



ANEXO A

Descarga e Instalación del Simulador V-rep

La descarga de este software se puede realizar desde la página oficial de COPPELIA ROBOTICS¹ empresa creadora de la herramienta V-rep.

Ubicado en la página web del simulador, dirigirse a la pestaña “downloads”.



Figura 1. Sitio web oficial de COPPELIAROBOTICS y su herramienta de simulación V-rep.

A continuación se mostrarán, en el sitio web, los diferentes tipos de licencias y las opciones de descarga según el sistema operativo en el que se desee instalar la aplicación. Para este proyecto se descargó la versión v3.1.3, para compatibilidad de los sistemas operativos Microsoft Windows y con licencia educativa V-rep^{pro-edu}, la cual no posee restricciones de ningún tipo.

¹ Sitio web oficial de COPPELIA ROBOTICS <http://www.coppeliarobotics.com/>



Al dar clic en esta elección automáticamente inicia la descarga del archivo (V-REP_PRO_EDU_V3_1_3_Setup.exe) que pesa alrededor de 100 Mbytes.



Anteriores versiones de V-REP están disponibles -> aquí <-

 ↓	 ↓	 32 ↓	 64 ↓	v-rep <small>pro^{edu}</small> V-REP PRO EDU v3.1.3 rev. 2b no limitada versión educativa. Gratis. -> Condiciones de licencia
 ↓	 ↓	 32 ↓	 64 ↓	v-rep <small>pro^{eval}</small> V-REP PRO EVAL v3.1.3 rev. 2b La versión comercial EVALUACIÓN. Gratis. Saving está desactivada. Ver más abajo para detalles de compra.
 ↓	 ↓	 32 ↓	 64 ↓	v-rep <small>player</small> V-REP rev v3.1.3 PLAYER. 2b Reproduzca sus simulaciones en cualquier computadora! Gratis. capacidades de edición limitada. puede ser distribuido libremente.
				v-rep <small>pro</small> V-REP rev PRO v3.1.3. 2b La versión comercial (es decir, no educativos). Póngase en contacto con nosotros o uno de nuestros socio distribuidores .

Figura 2. Opciones y licencias de descarga para V-rep.

Al finalizar la descarga del archivo, se procede a instalar el simulador ejecutando el archivo (V-REP_PRO_EDU_V3_1_3_Setup.exe). A continuación se inicia el asistente automático para instalación de V-rep sobre Microsoft Windows. Aparecerá una ventana de bienvenida y familiarización con el instalador. Seguidamente una



nueva ventana donde se encuentran los términos de la licencia, si se está de acuerdo con estos, clicar en “SI yo acepto los términos de licencia” y a continuación en siguiente.

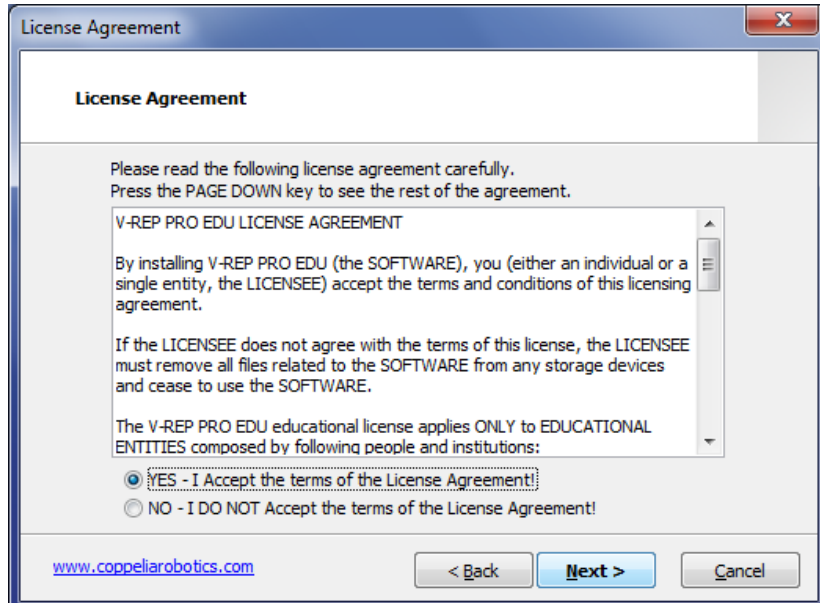


Figura 3. Instalación del simulador V-rep, ventana de acuerdos de licencia.

Posteriormente se despliega una nueva ventana, en la cual pregunta si el usuario desea crear accesos directos desde el menú inicio y desde el escritorio.

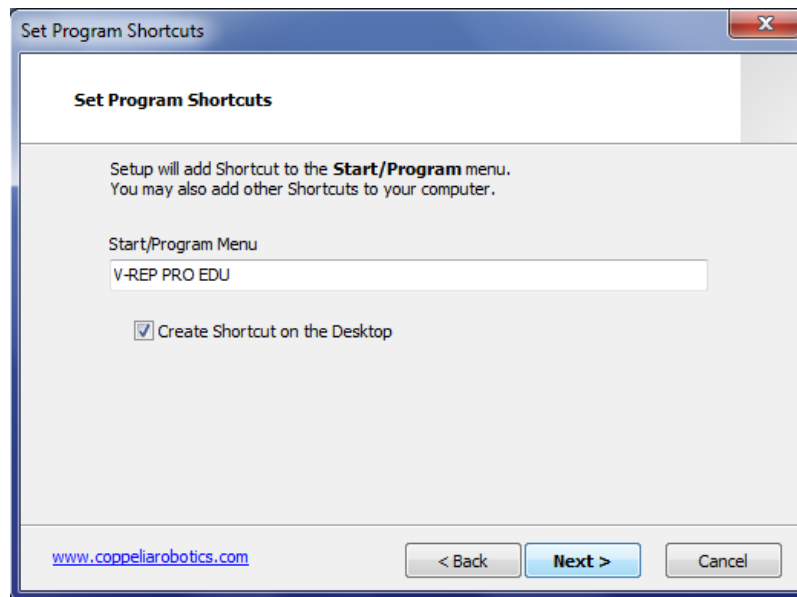


Figura 4. Instalación del simulador V-rep, ventana de accesos directos.



Por último una ventana en la cual pregunta por las configuraciones de instalación y si se está de acuerdo con ellas. Si es así, elegir siguiente y de esta manera se iniciará la instalación del simulador en el ordenador.

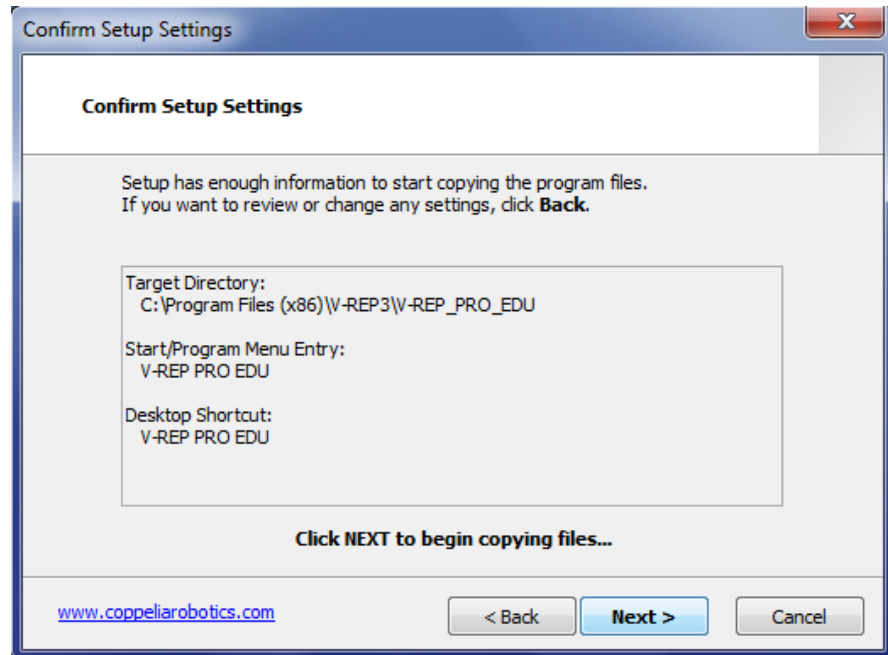


Figura 5. Instalación del simulador V-rep, ventana de confirmación de configuración.

Al terminar la instalación se mostrará un mensaje en el cual se pregunta si se desea ejecutar la aplicación inmediatamente, si es así verificar el cuadro de selección y clic en finalizar para arrancar el simulador V-rep.



ANEXO B

Entornos de Simulación para Robots

Al encontrarnos dentro de la era tecnológica, existe saturación de la información lo que obliga a ser bastante selectivos en el la elección de lo que realmente se necesita para este fin. En este sentido existe gran cantidad de soluciones software donde se pueden llevar a cabo simulaciones robóticas. Pero cada simulador tiene diversos factores que influirán a favor o en contra en la decisión final para concretar por uno de ellos. A continuación se presentan los simuladores más opcionados para cumplir la función de simulación del ciclo de marcha del robot bípedo Bioloid sobre un ambiente 3D.

1. Webots™

El entorno comercial 3D Webots², permite el modelado y la programación para la simulación de la locomoción robótica. Este simulador hace posible trabajar con diversos tipos de robots de manera independiente y bajo un mismo entorno. Además el usuario es libre de definir las propiedades físicas de cada objeto como su forma, color, textura, masa, entre otros. Del mismo modo Webots™ cuenta con una biblioteca de actuadores y sensores donde podrá ajustar de forma individual rango, ruido, el campo de visión etc. Este simulador incluye sensores de distancia (de infrarrojo y ultrasonido), telémetros, sensores de luz, sensores táctiles, sensor de posicionamiento (GPS), brújula, cámaras (1D, 2D a color o blanco y negro), receptores (radio y de infrarrojo), sensores de posición para los servos, encoders incrementales de ruedas, entre muchos más. Los algoritmos de programación desarrollados sobre este simulador podrán usarse en robots palpables.

