



Departamento de Ingeniería de Sistemas y Automática

Universitat Politècnica de València



# Simulación en robótica con *VirtualRobot*. Introducción a VRS

**Martin Mellado Arteche**

---



Escuela Técnica  
Superior de Ingeniería  
Informática

**Septiembre, 2012**

Este documento está regulado por la licencia *Creative Commons*



**Contenido**

<b>INTRODUCCIÓN AL SISTEMA DE SIMULACIÓN GRÁFICA DE ROBOTS VRS</b>	<b>3</b>
<b>1. INTRODUCCIÓN</b>	<b>3</b>
<b>2. VIRTUALROBOT SIMULATOR (VRS)</b>	<b>3</b>
<b>3. INTERFAZ GRÁFICO DE USUARIO DE VRS</b>	<b>4</b>
<b>4. VRS LOADER</b>	<b>9</b>
<b>5. VRS TEACHPENDANT</b>	<b>12</b>
<b>6. VRS TRACECONTROL</b>	<b>19</b>
<b>7. VRS IOCONNECTION</b>	<b>20</b>
<b>8. VRS PARTHANDLING</b>	<b>21</b>
<b>9. VRS SPEEDCONTROL</b>	<b>23</b>
<b>10. APLICACIONES DE DEMOSTRACIÓN</b>	<b>24</b>
<b>A. CONFIGURACIÓN DE VRS</b>	<b>25</b>

# Introducción al sistema de simulación gráfica de robots *VRS*

## 1. Introducción

El uso de la programación textual de robots de forma *off-line* corre el riesgo de que los programas desarrollados se prueban en el robot sin conocer a ciencia cierta si van a funcionar correctamente. De ahí surge la necesidad de disponer de sistemas de simulación gráfica que permitan conocer previamente el resultado de la ejecución de un programa para poder depurarlo. Si bien esta práctica reducirá enormemente la posibilidad de cometer errores en los programas, así como facilitará el desarrollo de los mismos, hay que tener claro que no garantiza que lo que se realiza en un mundo virtual en un ordenador se corresponda fielmente con lo que ejecute el robot en el mundo real, debido a las divergencias que hay entre el modelo del robot y éste mismo, así como de los elementos de su entorno.

El objetivo de este guión es conocer un sistema de simulación y programación *off-line* de robots para posteriormente, en guiones sucesivos, aprender a programarlo para poder desarrollar aplicaciones industriales típicas. Para ello se utilizará el programa *VirtualRobot Simulator (VRS)* que permite su programación en C/C++. En este guión se muestra un *tutorial* para el aprendizaje de dicho programa con una serie de tareas a realizar.

## 2. VirtualRobot Simulator (VRS)

*VirtualRobot Simulator (VRS)* es un simulador multipropósito diseñado para la simulación de robots, 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). Su objetivo está orientado tanto para aplicaciones industriales como para usos de investigación y propósitos educativos.

*VRS* es básicamente una aplicación de modelado gráfico con prestaciones específicas para la simulación de robots que dispone de la posibilidad de generar programas de aplicaciones externas (programas ejecutables Windows) para su control. De hecho, muchas de las opciones de su interfaz están desarrolladas como aplicaciones externas llamadas *VRS Tools*. Además de las posibilidades de simulación directa con el programa y las *VRS Tools*, permite la programación de las acciones de robots virtuales con programas desarrollados en C y/o C++. *VRS* Permite añadir un entorno al robot con piezas que el robot puede manipular actuando con una pinza y que se pueden tomar como referencia de movimientos de los robots.

*VRS* se desarrolló inicialmente con el objetivo de ser una herramienta de monitorización de robots controlados por el sistema de control Generis, desarrollado por el Joint Research Centre (JRC), por lo que permite conectarse con dicho sistema y representar gráficamente la situación en que se encuentra el robot controlado por Generis.

### 2.1. Instalación de *VRS*

El instalador del software **VirtualRobot** se encuentra disponible en la página web <http://robotica.isa.upv.es/virtualrobot>. Existen opciones de instalación para configurarla.

Para arrancar el programa *VRS*, en la versión por defecto del instalador se crea un acceso directo al mismo en **Inicio>>Programas>>Virtual Robot>>VRS**.

Se recomienda que antes de cerrar el programa VRS se cierren todas las aplicaciones externas que se encuentren en ejecución, incluidas la *VRS Tools*. En otro caso, algunas de estas aplicaciones pueden producir error de *Timeout*<sup>1</sup> en comunicación con *VRS*.

