV aborda importantes tópicos presentes en el estudio de esa ciencia de modo que los alumnos dominen elementos cruciales para el empleo de la robótica. Tópicos como cinemática (directa e inversa), dinámica (directa e inversa), representación espacial, control de robots, diseño de controladores y programación de robots, son impartidos en la asignatura, sobre todo haciendo énfasis en la robótica de manipulación. Por tanto, en este capítulo se plasmarán conceptos importantes de la robótica presentes en el plan de estudio de la UCLV y se abordarán los resultados del Marco Teórico sobre diferentes simuladores usados a nivel mundial para impartir clases de robótica en la enseñanza universitaria.

La revisión se orienta a señalar las principales características de los softwares encontrados que se destinen a la robótica de manipulación. Se demuestra cuál simulador puede ser usado en la asignatura Robótica de la UCLV como una útil herramienta educativa. Se enfatiza en sistemas exentos de pago, lo cual permite su empleo sin la necesidad de una licencia y que usen lenguajes de programación que se hayan empleado a lo largo de la carrera, principalmente Matlab.

1.1 Conceptos relacionados con la robótica

La robótica como ciencia depende de otras disciplinas como la Matemática, la Física, la Informática y la Electrónica para poder realizar un análisis preciso de la misma. A continuación, se enuncian varios conceptos necesarios en el estudio de la robótica, los cuales, además, rigen el plan docente empleado en la UCLV para el estudio de esta ciencia. El análisis de estos conceptos servirá para seleccionar un simulador de robótica de manipulación útil para la asignatura Robótica de la UCLV.

1.1.1 Robótica

La robótica es la ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia.

Un **robot** es una máquina controlada por un computador y programada para moverse, manipular objetos y realizar trabajos a la vez que interacciona con su entorno. Los robots son capaces de realizar tareas repetitivamente de forma más rápida, barata y precisa que los seres humanos.

Manipulador: según la Asociación Francesa de Normalización (AFNOR) es un “mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante un dispositivo lógico”. Además cada articulación se encuentra ubicada respecto a la anterior hasta la base y al final de la cadena se le llama Efecto Final (EF) (Corke, 2017, Barrientos, 2007).

Efecto final (EF) del inglés *End Effector*, es la parte del robot para manipular los objetos, y pueden ser neumáticos, hidráulicos, electro-neumáticos y electrónicos (servo-controlados). Por lo que el término efecto final se refiere a la mano o herramienta que está unida a la muñeca del robot, como por ejemplo ventosas, pinzas magnéticas, adhesivas, ganchos, etc. (Barrientos, 2007).

Grados de Libertad (GDL) Se puede definir como cada una de las variables necesarias para obtener los movimientos de un cuerpo en el espacio siendo 6 el máximo número posible. Son, además, cada uno de los movimientos independientes que en una articulación

puede realizar un elemento con respecto al otro (desplazamiento o rotación), (Barrientos, 2007).

1.1.2 Representacion espacial.

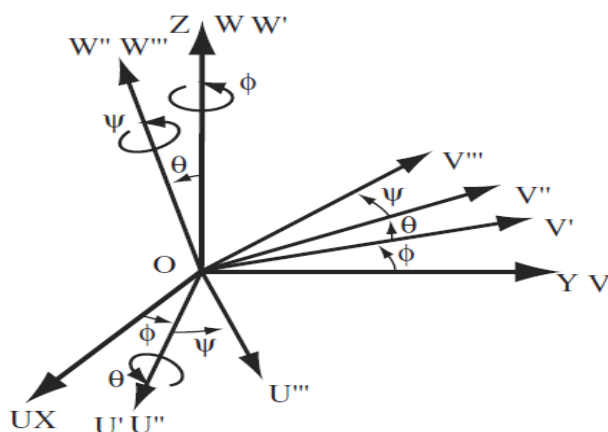
La representación espacial es el contenido de la robótica que estudia la forma de representar la posición y orientación de los elementos de un robot respecto a la base. La representación se realiza a través de herramientas matemáticas que permiten especificar la posición y orientación de cualquier objeto. Este contenido permite la comprensión de los restantes conceptos que se abordan en la asignatura (Barrientos, 2007).

La posición de un cuerpo rígido en un plano bidimensional precisa de, al menos, dos GDL, por lo que la posición queda definida por dos componentes independientes y tres en el caso tridimensional. La representación de una posición en el espacio se puede dar en coordenadas cartesianas, polares o esféricas, siendo expresadas en un vector columna.

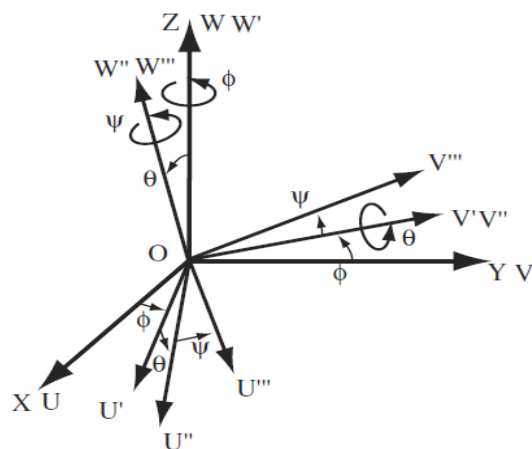
La orientación en el espacio tridimensional se definida por tres GDL o tres componentes independientes. Entre las formas que existen para representar la orientación se hallan las **matrices de rotación** y los **ángulos de Euler** (φ, θ, ψ).

Los ángulos de Euler representan los valores de los giros a realizar, de modo que de un sistema de origen OXYZ se obtenga un sistema OUVW, que se encuentra solidario a un cuerpo. Entre las diversas formas que se pueden obtener los ángulos de Euler se encuentran:

- Ángulos de Euler **WUW** (Figura 1.1), se refiere a los ángulos que se forman tras realizar la rotación del sistema OUVW, inicialmente coincidente con OXYZ un ángulo φ con respecto al eje OZ, formando el sistema $OU'V'W'$. Luego girar sistema resultante un ángulo θ con respecto al eje OU' , obteniéndose el sistema $OU''V''W''$. Por ultimo rotar el sistema anterior un ángulo ψ con respecto al eje OW'' convirtiéndose finalmente en el $OU'''V'''W'''$.

Figura 1.1 Ángulos de Euler **WUW**

- Ángulos de Euler **WVW** (Figura 1.2), en similitud con la forma anterior realizan los giros sobre ejes previamente girados. Su diferencia radica en el eje sobre el que se realiza el segundo giro, en este caso será sobre eje OV' .

Figura 1.2 Ángulos de Euler **WVW**

- Ángulos Euler **XYZ** (Figura 1.3), se refiere a los ángulos que se obtienen de girar el sistema OUVW con respecto al eje OX, rotar el sistema OUVW, respecto al eje OY y girar el sistema OUVW respecto al eje OZ. Estos giros sobre los ejes fijos se denominan guiñada, cabeceo y alabeo (Yaw, Pitch y Roll), respectivamente.

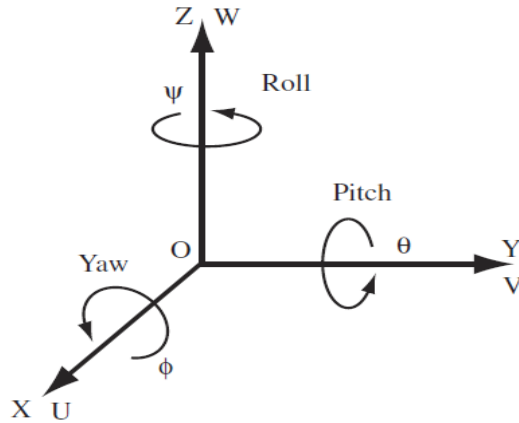


Figura 1.3 Ángulos de Euler **XYZ** (Yaw, Pitch y Roll).

En cada uno en los casos anteriores la secuencia de giro seguida no es conmutativa.

Matrices de Transformación Homogénea (MTH) se refiere a las matrices con las que se puede representar cualquier traslación y orientación entre dos sistemas de referencia. En el caso de la robótica se establece una matriz para cada transformación (rotación y/o traslación) ocurrida entre un sistema de referencia y el siguiente, desde la base hasta el EF. Al multiplicar las matrices anteriores se obtiene la matriz de transformación homogénea del sistema completo desde la base hasta el EF (Barrientos, 2007).

1.1.3 Cinemática

Cinemática, según el Diccionario de la Real Academia de la Lengua Española (DRAE) es “la parte de la mecánica que estudia el movimiento de un cuerpo o sistema de cuerpos sin considerar la acción de las fuerzas o las masas”. La cinemática en robótica se utiliza para el estudio analítico de la geometría del movimiento del manipulador con respecto a un sistema de coordenadas de referencia fijo. Se interesa por la descripción analítica del movimiento espacial como una función del tiempo a través de las relaciones entre la posición y la orientación del extremo final con los valores que toman sus coordenadas articulares (Cerrillo et al., 2015, Barrientos, 2007).

Cinemática directa es la determinación de la posición y orientación del extremo final respecto al sistema de referencia de la base, siendo conocidas las coordenadas articulares y los parámetros geométricos del brazo como la longitud de los eslabones. El resultado que se busca es obtener una MTH que expresa dichas relaciones y la resolución final del problema

cinemático directo mediante MTH está muy relacionada con los últimos tres pasos del algoritmo de Denavit-Hartenberg (Barrientos, 2007).

Parámetros Denavit & Hartenberg (D&H) propuestos en 1955 por Jacques Denavit y Richard Hartenberg son una metodología que permite establecer la ubicación de los sistemas de referencia de los eslabones en los sistemas robóticos articulados, ya sean prismáticas o de revolución, con cadenas cinemáticas abiertas, definiendo las transformaciones relativas entre eslabones con tan solo cuatro parámetros (Barrientos, 2007):

1. Una rotación en torno al eje z_{i-1} un ángulo θ_i .
2. Una translación a lo largo del eje z_{i-1} una cantidad d_i .
3. Translación a lo largo del eje x_i una distancia a_i .
4. Rotación en torno a x_i un ángulo α_i
 - θ_i : ángulo que forman los ejes x_{i-1} y x_i medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Es variable en articulaciones giratorias.
 - d_i : distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i . Es variable en articulaciones prismáticas.
 - a_i : distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .
 - α_i : ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i , utilizando la regla de la mano derecha.

Cinemática inversa es la determinación de todas las coordenadas articulares del brazo que permiten llevar el extremo del robot a una posición y orientación determinadas. El problema cinemático inverso es el más importante pues, lo más deseado es encontrar los valores de las coordenadas articulares para que el extremo del robot se posicione y oriente según una determinada localización espacial (Barrientos, 2007).

El Jacobiano o la matriz jacobiana: se usa para determinar las velocidades del extremo del robot a partir de las velocidades de cada articulación. Asimismo permite la relación entre la velocidad angular y lineal (Barrientos, 2007).

1.1.4 Trayectoria

Las trayectorias son las que definen el movimiento de un brazo robótico de un punto inicial a uno final, de modo que sea posible realizar una tarea determinada. Estas forman parte del control cinemático y a su vez del control dinámico. Las trayectorias empleadas en los robots pueden ser punto a punto o continuas. Las primeras, tienen la característica de que cada articulación evoluciona desde su posición inicial a la final sin tener en cuenta el estado o evolución de las demás articulaciones. Mientras que las segundas tienen en cuenta las velocidades que desarrollan las articulaciones, de manera que se calcula cuál es la más lenta para coordinar el movimiento de las articulaciones en un tempo común para todas.

Por otro lado, para la generación de las trayectorias se tiene en cuenta las localizaciones espaciales por las que el extremo del robot debe pasar, los instantes de tiempo, las velocidades y los tipos de trayectorias que se han de experimentar. Por ejemplo: hacer que el extremo del robot vaya desde un punto inicial a otro final, en línea recta y a una velocidad determinada.

El uso de interpoladores está determinado por la decisión del usuario de especificar una secuencia de localizaciones por las que se desea pasar el EF. Los interpoladores dan una expresión analítica de la evolución de cada coordenada y garantizan que en las secuencias se cumplan las condiciones de la configuración que toma el brazo y el tiempo de paso, así como las restricciones de velocidad y aceleración articulares. Entre los más utilizados se encuentran los lineales, los cúbicos, los de quinto orden y los trapezoidales (Barrientos, 2007).

1.1.5 Dinámica

La dinámica según el DRAE es “la rama de la física que estudia el movimiento en relación con las causas que lo producen y el conjunto de hechos o fuerzas que actúan con un fin determinado”. La dinámica del robot relaciona su movimiento y las fuerzas implicadas en el mismo, siendo así el modelado dinámico el que establece las relaciones matemáticas entre las coordenadas articulares, sus derivadas y las fuerzas y pares aplicados en las

articulaciones y los parámetros físicos del robot (masa de los eslabones, inercia, etc.) (Cerrillo et al., 2015).

Dinámica directa es usada para determinar el movimiento del brazo robótico en respuesta a las fuerzas y torques aplicados a las articulaciones. Para obtener las coordenadas articulares en cada instante de tiempo se necesita la integración de las aceleraciones y el conocimiento de la dinámica inversa (Barrientos, 2007).

Dinámica inversa se utiliza para conocer cuál es el par necesario en cada articulación para un conjunto de coordenadas, velocidades y aceleraciones de las articulaciones. Con ella se puede obtener el diseño mecánico del brazo, seleccionar los motores a instalar en el mismo y se usa para su control dinámico (Barrientos, 2007).

1.2 Simuladores de Robótica industrial (manipulación)

Para definir la robótica industrial en la literatura se pueden encontrar diversos términos devenidos de las diferencias entre los mercados japonés y euro-americano y de lo que es un robot y un manipulador. Según la Asociación de Industrias Robóticas (RIA, siglas en inglés), define como robot industrial a “un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas”.

Por otro lado, la ISO define al robot industrial como: “un manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”.

La FIR, por su parte, define un robot industrial como “una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento”.

Otro concepto que se puede encontrar sobre la robótica industria es el planteado por la AFNOR, expresando que robot es un: “manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos

especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material”.

Entre los simuladores usados para facilitar la enseñanza e investigación de la robótica industrial se encuentran: RoboAnalyzer (RA), RoboDK, Cosumir Educational, ROBOLAB, A Robotic Toolbox for Education (ARTE), entre otros. A continuación, se analizan las características de los mismos que permitan tomar una decisión de selección de software para ser usado en la asignatura Robótica de la UCLV.

1.2.1 RoboAnalyzer (RA)

RoboAnalyzer (Figura 1.4) es un software cuyas características según (Bahuguna et al., 2013) se emplean en el aprendizaje de la robótica por parte de los estudiantes con gran eficiencia. Tiene como filosofía facilitar el aprendizaje de los mecanismos robóticos- la matemática asociada - por parte de los estudiantes, programar y modelar robots a través de un Diseño Asistido por Computadoras (CAD, siglas en inglés) y potenciar el aprendizaje de modelación a través de las variaciones dinámicas y cinemáticas inversa y directa.

El software provee al usuario la posibilidad de cargar diferentes robots que se encuentran en su librería con GDL desde 1 hasta 7 (por ejemplo, el robot SCARA de 7-GDL), además, sus desarrolladores se mantienen creando modelos de robots de cualquier GDL para facilitar el trabajo de los investigadores (Bahuguna et al., 2013).