² Webots: <http://www.cyberbotics.com/>

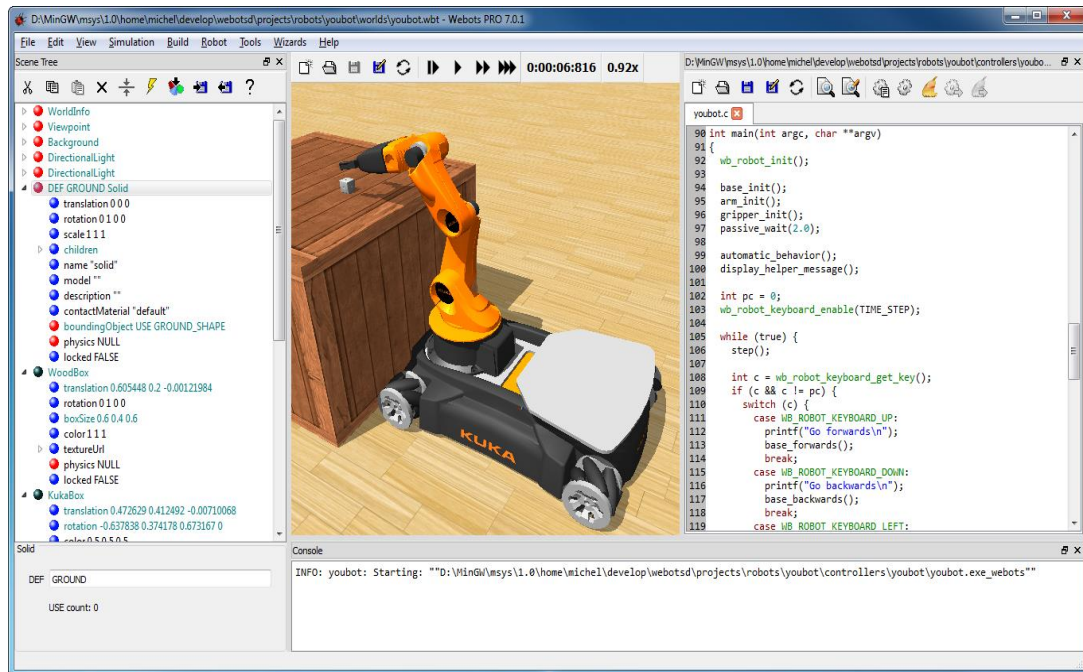


Figura 6. Simulador Webots™.

Otras características significativas de Webots™ son; que incluye una biblioteca completa y exclusiva de sensores y actuadores, Permite programar los robots en C, C++, Java y otros software de terceros a través de protocolos TCP / IP. Para la simulación de física realista, utiliza ODE (Open Dinámica Motor) ODE que es un motor de simulación de código abierto que contiene una biblioteca de alto rendimiento para la simulación dinámica de cuerpos rígidos. En este programa se pueden crear y exportar archivos AVI o MPEG para muestras multimedia de las simulaciones realizadas. Asimismo contiene por defecto muchos ejemplos de código fuente del controlador y de los modelos de robots comerciales a disposición del usuario [53].

2. Gazebo

Es un simulador Gazebo³ 3D de código abierto, que permite cargar diferentes tipos de robots, sensores y actuadores para desempeñarse especialmente en entornos exteriores, aunque también se utilicen sin mayor inconveniente los ambientes interiores. Ofrece herramientas sofisticadas para el fácil uso de sensores y actuadores. Es apto y completo para trabajar con robots en una interfaz sencilla y

³ Gazebo: <http://gazebosim.org/>



fluida, además de la posibilidad de desarrollar modelos online, con materiales, mallas y plugins, los cuales ofrece Gazebo con acceso a contenido creado por el usuario en la interfaz en la cual simplemente hay que arrastrar y soltar los elementos necesarios. Esta interfaz gráfica está rediseñada para proporcionar acceso a las propiedades de la simulación, la modificación de los modelos, y las propiedades físicas de los modelos. Cuenta con el sistema operativo para robots ROS, el cual proporciona el control de los robots y de sus sensores articulados, para aplicaciones especiales.



Figura 7. Simulador GAZEBO.

El simulador cuenta con un conjunto de plugins que crean una interfaz de comunicación perfecta entre Gazebo y ROS, permitiendo a los desarrolladores realizar cambios fácilmente entre el hardware y la simulación, además de aplicar una capa de abstracción física que permite a cuatro motores de simulación dinámica diferentes, (ODE⁴, Bullet⁵, SimBody⁶ y DART⁷), ser utilizados dentro de Gazebo, y permite a los plugins acceder a los parámetros de cada motor físico. Este software es una herramienta que nos permite desarrollar comportamientos jerárquicos para el

⁴ Open Dynamics Engine: <http://opende.sourceforge.net/>

⁵ Bullet: <http://bulletphysics.org/wordpress/>

⁶ SimBody: <https://simtk.org/home/simbody/>

⁷ DART: <http://dartsim.github.io/>



robot con el cual se desea trabajar, como es caso de las cadenas cinemáticas de un robot [54].

Gazebo ha logrado desempeñarse en diversos campos como:

- ✓ Custom UI: Creación de sus propias interfaces para la interacción del usuario con Gazebo personalizado, o estudios de investigación.
- ✓ Prototipado rápido: Diseño y desarrollo de robots junto con los controladores del robot con modelos desde Solidworks y Nodos ROS.
- ✓ Locomoción Legged: Diseño y prueba de la locomoción de extremidades de robots.
- ✓ Competencia de Robots: Ejecutar las competiciones de equipos en entornos de fútbol o al aire libre.
- ✓ Simulación de Personas: Utilizar los datos de captura del movimiento humano para animar personas virtuales.
- ✓ Pruebas de regresión: Las utiliza Gazebo para poner a prueba los sistemas de robots simples o complejos, para comprobar y comparar el código de la dinámica.

3. Microsoft Robotics Developer Studio RDS-4

Este software de simulación es libre y puede ser utilizado por desarrolladores, profesionales y aficionados para crear aplicaciones robóticas con una amplia variedad de hardware basado en la programación .NET, la cual es una plataforma de desarrollo lanzada al mercado por MicroSoft, y en la que ha estado trabajando durante los últimos años. Microsoft .NET es un conjunto de tecnologías de software compuesto de varios lenguajes de programación que se ejecutan bajo el Framework .NET. Es además un entorno completamente orientado a objetos y es capaz de ejecutarse bajo cualquier plataforma.



Microsoft Robotics Developer Studio⁸ 4, cuenta con un soporte para el sensor de Kinect y una plataforma de referencia definida conocida como Mark (Autonomous Robot Móvil usando Kinect). Puede soportar una amplia gama de plataformas robóticas, ya sea corriendo directamente sobre la plataforma (si tiene un PC integrado con Windows) o controlar el robot desde un PC con Windows a través de un canal de comunicación, tales como WiFi o Bluetooth. Además de proporcionar soporte para Microsoft Visual Studio 2010, Microsoft Robotics Developer Studio 4 ofrece el Lenguaje de Programación Visual (VPL) que permite a los desarrolladores crear aplicaciones simplemente arrastrando y soltando componentes en un lienzo y conectarlos entre sí, sin necesidad de hardware. El lenguaje de programación de Microsoft Visual (VPL) es un entorno de desarrollo de aplicación diseñada específicamente para trabajar con DSS (Distributed Software Services). Los programas se definen gráficamente en diagramas de flujo de datos en lugar de la típica secuencia de comandos e instrucciones que se encuentran en otros lenguajes de programación. VPL proporciona una manera fácil de definir rápidamente cómo fluyen los datos entre los servicios. Es útil para el comienzo de los programadores, ya que pueden especificar rápidamente su intención, pero también es muy adecuado para ayudar con la creación de prototipos y generación de código para más programadores experimentados.

⁸ RSD 4: <http://msdn.microsoft.com/en-us/library/bb881626.aspx>

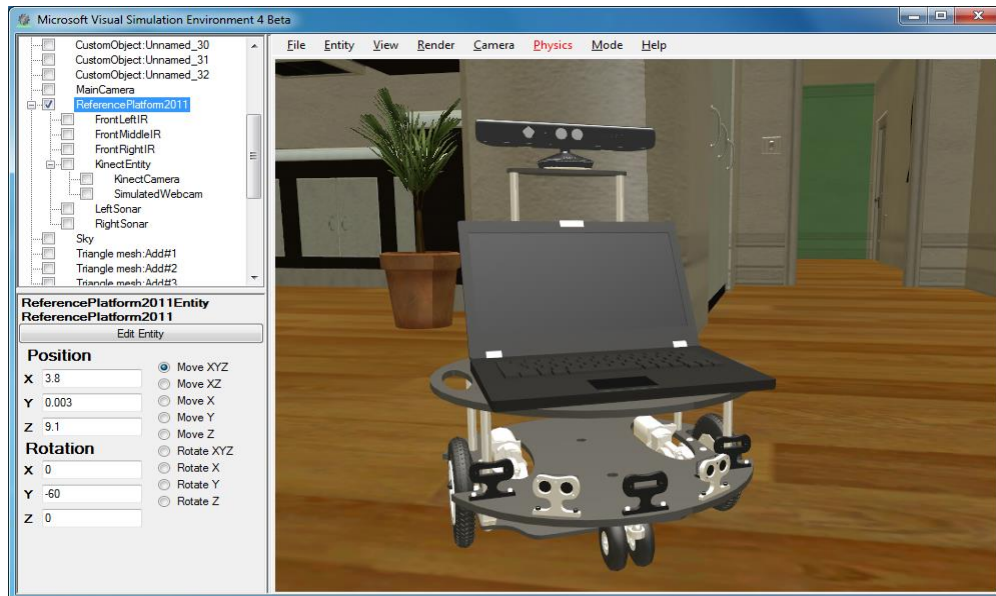


Figura 8. Microsoft Robotics Developer Studio (RDS 4).