## 2.2. Estructura de directorios creados para trabajar con VRS

En la instalación del software **VirtualRobot**, se habrá especificado el directorio de instalación del programa ejecutable de *VRS*. Denominando a este directorio *VRPATH*, por defecto el instalador lo toma como "C:\Archivos de programa\Virtual Robot\VRS". En el directorio *VRPATH* está el programa ejecutable de *VRS*. En carpetas se encuentran el resto de componentes del software *VirtualRobot*, destacando las siguientes:

- **Applications.** Donde se encuentran las diferentes aplicaciones que se ejecutan sobre *VRS*, como por ejemplo, *VRS Tools*, *VRM Tools*, *Demos*, etc guardadas en diferentes carpetas. En la carpeta *Applications\Users* se deben guardar las aplicaciones desarrolladas por el usuario del programa.
- **Models.** Donde se encuentran los modelos de robots, crafts, entornos y demás que se pueden utilizar en *VRS* guardados en diferentes carpetas. En la carpeta *Models\Users* (o cualquier subcarpeta con nombre *Users*) se deben guardar todos los modelos desarrollados por el usuario del programa.
- **SourceCode.** Donde se encuentran los ficheros de interface y definición de las librerías para desarrollar nuevas aplicaciones haciendo uso de ellas. En la carpeta *SourceCode\Users* se debe guardar los proyectos de las aplicaciones con los programas fuentes desarrollados por el usuario del programa.

**Nota:** Para evitar problemas en ordenadores compartidos, se recomienda copiar las carpetas *Models* y *SourceCode* a un directorio propio del usuario, donde se deben salvar sus ficheros. Siguiendo esta filosofía (guardar las aplicaciones, modelos y proyectos fuente en tus carpetas) para *transportar* el desarrollo de un usuario bastará con copiar el contenido de estas carpetas y copiarlo a cualquier otro ordenador en el mismo lugar.

## 3. Interfaz Gráfico de Usuario de VRS


### 3.1. Carga de robots

Al arrancar el programa, tras una ventana de presentación, se muestra la ventana de la aplicación vacía y la ventana *VRS Tool* denominada ***VRS Loader***<sup>2</sup>. Esta aplicación servirá para cargar robots. *VRS* es multi-robot, es decir, admite la simulación de múltiples robots simultáneamente.

Los robots, en *VRS*, vienen descritos mediante dos ficheros del mismo nombre. En el primero, con extensión *RKF* (Robot Kinematics File), se indican algunos parámetros generales así como una descripción de su cinemática. Este es el fichero que se debe cargar. El segundo fichero, con extensión *RGF* (Robot Geometric File), contiene la descripción geométrica del robot y su carga se realiza de forma automática junto con el fichero de la cinemática. La descripción de los ficheros está especificada en el documento *VRFFD*.

#### **Tarea 3.1:**

Carga un robot en *VRS* mediante los siguientes pasos:

- Selecciona el botón  de ***VRS Loader***
- Selecciona en la carpeta ***Models\Tutorial*** el fichero de robot ***CRS A465.rkf***.

El resultado final debe ser el mostrado en la Figura 1.

<sup>1</sup> En el apéndice A se muestra como ajustar los posibles valores de *Timeout*.

<sup>2</sup> En el apéndice A se muestra que las aplicaciones *VRS Tool* se pueden configurar para que se arranquen automáticamente o no, por lo que podría darse el caso de que no apareciera o no estuviera disponible, en cuyo caso habrá que reconfigurar *VRS* o lanzarlo tal como se explica posteriormente.