Para el análisis de la cinemática de los robots, la interfaz de usuario presenta los parámetros de D&H para cada articulación del robot cargado, las MTH y la configuración de las articulaciones para las transformaciones ocurridas a lo largo de una trayectoria dada. Las trayectorias que se utilizan en el software son coseno, cúbica y definidas por el usuario, esta última se carga de un archivo con extensión .csv creado con Microsoft Excel. Además, presenta una interfaz secundaria con la posibilidad de obtener las soluciones de la cinemática inversa de una serie de robots (16 en total) y mostrarlas en la interfaz principal.

Por otro lado, para el análisis de la dinámica de robots, el simulador presenta la posibilidad de analizar la dinámica directa del robot a partir de las fuerzas cargadas, lo cual se puede hacer desde un archivo *.csv* creado con Microsoft Excel donde se encuentren definidas. La dinámica inversa es analizada con los resultados obtenidos de la cinemática inversa.

Además, en RoboAnalyzer se pueden visualizar los gráficos del desarrollo de las coordenadas articulares y sus derivadas en función de una trayectoria dada y el análisis dinámico provee también los gráficos de los valores de fuerza aplicados a las articulaciones para su movimiento.

Por otro lado, RoboAnalyzer también integra un “Módulo de Robot Virtual” (VRM, siglas en inglés), que permite hacer un trabajo de *Teach-Pendant*, lo que facilita la asimilación de los contenidos de cinemática de robots por parte de los estudiantes. Este módulo se integra con Matlab a través de su *Robotics Toolbox* y con Microsoft Excel.

Las dificultades que presenta este software para su uso en la asignatura Robótica de la UCLV son que la librería de robots presenta actualmente un reducido número de modelos CAD, la imposibilidad de enseñar la programación a través del algún lenguaje clásico para estos fines (Lua, RAPID, entre otros), la dificultad que presenta la generación de trayectorias y que no es posible analizar el diseño de controladores.

El software puede ser descargado gratuito desde <http://www.roboanalyzer.com> para Windows 7, 8, 8.1 y 10 (Bahuguna et al., 2013).

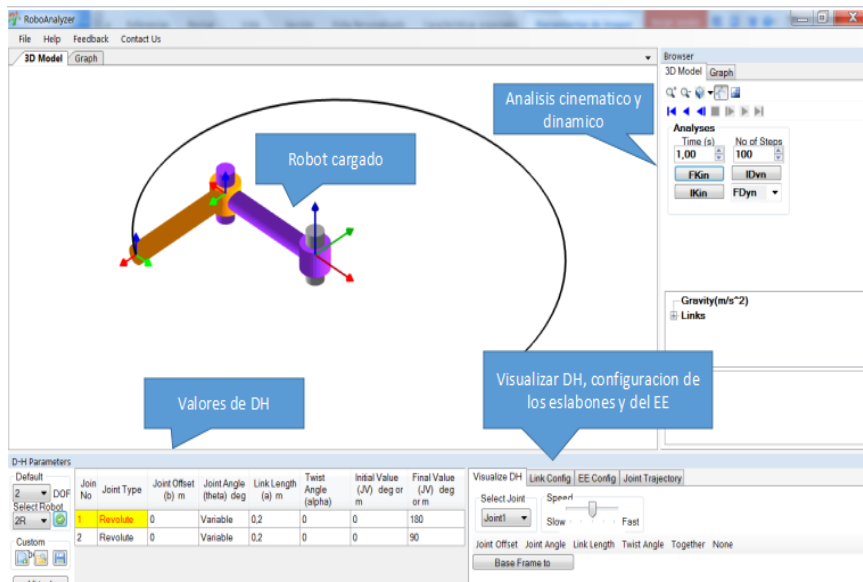


Figura 1.4 RoboAnalyzer

1.2.2 RoboDK

RoboDK (Figura 1.5) es un simulador para la robótica industrial que presenta entre sus principales características la posibilidad de ejecutar cualquier programa del simulador directamente en el robot, si este está conectado a la computadora. Además, este software es usado para simular robots universales, o sea que sirven para cualquier fabricante.

RoboDK utiliza para la programación de los robots de forma visual una interfaz gráfica nombrada *PolyScope* con la que se simulan los robots a través del *Teach Pendant*. Al mismo tiempo, este software presenta la posibilidad de generar los *scripts* de la programación realizada de forma visual y cargar códigos compilados en lenguajes de programación como Python y C#. Además, presenta diferentes funciones con las que se puede realizar una articulación con el software Matlab.

Para la ejecución de un programa creado en el software RoboDK en un robot real se necesita además de conectar el dispositivo a la computadora, que el robot posea una dirección IP conocida. Con RoboDK se pueden simular, exportar e importar programas de robots como ABB Rapid (*.mod*), Fanuc LS (*.LS* y *TP*), Kuka SRC (*.src*), Motoman (*.JBI*) y robots universales (sin *script*) (Alçaçarenho, 2016).

Su explotación no es gratuita, una licencia para usar el software para la enseñanza de la robótica está en los valores de €145.00, mientras que para su uso profesional la licencia es

de €2,995.00 (2018), haciendo difícil su explotación por parte de la UCLV.

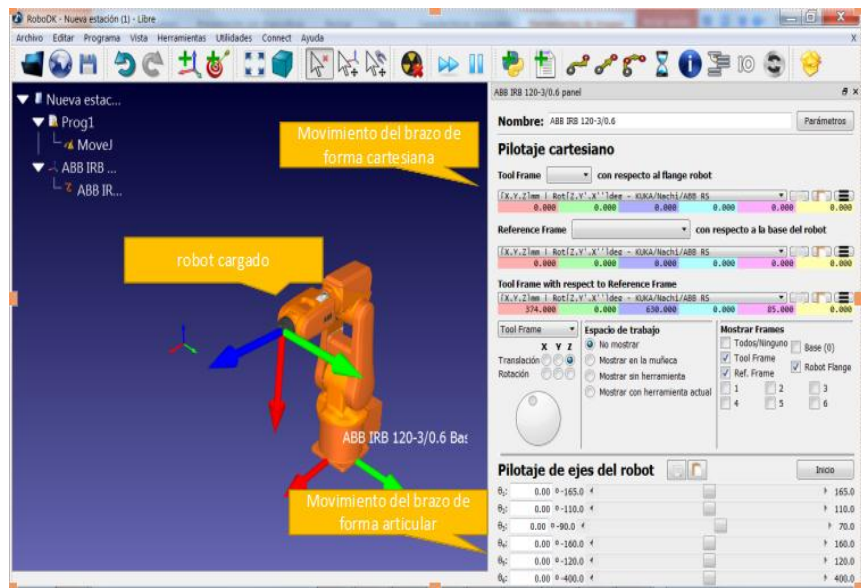


Figura 1.5 RoboDK

1.2.3 Cosumir Educational

Cosumir Educational (Figura 1.6), actualmente CIROS Automation Suite, según (Velásquez Costa, 2009) permite modelar, diseñar y simular sistemas de producción robotizados. Cuenta con una librería que contiene modelos de robots, pinzas, herramientas, alimentadores neumáticos, sensores, PLC, entre otros elementos. Presenta varias ventanas con las que se pueden realizar las simulaciones.

En la ventana de modelamiento se insertan los objetos que estarán presentes en el entorno de trabajo en 3D. La ventana de posiciones está diseñada para almacenar y grabar las coordenadas con las que el robot se moverá, dichas coordenadas son valores en *mm*. También almacena la información acerca de la orientación de las herramientas, así como su estado (abierto o cerrado), permitiendo almacenar hasta 999 posiciones. Emplea un modo *XYZ* cuando se desea mover el robot en el eje cartesiano, mientras el modo *Tool* se emplea para mover el robot en un plano inclinado (Velásquez Costa, 2009).

Presenta también una ventana de programación, donde se crea la secuencia de acciones para el robot a través de una serie de instrucciones que el usuario deberá plasmar. Las principales instrucciones usadas son: *MO* (*move*), *MS* (*move straight*), *MTS* (*move tool straight*), *DJ* (*draw joint*), *ED* (*end*), *GO* (*grip open*), *GC* (*grip close*), *IC* (incrementar

contador), GS (*go sub*), MJ (*move joint*), SP (velocidad), SC (establecer contador), TI (*timer*), (Velásquez Costa, 2009).

Las complicaciones de su uso para la impartición de clases de robótica en la UCLV son la difícil generación de trayectorias, la incapacidad de almacenar más de 999 posiciones del robot y que su uso está determinado solo para robots FESTO.

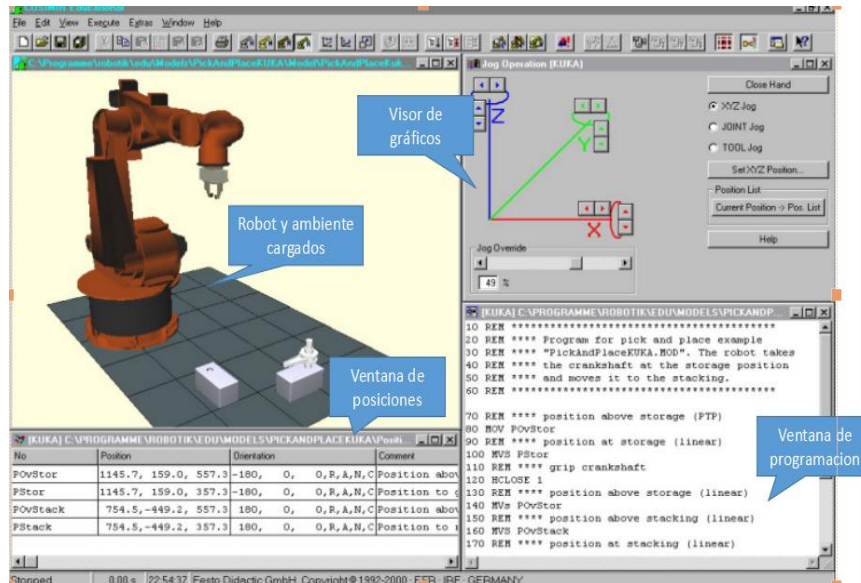


Figura 1.6 Cosumir Educational

1.2.4 ROBOLAB (RobUAlab)

ROBOLAB (Figura 1.7) es un software que permite la simulación de un robot industrial, específicamente uno real situado en un aula de la Universidad de Alicante en España. RobUAlab es la versión más actual de este software desarrollado hasta nuestros días por investigadores de la propia Universidad como herramienta del tipo Open-Source, (Jara et al., 2010).

El sistema puede trabajar en dos modos distintos. El modo simulación tiene el fin de operar sin conexión a Internet y donde el usuario manipula el robot con sus propias trayectorias generadas. Mientras el modo remoto es usado básicamente para validar el trabajo en el aula con el robot real situado en la Universidad de Alicante. Entre sus características pedagógicas e investigativas presenta la posibilidad de analizar la cinemática del sistema, la planificación de trayectorias tanto en el espacio articular como cartesiano, el análisis de la

dinámica del sistema, siendo posible calcular los pares-fuerzas según los GDL, la programación del robot utilizando el lenguaje Java, la posibilidad de usar una cámara virtual con la que el usuario puede observar el entorno de forma 3D y la posibilidad de realizar acciones de control sobre el sistema (Jara et al., 2010).

Además, ROBOLAB (actualmente RobUaLab) presenta la opción de grabar la evolución de las variables del sistema en un archivo con extensión *.m* que se puede usar en Matlab para representar las gráficas relativas a los experimentos y el software muestra los resultados de las matrices de transformación y otros valores referentes a la dinámica.

La principal dificultad que presenta este simulador para su uso en la asignatura Robótica es el lenguaje de programación que utiliza y tampoco se especifica la posibilidad de realizar pruebas a diseños de controladores.

ROBOLAB puede ser descargado desde <http://www.edd.ed.com> de forma gratis.

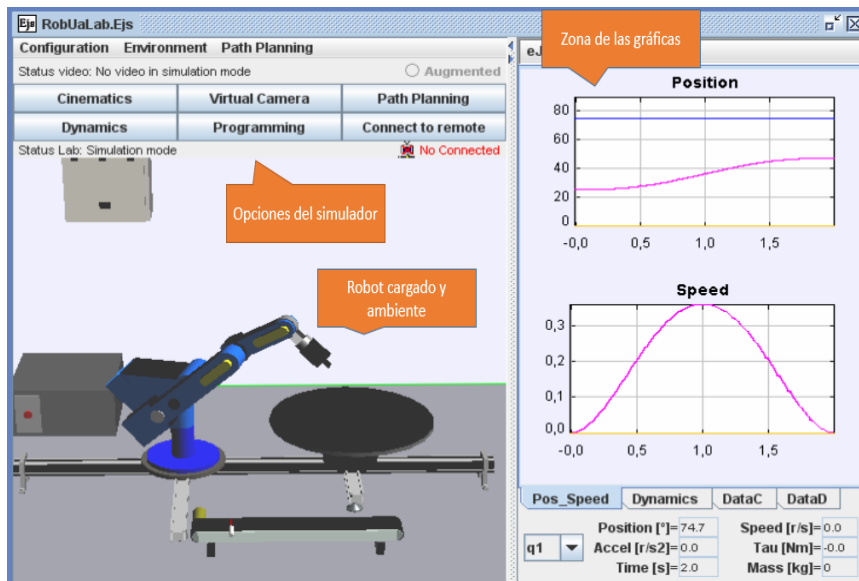


Figura 1.7 ROBOLAB (RobUaLab)

1.2.5 Virtual Robot Experimentation Platform (VREP)

V-REP (Figura 1.8) es un simulador de robótica desarrollado por la empresa Coppelia Robotics con gran capacidad y funcionalidad. Presenta un conjunto de funciones y métodos que pueden ser utilizados por otros softwares. Dispone una interfaz 3D donde los modelos u objetos pueden ser controlados por un *script* propio, un *plugin* o una aplicación cliente

remota, lo que posibilita trabajar en un entorno de multi-robot. Los controladores de los robots pueden ser escritos en C/C++, Python, Java, Lua, Matlab o Urbi.

Con el software se puede realizar la simulación de escenarios 3D de gran complejidad tales como sistemas de fabricación automatizados como una aplicación de la robótica de manipulación y la simulación con robots móviles y humanoides, así como una gran cantidad de sensores. Presenta la posibilidad de realizar cálculos de la cinemática directa e inversa de los robots, así como de su dinámica. Además, dispone de un módulo para simular e interpretar condiciones reales, otro con el que se puede gestionar las colisiones y simulaciones de partículas y otro destinado a la planificación de movimientos (Freese et al., 2010).

La dificultad que presenta el uso de este simulador para su uso en la asignatura está dada principalmente a la complejidad de su articulación con Matlab y que para su uso se necesitan de conocimientos específicos no solo de robótica sino también de diferentes lenguajes de programación.

VREP se encuentra disponible para ser descargado de forma gratis desde <http://www.coppeliarobotics.com/downloads.html> y es del tipo libre.