Este entorno de simulación trabaja con Visual Simulation Environment (VSE) que proporciona un entorno de simulación de alta fidelidad impulsado por motor NVIDIA PhysX™ para ejecutar simulaciones en 3D en calidad de video juego, con las interacciones de la física del mundo real.

Para ayudar a los principiantes, Robotics Developer Studio contiene una amplia documentación y una amplia serie de ejemplos y tutoriales que muestran cómo escribir aplicaciones que van desde el simple "Hola Robot", hasta aplicaciones complejas que se ejecutan de forma simultánea en múltiples robots.

RDS4 es compatible con Windows 7 por lo que además de utilizar Robotics Developer Studio como un entorno de desarrollo independiente también puede ser utilizado con cualquiera de las versiones de Visual Studio, incluyendo la versión Express. Para que los nuevos usuarios instalen el servicio RSD4, se debe instalar Visual Studio antes del Robotics Developer Studio. Microsoft RDS también permite interactuar con los robots que utilizan Windows o interfaces de usuario basadas en Web. Usted puede monitorear y controlar remotamente un robot usando HTML y JavaScript en la Web [55].



4. USARSim

El simulador USARsim (Sistema Unificado de Automatización y Simulación Robot), fue creado con el propósito de brindar diversos escenarios donde se requieran realizar rescates, operaciones militares, entre muchas más acciones que sean necesarias efectuar en casos de emergencias, que representen un alto riesgo para la vida humana y sea necesario utilizar robots que puedan suplir estas tareas complicadas y de alto riesgo para el hombre. En este sentido USARSim⁹ puede representar entornos y tareas de coordinación a gran escala en ambientes peligrosos, extremos, de difícil acceso y bajo la presión del tiempo, en fin todas aquellas situaciones que puedan terminar en desastres naturales, accidentes o fenómenos de este estilo. Brindando la oportunidad de estudiar posibles estrategias para afrontar estas situaciones que rara vez se presentan en la vida real y disminuir la incertidumbre a la hora de actuar bajo esas circunstancias.

USARSim ofrece una simulación de alta fidelidad donde los robots y diversos entornos son destinados como herramienta de investigación para el estudio de la interacción humano-robot (HRI) y la coordinación multirobot para búsqueda y rescate urbano (USAR). Desde su lanzamiento inicial, se ha ampliado para admitir diversos ambientes, incluyendo los robots de autopistas, el desafío urbano de DARPA, fútbol robótico, submarinos, humanoides y helicópteros. USARSim fue diseñado como un compañero de simulación para el Instituto Nacional de Estándares (NIST) para pruebas de desempeño de Robots móviles programados para Búsqueda y Rescate Urbano. El test de NIST USAR presenta un ambiente estandarizado de desastres que consiste en tres escenarios: Amarillo, Naranja, Rojo de acuerdo al ambiente simulado. El proyecto USAR se centra en el estudio y ejecución del comportamiento de robots y la interacción física con entornos llenos de obstáculos y escombros regados por todo el escenario. USARSim soporta HRI por su fina representación de elementos dentro de la interfaz de usuario (en particular, la cámara de vídeo), la simulación realista de robots móviles y su desempeño, además de la simulación del control remoto humano-robot.

⁹ USARsim : <http://sourceforge.net/projects/usarsim/>



Figura 9. Simulación Nivel Amarillo en USARSim V3.1.3.

La alta fidelidad a un bajo costo es posible gracias a la construcción de la simulación sobre un motor diseñado para video juegos. Es una plataforma comercial de gran volumen que proporciona representación visual de alta calidad y el modelado físico, este software realiza un dedicado esfuerzo para realizar las tareas de modelado de prototipos robóticos, sistemas de control, sensores, herramientas de interfaz y entornos. Estas tareas son, a su vez, aceleradas por las herramientas de edición y desarrollo avanzadas, las cuales están integradas con el motor del mismo.

La versión actual de la simulación consiste en diferentes escenarios, los modelos de robots comerciales y experimentales, y los modelos de sensores. Igualmente se suministran las interfaces de usuario, la automatización y la coordinación lógica que desee probar el usuario en sus aplicaciones [56].



5. V-REP

El entorno de simulación V-REP, figura 10, permite el diseño y construcción de animaciones CAD basadas en la dinámica real de los sistemas, y esto es posible gracias a que cuenta con tres motores de simulación, Bullet, ODE y VORTEX aunque este último permite realizar la simulación durante algunos segundos, el usuario escoge el motor que mejor se adapte a sus requerimientos de simulación.

ODE es una plataforma de código abierto con librerías de alto rendimiento para la simulación dinámica de cuerpos rígidos. Está totalmente equipado, estable y robusto, con una plataforma fácil de utilizar en C/C++ API. Este ha avanzado en tipos de articulaciones y detección de colisiones en conjunto con la fricción. ODE es útil para la simulación de vehículos, objetos en entornos de realidad virtual y criaturas virtuales. Actualmente se utiliza en muchos juegos para computador, herramientas de creación 3D y herramientas de simulación.

Por otra parte la biblioteca de física Bullet, es un motor de código abierto que ofrece detección de colisiones 3D, dinámicas de cuerpos rígidos, y la dinámica de cuerpos blandos (característica que actualmente no se admite en V-REP). Se utiliza en los juegos, y en los efectos visuales en películas. A menudo se considera como un motor de física del video juegos.

Vortex es un motor de física comercial para reproducir simulaciones dinámicas de alta fidelidad. Vortex ofrece parámetros del mundo real para un gran número de propiedades físicas, por lo cual este motor es realista y preciso. Vortex se utiliza principalmente para aplicaciones de alto rendimiento y precisión, perfecto para aplicaciones industriales y de investigación. El plugin de Vortex por defecto para V-REP es un producto comercial, y su versión gratuita soporta actualmente la simulación durante 20 segundos.



Asimismo V-rep¹⁰ realiza internamente la consecuente inyección de fuerzas en el modelo dinámico directo con el fin de evitar la interpenetración entre los cuerpos. V-REP además permite importar gran variedad de formatos CAD, lo cual facilita la personalización de piezas y sistemas completos. V-REP maneja seis enfoques de programación dentro de los cuales tres son remotos cubriendo así una gran gama de necesidades del usuario, además de que estos enfoques son compatibles entre sí, como también ofrece una extensa lista de funciones de programación. Adicionalmente se pueden implementar algoritmos de control en lenguaje C / C++, Python, Java, Lua, Matlab o Urbi.

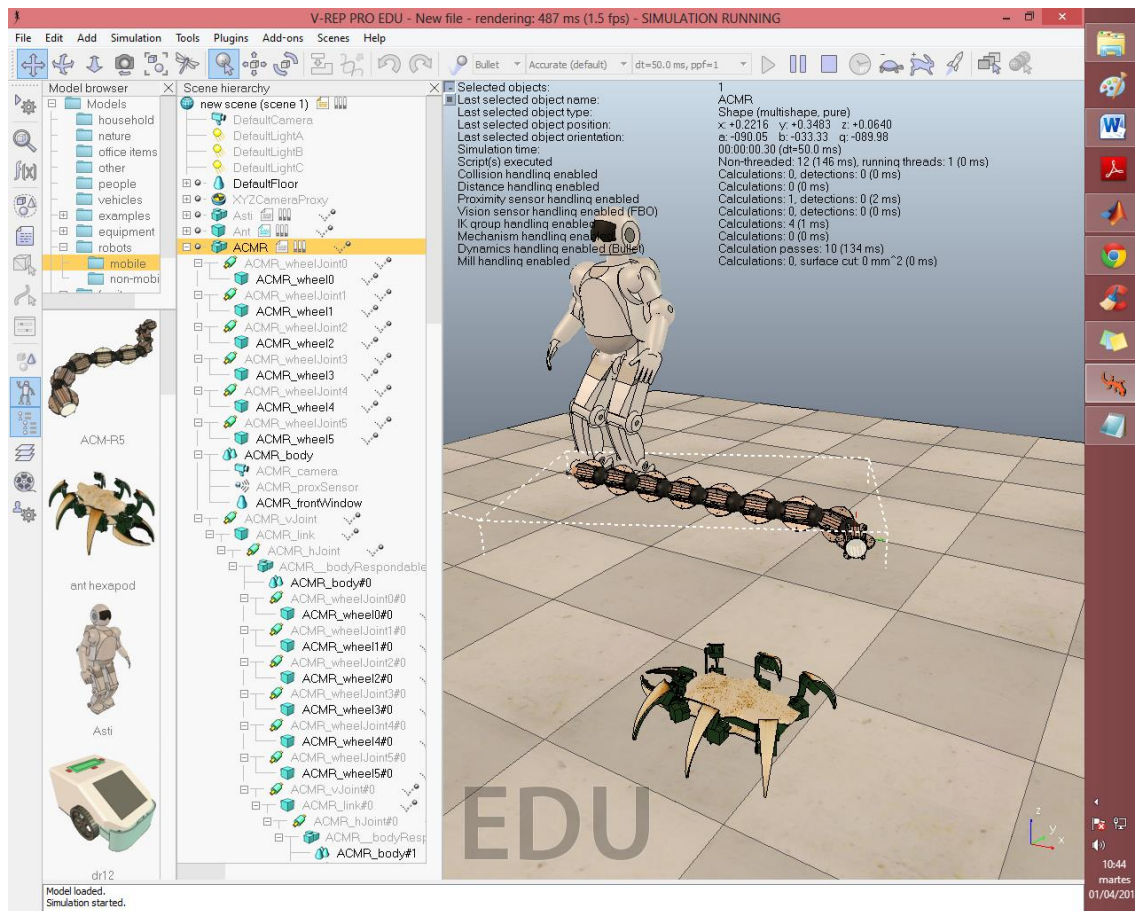


Figura 10. Entorno de simulación V-REP versión 3.1.1

¹⁰ V-rep (Virtual Robot Experimentation Platform): <http://www.coppeliarobotics.com/>



Varios sensores están disponibles, incluyendo sensores de proximidad, infrarrojos y en las últimas versiones la simulación del sensor Kcine¹¹. También cuenta con módulos encargados de realizar el cálculo de cinemática inversa, dinámica de los cuerpos, detección de colisión, cálculos de distancia mínima, planificación de trayectorias, entre muchas más funciones. Además posee una arquitectura de control distribuido de modo que se puede efectuar un número ilimitado de secuencias de comandos de control y varios mecanismos de extensión como plugin y aplicaciones de cliente personalizadas [57] [58].