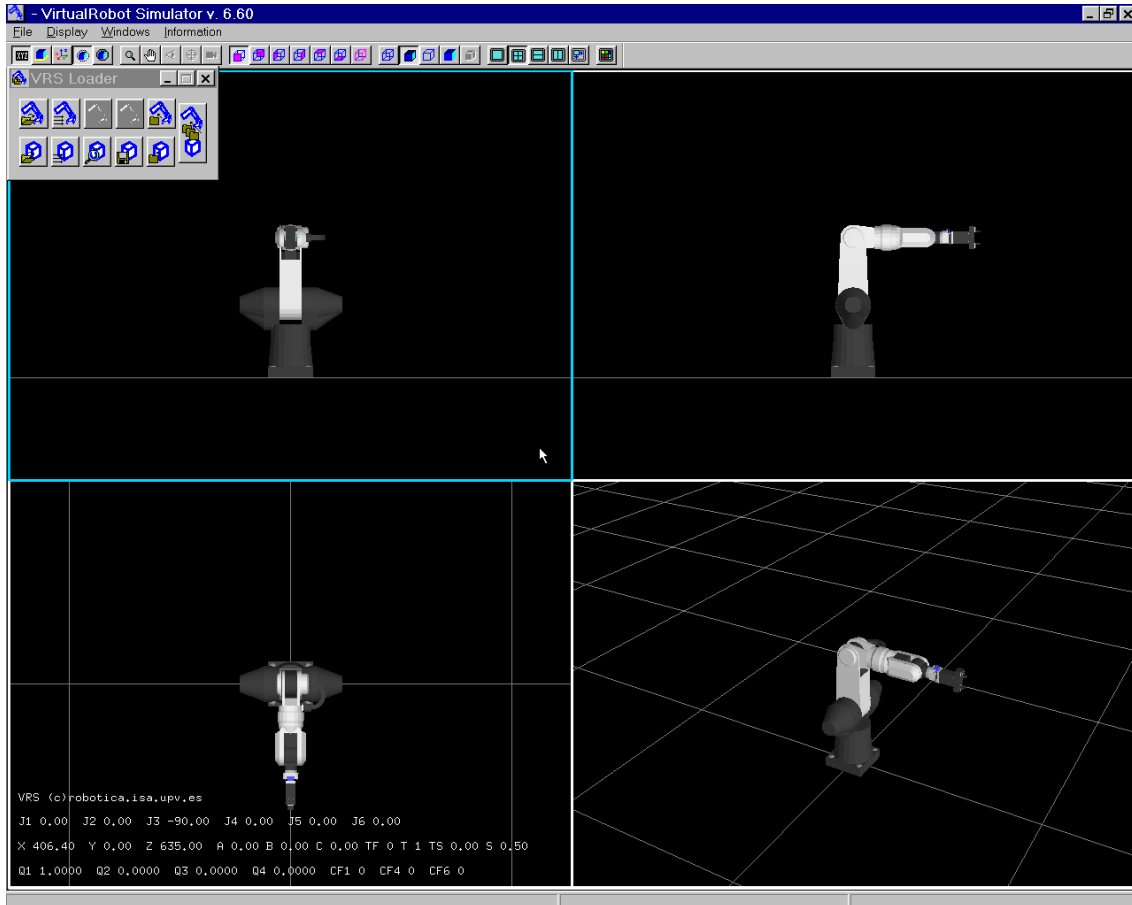


Figura 1. Robot CRS A465 en VRS

La aplicación dispone de:

- Una lista de menús en la parte superior donde se encuentran las opciones organizadas jerárquicamente.
- Una barra de herramientas, **Toolbar**<sup>3</sup>, en la parte superior para el control de la ventana gráfica.
- Una barra de estado, **Statusbar**, en la parte inferior para mostrar mensajes de estado y error así como el nombre del robot activo y su estado.
- Una ventana gráfica en la parte central que en principio dispone de cuatro marcos gráficos con planta, alzado, perfil y perspectiva.

En la ventana gráfica se muestra el robot cargado y en la parte inferior izquierda, en todo momento<sup>4</sup> aparecerá sobrescrito la información dinámica del robot activo:

- el valor actual de las variables de articulación, mostradas como J1 valor, J2 valor, ...
- la localización del sistema de herramienta activo (*tool frame*) respecto al sistema origen de programación (con orientación representada por ángulos de Euler tipo 2), mostrada como X valor, Y valor, Z valor, A valor, B valor, C valor
- el sistema de herramienta activo (TF valor), la herramienta activa (T valor), el estado de la herramienta activa (TS valor) y el estado de la velocidad del robot (S valor)
- según la configuración del programa puede aparecer la orientación en cuaterniones (Q1 valor, ... Q4 valor) y la descripción de la configuración del robot (CF1 valor, CF4 valor, CF6 valor).

<sup>3</sup> Tanto **Toolbar** como **Statusbar** se pueden ocultar o hacer visibles con las opciones correspondientes del menú **Windows**. Además, **Toolbar** puede arrastrarse y manejarse como una ventana Windows.

<sup>4</sup> Posteriormente se verá que esta visualización se puede activar/desactivar.

Nótese que, según cómo esté configurado *VRS*, en la pinza del robot pueden aparecer el sistema de coordenadas activo de la herramienta activa mediante tres ejes con los colores rojo, verde y azul (RGB), que representan los ejes X, Y y Z respectivamente de dicho sistema. Posteriormente se verá que esta visualización se puede activar/desactivar.

### 3.2. Opciones de visualización

*VRS* dispone de una barra de herramientas, denominada **Toolbar**, con controles para configurar la forma de mostrar la información gráfica. Estos controles tienen su equivalente en los menús **Display** o **Windows** (además algunas disponen de teclas función para su acceso). Las principales opciones son las siguientes:

#### Configuración de la ventana gráfica

La ventana gráfica se encuentra dividida en cuatro regiones iguales. Esto permite disponer de cuatro puntos de vista distintos de la misma escena, con lo que se logra una completa visualización de los distintos elementos presentes en la simulación. Debido a diversos aspectos de la simulación, es posible que surja la necesidad de configurar de otra forma la ventana gráfica. *VRS* ofrece cuatro formas de configuración: Una sola vista que ocupa toda la ventana gráfica, cuatro vistas que dividen la ventana en cuatro partes iguales, dos vistas que dividen la ventana gráfica horizontalmente en dos partes iguales y dos vistas que dividen la ventana gráfica verticalmente en dos partes iguales. Estas opciones son seleccionables desde el menú principal, dentro del menú **Windows**, y desde **Toolbar**:



Independientemente de la configuración elegida, siempre hay una de las vistas o regiones que se encuentra seleccionada. Esta selección se muestra resaltando en color blanco el marco que rodea cada vista, e indica sobre que región se van a efectuar los cambios de visualización que elijamos, como por ejemplo la opción **Windows>>Maximize**. **Maximize** permite ampliar la ventana seleccionada hasta ocupar la total extensión de la ventana gráfica. Desactivando esta opción se vuelve a la configuración que estuviera anteriormente.