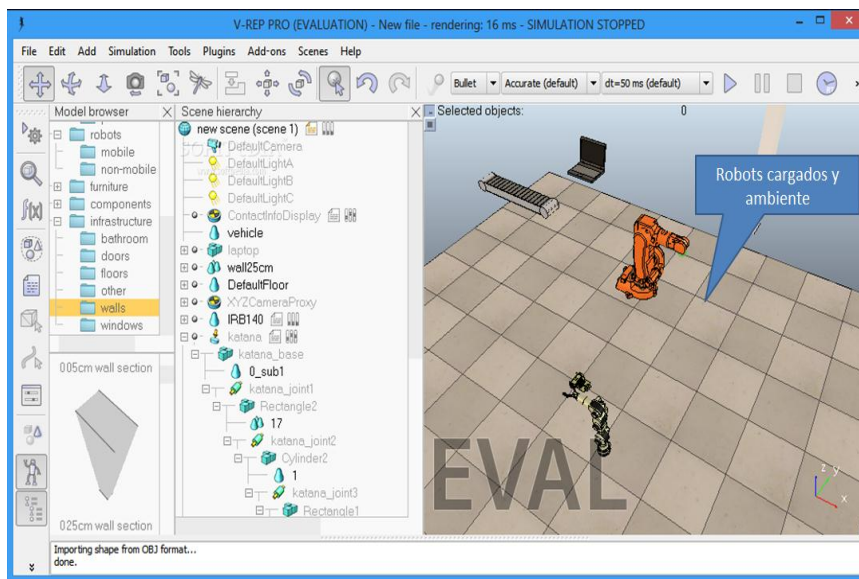


Figura 1.8 Virtual Robot Experimentation (VREP)

1.2.6 Robotics Toolbox de Peter Corke

El Robotics Toolbox (RTB), (Figura 1.9) ha sido desarrollado por Peter Corke desde el año 1996 hasta nuestros días, teniendo diferentes actualizaciones que dotan de mayor funcionalidad al simulador. RTB es un *toolbox* de Matlab que contiene funciones y clases que permiten la representación espacial en 2D y 3D a través de matrices, cuaternos, ángulos de giro, la manipulación y la conversión entre diferentes tipos de datos como vectores y MTH. Este *toolbox* es usado como un método para la representación de la cinemática y la dinámica de manipuladores seriados como un objeto de Matlab, los cuales se encuentran dentro de la librería del simulador tales como el Puma 560 o a su vez, pueden ser creados por los usuarios. El RTB también permite el trabajo con robots móviles y la generación de trayectorias.

Para el uso del simulador está disponible su código fuente lo que beneficia su uso para la enseñanza y el entendimiento de la robótica. Las dificultades que presenta este simulador en relación con su uso para la asignatura Robótica de la UCLV son pocas, solo destaca su deficiente interfaz en la mayoría de sus modelos de robots y la engorrosa implementación de varias de sus funciones al realizar labores relacionadas con el estudio de la dinámica y la cinemática (Corke, 2017).

El *toolbox* se encuentra disponible en Internet para ser descargado de forma gratis en <http://www.petercorke.com/wordpress/?ddownload=412>.

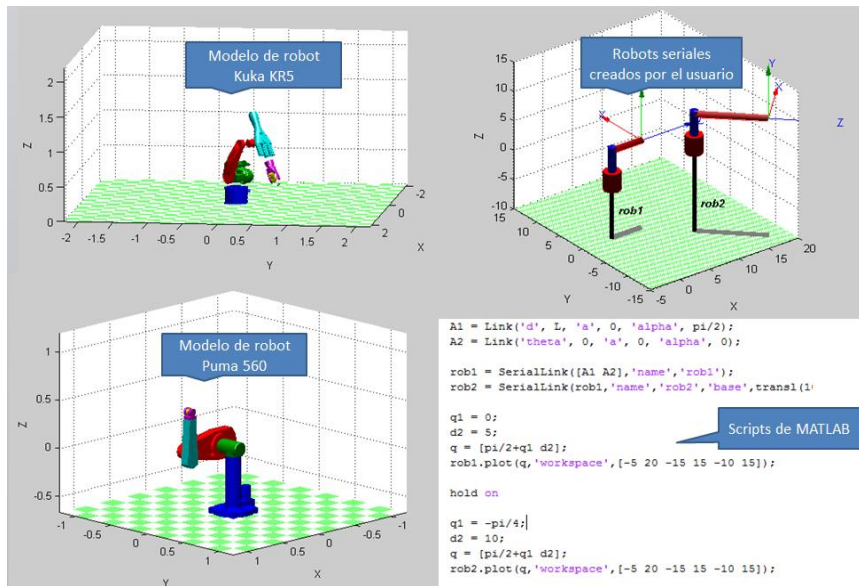


Figura 1.9 Robotics Toolbox (RTB)

1.2.7 A Robotic Toolbox for Education (ARTE)

ARTE (Figura 1.10) es un software desarrollado por Arturo Gil y estudiantes de robótica de la Universidad Miguel Hernández de España desde el 2012. Se presenta como un *toolbox* de Matlab enfocado en la robótica de manipulación, tanto serie como paralela. Es un simulador gráfico, donde se cargan robots industriales de diferentes fabricantes, así como diferentes ambientes de trabajo (pintura, manipulación, etc.) (Gil et al., 2015).

Permite con alta eficacia el análisis cinemático de los sistemas robóticos, por tener la posibilidad de obtener las soluciones de la cinemática tanto directa como inversa, representar los parámetros de D&H, obtener la matriz Jacobiana, así como la velocidad de las articulaciones y del EF. En este sentido, ARTE permite visualizar a través de gráficos el desarrollo de la posición, velocidad y aceleración de las articulaciones, además de posibilitar la generación de trayectorias y la simulación de las mismas.

Por otro lado, también se puede realizar el análisis dinámico de los robots tanto directo como inverso, representar mediante gráficas el comportamiento de las fuerzas y torques presentes en el sistema simulado, combinar diferentes configuraciones de las características dinámicas del robot cargado (masas de los eslabones, momentos de inercia, etc.) y realizar una selección de motores de acuerdo a las intenciones docentes que se tengan.

ARTE presenta una interfaz de usuario donde a través del uso de *Teach Pendant* se puede simular el movimiento del robot y llevar a cabo acciones de control. Con esa herramienta se pueden generar los puntos necesarios para una simulación de control de trayectoria, visualizar los resultados de la cinemática directa (MTH) y la cinemática inversa. Para la simulación de los controladores (P, PI, PID, etc.), ARTE usa las facilidades de Matlab a través de la herramienta *Simulink*, donde se puede visualizar el comportamiento de las diferentes variables presentes en el control de robots (posición, velocidad, torques aplicados, salida de los controladores). La programación de los robots se realiza en Matlab, de donde se exporta el código obtenido al lenguaje ABB RAPID, por lo que el robot real usará como programa el código escrito en Matlab, (Gil et al., 2015).

Este simulador utiliza también las facilidades que brinda el Robotics Toolbox de Peter Corke en un ambiente de simulación más amigable para los usuarios. ARTE se distribuye bajo licencia libre y para comenzar a ser usado se puede descargar de forma gratuita desde: <http://arvc.umh.es/arte/arte.zip>.

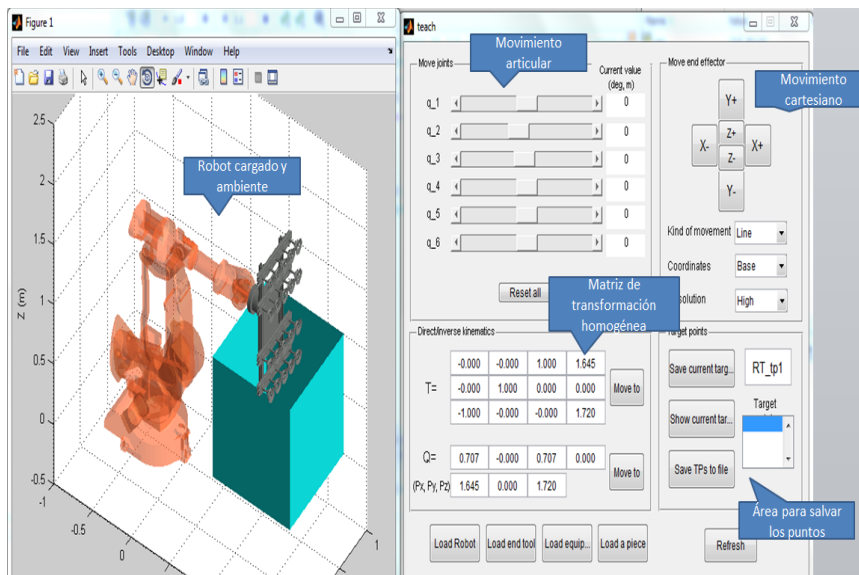


Figura 1.10 A Robotic Toolbox for Education (ARTE)

1.3 Consideraciones finales del capítulo.

- El simulador que más satisface los intereses educativos de la asignatura Robótica impartida en la carrera Ingeniería Automática de la UCLV es ARTE. Este simulador contiene elementos que lo hacen útil para el estudio de la robótica en el ámbito universitario y en un ambiente de simulación. Permite el análisis de elementos que van desde la representación espacial hasta el diseño de controladores; y en caso de disponer de un robot real obtener los códigos necesarios para su implementación.
- Este simulador presenta la posibilidad de usar las herramientas que contiene el Robotics Toolbox de Peter Corke lo que permite una mayor versatilidad de ARTE para ser usado en el estudio de la robótica de manipulación.
- Tanto ARTE como el Robotics Toolbox son simuladores gratis, lo que permite su uso de forma gratis por parte del Departamento Automática y Sistemas Computacionales de la UCLV.

CAPÍTULO 2. ANÁLISIS DE ARTE PARA LA DOCENCIA.

Siendo ARTE vinculado al RTB de Peter Corke, el simulador seleccionado para ser usado como herramienta para su uso en la enseñanza de la robótica en la carrera Ingeniería Automática de la UCLV, en este capítulo se explica su uso en los diferentes contenidos abordados en el plan de la asignatura. Elementos como la instalación, la labor con la representación espacial, el análisis cinemático y dinámico, la programación de robots, y el diseño de controladores, conforman el contenido del capítulo. Toda esta documentación permitirá el uso adecuado del simulador por parte de estudiantes, docentes e investigadores de la UCLV.

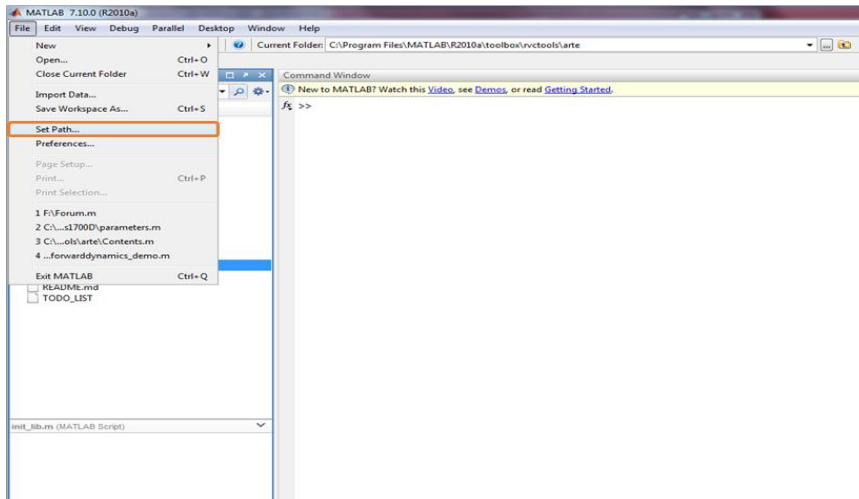
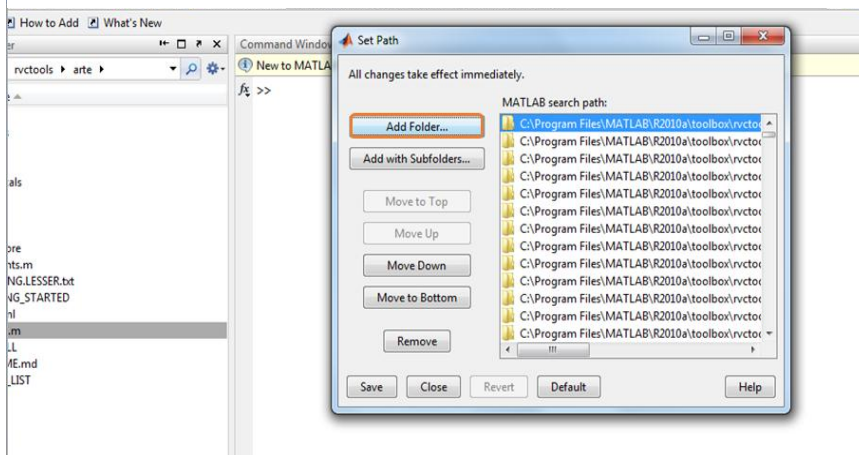
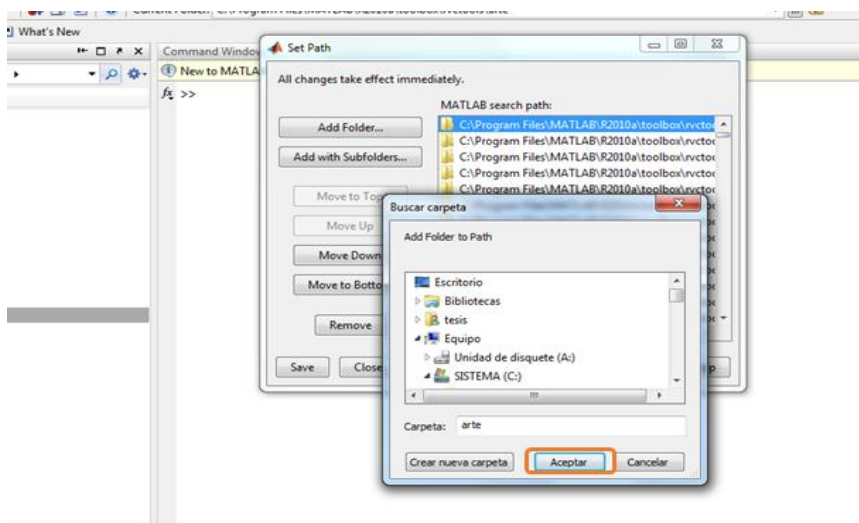
2.1 Instalación y elementos básicos

A continuación se enuncian los elementos básicos destinados al uso de ARTE.

2.1.1 Instalacion

Para realizar la instalación de ARTE, primero se descarga el archivo que se encuentra en <http://arvc.umh.es/arte> y se copia la carpeta que está contenida en el mismo en la dirección donde se encuentran los *toolbox* del Matlab instalado. Por ejemplo: “C:\Program Files\MATLAB\R2010a\toolbox”.