¹¹ COPPELIA ROBOTICS, “Virtual Robot Experimentation Platform USER MANUAL”, V_Rep Pro/V_Rep Pro edu, version 3.0.4.



ANEXO C

Principales Funciones de la API Remota entre Matlab y V-rep

Como se necesita comunicar a Matlab con V-rep para manipularlo desde el lado del cliente, se usaron unas funciones que V-REP ya tiene definidas para desarrollar la API remota con Matlab. Tabla 1.

1	remApi()	
Descripción	Llama los archivos remApi.m, remoteApiProto.m, remoteApi.dll, extApi.h y crea el objeto con la etiqueta vrep.	
Sinopsis	vrep= remApi ('remoteApi','extApi')	
Parámetros	Debe copiar dentro de la misma carpeta del main los archivos remApi.m, remoteApiProto.m que se encuentran en el siguiente destino: C:\ProgramFiles\V_REP3\V_REP_PRO_EDU\programing\remoteApiBindings\matlab\matlab. remoteApi.dll que se encuentra dentro de la la siguiente direccion: C:\ProgramFiles\V_REP3\V_REP_PRO_EDU\programing\remoteApiBindings\\lib\lib\32Bit o de 64Bit según las características de su equipo. Y el archivo de cabecera extApi.h que se encuentra en: C:\Program Files\V-REP3	
2	simxFinish()	
Descripción	Termina todos los hilos de comunicación existentes	
Sinopsis	simxFinish(number clientID)	



general	
Sinopsis implementada	vrep.simxFinish(-1)
Parametros	-1 se pone para terminar todos los hilos de comunicación
3	simxStart()
Descripción	Inicia un hilo de comunicación. Esta función devuelve un 0 si la comunicación se logró, y un -1 si no se pudo comunicar.
Sinopsis general	[number_clientID]= simxStart (string_connectionAddress, number_connectionPort ,Boolean_waitUntilConnected, Boolean_doNotReconnectOnceDisconnected,number_timeOutInMs, number_commThreadCycleInMs)
Sinopsis implementada	clientID=vrep. simxStart ('127.0.0.1',19999,true,true,5000,5)
Parametros	number connectionPort : Se debe asignar un número de puerto para iniciar la conexión. Asegúrese de utilizar el mismo número de puerto tanto en el cliente Matlab, y en el archivo remoteApiConnections.txt. Los demás valores se dejan por defecto como lo recomienda el instructivo de V-Rep.
Sinopsis bloque de	Finalmente así queda el bloque de comunicación escrito en Matlab vrep=remApi('remoteApi','extApi');



comunicación	vrep.simxFinish(-1); clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5)
4	simxPauseCommunication()
Descripción	Permite cortar temporalmente el hilo de comunicación.
Sinopsis general	simxPauseCommunication(boolean pause)
Sinopsis implementada	vrep.simxPauseCommunication(clientID,1)
Parametros	boolean: se pone la etiqueta fijada en la funcion simxStart() pause: 1

Tabla 1A.Bloque Para Iniciar La Comunicación Desde La Api Remota Con Vrep

5	Servicio de servidor API remota continua
Descripción	Con este método de funciones de la API remota siempre se ejecutarán en el lado del servidor, incluso si la simulación no se está ejecutando. Lo cual habilita la funcion simxStartSimulation() para poder correr la simulacion desde la GUI de Matlab.
Sinopsis implementada	portIndex1_port=19999 portIndex1_debug=false portIndex1_syncSimTrigger=true
Parámetros	Asegurese de asignarle a portIndex1_port el mismo puerto que definió en simxStart() tanto en el main, como en el archivo remoteApiConections.txt que se encuentra en C:\Program Files\V-REP3\V-REP_PRO_EDU.



6	simxStartSimulation()	
Descripción	Corre la simulación en V-Rep	
Sinopsis general	simxStartSimulation(number clientID,number operationMode)	
Sinopsis implementada	Vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot)	
7	simxPauseSimulation()	
Descripción	Pausa la simulación en V-Rep	
Sinopsis general	simxPauseSimulation(number clientID,number operationMode)	
Sinopsis implementada	vrep.simxPauseSimulation(clientID,vrep.simx_opmode_oneshot)	
8	simxStopSimulation()	
Descripción	Para la simulación en V-Rep	
Sinopsis general	simxStopSimulation(number clientID,number operationMode)	
Sinopsis implementada	vrep.simxStopSimulation(clientID,vrep.simx_opmode_oneshot)	

Tabla 1B. Grupo de funciones para iniciar, pausar y parar la simulación.

9	simxDisplayDialog()	
Descripción	Despliega un mensaje en el simulador V-REP	
Sinopsis general	simxDisplayDialog(number clientID,string titleText,	



	string mainText,number dialogType,string initialText, array titleColors,array dialogColors,number operationMode)
Sinopsis implementada	vrep.simxDisplayDialog(clientID,'ESTADO DE COMUNICACION', 'CONECTADO',vrep.sim_dlgstyle_message,'string initialText', [],[],vrep.simx_opmode_oneshot_wait)
10	simxEndDialog()
Descripción	Quita el o los mensajes que se haya desplegado en el servidor V_REP
Sinopsis general	simxEndDialog(number clientID,number dialogHandle,number operationMode)
Sinopsis implementada 1	vrep.simxEndDialog(clientID,vrep.sim_dlgret_still_open,vrep.simx_opmode_oneshot_wait)
Sinopsis implementada 2	vrep.simxEndDialog(clientID,vrep.sim_dlgret_ok,vrep.simx_opmode_oneshot_wait)

Tabla 1C. Grupo de funciones para desplegar mensajes en V-Rep

11	simxGetObjectHandle()
Descripción	Esta función enlaza un objeto que se encuentra en una escena del servidor V-REP con el cliente Matlab para que pueda ser manejada desde la GUI.
Sinopsis general	[number errorCode,number handle]= simxGetObjectHandle(number clientID,string objectName,number operationMode)
Sinopsis	[errorC,ART_1]=vrep.simxGetObjectHandle(clientID,'ART_1',



implementada	vrep.simx_opmode_oneshot_wait)
Parámetros	<p>string objectName: En este parametro se debe poner el mismo nombre que tiene el objeto en la escena del servidor V-REP. En este caso el nombre de la articulacion que se va a manejar (ART_1).</p> <p>Esta funcion devuelve un codigo numerico a una etiqueta que para manejar un optimo orden deberia tener el mismo nombre de el objeto que se esta manejando. En este caso particular, la junta ART_1.</p> <p>Finalmente este valor retornado le permitirá a las demas funciones modificar las características de las articulaciones que se encuentre en una escena del servidor V-REP desde la GUI de Matlab.</p>
12	simxSetJointForce()
Descripción	Establece el máximo torque que una junta puede ejercer.
Sinopsis general	simxSetJointForce(number clientID,number jointHandle,number force,number operationMode)
Sinopsis implementada	vrep.simxSetJointForce(clientID,ART_1,1.5,vrep.simx_opmode_oneshot)
Parámetros	<p>number jointHandle: Se le asigna el nombre de la articulacion a la cual se le va a asignar el torque.</p> <p>number force: Se le asigna la fuerza de torque en [N.m]. En este caso 1.5[N.m] que indica el maximo torque del motor AX-12+.</p>
13	simxSetJointTargetVelocity()
Descripción	Establece la velocidad intrínseca del objeto. En este caso representa la velocidad del motor AX-12+.
Sinopsis general	simxSetJointTargetVelocity(number_clientID,number_jointHandle,



	number_targetVelocity,number operationMode)
Sinopsis implementada	vrep.simxSetJointTargetVelocity(clientID,ART_1,0.05,vrep.simx_opmode_oneshot)
Parámetros	number_targetVelocity: Se establece la velocidad relacionada a la junta articular ART_1. En este caso 0.05.
14	simxSetObjectFloatParameter()
Descripción	Envía un valor de punto flotante a un objeto determinado. En este caso se usa esta función para asignar las constantes PID a cada junta articular, simulando el control PID que posee el motor AX-12.
Sinopsis general	simxSetObjectFloatParameter(number_clientID,number_objectHandle ,number_parameterID,number_parameterValue, number_operationMode)
Sinopsis implementada	vrep.simxSetObjectFloatParameter(clientID,ART_1,PID_P,Kp,vrep.simx_opmode_oneshot)
Parámetros	number_parameterID: A cada constante P,I Ó D, V-REP le tiene asignado un numero de identificación, el cual debera colocarse en este parametro. number_parameterValue: Se asigna el valor de la constante P,I Ó D
Sinopsis de bloque de constantes PID	Asi queda el codigo para asignar las constantes PID, para la articulación 1 (ART_1): PID_P =2002 PID_I =2003 PID_D =2004



	$K_p = 0.1$ $K_i = 0$ $K_d = 0$ vrep.simxSetObjectFloatParameter(clientID,ART_1,PID_P,Kp,vrep.simx_opmode_oneshot) vrep.simxSetObjectFloatParameter(clientID,ART_1,PID_I,Ki,vrep.simx_opmode_oneshot) vrep.simxSetObjectFloatParameter(clientID,ART_1,PID_D,Kd,vrep.simx_opmode_oneshot)
15	simxSetJointTargetPosition()
Descripción	Se Establece la posición articular deseada.
Sinopsis general	simxSetJointTargetPosition(number_clientID, number_jointHandle,number_targetPosition, number_operationMode)
Sinopsis implementada	vrep.simxSetJointTargetPosition(clientID,ART_1,radianes,vrep.simx_opmode_oneshot)
Parámetros	number_targetPosition: La posición de la junta articular se debe asignar en radianes.