#### Color del fondo

El color de fondo la ventana gráfica, que inicialmente aparece en negro, es modificable, aunque este es siempre igual para todas las vistas o regiones de la misma. El comando que permite esta configuración se encuentra en el menú **Windows**, así como en la barra de herramientas:



#### Visualización de la información dinámica

La información dinámica del robot activo se puede activar o desactivar mediante la opción **Display>>Dynamic Info** o el botón equivalente  del **Toolbar**.

#### Selección de la ventana gráfica













Varias de las opciones sobre ventanas gráficas se aplican sobre la ventana seleccionada. Para seleccionar una ventana hay que pulsar el botón derecho del ratón (para cancelar cualquier acción activa) y pulsar con el botón izquierdo sobre la ventana que se desea seleccionar.

### Selección y modificación de los puntos de vista

Al inicializar VRS, la ventana gráfica aparece dividida en cuatro regiones. Al cargar algún elemento en el simulador, se puede comprobar que el tipo de proyección y el punto de vista varía entre las diferentes vistas. Inicialmente la disposición es la siguiente:

Región	Proyección	Punto de Vista
Superior Izquierda	Ortogonal	Izquierda
Superior Derecha	Ortogonal	Frontal
Inferior Izquierda	Ortogonal	Superior
Inferior Derecha	Perspectiva	Libre <sup>5</sup>





La proyección y el punto de vista son modificables desde el menú **Display** y desde **Toolbar**. Los comandos de cambio de proyección y puntos de vista predefinidos se ven enriquecidos con opciones que permiten ajustar la visión según la necesidad del caso a tratar. Estas opciones (**Zoom**, **Scroll**, **Reference Point**, **Point of View...**), junto con los comandos de cambio de proyección se resumen en la siguiente tabla:

Menú Display	Toolbar	Acción
<b>Zoom</b>		Acerca el punto de vista al plano de proyección, desplazando el puntero de abajo a arriba mientras se mantiene pulsado el botón izquierdo del ratón
<b>Scroll</b>		Desplaza la imagen que se está visualizando. Desplazamientos del ratón, con el botón izquierdo pulsado, a derecha/izquierda o arriba/abajo, desplazan la imagen en ese sentido
<b>Point of View</b>		Modifica el punto de vista en la ventana seleccionada si ésta posee proyección en perspectiva. Mientras se mantiene pulsado el botón izquierdo del ratón, realizar desplazamientos laterales para modificar la longitud y verticales para la latitud
<b>Reference Point</b>		Modifica el punto de interés en la ventana seleccionada si ésta posee proyección en perspectiva. Con movimientos del ratón (con el botón izquierdo pulsado) se varía el punto de interés
<b>Display Parameters...</b>		Permite modificar los parámetros de vista de la ventana seleccionada, si ésta posee proyección en perspectiva. Al ejecutar esta opción se despliega un diálogo donde se pueden modificar el punto de interés, el punto de vista, la relación entre ellos en polares (tecta,fi,ro) y los parámetros de cámara (near plane, far plane y fov)
<b>Projections&gt;&gt;</b>		
<b>Orthogonal&gt;&gt;Front</b>		Muestra la vista frontal, con proyección ortogonal
<b>Orthogonal&gt;&gt;Back</b>		Muestra la vista trasera, con proyección ortogonal
<b>Orthogonal&gt;&gt;Left</b>		Muestra la vista lateral izquierda, con proyección ortogonal
<b>Orthogonal&gt;&gt;Right</b>		Muestra la vista lateral derecha, con proyección ortogonal
<b>Orthogonal&gt;&gt;Up</b>		Muestra la vista superior, con proyección ortogonal
<b>Orthogonal&gt;&gt;Down</b>		Muestra la vista inferior, con proyección ortogonal
<b>Perspective</b>		Muestra una proyección en perspectiva


VRS se inicializa por defecto con el modo de sombreado sólido. El simulador posee también la opción de visualizar los elementos cargados en modo alámbrico, con líneas ocultas o en modo render (únicamente para los robots o elementos del entorno que contienen información de renderizado en sus ficheros de definición). El modo de sombreado se puede modificar de forma independiente para cada ventana. Esta operación se realiza seleccionando la ventana a la que

<sup>5</sup> VRS puede cambiar automáticamente el punto de vista (desde donde mira la cámara) y el punto de referencia (punto al que mira la cámara) cada vez que se cargue un robot o entorno en función del tamaño del robot o entorno cargado.

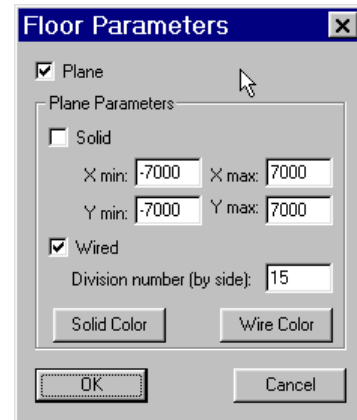
se desea cambiar el modo y ejecutando a continuación unas de las opciones, bien sean dentro de la barra de herramientas o bien en el menú **Windows**, que se muestran a continuación.