Luego se abre Matlab y, una vez allí se ejecuta el archivo *init_lib.m* y se salva el *path* en: “File/Set path/Add Folders/Save” (ver Figura 2.1, Figura 2.2, Figura 2.3). Cabe señalar que es necesario ejecutar siempre al inicio del trabajo con ARTE el archivo *init_lib.m*. (ver Figura 2.4)

Figura 2.1 Salvar la carpeta en el *path* (1)Figura 2.2 Salvar la carpeta en el *path* (2)Figura 2.3 Salvar la carpeta en el *path* (3)

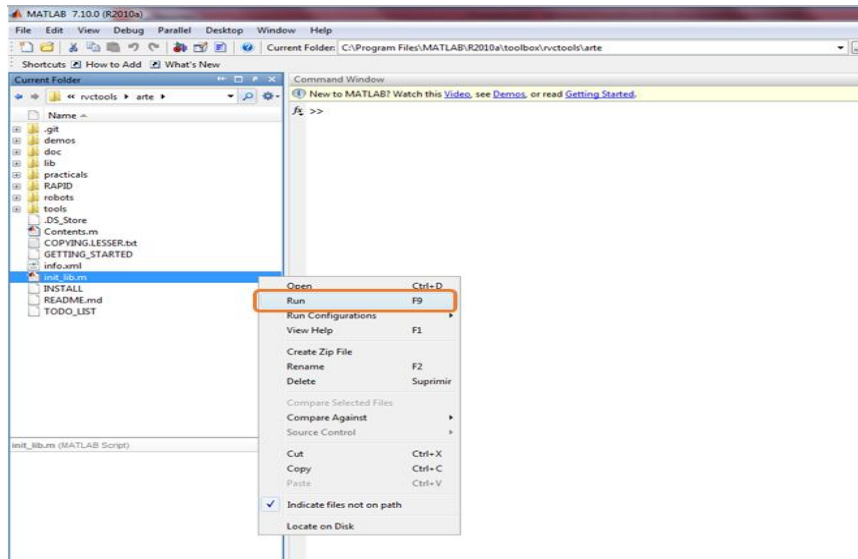


Figura 2.4 Ejecutar el archivo "init_lib.m"

2.1.2 Cargar un robot y su representación CAD

Para cargar un robot en ARTE se puede realizar de diferentes maneras. A través de la función `load_robot` o bien mediante la función `teach` haciendo clic en el botón “Load Robot” (Figura 2.5)

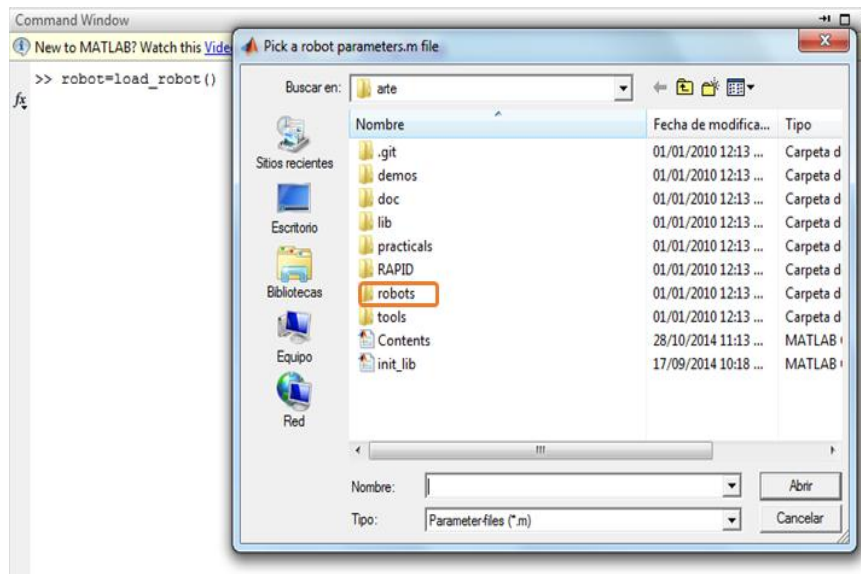


Figura 2.5 Cargar un robot

Para cargar un robot específico con el uso de la función `load_robot()` (ver Figura 2.6(a)) se escribirá:

```
robot = load_robot('abb','irbl40');
robot = load_robot('abb/irbl40');
```

La variable `robot` contiene una serie de atributos del robot cargado entre los que destacan: estructura del ambiente (equipamiento, herramienta y pieza), de los parámetros cinemáticos (posiciones y velocidades articulares, D&H) y dinámicos, entre otros ([Anexo I](#)).

Por tanto para cambiar alguno de los parámetros del robot solo es preciso acceder al mismo. Por ejemplo, para cargar un ambiente (ver Figura 2.6(b)) se escribe:

```
robot.equipment = load_robot('equipment','tables/table_two_areas');
```

Para cargar una herramienta, ver Figura 2.6(c)

```
robot.tool= load_robot('equipment','end_tools/parallel_gripper_0');
```

Para cargar una pieza, ver Figura 2.6(d)

```
robot.piece=load_robot('equipment','cylinders/cylinder_tiny');
```

Con la función `drawrobot3d(robot,q)` se muestra el robot cargado en una posición articular determinada.

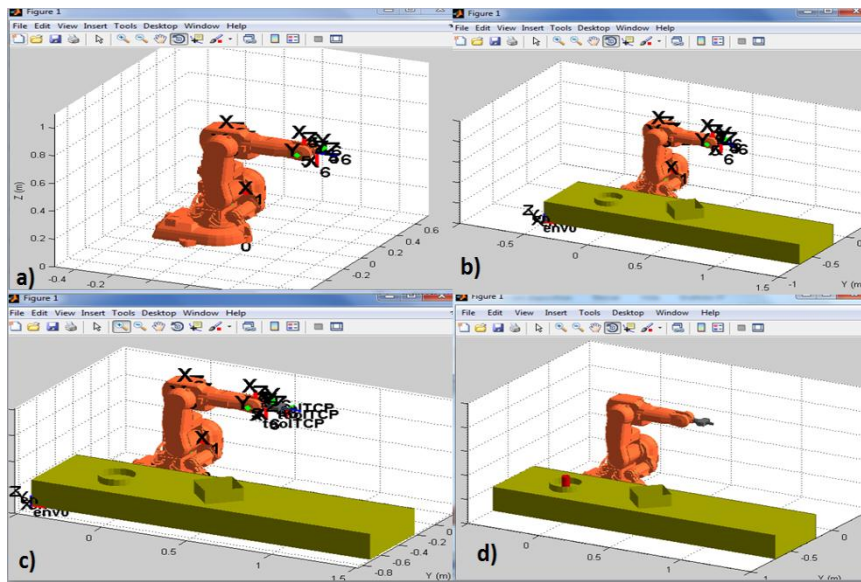


Figura 2.6 Diferentes elementos que se pueden cargar para crear un ambiente de trabajo con el robot

2.1.3 Representación espacial

La representación espacial, tal como se enuncia en el sub-epígrafe 1.1.2 es una temática que resulta compleja de explicar debido a la carga matemática que tiene. La localización espacial y su respectiva representación es uno de los contenidos primarios que se imparte en la asignatura Robótica de la UCLV.

Para la representación espacial se puede usar varias de las funciones implementadas en el Robotics Toolbox lo que facilita el aprendizaje de este contenido por parte de los estudiantes pues permite la visualización de las transformaciones.

`TR = transl(dx,dy,dz);` MTH (4x4) que representa una traslación pura en el espacio (Figura 2.7 a).

`TR = trotx(te); troty(te); trotz(te);` MTH (4x4) que representa una rotación pura en el espacio (Figura 2.7 b).

`TR = [[rpy2r(roll,pitch,yaw) [0 0 0]']; 0 0 0 1];` MTH (4x4) obtenida a través de los ángulos Roll, Pitch, Yaw (Figura 2.7 c).

`rpy = tr2rpy(TR);` forma de obtener los ángulos de Euler de la forma **XYZ** (Roll, Pitch, Yaw) desde una MTH (Figura 2.7 d).

`TR = [[eul2r(a,b,c) [0 0 0]']; 0 0 0 1];` MTH (4x4) obtenida a través de los ángulos de Euler (Figura 2.8 a).

`abc = tr2eul(TR);` forma de obtener los ángulos de Euler de la forma **ZYZ** a través desde una MTH (Figura 2.8 b).

`trplot(TR, 'color', 'k', 'frame', 'a', 'view', 'auto');` plotea el sistema coordenado en el espacio (Figura 2.8 c).


```

>> TR=transl(5,-5,5)
TR =
    1    0    0    5
    0    1    0   -5
    0    0    1    5
    0    0    0    1
a)

>> TR=trotz(-pi/2)
TR =
    0.0000    1.0000    0    0
   -1.0000    0.0000    0    0
    0         0    1.0000    0
    0         0    0    1.0000
b)

>> roll=pi;pitch=pi/4;yaw=3*pi/4;
TR = [[rpy2r(roll,pitch,yaw) [0 0 0]']; 0 0 0 1]
TR =
   -0.5000   -0.5000    0.7071    0
   -0.7071    0.7071   -0.0000    0
   -0.5000   -0.5000   -0.7071    0
    0         0         0    1.0000
c)

>> rpy = tr2rpy(TR)
rpy =
    3.1416    0.7854    2.3562
d)

```

Figura 2.7 Funciones para l Representación espacial en RTB

```

>> a=pi*2;b=5*pi/4;c=5*pi/6;
TR = [[eul2r(a,b,c) [0 0 0]']; 0 0 0 1]
TR =
    0.6124    0.3536   -0.7071    0
    0.5000   -0.8660    0.0000    0
   -0.6124   -0.3536   -0.7071    0
    0         0         0    1.0000
a)

>> abc = tr2eul(TR)
abc =
    3.1416    2.3562   -0.5236
b)

>> trplot(eye(4),'color','k','frame','0','view','auto');hold on
>> trplot(TR,'color','b','frame','b','view','auto')
>> tranimate(eye(4),TR)
c)

```

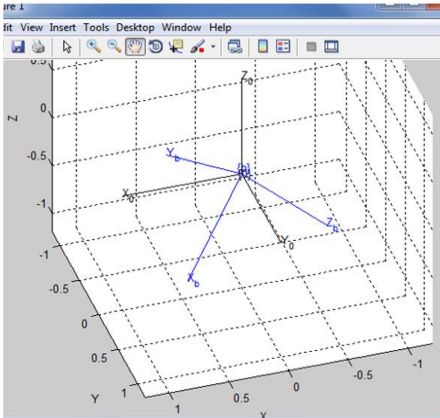


Figura 2.8 Funciones para representación espacial en RTB

2.2 Cinemática

Como se explica en el sub-epígrafe 1.1.3, el problema de la cinemática es el estudio del movimiento del extremo (y resto de eslabones) respecto de un sistema de referencia (base), donde se analiza posición, orientación, velocidades y aceleraciones del extremo del robot (EF).

Como materia imprescindible en el estudio de la robótica, el análisis de la cinemática forma parte esencial del programa de la asignatura. En este estudio se abordan tanto la solución al problema cinemático directo como al inverso.

2.2.1 Cinemática directa

Determinación de la posición y orientación del extremo respecto del sistema de referencia de la base, siendo conocidas las coordenadas de las articulaciones $q = [q1, q2, q3, \dots, qn]$ y los parámetros geométricos del brazo (longitud de los eslabones). El resultado de la cinemática directa es una MTH que represente dicha posición y orientación del EF, lo que resulta altamente laborioso en la medida en que aumenta el número de GDL del robot. Con la existencia de una herramienta con la que se pueda ejercitar la teoría, los estudiantes comprenden de manera más clara este contenido.

La cinemática directa forma parte de los tópicos de la asignatura Robótica. Con el uso del *Virtual Robot Simulator* se ejercita de forma práctica la teoría recibida, lo que posibilita que los estudiantes asimilen este contenido. El simulador ARTE permite el análisis de la cinemática directa eficazmente, así como el estudio de forma más amena de los parámetros de D&H y las MTH.

Para resolver este problema en ARTE se usa la función `directkinematic()`. La función recibe la variable `robot` que contiene los parámetros del robot cargado y la variable `q` que contiene las posiciones de las articulaciones del robot para las que se desea resolver el problema de la cinemática directa. El resultado de la función será la MTH que expresa las relaciones de posición y orientación del extremo del robot respecto a un sistema de referencia fijo situado en su base. Por ejemplo:

```
robot=load_robot('abb','irbl40');
q=[pi pi/2 0 0 pi/4 pi];
T = directkinematic(robot, q)
```

Resulta en $T =$

-0.7071	-0.0000	0.7071	-0.3840
-0.0000	1.0000	-0.0000	-0.0000
-0.7071	-0.0000	-0.7071	-0.0740
0	0	0	1.0000

Con los parámetros de D&H se pueden definir las matrices que permiten el paso de un sistema de referencia asociado a una articulación al siguiente y por tanto definen las MTH entre las articulaciones, dando paso a la MTH del EF respecto a la base. Este cálculo resulta largo y tedioso, así que una vez entendida la teoría se puede hacer uso del simulador para profundizar el contenido.

Con la función `eval(robot.DH.el_parametro_DH)` se obtienen los parámetros de D&H presentes en el robot. Esta función devuelve para cada parámetro un vector que contiene los valores para cada transformación ocurrida desde la base hasta el EF. Por ejemplo:

```
theta = eval(robot.DH.theta)
d=eval(robot.DH.d)
a=eval(robot.DH.a)
alpha=(robot.DH.alpha)
```

Luego mediante la función `dh(robot, q, #de_trasformacion)` se puede obtener las diferentes MTH que relaciona la posición y orientación del EF respecto a la articulación anterior. Por ejemplo:

```
T01 = dh(robot, q, 1); T02 = dh(robot, q, 2); T03 = dh(robot, q, 3);
T04 = dh(robot, q, 4); T05 = dh(robot, q, 5); T06 = dh(robot, q, 6);
T=T01*T02*T03*T04*T05*T06 % resultado de las transformaciones desde la
base hasta el extremo del robot
```

Con el uso de la función `teach` se puede observar la MTH que se obtiene luego de posicionar las articulaciones del robot a través de los *sliders* de movimiento articular. Además se puede observar la matriz de los cuaternos en el cuadro “Q”, que define la orientación del EF y en el cuadro “(Px, Py, Pz)” la posición del EF.

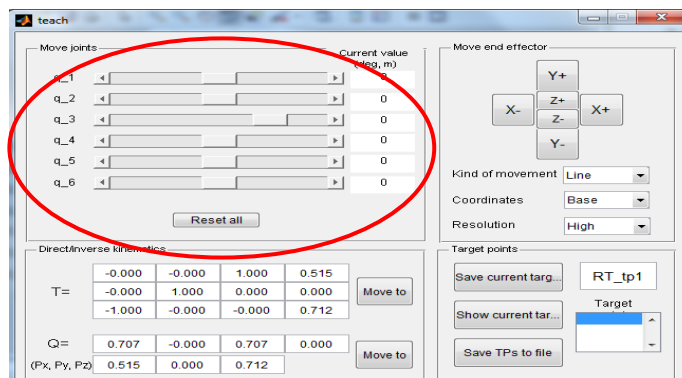


Figura 2.9 Sliders para establecer una posición articular

2.2.2 Cinemática inversa

Determinación de las coordenadas articulares $q = [q_1, q_2, q_3, \dots, q_n]$ del brazo que permiten llevar el extremo del robot a una posición y orientación determinadas. Este contenido es uno de los más importantes en el estudio de la robótica, pero a su vez resulta altamente complejo. Para hallar los valores de las posiciones articulares se puede usar el método geométrico, la resolución a partir de MTH, el desacoplamiento cinemático, entre otros. La carga matemática aplicada a solucionar este problema hace difícil su entendimiento por parte de los estudiantes por lo que se hace necesario utilizar una herramienta que ayude a la comprensión de dicho contenido.

Este tópico es abordado en la asignatura Robótica de forma teórica y práctica mediante el uso del *Virtual Robot Simulator*, permitiendo a los alumnos apropiarse de este contenido. Sin embargo, ARTE presenta otras formas de realizar el análisis cinemático inverso, lo que permite la enseñanza de este contenido con más dinamismo.