Tabla 1D. Bloque de funciones para la asignación de propiedades a las articulaciones.

La anterior tabla fue investigada mediante el Manual de usuario de V-rep¹², este documento se encuentra en la página oficial de COPELIA ROBOTICS.

¹² The user manual is included in the downloadable V-REP packages, Virtual Robot Experimentation Platform USER MANUAL, V-rep pro edu, Version 3.1.3



SIMULACIÓN DE CICLO DE MARCHA DEL ROBOT BÍPEDO BIOLOID EN UN ENTORNO VIRTUAL 3D



ANEXO D

Funciones de Matlab

También fue necesario recurrir a algunas funciones de Matlab con el fin de cargar archivos, convertir datos numéricos a caracteres y viceversa, generar retardos y desplegar mensajes en la ventana de comandos de Matlab. Tabla 2.

1	[FileName Path]=uigetfile({'*.tst'}, '')
Descripción	Esta función nos permite buscar un archivo con un formato especificado, y retorna su respectiva dirección y el nombre del archivo.
2	strcat(Path,FileName)
Descripción	Concatena strings.
3	winopen()
Descripción	Recibe una dirección de destino y procede a abrir el archivo.
4	image(a)
Descripción	Muestra la matriz (a) como una imagen.
5	disp()
Descripción	Despliega una cadena de caracteres o un valor numérico en la ventana de comandos de Matlab.
6	pause()
Descripción	Genera un retardo en segundos.
7	str2num()
Descripción	Convierte una cadena de caracteres a un valor numérico.









Tabla 2. Principales Funciones De Matlab



ANEXO E

Herramientas para Diseño de GUI Matlab

En el presente cuadro se da una descripción general de las herramientas usadas para el diseño de la interfaz de usuario. Tabla 3.

COMPONENTE	ÍCONO	DESCRIPCIÓN
Push Button		Invoca un evento inmediatamente y ejecuta lo que se haya definido dentro de la función name_Callback
Slider		El usuario puede deslizar una barra de desplazamiento que está relacionada a un rango de valores.
Check Box		Indicador de si ocurre o no un evento.
Edit Text		Muestra y lee una cadena de caracteres en un cuadro de texto. Útil para leer datos y desplegar mensajes.
Static Tex		Muestra una cadena de caracteres en un cuadro de texto.
Pop-Up Menu		Despliega una lista de opciones al usuario. Generalmente va asociada a un Switch case en el main.
Axes		Permite plotear gráficos y mostrar imágenes.
Panel		Los paneles sirven para organizar los componentes de la GUI en grupos y hacer más comprensible la interfaz al usuario.




Menu Editor		Crea una barra con pestañas que pueden desplegar una lista de opciones que cumplen la misma función que un Push Button.
-------------	---	---

Tabla 3. Herramientas de GUI Matlab



ANEXO F

Funciones del “main principal” Asociadas a la GUI de Matlab

En la tabla presentada a continuación se encuentran las funciones generadas al crear la interfaz y otras que se implementaron con el fin de indicar, manipular, ingresar o extraer datos de la interfaz para ser procesados dentro del main. Tabla 4.

1	function name_OpeningFcn(hObject, eventdata, handles, varargin)
Descripción	Esta subrutina aparece dentro del .m asociado a la interface creada. Dentro de esta función se recomienda realizar la inicialización de variables.
2	function name_Callback(hObject, eventdata, handles)
Descripción	Esta función se crea automáticamente cuando hemos puesto algún elemento en la interfaz. Esta subrutina se activa cuando interactuamos en la interfaz con el elemento que está asociado a ella, ya sea un botón, un cuadro de texto entre otros. Y hará lo que hayamos programado previamente dentro de ella.
3	get(hObject,'string') get(hObject,'value')
Descripción	Estas funciones obtienen el valor o una cadena de caracteres respectivamente entregados por el name_Callback cuando el usuario interactúa con el elemento asociado a esta función desde la interface.
4	handles.output = hObject
Descripción	Todos los valores de las propiedades de los elementos (color, valor, posición, string, entre otros) y los valores de las variables transitorias del programa se almacenan en una estructura, los cuales son



10	<code>set(handles.name,'Enable','on')</code> <code>set(handles.name,'Enable','off')</code>
Descripción	Esta sentencia le permite o no al usuario interactuar con un elemento de la interfaz.
11	<code>axes(handles.name_axes)</code>
Descripción	Con esta función especificamos el Axes en el que se mostrará alguna imagen, o se ploteará algún gráfico en la interfaz.

Tabla 4. Principales Funciones De Gui Matlab

La anterior tabla fue investigada mediante el documento Creating Graphical User Interfaces, citado de la página oficial de MATLAB



ANEXO G

Manual de Usuario de la Aplicación Completa

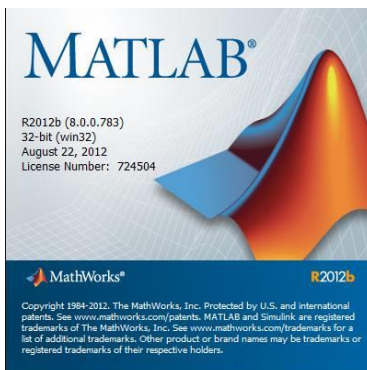
La interfaz diseñada permite manejar diversas propiedades del entorno de simulación V-Rep. Esto hace posible controlar la escena cargada que contiene el modelo virtual del robot bípedo Bioloid Premium Kit. La interfaz cuenta con dos Modos. El modo manual y el modo automático.

El modo manual existe con el fin de familiarizarse con las articulaciones del robot y el modo automático permite ingresar trayectorias articulares sinusoidales al bípedo virtual, con el fin de experimentar diversos ciclos de marcha. Dentro de la carpeta de la interfaz de usuario también se encuentra un script donde se podrán implementar trayectorias más avanzadas como son las polinomiales.

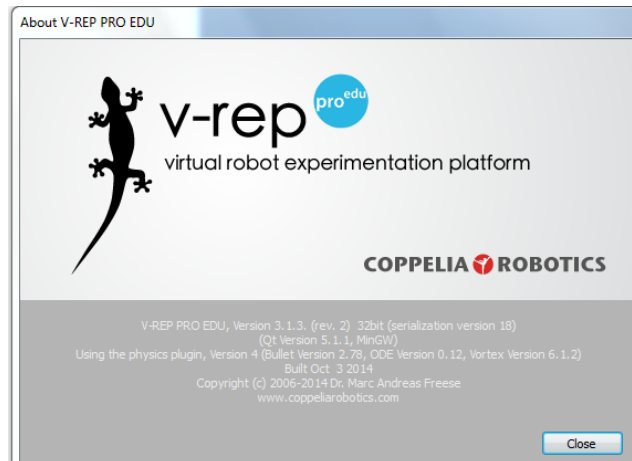
Instrucciones para habilitar la interfaz de usuario

A continuación se dan unas instrucciones que se deben seguir para poder configurar la comunicación REMOTE API CONTINUA tipo cliente servidor entre Matlab y V-Rep. Estas instrucciones solo necesitan realizarse una sola vez.

1. Instalar el software de cálculo Matlab de 32 bits.



2. Instalar V-REP PRO EDU compatible con el sistema operativo de Windows 7 de 32 bits (ver anexo A).



3. Instalar el compilador de lenguaje C, Windows SDK for Windows 7 and .NET Framework 4. <http://www.microsoft.com/en-us/download/details.aspx?id=8279>.