Menú Display	Toolbar	Modo de Sombreado
Wired		Alámbrico
Shaded		Sombreado
Hidden Lines		Líneas Ocultas
Rendered		Renderizado

### Visualización del suelo

Con la opción **Display>>Floor Parameters...** (o el botón  del **Toolbar**) se permite seleccionar el modo de visualización del suelo. El diálogo de control de esta opción es el que se muestra a la derecha, con las opciones siguientes:


- Con la opción **Plane** activa se mostrará un suelo, bien en modo sólido (**Solid** activo), alámbrico (**Wired** activo) o ambos (ambos activos).
- El tamaño del suelo se elige con los campos **X min**, **X max**, **Y min**, **Y max**. El número de divisiones alámbricas se elige con el campo **Divisions number (by side)**:
- Con los botones **Solid Color** y **Wire Color** se puede elegir el color del suelo y de las líneas alámbricas.



### Visualización de sistemas de coordenadas

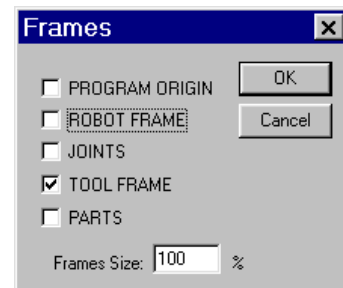
Un robot dispone de los siguientes sistemas de coordenadas:

- Sistema de Coordenadas del Robot (**ROBOT FRAME**).
- Sistema de Coordenadas Origen de Programación (**PROGRAM ORIGIN**).
- Un Sistema de Coordenadas propio de cada articulación (**JOINTS**).
- Los sistemas de coordenadas de herramienta (**TOOL FRAME**). En el fichero rkf de descripción de un robot pueden haberse definido varios, pero el cargado como activo por defecto siempre es el **Tool Frame 0**.

Por defecto, sólo se visualiza en pantalla el sistema de coordenadas de la herramienta. Para ver donde están otros sistemas de coordenadas, o incluso para no ver el sistema de la herramienta, se dispone del botón  del **Toolbar** (opción del menú **Display>>Frames...**) que abre un diálogo que permite seleccionar los sistemas de coordenadas a mostrar así como escalar su tamaño.

El diálogo se muestra a la derecha. Los sistemas de coordenadas son representados con los ejes X,Y,Z de color rojo, verde y azul (RGB). Además de las opciones enumeradas anteriormente, la opción **PARTS**, que se comentará posteriormente, posibilita ver los sistemas de coordenadas de operación de las piezas del entorno.









La opción **Frames Size: 100 %** permite ajustar el tamaño de los ejes en visualización.



### Tarea 3.2:

- Pulsa el botón derecho del ratón para cancelar cualquier opción activa.
- Pulsa el botón izquierdo en la vista de perspectiva (inferior derecha) para seleccionarla (marco debe quedar resaltado en azul).



- Maximiza la ventana de perspectiva (  ).
- Prueba la visualización en alámbrico (  ), líneas ocultas (  ) y sombreado (  ).
- Activa el zoom (  ) y pulsando el botón izquierdo mueve el ratón arriba y abajo.
- Activa el punto de vista (  ) y pulsando el botón izquierdo mueve el ratón libremente.
- Pulsa el botón de punto de referencia (  ) y pulsando el botón izquierdo mueve el ratón libremente.
- Pulsa el botón de parámetros de visualización (  ) e introduce como punto de referencia las coordenadas donde se encuentra el sistema de coordenadas de la herramienta (aparecen listadas en X,Y,Z en la parte inferior izquierda de la ventana gráfica).
- Prueba las opciones zoom y punto de vista para comprobar que la cámara está mirando a dicho sistema de coordenadas.
- Desactiva la visualización del suelo.
- Visualiza los sistemas de coordenadas del robot y la herramienta a tamaño 500% y deduce donde se encuentran localizados.
- Activa la visualización del suelo, probando como se muestra el plano, cambiando los colores y dejando al final la configuración de partida.

### 3.3. Ejecución de una aplicación externa en VRS

En *VRS* se pueden ejecutar aplicaciones externas generados en ficheros ejecutables que se comuniquen con *VRS* mediante la librería *VREAL* (ver documentación *VREAL*). Para ello se deben lanzar los programas con la opción **File>>Run Application** que muestra el diálogo de abrir fichero<sup>6</sup>. Estas aplicaciones se generan tal como se muestra en el guión titulado “Programación de aplicaciones para *VRS*”. Básicamente hay cuatro tipos de aplicaciones:

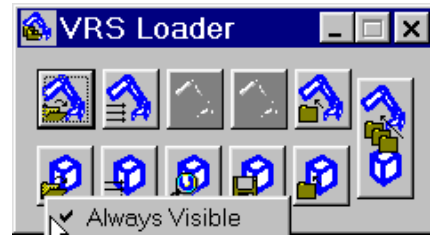
- Demostraciones (**Demos**). Son aplicaciones de demostración de procesos industriales realizados en *VRS*. Los ejecutables se encuentran en la carpeta **Applications\Demos**. Normalmente se ejecutarán desde la opción **File>>Demos**.
- Herramientas *VRS* (*VRS Tools*). Son aplicaciones útiles para el manejo de *VRS* (unidad de programación, grabación de vídeos, ...). Los ejecutables se encuentran en la carpeta **Applications\VRSTools**. Normalmente se ejecutarán desde la opción **File>>VRS Tools**.
- Herramientas *VRM* (*VRM Tools*). Son aplicaciones útiles para el modelado de objetos, piezas y entornos en *VRS*. Los ejecutables se encuentran en la carpeta **Applications\VRMTools**. Normalmente se ejecutarán desde la opción **File>>VRM Tools**.
- Aplicaciones de usuario. Son las desarrolladas por los usuarios, cuyos ejecutables es aconsejable que se copien en una carpeta propia del usuario dentro de la carpeta **Applications\Users**. Se deben ejecutar con la opción **File>>Run Application**.

## 4. *VRS Loader*

Para la gestión de las operaciones de carga de robots, de su localización en el espacio y su cierre o eliminación de *VRS*, así como para la realización de las mismas operaciones sobre el entorno, se dispone de la *VRS Tool* llamada ***VRS Loader***. ***VRS Loader*** es una aplicación externa que sirve para cargar, localizar y cerrar tanto robots como el entorno en que se integran. Como programa externo, su ejecutable ***VRSLoader.exe*** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **Applications\VRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Loader** y se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas. Si ***VRS Loader*** está en ejecución se volverá a abrir en caso de lanzarse a ejecución de nuevo.




<sup>6</sup> También se pueden arrancar directamente los ficheros ejecutables desde el sistema operativo.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *pop-up*, accesible al pulsar sobre la ventana (fuera de los botones) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*.



#### 4.1. Carga, localización y cierre de robots

Como ya se ha comentado, con **VRS Loader** se pueden cargar robots en el simulador, localizarlos en el espacio y cerrarlos. Los botones de cada una de estas opciones son los siguientes:

<b>VRS Loader</b>	<b>Acción</b>
	Permite seleccionar un fichero rkf y cargar el robot correspondiente
	Permite localizar un robot en el espacio
	Permite cerrar uno de los robots cargados en <i>VRS</i>

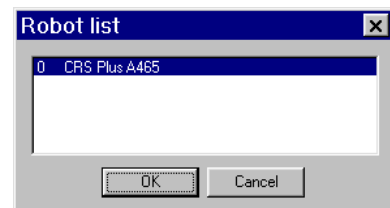
La carga de robots ya se ha explicado y usado al inicio de este guión. Los robots cargados se identifican por su nombre<sup>7</sup> y un identificador entero asignado internamente, siendo el primer robot el número 0 y los sucesivos serán 1,2,3... Cuando se carga un robot y su nombre ya está utilizado por uno de los robots cargados, *VRS* añade un número (1,2,3,...) al nombre del nuevo robot. Existe un límite en el número de robots cargados en *VRS*, por lo que si se intenta cargar otro robot se mostrará un mensaje de error.

En todo momento *VRS* mantiene un robot como el activo. Cuando se carga un robot, éste pasa a ser el robot activo. El nombre del robot activo se visualiza en la barra de estado y es de éste robot del que se muestra la información dinámica.

Los robots se cargan inicialmente con su sistema de coordenadas de robot coincidente con el del mundo de *VRS*. Sin embargo, esta localización se puede modificar haciendo uso del botón correspondiente de **VRS Loader**, que si hay algún robot cargado en *VRS*, mostrará los diálogos siguientes:

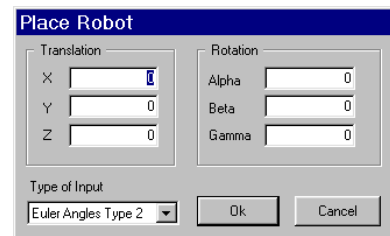
##### Robot List:

Muestra la lista de robots cargados en *VRS*. Se podrá cancelar el comando con el botón **Cancel** o seleccionar un robot de la lista y pulsar **OK** para localizarlo. Si sólo hay un robot cargado en *VRS* no se mostrará este diálogo, pasándose directamente al diálogo siguiente. El robot seleccionado se resaltará en pantalla.



##### Place Robot:

Solicita una localización mediante tres valores de traslación (X,Y,Z) y tres valores de rotación (ángulos alfa, beta y gamma). Se puede seleccionar el tipo de ángulos de Euler de la entrada entre los tipos 1, 2 y 3 con el campo **Type of Input**. Se realizará la orden de localización con **Ok** o se cancelará con **Cancel**.



Para cerrar los robots, es decir, eliminarlos de *VRS*, tras seleccionar el botón correspondiente de **VRS Loader**, se mostrará la lista de robots para seleccionar el robot a cerrar, incluso cuando sólo haya un robot cargado en *VRS*. Se debe seleccionar un robot y pulsar **OK** para cerrarlo o se puede cancelar el comando con **Cancel**.