Mediante la función `inversekinematic()` se resuelve el problema de la cinemática inversa. Esta función recibe la variable `robot` y la variable `T` que es la MTH que se obtuvo en la solución de la cinemática directa u otra MTH que se desee definir. Por ejemplo:

```
qinv=inversekinematic(robot, T)
```

Siendo `T = directkinematic(robot, q)`

El resultado de la función `inversekinematic` es un conjunto de vectores presentados en forma de matriz en la que cada columna contiene una solución de la cinemática inversa del robot. Además, se pueden obtener las MTH para cada una de las soluciones de la cinemática inversa, así como mostrar la configuración del robot para cada solución en un CAD a través de las siguientes sentencias:

```
Tl=directkinematic(robot, qinv(:,1))
```

```
drawrobot3d(robot, qinv(:,1))
```

A través del *Teach Pendant* también se puede realizar un análisis de la cinemática inversa. Por ejemplo: si se rellena la matriz “T” y se hace clic en “Move to” se puede visualizar la configuración del robot en el CAD.

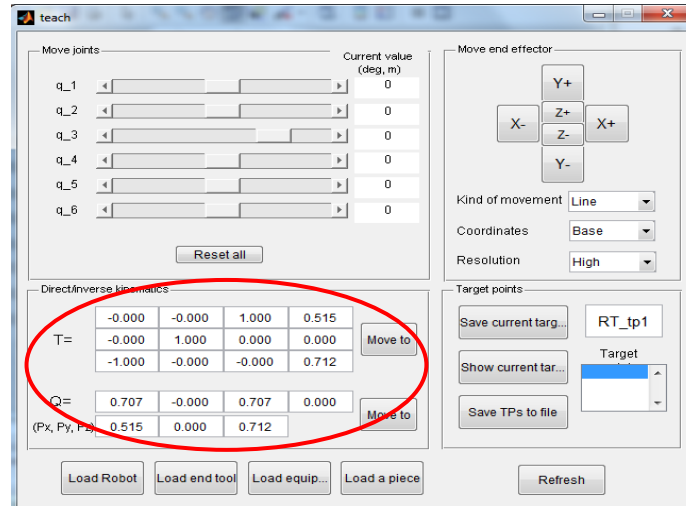


Figura 2.10 Visualización de la posición y orientación del EF en forma de MTH.

Para el estudio de la velocidad en el problema cinemático se utiliza la matriz Jacobiana, la cual permite conocer las velocidades del extremo del robot a partir de las velocidades de cada articulación. Con el uso del Jacobiano inverso se puede conocer las velocidades en las articulaciones a partir de las velocidades en el extremo del robot. Estas matrices pueden ser de dimensión 6 lo que implica que son complejas de trabajar algebraicamente.

En ARTE para el trabajo con la matriz Jacobiana y el estudio de la velocidad en el robot usa las funciones:

- `compute_conventional_jacobian(robot, q)` que devuelve una matriz que contiene el Jacobiano del robot cargado en función de su posición articular.
- `compute_end_velocity(robot, q, qd)` que devuelve un vector que contiene la velocidad del EF de un robot cargado con una posición y velocidad determinada de sus articulaciones. Por tanto el resultado será $V=[v_n \ w_n]'$ donde la primera columna será para la velocidad lineal y la segunda representará la velocidad angular que tenga el EF.

- `compute_joint_velocity(robot,q,V)` devuelve la velocidad de las articulaciones de un robot cargado, en función de una posición articular determinada y con una velocidad del EF determinada.

2.3 Generación de trayectorias

La generación de trayectorias, tal como se explica en el sub-epígrafe 1.1.4 se realiza definiendo los puntos por donde se desea que pase el EF del robot en una tarea o experimento. Dichos puntos pueden ser representados mediante MTH, determinando así tanto la posición del EF como la configuración que alcanza el brazo robótico.

Este contenido dentro de la asignatura Robótica se imparte de forma teórica y hasta el momento en el *Virtual Robot Simulator* también se realizaban actividades prácticas con poca profundidad. Sin embargo, ARTE presenta más opciones para generar trayectorias lo que enriquece la cultura de los estudiantes en esta materia de la robótica.

La generación de trayectorias en ARTE se puede realizar de varias maneras. Una vía es a través de la obtención de MTH escritas directamente que representen tanto la posición como la orientación del EF. Otra es a través de declarar la posición articular del brazo en cada período de muestreo.

A través de las funciones `jtraj` y `ctrj` pertenecientes al Robotics Toolbox se obtienen matrices en las que cada columna representa la posición de las articulaciones en cada muestra. Por ejemplo:

```
[q,qD,qDD] = jtraj(q0, qf, N)
```

Devuelve la variación de la posición, velocidad y aceleración de las articulaciones desde una posición inicial hasta una final en N pasos.

```
TC = ctrj(T0, T1, N)
```

Devuelve una trayectoria cartesiana ($4 \times 4 \times N$) con N puntos desde la posición T0 hasta la T1, donde las T son MTH que definen la configuración del brazo en esas posiciones. La trayectoria cartesiana es una secuencia de transformación homogénea y contiene el índice de puntos a lo largo de la trayectoria.

Otra manera de generar los puntos para una trayectoria es a través del *Teach Pendant*, donde se definen las estructuras de los puntos por donde se ha de mover el brazo robótico.

Los puntos son una estructura que contiene como primer vector la posición del EF, el segundo es la matriz de los cuaternos o sea la orientación del EF, el tercero es la configuración del brazo robótico y por último un vector que permite sincronizar la posición con los 6 ejes externos. Estos puntos se obtienen de la siguiente manera:

- 1- Mover el EF hasta la primera posición deseada y hacer clic en el botón *Save current target*.
- 2- Mover el EF hasta la siguiente posición deseada y hacer clic en el botón *Save current target*.
- 3- Al concluir la definición de puntos hacer clic en el botón *TPs to file* y guardar los puntos generados en un archivo de extensión *.m*.

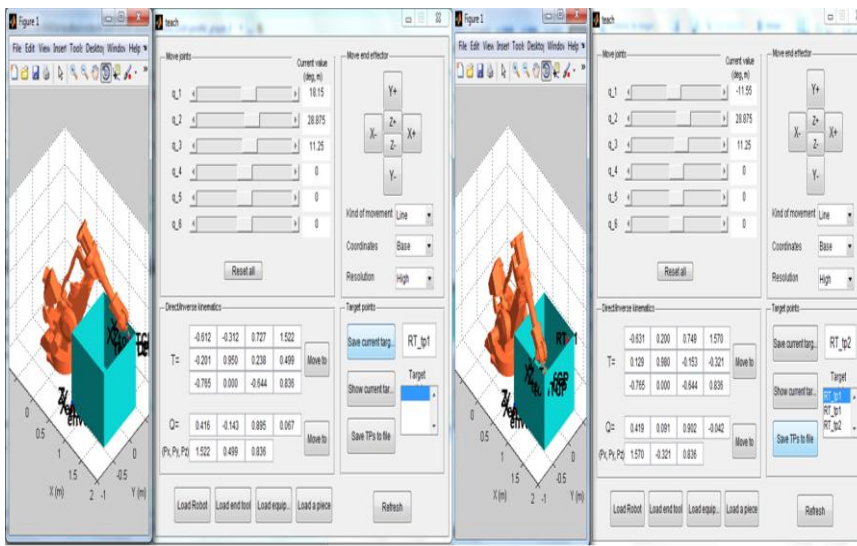


Figura 2.11 Puntos de una trayectoria definidos usando *Teach Pendant*

2.4 Dinámica

El estudio de la dinámica de un robot es uno de los aspectos más importantes de la robótica pues luego de conocer la relación entre el movimiento del robot y las fuerzas y pares que intervienen en el mismo, se puede conseguir simular el movimiento del robot, dimensionar los actuadores y realizar el diseño y evaluación del control dinámico. Dada la complejidad de este tópico su explicación en clases resulta compleja. Mediante las formulaciones de Euler-Lagrange y Newton-Euler se llega a la misma expresión con la que se define el modelo dinámico (Ecuación 2.1) de cualquier robot:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

Ecuación 2.1 Modelo dinámico directo

Donde:

- M es la matriz de los momentos de inercia de las articulaciones del robot.
- C es la matriz que agrupa las fuerzas de Coriolis y Centrífetas.
- q es la posición articular.
- G es la matriz que define la acción de la gravedad sobre el sistema.
- τ es la matriz que agrupa los valores de fuerza a aplicar en las articulaciones (Barrientos, 2007).

Por tanto, determinar los torques a aplicar a cada articulación para que las mismas tengan una evolución en el tiempo determinada y viceversa es difícil de explicar y realizar, pero con el uso de ARTE se puede facilitar el entendimiento de esta materia en el estudio de la robótica.

2.4.1 Dinámica directa

La dinámica directa responde a la pregunta: ¿Cuál es el movimiento del brazo del robot si se aplican unos pares de fuerza determinados? En otras palabras, expresa la evolución temporal de las coordenadas articulares del robot en función de las fuerzas y pares aplicados. Este proceso resulta complejo desde el punto de vista matemático y analítico. En la asignatura, la enseñanza de este contenido en robots industriales se realiza de forma teórica, lo que dificulta el aprendizaje de la materia. Con el uso de la herramienta ARTE se puede llevar a los alumnos al análisis de este tópico de forma práctica.

ARTE usa la función `forwarddynamic` para solucionar el problema dinámico directo. La función recibe como parámetros la variable `robot`, una variable que contenga el tiempo de la simulación, otra con el valor de las coordenadas articulares iniciales, otra con la velocidad de inicial de las articulaciones, otra con las fuerzas y pares aplicados a cada articulación, otra con la acción de la gravedad en los ejes X, Y, Z y otra que contiene las cargas aplicadas al EF.

La función `forwarddynamic` devuelve un vector de tiempo, y otro dos que contienen el desarrollo tanto de la posición como de la velocidad de las articulaciones en función de dicho tiempo. Con la función `animate` se puede visualizar en un CAD el movimiento del robot. Por ejemplo se puede escribir las siguientes sentencias y esperar la simulación:

```
[t q qd]=forwarddynamic(robot,tiempo_de_simulacion, q0, qd0, tau, g',[[]]
animate(robot, q).
```

2.4.2 Dinámica inversa

Por otro lado, la dinámica inversa se emplea para responder: ¿Cuál es el par en cada articulación para un conjunto de posición, velocidad y aceleración del EF? Por lo que expresa las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares y sus derivadas. La asignatura Robótica incluye un tópico para analizar el problema de la dinámica inversa, labor que hasta hoy se realiza solo de forma analítica.

Para dar solución a la dinámica inversa ARTE usa la función `inversedynamic()`. Esta devuelve el valor de la fuerza a aplicar en cada articulación para obtener una posición articular, con una velocidad y una aceleración de las articulaciones determinadas. Además, contiene un vector columna que almacena la acción de la gravedad en los ejes x , y , z y otro vector columna que contiene el valor de las fuerzas externas que están presentes en el sistema. Los valores de la posición, velocidad y aceleración pueden ser un conjunto de vectores que describan el movimiento deseado para el brazo robótico.

```
inversedynamic(robot, q, qd', qdd', g',fext' )
```

Con la función `accel()` se obtiene el valor de la aceleración instantánea de cada articulación de un robot con una posición articular, una velocidad de las articulaciones y las fuerzas aplicadas para que el robot se comporte con las variables anteriores.

```
accel(robot, q, qd, tau).
```

2.5 Programación (RAPID)

El lenguaje RAPID es un lenguaje para programar robots ABB. Con la programación en el lenguaje RAPID a través de Matlab se pueden programar diferentes labores y/o experimentos con los robots. Estos programas pueden ser usados en el ámbito de la docencia a través de la simulación y en el escenario investigativo donde además de trabajar

en un ambiente de simulación si se cuenta con un robot real se pueden llevar el código compilado en Matlab a un tipo de archivo que puede ser usado en robot real.

La programación de robots en la asignatura Robótica no se ha podido materializar como un contenido para el dominio de los estudiantes. Por tanto, el empleo del simulador ARTE permite de forma práctica la asimilación de la teoría que se pueda impartir en la asignatura.

Luego de haber cargado un robot en un ambiente de trabajo determinado (sub-epígrafe 2.1.2) y poseer las posiciones del brazo para la realización de la tarea (epígrafe 2.3), se puede realizar la programación del robot.

Para hacer efectiva la implementación del programa en ARTE, se escribe una función que contendrá la secuencia de pasos de una tarea. Dicha función se nombra *main*, y en ella se crean primeramente variables globales con el nombre de los puntos salvados y la estructura de la herramienta.

A través de las funciones `MoveJ`, `MoveC`, `MoveAbsJ`, `MoveL` y `Offs` se puede simular el movimiento deseado del brazo robótico. El argumento de estas funciones lo compone:

- el punto de destino: contiene la posición y orientación, la configuración del robot y la posición de los ejes externos.

```
Punto_destino=[0.5472,-0.2706,0.5185],[0.5520,0.2727,0.7850,-  
0.0693],[-1,-1,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

- la velocidad: contiene la velocidad para los movimientos de rotación y traslación expresado en *mm/s* (*v100*, *v200*, ...).
- la precisión: indica la máxima diferencia en *mm* entre la posición del robot y el punto de destino (*z10*, *z20*, *z50*, *fine*).
- la herramienta: contiene los parámetros geométricos de la herramienta (sistema de coordenadas) y de la carga (peso, centro de gravedad y momentos de inercia).