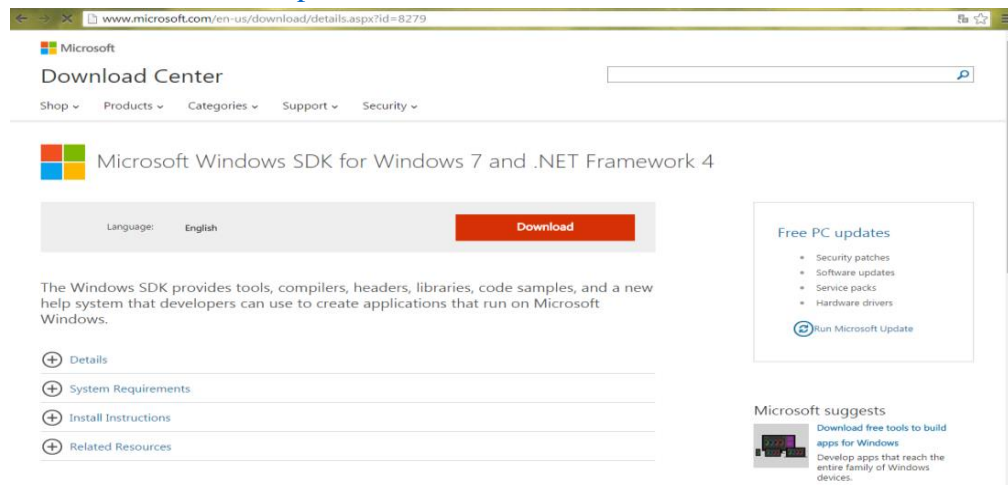


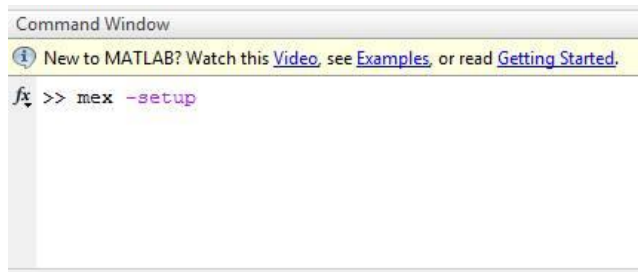
Figura 11. Microsoft Windows SDK for Windows 7 and .NET Framework 4.

4. Instalar el complemento Microsoft Visual C++ 2010 Service Pack 1 Compiler Update for the Windows SDK 7.1. <http://www.microsoft.com/en-us/download/details.aspx?id=4422>.



Figura 12. Microsoft Visual C++ 2010 Service Pack 1 Compiler Update for the Windows SDK 7.1.

5. Debe copiar dentro de la misma carpeta de la interfaz de usuario unas librerías y archivos de cabecera que se encuentran en la carpeta que contiene el software V-REP3.
 - a. remApi.m, remoteApiProto.m C:\ProgramFiles\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\matlab\matlab.
 - b. remoteApi.dll. C:\ProgramFiles\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\32Bit.
 - c. extApi.h. C:\Program Files\V-REP3.
6. Posteriormente en la ventana de comandos de Matlab digite **mex -setup** y presione la tecla **ENTER**.



En seguida le aparecerá un párrafo de bienvenida y una pregunta al final.



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

>> mex -setup

Welcome to mex -setup. This utility will help you set up
a default compiler. For a list of supported compilers, see
http://www.mathworks.com/support/compilers/R2012b/win32.html

Please choose your compiler for building MEX-files:

A Would you like mex to locate installed compilers [y]/n? |
```

Responda yes digitando la letra (y) presionando **ENTER**. Finalmente aparece una lista de compiladores. Seleccione el compilador Microsoft Software Development Kit (SDK) 7.1 digitando el número (2) y presionando **ENTER**.

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

>> mex -setup

Welcome to mex -setup. This utility will help you set up
a default compiler. For a list of supported compilers, see
http://www.mathworks.com/support/compilers/R2012b/win32.html

Please choose your compiler for building MEX-files:

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:
[1] Lcc-win32 C 2.4.1 in E:\MATLAB\1\sys\lcc
[2] Microsoft Software Development Kit (SDK) 7.1 in C:\Program Files\Microsoft Visual Studio 10.0
[0] None

Compiler: 2
```

Se realizará una última pregunta a la cual deberá contestar yes digitando la letra (y) y presionando **ENTER**.

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:
[1] Lcc-win32 C 2.4.1 in E:\MATLAB-1\sys\lcc
[2] Microsoft Software Development Kit (SDK) 7.1 in C:\Program Files\Microsoft Visual Studio 10.0
[0] None

Compiler: 2

Please verify your choices:

Compiler: Microsoft Software Development Kit (SDK) 7.1
Location: C:\Program Files\Microsoft Visual Studio 10.0