<sup>7</sup> El nombre que se especifica en el campo Name de la sección GENERAL de los ficheros rkf y rgf (no el nombre de los ficheros).

Evidentemente, los botones de localización o cierre de un robot no tienen ningún efecto si no existe un robot cargado.

#### 4.2. Carga, localización y cierre del entorno

Los robots se sitúan en celdas de trabajo donde, además de otros robots (incluidos cualquier elemento articulado como cintas transportadoras, mesas giratorias, ...), puede haber diversos objetos que forman el entorno del robot. En *VRS* el entorno del robot está definido en un fichero ENF (ENvironment File, descrito en el documento VRFFD). Estos elementos cumplen dos funciones, actuar como entorno del robot, haciendo más realista la simulación y definir las piezas susceptibles de ser manipulados u operados por los robots. La inclusión del modelo del entorno en una simulación facilita las tareas de la programación *off-line*.

Como ya se ha comentado, con **VRS Loader** se puede cargar un entorno en el simulador, localizarlo en el espacio, buscarlo, salvarlo y cerrarlo. Los botones de cada una de estas opciones son los siguientes:

<b>VRS Loader</b>	<b>Acción</b>
	Permite seleccionar un fichero enf y cargar el entorno correspondiente
	Permite localizar el entorno en el espacio
	Permite cerrar el entorno cargado en <i>VRS</i>
	Permite buscar un entorno y cargarlo en <i>VRS</i>
	Permite salvar el entorno cargado en <i>VRS</i>
	Permite cerrar el entorno y todos los robots cargados en <i>VRS</i>

Para cargar un entorno se debe llamar a la opción correspondiente de **VRS Loader**. Sólo es posible tener un entorno cargado en *VRS*, por lo que se pedirá confirmación en caso de que ya exista uno. A partir de entonces, el entorno se pueden localizar y cerrar de la misma forma que se hace con los robots, teniendo en cuenta que no aparecerá ninguna lista de entornos al ser éste único. Para localizar el entorno, se dispone del diálogo **Place Environment** similar al de **Place Robot**. Para cerrar el entorno se pide confirmación. Para cualquiera de estas opciones, todos los elementos del entorno se resaltarán en pantalla. Evidentemente, los botones de localización o cierre del entorno no tienen ningún efecto si no existe un entorno cargado. El botón para buscar entornos abre una nueva aplicación (**VRM Visor**) que permite, moviéndose por los directorios con un explorador, visualizar los ficheros entornos para cargarlos en *VRS*. El botón para salvar entornos **Save Environment** salva el entorno en un fichero.

Finalmente, el último botón cierra, tras pedir confirmación, todos los robots y el entorno que estén cargados en *VRS*. Todos los elementos a borrar se resaltarán en pantalla.

##### **Tarea 4.1:**

- Carga el entorno **CRS Table.enf** de la carpeta **Models\Tutorial**.
- Desplaza el robot **CRS A465** 800 milímetros en Z para que quede encima de la mesa.
- Observa, activando la visualización de sistemas de coordenadas necesarias, que el sistema de coordenadas origen de programación sigue siendo el mismo con relación al robot, pero está situado en otro lugar respecto al sistema universal de *VRS*.
- Observa los sistemas de coordenadas de operación de las piezas (Parts) activando dicha opción.

##### **Tarea 4.2:**

- Cierra todo lo cargado en *VRS*.
- Carga el robot **CRS A465**.
- Vuelve a cargar otra vez el mismo robot.
- Localiza uno de los robots de forma que quede encarado pero sin solaparse con el otro.

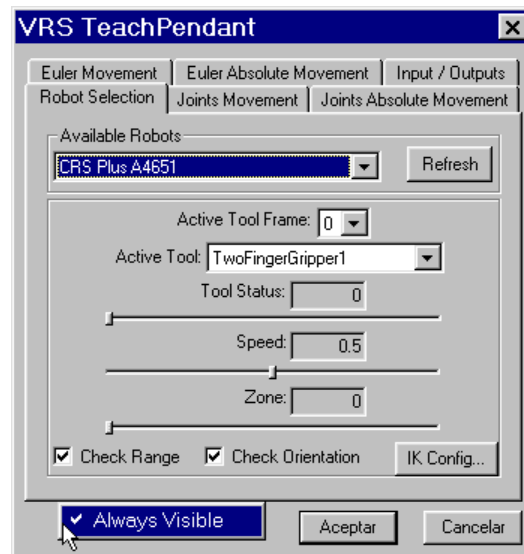
## 5. VRS TeachPendant

Para la gestión de un robot (su movimiento, control de su herramienta, gestión de sus entradas y salidas, ...) se dispone de la *VRS Tool* llamada **VRS TeachPendant**. **VRS TeachPendant** es una aplicación externa que sirve para gestionar robots. Como programa externo, su ejecutable **VRSTeachPendant.exe** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **Applications\VRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Teach Pendant**, si bien no se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas.