```
TD_tool=[1,[0,0,0.125],[1,0,0,0]],[0.1,[0,0,0.100],[1,0,0,0],0,0,0]];
```

- el objeto o pieza: contiene el peso, el centro de gravedad y los momentos de inercia.

`MoveJ(RT_tpl, 'vmax' , 'fine' , tool0, 'wobj0');` se usa para mover el EF del robot con un movimiento independiente de las articulaciones.

`MoveL(RT_tpl, 'vmax' , 'fine' , tool0, 'wobj0');` mueve el EF del robot en una línea del espacio.

`MoveAbsJ(RT_tp1, 'vmax' , 'fine' , tool0, 'wobj0');` mueve el EF del robot a una posición articular absoluta.

`MoveC` mueve el EF del robot en una trayectoria circular.

`Offs(robtarget, X, Y, Z)` programa un desplazamiento relativo desde un destino especificado.

`MoveJ (Offs(robtarget, X, Y, Z), 'vmax' , 'fine' , tool0, 'wobj0');`

Mientras que para las tareas se utilizan las siguientes sentencias:

```
simulation_open_tool; abrir la herramienta
simulation_close_tool; cerrar la herramienta
simulation_grip_piece; coger la pieza
simulation_release_piece; soltar la pieza
```

Con el programa completo y simulado se puede llevar dicho código al lenguaje RAPID desde Matlab. Con la función `matlab2RAPID` se obtiene un programa listo para ser implementado en un robot real en caso de que se cuente con el mismo.

2.5.1 Pequeño ejemplo

```
function main()
global TD_gripper RT_tp1 RT_tp2
%abrir la herramienta
simulation_open_tool;

% mover hasta la posición inicial
MoveJ(RT_tp1, 'vmax' , 'fine' , TD_gripper, 'wobj0');
simulation_close_tool; %Mover hasta el punto donde se va a comenzar a pintar
simulation_grip_piece %coger la pieza
simulation_close_tool;
%simulation_spot_piece;
MoveJ(RT_tp2, 'vmax' , 'fine' , TD_vacuum, 'wobj0');
simulation_release_piece %soltar la pieza
end
```

Ver [Anexo III](#)

2.6 Simulación de controladores

Los controladores en la robótica se utilizan para hacer que los actuadores pongan al sistema en las condiciones que se determinen (posición, orientación, velocidad, etc.). A pesar que en nuestros días se han desarrollado diferentes estrategias de control que tienden a hacer

más eficientes los procesos, sigue siendo el PID el controlador más usado. La robótica no escapa a su empleo y su estructura compleja hace posible que los investigadores lo utilicen para validar sus investigaciones. En la enseñanza de la robótica, el diseño y prueba de los controladores es la pieza que cierra la cadena en el control de los robots. Hasta el momento ha sido imposible simular el diseño de controladores en las clases de robótica, por lo que ARTE resulta de eficaz ayuda en la impartición de este contenido culminante.

Además, ARTE permite probar los diseños de controladores para robots industriales (2 GDL, 4 GDL y 6GDL hasta el momento) en un ambiente de simulación, labor que anteriormente solo se impartía para la robótica móvil.

Para realizar una simulación en ARTE, primeramente se carga un robot que corresponda al modelo *Simulink* con el que se trabaje. Por ejemplo:

```
robot=load_robot('example','2dofplanar')
```

Luego se ejecuta el archivo *simulate_robot_and_controller2DOF.mdl* y se editan los parámetros del controlador ajustándose así a los objetivos de cada clase o ejercicio propuesto. También se pueden editar los parámetros dinámicos del robot (momentos de fricción, centros de masa, matrices de inercia y masa de los eslabones) y/o los parámetros de los motores para trabajar con diferentes configuraciones.

Es importante tener en cuenta que para trabajar con cualquier modelo de robot es necesario que este contenga parámetros dinámicos.

En el modelo *Simulink* se cargan y obtienen los parámetros del robot a través de las funciones *call_direct_dynamics_2dofplanar_fast* y *call_inverse_dynamics_2dofplanar_fast* (ya implementadas y reproducibles para cualquier modelo de robot), lo que hace posible el control dinámico.

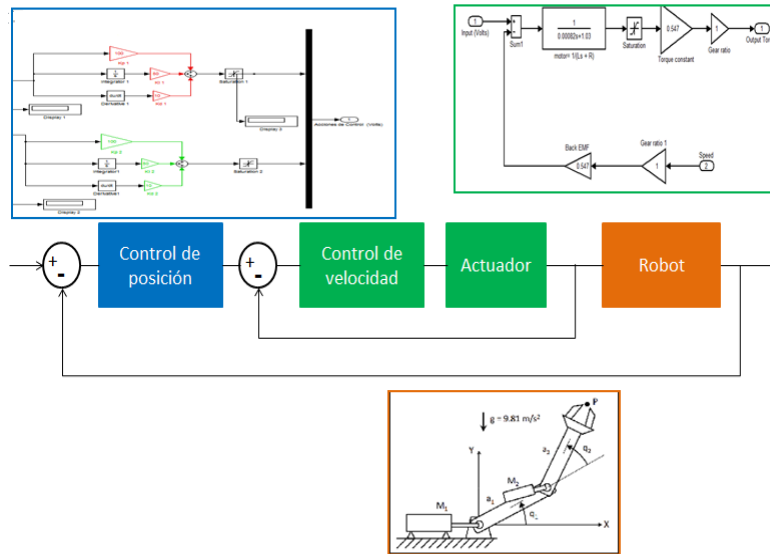


Figura 2.12 Control de robot 2 GDL.

2.7 Consideraciones finales del capítulo

- ARTE cuenta con un conjunto de funciones (`ctray`, `directkinematic`, `inversekinematic`, `forwarddynamic`, `inversedynamic`, entre otras) que permiten el estudio de la cinemática y la dinámica directa e inversa, la generación de trayectorias, la representación espacial, la programación de robots y la simulación de controladores.
- La herramienta *Teach Pendant* de ARTE posibilita la familiarización de los estudiantes con el control directo de robots además de propiciar con mayor efectividad el estudio de la cinemática en la robótica.
- Los parámetros de las funciones pueden ser editados por el usuario de forma que responda a sus exigencias, dígame cambiar la estructura de los robots, crear nuevos mecanismos, estructuras, etc.
- Para la asignatura Robótica ARTE cumple con los objetivos de la misma y además permite la implementación de contenidos que hasta el momento no era posible tratarlos en la carrera. Tal es el caso de la simulación de controladores y la programación.

CAPÍTULO 3. IMPLEMENTACIÓN DE LA ASIGNATURA ROBÓTICA CON ARTE.

Tomando en consideración los objetivos de la asignatura Robótica y luego de haber analizado las posibilidades que brinda ARTE para la enseñanza de esta ciencia en la UCLV, a continuación, se presenta una propuesta para la ejercitación de los diferentes contenidos abordados en la asignatura Robótica. Dicha propuesta permitirá evaluar al simulador ARTE como una herramienta eficaz para la enseñanza de la robótica en la carrera Ingeniería Automática.

3.1 Contenidos de cinemática

La cinemática como parte del estudio de la robótica contiene determinados contenidos que resultan complejos de explicar de forma teórica, como ya se ha abordado en capítulos anteriores. La práctica de los estudiantes en esta materia les permite apropiarse del contenido con más facilidad.

3.1.1 Representación espacial.

En la asignatura Robótica, luego de realizar una explicación teórica de la representación en el espacio de los objetos respecto a determinados sistemas de coordenadas, se pasa a una actividad práctica en la que los estudiantes aprecian los resultados de realizar diferentes transformaciones y cómo se representan las mismas en el espacio (2.1.3). Los siguientes ejercicios son una muestra de lo que se puede implementar como actividad práctica para este contenido.

1-En la figura se muestra un sistema $O'U'V'W'$ rotado y trasladado un vector $p = (5, -5, 5)$ respecto al sistema $OXYZ$. Encuentre la matriz de transformación entre los sistemas O' y O .

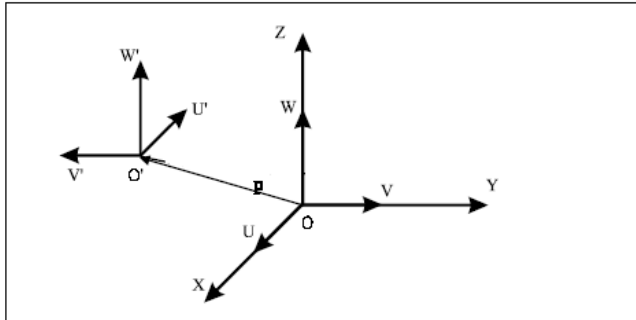


Figura 3.1 Representación gráfica de las transformaciones

```
T0=eye(4) % posición inicial en 000
T=trotz(-pi/2)*transl(5,-5,5) % transformaciones ocurridas
trplot(T0,'color','k','frame','a','view','auto'); axis([-10 20 -10 20 0
20]); hold on
trplot(T,'color','b','frame','b','view','auto'); axis([-10 20 -10 20 0
20]); hold on

trplot(T0,'color','k','frame','a','view','auto'); axis([-10 20 -10 20 0
20]); hold on
tranimate(T0,T) % visualizar transformaciones ocurridas
```

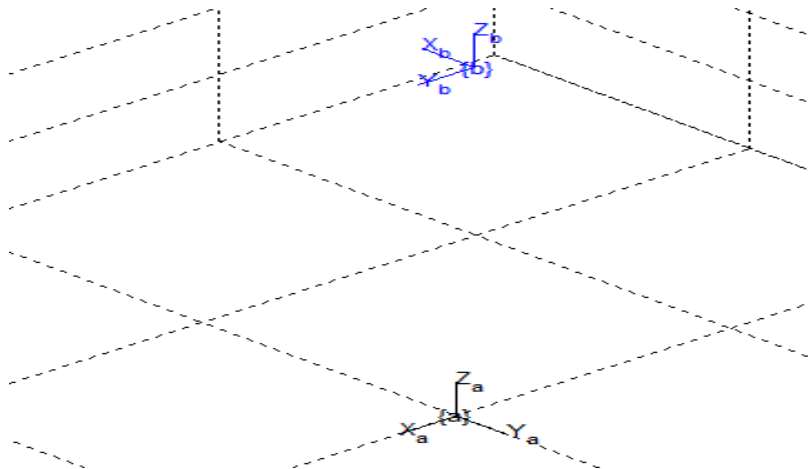


Figura 3.2 Respuesta del ejercicio

2-En la Figura se muestra un robot donde el sistema de referencia asociado a la herramienta tiene la misma orientación que el de la base. El robot cargado es el “ABB

IRB4400” con la herramienta “*parallel_gripper_0*”, en el ambiente de trabajo “*box_big*”. Obtenga los ángulos de Euler (en la forma XYZ) necesarios para colocar el EF del robot en las paredes del cubo con el eje “Z” perpendicular a la pared y el eje “X” paralelo al piso. Realice las operaciones con la herramienta *Teach Pendant*.

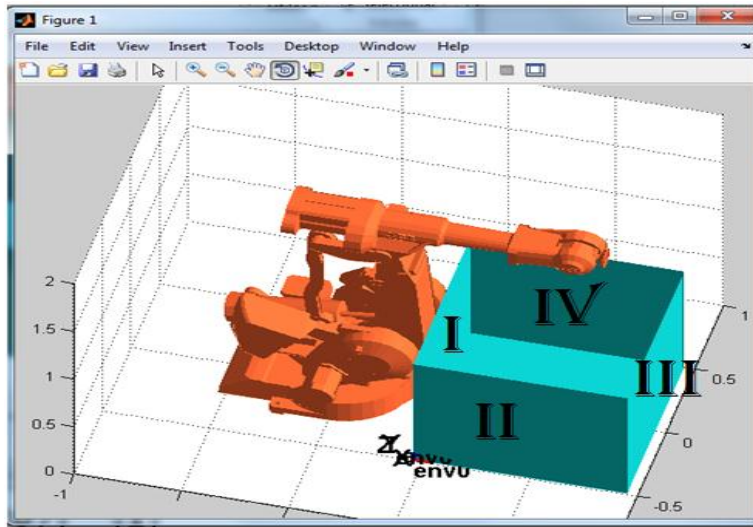


Figura 3.3 Caja para hallar ángulos de Euler

```
robot=load_robot('abb','irb4400'); % cargar el robot
robot.tool=load_robot('equipment','end_tools/parallel_gripper_0') % cargar
la herramienta
robot.equipment=load_robot('equipment','box_big');
q=[0 0 0 0 0 0];
robot.graphical.draw_transparent=0;
robot.graphical.draw_axes=0;
robot.equipment.T0= [1 0 0 0.45;0 1 0 -0.5;0 0 1 0;0 0 0 1]; % para
cambiar la posición inicial de la caja y poder trabajar con ella
drawrobot3d(robot,q)
adjust_view(robot);
teach
```

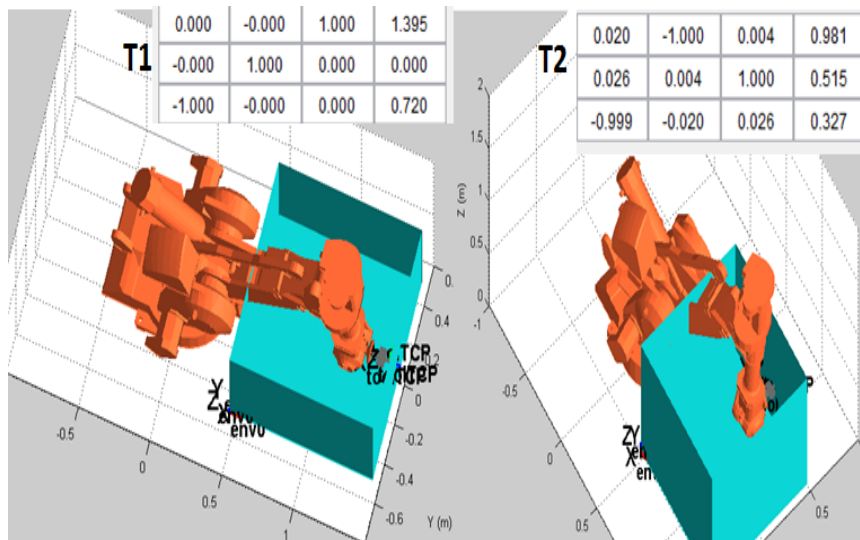



Figura 3.4 Resultado del ejercicio

```
T1=[0 0 1 1.393;0 1 0 0;-1 0 0 0.72; 0 0 0 1]; % define posición y
orientación del EF en el punto 1 (cara III)
rpy = tr2rpy(T1) % para obtener roll, pitch, yaw
T2=[0.02 -1 0.004 0.981; 0.026 0.004 1 0.515; -1 -0.02 0.026 0.327]; %
define posición y orientación del EF en el punto 2 (cara IV)
r2p2y2 = tr2rpy(T2)
```

3.1.2 Cinemática directa

Como se expuso en el Capítulo 2, epígrafe 2.2.1 la cinemática directa en la asignatura Robótica tiene una componente teórica y otra práctica. Con el uso de ARTE se puede realizar una explicación de la teoría de forma más clara. Sin embargo, la actividad práctica a través del simulador permite que se ejerciten los contenidos relacionados con la obtención de MTH entre una articulación y la siguiente y entre la base y el EF, el análisis de los parámetros de D&H y el control de robots a través de *Teach Pendant*. Por ejemplo, los siguientes ejercicios validan la eficacia de ARTE para la enseñanza de este contenido.

- 1-Obtenga la Matriz de Transformación Homogénea que relaciona la base con el extremo de un robot ABB IRB140 que tiene posiciones articulares $q=[0 \ 0 \ 0.1 \ 0 \ 0 \ 0]$ y muestre al robot en dicha posición.

```
q=[0 0 0.1 0 0 0];
robot=load_robot('abb','irb140');
T=directkinematic(robot, q)
drawrobot3d(robot,q)
```

2-Obtenga los parámetros de D&H que tiene el robot y halle las MTH que relacionan una articulación con la siguiente comenzando desde la base hasta el extremo del robot. Compruebe que la multiplicación de dichas MTH da como resultado la matriz del ejercicio anterior.