A Are these correct [y]/n? y
```

De esta manera Matlab ya trabajará con el compilador de lenguaje C Microsoft Software Development Kit (SDK) 7.1.



7. Servicio de servidor API remota continua:

Con este método las funciones de la API remota implementadas en el cliente Matlab siempre se ejecutarán en el lado del servidor V-Rep, incluso si la simulación no se está ejecutando. Para lo cual debe modificar el archivo de texto remoteApiConnections.txt que se encuentra en C:\Program Files\V-REP3\V-REP_PRO_EDU asignando el puerto de enlace (**19999**) y guarde los cambios.

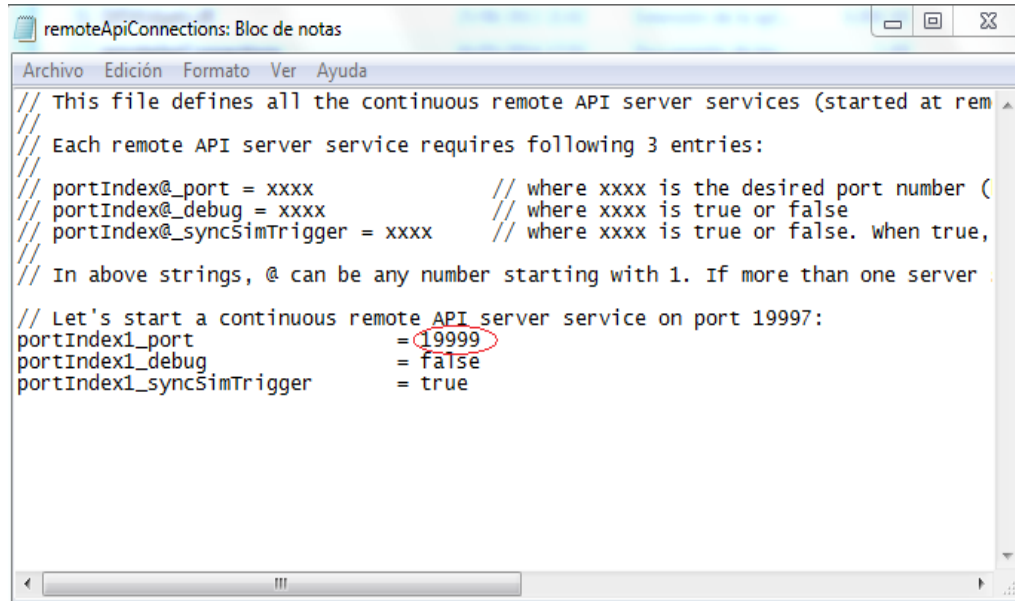
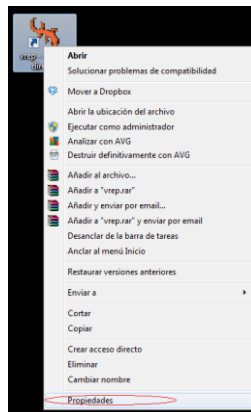
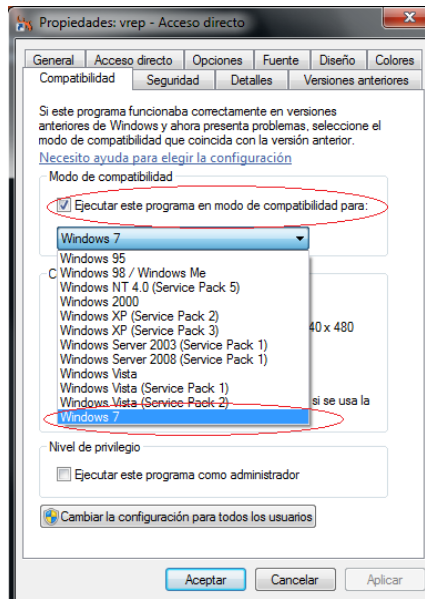


Figura 13. Archivo de configuración y conexión de la aplicación.

8. Vaya al icono de V-REP y de clic derecho y seleccione propiedades.



En la ventana que aparece, diríjase a la pestaña de compatibilidad y seleccione Ejecutar este programa en modo de compatibilidad, y escoja la versión de Windows que maneja su ordenador.



MODO MANUAL

El Modo Manual le permite al usuario ingresar las 18 posiciones articulares que maneja el robot, con el fin de conocer el sentido de rotación, y familiarizarse con sus movimientos.

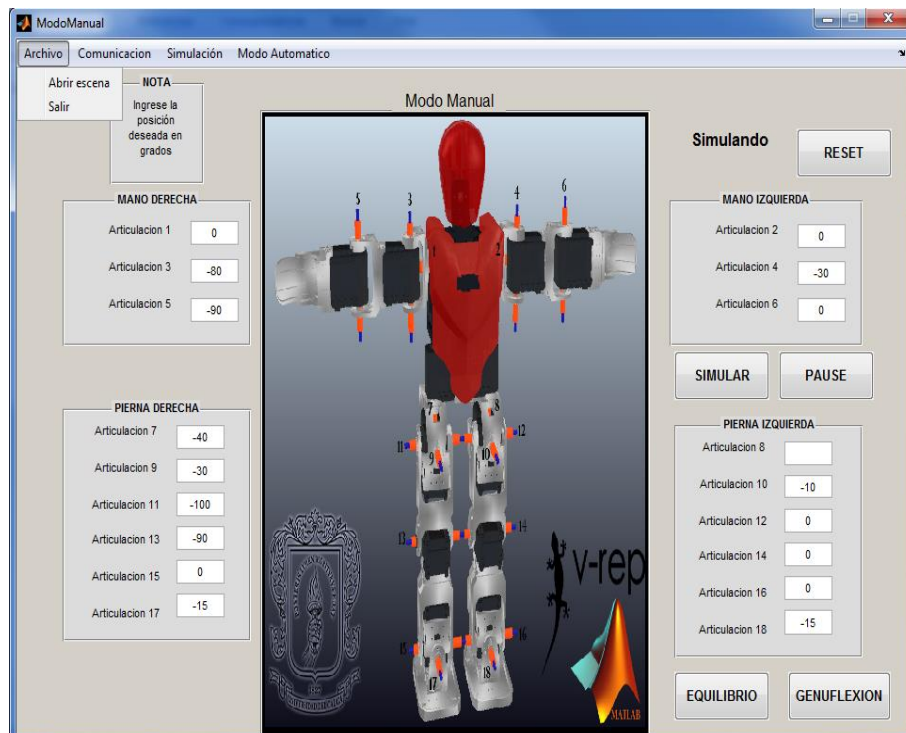


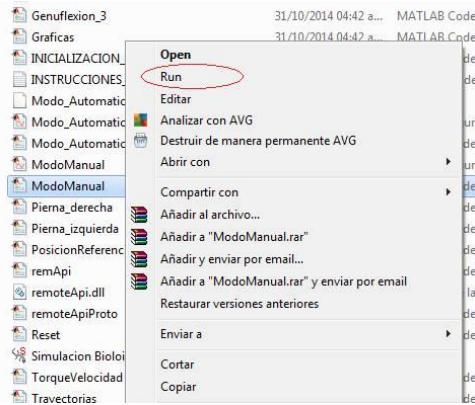
Figura 14. Ventana modo manual.



Instrucciones de inicio

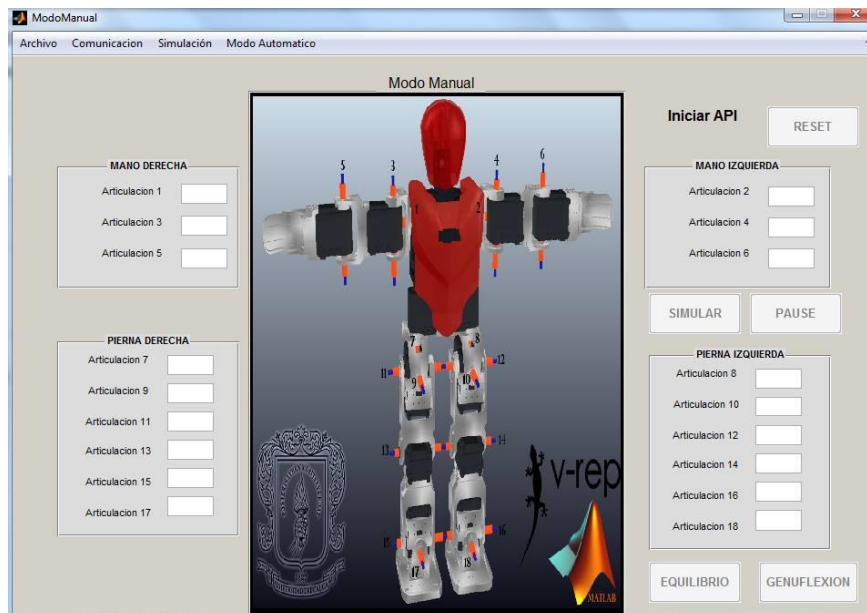
La explicación de los elementos que conforman el modo manual se encuentra al final de estas instrucciones de inicio.

- 1) Dentro de la carpeta que contiene todos los archivos de la interfaz de usuario, haga click derecho sobre el archivo **ModoManual.m** y elija la opción **Run**.



De esta manera se abrirá la interfaz gráfica llamada Modo Manual.

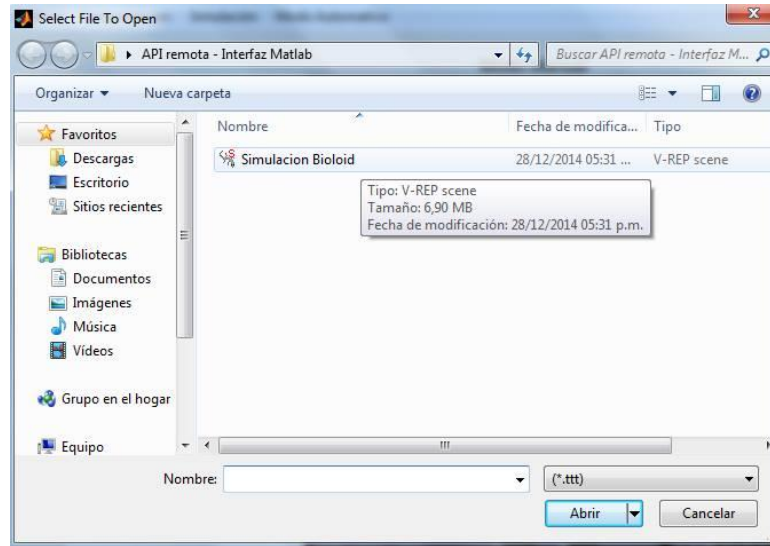
Inicialmente se encontraran bloqueados todos los botones.



Para desbloquearlos debe:



- 2) Ir a la pestaña de **Archivo**, elija la opción **Abrir escena**. Esto lo llevara a una ventana donde deberá dar doble click sobre el archivo **Simulación Bioloid**.



De esta forma se abrirá la escena de simulación en V-Rep donde se encuentra el modelo CAD de Bioloid.

- 3) Vaya a la pestaña de **Comunicación** y haga click en la opción **Iniciar API**. Si se establece el enlace, aparecerá el mensaje de **CONECTADO** habilitando todas las características del modo manual al usuario. De lo contrario aparecerá por un instante un mensaje de **Falló comunicación** que generalmente ocurre cuando no se ha realizado algún paso anterior.



- 4) Para pasar del modo manual al modo automático de click en la pestaña final de la interfaz. Cada vez que cambie de un modo al otro debe desbloquear los botones realizando el paso 3.



Barra de Pestañas: En la parte superior izquierda se encuentra una barra con 4 pestañas.



Pestaña de archivo

- Abrir escena: Permite buscar y cargar la escena de V-Rep que se va a manejar.
- Salir: Cierra la interfaz de usuario.

Pestaña Comunicación

- Iniciar API: Realiza la comunicación entre Matlab y el simulador V-Rep.

Pestaña Simulación

- Iniciar simulación: Corre la simulación en V-Rep.
- Pausar simulación: Pausa la simulación en V-Rep.

Pestañas Modo Automático/Manual

Cambia al modo Automático o Manual respectivamente.

Botón EQUILIBRIO:

Da las órdenes para que el bípedo realice equilibrio en una sola pierna con el ánimo de que el usuario pueda observar su comportamiento y demostrar de forma práctica que el entorno de simulación V-Rep se basa en la dinámica real de los sistemas.



Botón GENUFLEXIÓN:

Deja al bípedo en posición de genuflexión que es la postura con la que el robot real inicia su ciclo de marcha.

Botón RESET:

Detiene la simulación y asigna los valores iniciales a las articulaciones.

Botón SIMULAR:

Corre la simulación en V-Rep.

Botón PAUSE:

Pausa la simulación. Lo cual es útil cuando se desea ingresar más de una posición articular a la vez.

Ingreso De Parámetros:

El ingreso de parámetros se realiza mediante los cuadros de edición de texto donde el usuario puede digitar las 18 posiciones angulares en grados.



MODO AUTOMÁTICO

El objetivo principal del modo automático es utilizar el método de trayectorias sinusoidales acopladas (ver archivos de Matlab de la aplicación), para encontrar las 18 trayectorias articulares y lograr que el bípedo realice una marcha estable.

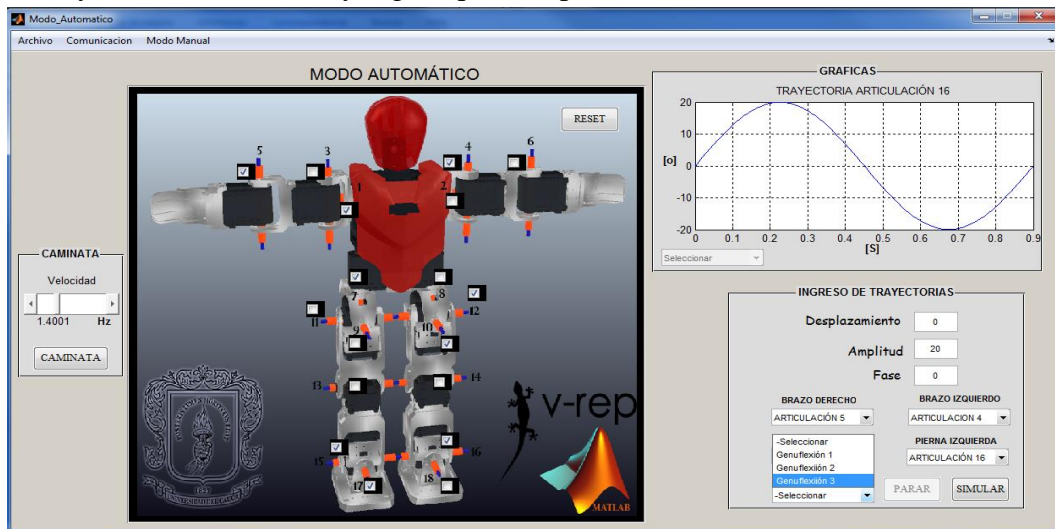


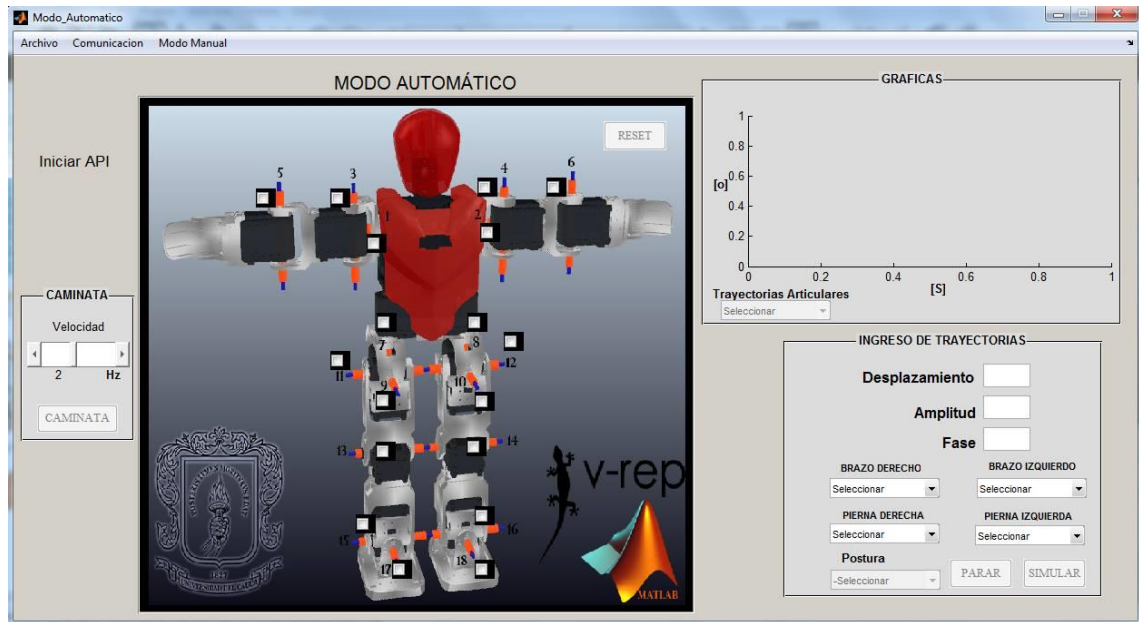
Figura 15. Ventana modo automático.

Instrucciones de inicio

Si cargó el modo automático mediante el paso 4, indicado en el modo manual, solo debe realizar el paso 3. La explicación de los elementos que conforman el modo automático se encuentra al final de estas instrucciones de inicio.

- 1) Dentro de la carpeta que contiene todos los archivos de la interfaz de usuario, haga clic derecho sobre el archivo ModoAutomatico.m y elija la opción **Run**. De esta manera se abrirá la interfaz gráfica llamada Modo Manual.

Inicialmente se encontraran bloqueados todos los botones.



Para desbloquearlos debe:

- 2) Ir a la pestaña de **Archivo**, elija la opción **Abrir escena** y de doble click sobre el archivo **Simulación Bioloid**. De esta forma se abrirá la escena de simulación en V-Rep donde se encuentra el modelo CAD de Bioloid.
- 3) Vaya a la pestaña de **Comunicación** y haga click en la opción **Iniciar API**. Si se establece el enlace, aparecerá el mensaje de **CONECTADO** en la interfaz habilitando todas las características del modo automático al usuario. De lo contrario aparecerá por un instante un mensaje de **Falló comunicación** que generalmente ocurre cuando no se ha realizado algún paso anterior.
- 4) Para pasar del modo automático al modo manual de click en la pestaña final de la interfaz. Cada vez que cambie de un modo al otro debe desbloquear los botones realizando el paso 3.

Barra de Pestañas: En la parte superior izquierda se encuentra una barra con 3 pestañas explicadas previamente en el modo manual. El modo automático cuenta con dos paneles.

Panel de caminata

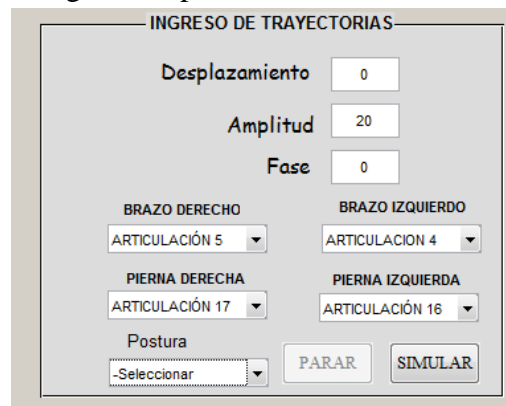
Las trayectorias realizadas mediante el método de trayectorias sinusoidales acopladas están implementadas dentro de la subrutina asociada al Botón **CAMINATA**.



- **Botón CAMINATA:** Al hacer clic sobre este botón el bípedo virtual realiza un ciclo de marcha a partir de la posición de genuflexión.
- **Barra deslizante de Velocidad:** El usuario puede modificar la frecuencia de la señal variando así, la velocidad marcha del robot.

Panel de INGRESO DE TRAYECTORIAS

Este panel permite aplicar el método de trayectorias sinusoidales acopladas para encontrar las 18 trayectorias articulares y lograr que el bípedo realice una marcha estable. Para ello siga los siguientes pasos:

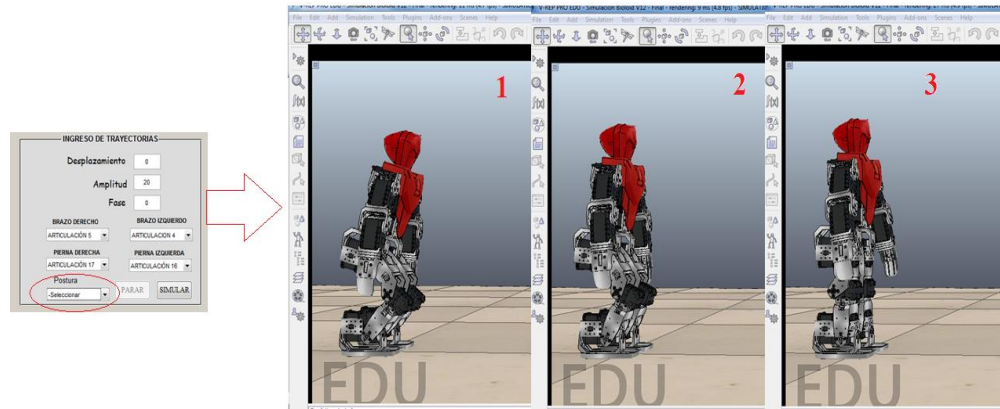


1. Barra de menú Postura:

El usuario puede escoger 3 diferentes posturas de genuflexión e implementar a partir de la postura seleccionada las trayectorias sinusoidales, utilizando el método de trayectorias sinusoidales acopladas.



SIMULACIÓN DE CICLO DE MARCHA DEL ROBOT BÍPEDO BIOLOID EN UN ENTORNO VIRTUAL 3D

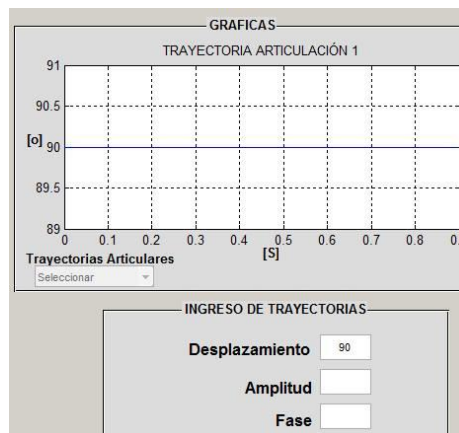


El ingreso de parámetros debe realizarse en grados.



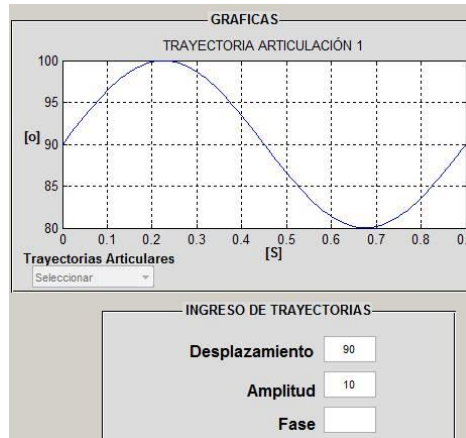
2. Ingreso de parámetros de Desplazamiento:

Se establece la posición angular a partir de la cual oscilará la articulación según la amplitud definida.





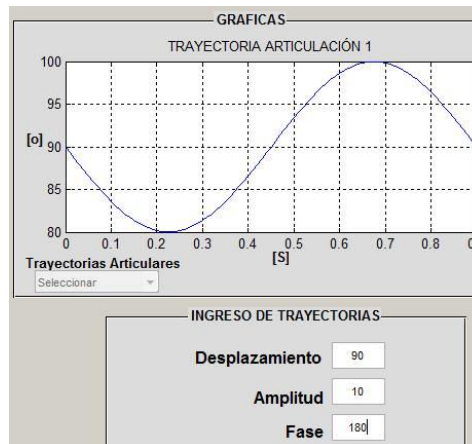
3. Ingreso de parámetros de Amplitud:



Se define el desplazamiento angular máximo que realizará la articulación.

4. Ingreso de parámetros de Fase:

Realiza un desfase de la señal sinusoidal.



5. Barra de menú de las articulaciones:

Permite ingresar la trayectoria sinusoidal implementada a cada una de las 18 articulaciones del robot mediante las barras de menú de brazo derecho, brazo izquierdo, pierna derecha, pierna izquierda.





- 6. Botón SIMULAR:**
Corre la simulación en V-Rep.
- 7. Botón PARAR:**
Para la simulación sin reiniciar variables.
- 8. Botón RESET:**
Para la simulación y reinicia las variables.

Panel de GRÁFICAS

En este panel se pueden observar las trayectorias implementadas, tanto en el módulo de caminata, como en el panel de ingreso de trayectorias.



- 9. Barra de menú Trayectorias Articulares:**
El usuario puede escoger la trayectoria que desee visualizar.