```
theta = eval(robot.DH.theta); d=eval(robot.DH.d); a=eval(robot.DH.a);
alpha=(robot.DH.alpha)

T01 = dh(abb, qabb, 1); T02 = dh(abb, qabb, 2); T03 = dh(abb, qabb, 3);
T04 = dh(abb, qabb, 4); T05 = dh(abb, qabb, 5); T06 = dh(abb, qabb, 6)
T=T01*T02*T03*T04*T05*T06
```

3-Con el uso de la herramienta *Teach Pendant* obtenga la MTH de la base con el EF para una posición articular $q=[-4.77 \ 46 \ 29.7 \ -5 \ -73.2 \ 1.24]$.

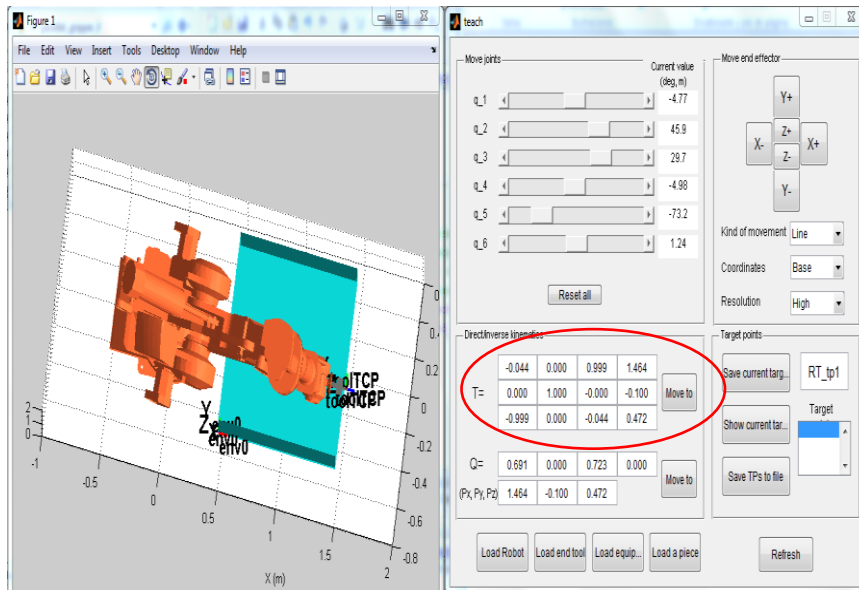


Figura 3.5 Resultado del ejercicio

3.1.3 Cinemática inversa

El contenido de la cinemática inversa, tal y como ocurre con la directa tiene una parte teórica y otra práctica como se explica en el sub-epígrafe 2.2.2 del Capítulo 2. Por lo compleja que resulta la teoría de este tema la actividad práctica es necesaria para que los estudiantes adquieran de forma más clara los objetivos. Los siguientes ejercicios son un ejemplo para ser empleados en las prácticas de la cinemática inversa a través de ARTE en la UCLV.

- 1- Obtenga las soluciones de cinemática inversa para un robot ABB IRB140 si su EF ha sido rotado $\pi/2$ en el eje “X”, $\pi/6$ en el eje “Z” y trasladado 0.5, 0.7, 0.6 en XYZ, con respecto a la base. Halle los diferentes valores que pueden tomar las articulaciones para que el robot tenga la configuración descrita. Luego muestre el robot en 3 de esos resultados, de ser posible.

```
T=trotx(pi/2)*troty(pi/6)*transl(0.5,0.7,0.6)
robot=load_robot('abb','irb140');
qinv=inversekinematic(robot, T) %se obtienen las soluciones de la
cinemática inversa
drawrobot3d(robot, qinv(:,1)) %muestra en 3D la primera solución
```

- 2- Con el uso de la herramienta *Teach Pendant* obtenga los valores de las posiciones articulares usando la matriz que representa las siguientes transformaciones del EF respecto a la base: rotación en el eje “Y” de $\pi/4$, traslación en el eje “Z” de 1, rotación en el eje “X” de $\pi/3$ y traslación en el eje “Y” de 0.4.

```
T= troty(pi/4)*transl(0,0,1)*trotx(pi/3)*transl(0,0.4,0)
teach
```

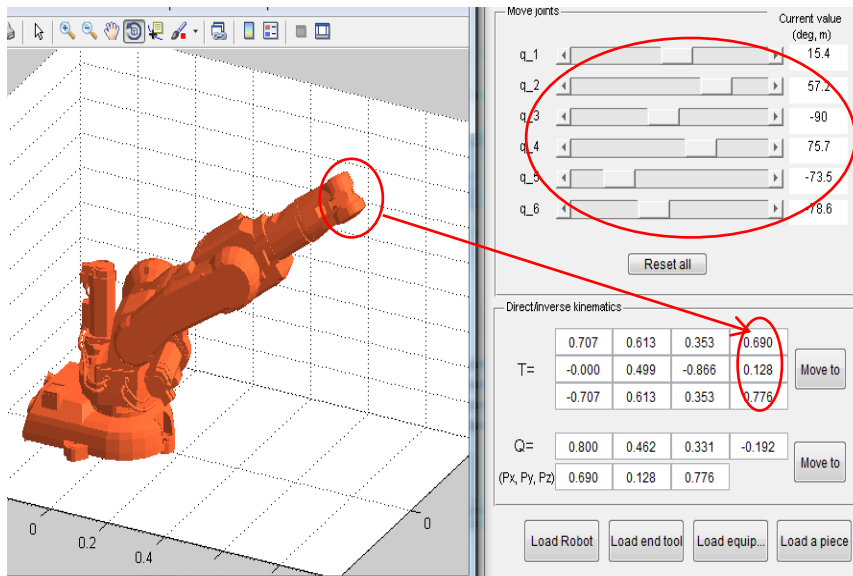


Figura 3.6 Resultado del ejercicio

3.1.4 Generación de trayectoria

Como se explica en el epígrafe 2.3, el contenido de generación de trayectorias en la asignatura Robótica se imparte de manera teórica y práctica. Sin embargo, en la forma

práctica solo se obtienen puntos de una trayectoria definidos a través de la herramienta *Parrrt Handling* en el *Virtual Robot Simulator*. Los siguientes ejercicios evidencian la versatilidad de ARTE para la enseñanza de este contenido.

- 1- Obtenga 100 valores de la posición de cada articulación para llevar el brazo robótico desde la posición $q_i=[0\ 0\ 0\ 0\ 0\ 0]$ hasta la posición $q_f=[10\ 2\ 0\ 3\ 4\ 8]$

```
q_i=[0 0 0 0 0 0]; q_f=[10 2 0 3 4 8]
```

```
[q,qD,qDD] = jtraj(q_i, q_f, 100)
```

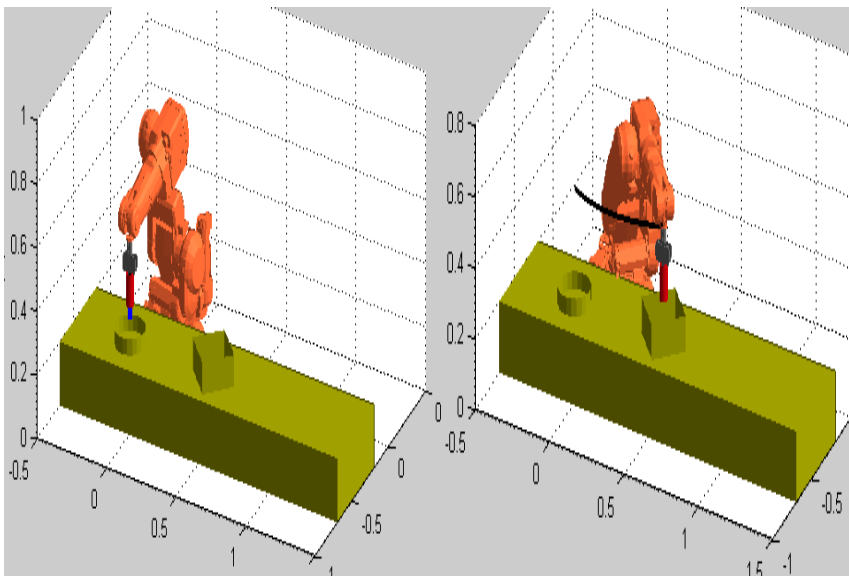
- 2- Obtenga 100 posiciones de la trayectoria cartesiana que va desde el origen de coordenadas hasta la posición definida por una rotación de 90° en el eje “X” y una traslación de 2 unidades en el eje “X” y 3 en el eje “Y”.

```
T_i=eye(4); T_f=trotx(pi)*transl(2,3,0);
```

```
TC = ctraj(T_i, T_f, 100)
```

- 3- Cargue el robot ABB IRB140 con la herramienta *grippe* en el espacio de trabajo *table_two_areas* y con la pieza cilindro. Luego con el uso de la herramienta *Teach Pendant* genere los puntos que permitan llevar al cilindro desde el círculo hasta el rombo.

```
robot = load_robot('abb','irb140');
robot.equipment = load_robot('equipment','tables/table_two_areas');
robot.tool= load_robot('equipment','end_tools/parallel_gripper_0');
robot.piece=load_robot('equipment','cylinders/cylinder_tiny');
robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]'; %cambia la posición de la pieza
drawrobot3d(robot, robot.q);
teach
```



3.2 Contenidos de dinámica

La dinámica en la robótica es un contenido complejo de explicar y asimilar. Como parte del plan de la asignatura Robótica se estudia la teoría asociada a la dinámica para robots móviles e industriales. Sin embargo, la parte práctica solo contiene lo relacionado con la robótica móvil, lo que hace necesario el uso de una herramienta que permita analizar la dinámica de robots industriales. En ese sentido ARTE presenta los elementos necesarios para ser usados en la ejercitación de los contenidos de dinámica por parte de los estudiantes.

3.2.1 Dinámica directa

Figura 3.7 Manipulación del cilindro

La ejercitación de forma práctica de los contenidos relacionados con la dinámica directa de robots industriales no forma parte actualmente de las prácticas en la asignatura por no contar con un simulador que permita dicha labor. Como se explica en el sub-epígrafe 2.4.1, ARTE presenta las herramientas para realizar una práctica por parte de los estudiantes que posibilite la asimilación de la teoría. Los ejercicios siguientes son una propuesta que puede ser usada en clases prácticas de esta materia:

- 1- Dado el vector de fuerzas aplicadas $\tau = [12 \ 2 \ 1 \ 4 \ 21 \ 14]$ y conociendo que la posición y velocidad inicial de las articulaciones es 0. Halle el desarrollo de la posición de las articulaciones en un tiempo de simulación de 2 segundos. Tenga en cuenta que la gravedad solo actúa en el eje “Z”.

```
total_simulation_time=2;
q0=[1 2 0 2 3 1];
qd0=[1 2 1 0 1 2];
g=[0 0 9.8]';
tau=[1 2 1 4 1 4];
[t q qd] = forwarddynamic(robot, total_simulation_time, q0, qd0, tau',
g, []);
```

- 2- Muestre el desarrollo de la posición y la velocidad del ejercicio anterior.

```
animate(robot,q)
```

```
figure, plot(t, q), grid, title('Position vs. time')
xlabel('time (s)'), ylabel('Position (rad)')
legend('q_1', 'q_2', 'q_3', 'q_4', 'q_5', 'q_6');
```

```
figure, plot(t, qd), grid, title('Speed vs. time')
xlabel('time (s)'), ylabel('Speed (rad/s)')
legend('qd_1', 'qd_2', 'qd_3', 'qd_4', 'qd_5', 'qd_6');
```

3.2.2 Dinámica inversa

Tal como ocurre con la dinámica directa, el contenido de dinámica inversa solo es posible abordarlo de forma teórica en las clases. Pero como se explica en el sub-epígrafe 2.4.2, el simulador ARTE presenta diferentes funciones que posibilitan la asimilación de forma práctica de dichos contenidos por parte de los alumnos. Los siguientes ejercicios son un ejemplo que puede ser usado como actividad práctica en la asignatura:

- 1- Cargue el robot ABB IRB 140. Obtenga el valor de fuerzas a aplicar en cada articulación para que el EF tenga una posición articular $q=[1 \ 2 \ 0 \ 3 \ 2 \ 1]$, una velocidad en las articulaciones de $qd=[3 \ 5 \ 1 \ 0 \ 2 \ 1]$ m/s y una aceleración en m/s^2 $qdd=[1 \ 0 \ 2 \ 3 \ 6 \ 5]$. Tenga en cuenta que la acción de la fuerza de gravedad solo tiene lugar en el eje “Z” y no existen fuerzas externas que influyan en el movimiento.

```
robot=load_robot('abb','irb140')
q=[1 2 0 3 2 1];
qd=[3 5 1 0 2 1];
qdd=[1 0 2 3 6 5];
g=[0 0 9.8]';
fm=[0 0 0 0 0 0];
tau = inversedynamic(robot, q, qd, qdd, g, fm)
```

- 2- Con el mismo robot del ejercicio anterior, obtenga el valor de fuerzas a aplicar en cada articulación para llevar el EF desde la posición $q0=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ hasta la posición $qf=[1 \ 2 \ 1 \ 0 \ 4 \ 5]$. Para generar la posición, velocidad y aceleración de las articulaciones use la función `jtraj` del RTB.

```
q0=[0 0 0 0 0 0];
g=[0 0 9.8]';
```

```
fm=[0 0 0 0 0 0];
qf=[1 2 1 0 4 5];
[Q,QD,QDD] = jtraj(q0, qf, 100);
tau = inversedynamic(robot, Q, QD, QDD, g, fm)
```

3.3 Programación de robots y simulación de controladores

3.3.1 Programación de robots

La programación de robots es uno de los contenidos que actualmente no son impartidos en la asignatura Robótica debido a la carencia de un simulador que permita generar códigos validos en alguno de los lenguajes de programación de robots básicos (RAPID, Lua, Octave, etc.). Dado que ARTE presenta la posibilidad de programar robots sobre la base del lenguaje RAPID para ABB, permite la inclusión de este importante tópico en el estudio de la robótica en la carrera Ingeniería Automática de la UCLV. El siguiente ejercicio evidencia la efectividad de ARTE para que los estudiantes dominen al menos el lenguaje RAPID:

- 1- Escriba una función que permita desarrollar una tarea de manipulación de un cilindro sobre una mesa con un robot ABB IRB 140 equipado con una herramienta *gripper*

```
function Respuestas
% primero se declaran las variables que intervienen
global robot TD_gripper RT_inicio RT_cerca_circulo RT_coger
RT_cerca_rombo RT_soltar
% luego se carga el robot y su ambiente
robot = load_robot('abb','irb140');
robot.equipment = load_robot('equipment','tables/table_two_areas');
robot.tool= load_robot('equipment','end_tools/parallel_gripper_0');
robot.piece=load_robot('equipment','cylinders/cylinder_tiny');
% se inicializa la posición de la pieza para la simulación
robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]';
drawrobot3d(robot, robot.q);
% se llama a teach para poder definir los puntos
teach
% se define la herramienta Anexo
TD_gripper;
```

```
%definir los puntos para la simulación (los puntos tienen la estructura
mencionada en el capítulo 2) Anexo
RT_inicio; RT_cerca_circulo; RT_coger; RT_cerca_rombo; RT_soltar;
% se llama la función main
main
end
```

El código del programa completo se encuentra en el [Anexo III](#).

3.3.2 Simulación de controladores

Las estrategias de diseño de controladores en la asignatura Robótica se abordan desde el punto de vista teórico y práctico solo para robots móviles. Como se expone en el epígrafe 2.6 del Capítulo anterior, el simulador ARTE tiene herramientas que permiten simular los controladores en el ámbito de la robótica de manipulación. Por tanto, al contar con la posibilidad que brinda el software, la simulación de controladores se convierte en un contenido que se puede impartir de forma práctica. Los siguientes ejercicios sirven como muestra de una clase práctica para ejercitar dicho contenido.

- 1- Cargue el robot de 2 GDL que se encuentra en la librería de ARTE. Escriba diferentes valores para sus parámetros dinámicos, diseñe un controlador de posición que cumpla con 3 segundos de tiempo de establecimiento con cero error en estado estable ante una entrada paso de valor 0.8. Luego ejecute el archivo *simulate_robot_and_controller2DOF.mdl* de manera que pueda comprobar su diseño.

```
robot=load_robot('example','2dofplanar');
robot.dynamics.masses=[1 1];
robot.dynamics.Inertia=[0 0.2 0.2 0 0 0;0 0.2 0.2 0 0 0];
>> simulate_robot_and_controller2DOF.mdl
```

- 2- Con los mismos parámetros del robot, realice un control en cascada diseñando el controlador interno de velocidad y el externo de posición. Valore los resultados.

3.4 Análisis económico medioambiental

Con la ejecución de este proyecto se garantiza la continuidad del estudio de la robótica en la carrera Ingeniería Automática de la UCLV. Esto posibilita la formación de profesionales con conocimientos capaces de explotar estas tecnologías con una mayor eficiencia y eficacia en nuestro país.

Además, se obtiene un producto que puede ser usado por los investigadores del Departamento de Ingeniería Automática de dicha Universidad para validar, mediante simulación, el resultado de sus investigaciones. Por lo que el uso del simulador ARTE en la UCLV tendrá un alto impacto científico e investigativo

El impacto académico que contiene esta investigación es relevante, pues contribuye con mayor eficiencia al aprendizaje de los contenidos de la robótica por parte de los estudiantes de dicha carrera.

Debido al precio que posee un robot industrial en el mercado (aproximadamente €14,000) se dificulta su obtención por parte de la institución. Por tanto, el empleo de este simulador se presenta como una excelente solución para la actual situación que se presenta en el centro en cuanto a la docencia y la investigación.

Por otro lado, el empleo del simulador ARTE en apoyo a la enseñanza de la robótica en la UCLV posibilita que los profesionales formados en dicha institución posean los conocimientos necesarios para realizar un uso adecuado de los robots industriales. Esto garantiza que la explotación de estas tecnologías sea hecha de una forma científica y en armonía con el Medio Ambiente.

3.5 Consideraciones finales del capítulo

- El simulador ARTE presenta funciones y herramientas con los que se pueden realizar las prácticas de los contenidos de la asignatura Robótica de forma efectiva.
- Contenidos como la dinámica directa e inversa y la simulación de controladores de robots industriales que hasta el momento solo se imparten de forma teórica, pueden ser desarrollados de forma práctica a través de ARTE.

- El simulador ARTE permite la inserción de la programación de robots industriales en los tópicos que se imparten en la asignatura.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- 1 Para desarrollar el estudio de la robótica de manipulación de forma eficaz se precisa del uso de simuladores. Estos pueden ser de uso propietario como RoboDK y otros como RoboAnalyzer, VREP, ROBOLAB, Robotics Toolbox y ARTE tienen la posibilidad de ser explotados de forma gratuita.
- 2 En la enseñanza de la robótica en la carrera Ingeniería Automática de la UCLV, ARTE destaca hasta el momento por encima de otros simuladores mencionados anteriormente.
- 3 ARTE presenta una serie de funciones y herramientas con las que se puede impartir los contenidos de la asignatura (representación espacial, cinemática directa e inversa, dinámica directa e inversa, generación de trayectorias, la programación de robots y la simulación de los controladores diseñados) de forma teórica y práctica, lo que posibilita que los alumnos asimilen de forma eficaz dichas materias.
- 4 Con el uso de ARTE se puede dar continuidad al estudio de la robótica en la UCLV, con la implementación de contenidos que anteriormente no se podían impartir.

Recomendaciones

- 1 Profundizar en el estudio del software de modo que puedan ser utilizadas otras funciones con las que cuenta el simulador.
- 2 Conformar una serie de ejercicios más elaborados que puedan ser empleados en las clases prácticas para que el estudiante se adapte al rigor real de cada contenido.

- 3 Hacer uso del simulador tanto para cursos de postgrado como en las investigaciones del Departamento de Automática referentes al estudio de la robótica de manipulación.
- 4 Comenzar la implementación del simulador en el software Scilab con el fin de garantizar la continuidad del uso de ARTE en la UCLV en software libre.

REFERENCIAS BIBLIOGRÁFICAS

- AGUILERA, O. A. V., VEGA, V. M., MENDOZA, Y. C., BÁRBARA, R. & AGUILERA, V. 2014. Planificador para la navegación en la robótica móvil basada en agentes en entornos estructurados Navigation planner for mobile robotics based on agents for structured enviropment. 39-53.
- ALCAÇARENHO, N. M. N. 2016. *Plataforma de Simulação de Sistemas Distribuídos de Manufatura*.
- ARTECHE, M. M. 2004. *Simulación en robótica mediante" virtual robot"*, Universidad Politécnica de Valencia.
- BAHUGUNA, J., CHITTAWADIGI, R. G. & SAHA, S. K. Teaching and learning of robot kinematics using RoboAnalyzer software. Proceedings of Conference on Advances In Robotics, 2013. ACM, 1-6.
- BARRIENTOS, A. 2007. Fundamentos de robótica. 2da Edition ISBN: 978-84-481-5636-7 ed. Madrid, España: MCGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
- CERRILLO, F. A. A., CASTRO, A. L., MENDOZA, E. G. & PALAFOX, M. B. 2015. PROTOTIPO PARA APRENDIZAJE-ENSEÑANZA DE LA ROBÓTICA. *Multidisciplinas de la Ingeniería*.
- CORKE, P. 2017. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, EE.UU, Springer.
- FREESE, M., SINGH, S., OZAKI, F. & MATSUHIRA, N. Virtual robot experimentation platform v-rep: a versatile 3d robot simulator. International Conference on Simulation, Modeling, and Programming for Autonomous Robots, 2010. Springer, 51-62.
- GARCIA GARCIA, D., VALERIANO MEDINA, Y., PORTAL LINARES, J. A. & HERNÁNDEZ SANTANA, L. 2015. Sistema de navegación basado en modelo dinámico no lineal de Vehículo Autónomo Sumergible. *Ingeniería Electrónica, Automática y Comunicaciones*, 36, 83-97.
- GIL, A., REINOSO, O., MARIN, J. M., PAYA, L. & RUIZ, J. 2015. Development and deployment of a new robotics toolbox for education. *Computer Applications in Engineering Education*, 23, 443-454.

- GONZÁLEZ-RODRÍGUEZ, R., SANTANA, L., SALÍ, H., RUBIO, E. & GUERRA, Y. 2011. Control monocular 3D dinámico basado en imagen. *Ingeniería Electrónica, Automática y Comunicaciones*, 32, 15-30.
- HERNÁNDEZ, L., GONZÁLEZ, R., SAHLI, H., RUBIO, E. & GUERRA, Y. A decoupled control for visual servoing of camera-in-hand robot with 2d movement. *Electronics, Robotics and Automotive Mechanics Conference*, 2008. CERMA'08, 2008. IEEE, 304-309.
- IFR 2017. Executive summary: world robotics 2017 industrial robots. *Available online on <http://www.worldrobotics.org>. Accessed in 20 de febrero 2018.*
- IZAGUIRRE CASTELLANOS, E. 2012. *Control cinemático en el espacio de tareas de robot paralelo neumático en aplicación de simulador de movimiento*. Ph.D, Universidad Central "Marta Abreu" de Las Villas, Facultad de Ingeniería Eléctrica, Departamento de Automática y Sistemas Computacionales.
- IZAGUIRRE, C. E., HERNÁNDEZ, S. L., RUBIO, R. E., PRIETO, E. P. J. P. & PASCUAL, O. U. 2011. Análisis cinemático y control articular aplicado a simulador de movimiento de estructura paralela. *Revista Ingeniería Electrónica, Automática y Comunicaciones ISSN: 1815-5928*, 32, 1-12.
- JARA, C. A., CANDELAS, F. A., PUENTE, S. T., POMARES, J. & TORRES, F. 2010. Practical experiences using RobUALab. ejs: a virtual and remote laboratory for Robotics e-learning. *IFAC Proceedings Volumes*, 42, 1-6.
- MAZAIRA-MORALES, I. & CHANG-MUMAÑ, F. 2010. CONTROL COORDINADO DE ROBOTS MANIPULADORES. UN ENFOQUE DE LA TEORÍA DE REGULACIÓN NO LINEAL. *Ciencia en su PC*, 11-23.
- SÁNCHEZ, F. Á. B. & GUZMÁN, A. F. 2012. La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 13, 120-136.
- VELÁSQUEZ COSTA, J. A. 2009. Software para el modelamiento, simulación y programación de aplicaciones robotizadas. *Industrial Data*, 12, 032-036.

ANEXOS

Anexo I Parámetros de la variable robot

```

robot =

    name: 'ABB_IRB140_M2000'
    DH: [1x1 struct] % parámetros de D&H
    J: [] % Jacobiano del robot
    inversekinematic_fn: 'inversekinematic_irb140(robot, T)'
    directkinematic_fn: 'directkinematic(robot, q)'
    DOF: 6 % Grados de libertad
    kind: 'RRRRRR' % Tipo de articulaciones
    maxangle: [6x2 double]
    velmax: [6x1 double]
    accelmax: [6x1 double]
    linear_velmax: 2.5000
    T0: [4x4 double] % posición inicial de la base del robot
    debug: 0
    q: [6x1 double] % posición articular
    qd: [6x1 double] % velocidad articular
    qdd: [6x1 double] % aceleración articular
    time: []
    q_vector: []
    qd_vector: []
    qdd_vector: []
    last_target: [4x4 double]
    last_zone_data: 'fine'
    tool0: [1 0 0 0 1 0 0 0 0.1000 0 0 0 1 0 0 0 0 0 0] % configuración de la
herramienta
    wobj0: []
    tool_activated: 0
    path: [1x70 char]
    graphical: [1x1 struct] % variables graficas
    axis: [-0.5000 0.7500 -0.7500 0.7500 0 1.1000]
    has_dynamics: 1 % contiene variables dinámicas
    dynamics: [1x1 struct] % estructura dinámica
    motors: [1x1 struct] % variables de los motores

```

Anexo II Contenido de robots presentes en el simulador

Tabla 1 Modelos de robots presentes en ARTE

Fabricante	Modelo	Cinemática	Dinámica	Modelo 3D	GDL
ABB	IRB120	X	-	X	6
	IRB140	X	X	X	6
	IRB1600_6	X	-	X	6
	IRB1600ID	X	X	X	6
	IRB2400	X	X	X	6
	IRB4400	X	-	X	6
	IRB4600	X	-	X	6
	IRB52	X	-	X	6
	IRB6620	X	-	X	6
	IRB6620LX	X	X	X	6
	IRB6650S_125	X	-	X	6
	IRB6650S_200	X	-	X	6
	IRB6650S_90	X	-	X	6
	IRB760	X	-	-	6
	IRB7600_150	X	-	X	6
	IRB7600_400	X	X	X	6
	IRB7600_500	X	-	X	6
ADEPT	Viper_s1700D	X	X	X	6
EPSON	Prosix_C3	X	-	X	6
FANUC	LR_MATE_200iC	X	-	X	6
KUKA	KR_16_arc	X	-	X	6
	KR_30_L16_2	X	-	X	6
	KR_1000_1300	X	-	X	6
	KR_2ARC_HW	X	-	X	6
	KR_arc	X	-	X	6
	KR_scara_R350	X	X	X	4
	KR5_sixx_R650	X	-	X	6
	KR5_sixx_R850	X	-	X	6

	KR6_2	X	-	X	6
	KR30_jet	X	-	X	6
	KR90_R2700	X	-	X	6
	KR90_R3100	X	X	X	6
MITSUBISHI	PA-10	X	-	X	6
	rv-2aj	X	-	-	6
	rv-6s	X	-	X	6
STAUBLI	RX160L	X	X	X	6
	RX170BL	X	-	X	6
	TP80	X	X	X	6
UNIMATE	Puma560	X	X	X	6
EXAMPLE	1dofplanar	X	X	X	1
	2dofplanar	X	X	X	2
	2dofplanarRP	X	X	X	2
	3dofplanar	X	X	X	3
	3dofspherical	X	X	X	3
	3dofsphericalb	X	X	X	3
	3RPR*	X	-	-	6
	3RRR*	X	-	-	6
	5R*	X	-	-	5
	Delta*	X	X	X	3
	Maryland*	X	X	X	3
	Scara	X	-	-	4
	Stanford	X	X	-	6

* Robots industriales paralelos.

Anexo III Programa completo de manipulación de objetos.

```
function Programa
global robot TD_gripper RT_inicio RT_cerca_circulo RT_coger
global RT_cerca_rombo RT_soltar

% Cargando el robot en su ambiente y con las herramientas
robot = load_robot('abb','irb140');
```

```

robot.equipment = load_robot('equipment','tables/table_two_areas');
robot.tool= load_robot('equipment','end_tools/parallel_gripper_0');
robot.piece=load_robot('equipment','cylinders/cylinder_tiny');

robot.graphical.draw_axes=0; % no mostrar los ejes del robot
robot.tool.graphical.draw_axes=0; %no mostrar los ejes de la herramienta
robot.equipment.graphical.draw_axes=0; %no mostrar los ejes de la mesa
robot.piece.graphical.draw_axes=0;% mostrar los ejes de la pieza
%inicializar la posición de la pieza para realizar la simulación.
robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]';

drawrobot3d(robot, robot.q);%mostrar el robot con todo el ambiente
adjust_view(robot); %ajustar la vista

teach %llamar a la función teach para establecer las posiciones deseadas
%definición de la herramienta mediante la estructura de datos de RAPID
TD_gripper=[1, [[0,0,0.125],[1,0,0,0]], [0.1,[0,0,0.100],[1,0,0,0],[0,0,0]]];

%definir los puntos para la Simulación obtenidos mediante teach
RT_inicio=[[0.5000,-0.2500,0.5000],[0.7071,-0.0000,0.7071,0.0000],[-1,-1,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
RT_cerca_circulo=[[-0.1000, -0.5000, 0.4000],[0.0000, 0.7071, 0.7071, 0.0000],[-2, -1, -1, 0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
RT_coger=[[-0.1000, -0.5000, 0.2800],[0.0000, 0.7071, 0.7071, -0.0000],[-2, -1, -1, 0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
RT_cerca_rombo=[[0.5000, -0.5000, 0.4000],[0.0000, 0.7071, 0.7071, -0.0000],[-1, -1, 0, 0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
RT_soltar=[[0.5000, -0.5000, 0.2800],[0.0000, 0.7071, 0.7071, 0.0000],[-1, -1, 0, 0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

main %llamado a la función main

end
%función principal del programa
function main()

global TD_gripper RT_inicio RT_cerca_circulo RT_coger RT_cerca_rombo
RT_soltar
%cerrar la herramienta
simulation_close_tool;

%mover al punto inicial
MoveJ(RT_inicio, 'vmax' , 'fine' , TD_gripper, 'wobj0');

% abrir la herramienta
simulation_open_tool; %Reset dol;

%Mover el EF a un punto aproximado
MoveJ(RT_cerca_circulo, 'vmax' , 'fine' , TD_gripper, 'wobj0');
%mover hasta el punto donde está la pieza
MoveL(RT_coger, 'vmax' , 'fine' , TD_gripper, 'wobj0');

%luego cerrar la herramienta y coger la pieza
simulation_close_tool;
simulation_grip_piece;

```

```
%mover hasta el punto donde cercano al que se llevo anteriormente.
MoveL(RT_cerca_circulo, 'vmax' , 'fine' , TD_gripper, 'wobj0');

%Mover hasta un punto cercano del lugar de destino de la pieza
MoveJ(RT_cerca_rombo, 'vmax' , 'fine' , TD_gripper, 'wobj0');
%descender hasta el lugar donde se desea colocar la pieza
MoveL(RT_soltar, 'vmax' , 'fine' , TD_gripper, 'wobj0');

%entonces abrir la herramienta y soltar la pieza
simulation_open_tool;
simulation_release_piece;

%mover arriba al punto aproximado anterior
MoveL(RT_cerca_rombo, 'vmax' , 'fine' , TD_gripper, 'wobj0');

% regresar al punto inicial
MoveJ(RT_inicio, 'vmax' , 'fine' , TD_gripper, 'wobj0');

end
```