Nótese que esta aplicación no se puede ejecutar si no hay ningún robot cargado.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *pop-up*, accesible al pulsar sobre la ventana (fuera de la zona de pestañas) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*<sup>8</sup>.

La aplicación dispone de seis diálogos organizados por pestañas, un botón de aceptar y uno de cancelar. Para pasar de un diálogo a otro, hay que pulsar en la pestaña situada en la parte superior que contiene el nombre de la misma.



A diferencia del resto de *VRS Tools*, **VRS TeachPendant** se puede abrir varias veces para poder controlar diferentes robots simultáneamente.

### 5.1. Diálogo Robot Selection

El diálogo **Robot Selection** permite seleccionar uno de los robots disponibles y configurar su estado. Los campos disponibles se listan a continuación:

- **Available Robots:** muestra el robot que gestionará **VRS TeachPendant**, permitiendo seleccionar otro de la lista de robots cargados en *VRS*.
- **Refresh:** actualiza la lista de robots. Si con otra aplicación, por ejemplo **VRS Loader**, se cargan nuevos robots, resulta necesario actualizar en **VRS TeachPendant** la lista de robots disponibles con el botón **Refresh**. La actualización de la lista se realiza de forma automática al pasar de otro diálogo al diálogo **Robot Selection**, sin necesidad en este caso de pulsar el botón **Refresh**.
- **Active Tool Frame:** muestra el sistema de coordenadas de herramienta activo, permitiendo seleccionar otro de la lista de sistemas de coordenadas de herramientas definidos para el robot.
- **Active Tool:** muestra la herramienta activa, permitiendo seleccionar otra de la lista de herramientas definidas para el robot.
- **Tool Status:** muestra el valor del estado de la herramienta, permitiendo cambiarlo entre 0.0 y 1.0 con una barra de desplazamiento horizontal (0 a izquierda y 1 a derecha).
- **Speed:** muestra el valor de velocidad nominal de movimientos del robot, permitiendo cambiarlo entre 0.0 y 1.0 con una barra de desplazamiento horizontal (0 a izquierda y 1 a derecha).

<sup>8</sup> Dado el tamaño de esta ventana, puede ocultar diálogos que muestren otras aplicaciones, dando la sensación de que dichos diálogos no aparecen.

- **Zone:** muestra el valor del parámetro de zona (también llamado vuelo –fly- o de precisión) de movimientos del robot, permitiendo cambiarlo entre 0.0 y 1.0 con una barra de desplazamiento horizontal (0 a izquierda y 1 a derecha). Este campo no tendrá efecto en movimientos del robot con el **VRS Teach Pendant**, ya que sólo lo tiene cuando se realizan movimientos consecutivos.
- **Check Range:** permite activar o desactivar el chequeo de rangos del robot. Cuando está activo, el robot no podrá moverse fuera de los rangos de las articulaciones definidos en su fichero rkf. Cuando está desactivo, podrá moverse.
- **Check Orientation:** permite activar o desactivar el chequeo de orientación del robot. Cuando está activo, el robot no podrá moverse por localizaciones con orientación no alcanzable. Cuando está desactivo, podrá moverse por localizaciones con orientación no alcanzable haciendo caso solamente de la posición de la localización.
- **IK Config...:** Abre un nuevo diálogo para configurar la configuración deseada en la resolución de la cinemática inversa.

Los últimos cinco campos tienen su utilidad a la hora de realizar movimientos del robot con el **VRS TeachPendant**, ya que son opciones de configuración de movimientos, por lo que se irán comentando con más detalle en secciones posteriores.

### Control de herramienta

Los robots llevan herramientas acopladas para operar con ellas. Las herramientas pueden tener sus propios movimientos. En *VRS* se deben modelar los diferentes estadios con que se quieran representar el estado<sup>9</sup> de la herramienta en el fichero rgf. Posteriormente, se podrá controlar el estado de la herramienta mediante un valor numérico de 0.0 a 1.0, de forma que se interpole entre los posibles estadios de la herramienta. Es decir, si la herramienta se ha modelado con dos estadios, cuando el estado de la herramienta sea menor que 0.5 se representará con el primer estadio, mientras que cuando sea igual o mayor que 0.5, se representará con el segundo estadio. La misma filosofía se aplica cuando hay más estadios modelados.

Un robot puede tener más de una herramienta, acopladas a un adaptador. Para seleccionar la herramienta activa se dispone del campo **Active Tool** que permite seleccionar una de las herramientas disponibles (la opción no está disponible si el robot no tiene herramienta). Para cambiar el estado de la herramienta, se dispone del campo **Tool Status** con una barra de desplazamiento que permite introducir valores entre 0.0 y 1.0. La herramienta activa y el estado de la herramienta se muestran en la información dinámica (parte inferior izquierda de la ventana gráfica) con los identificadores **T** y **TS** respectivamente.

Otro aspecto importante a controlar es el sistema de coordenadas de herramienta, que sirve para mandar movimientos u obtener la configuración del robot. La localización en que se encuentra una configuración del robot se define mediante la localización de un sistema de coordenadas asociado a la herramienta respecto al sistema origen de programación del robot. En el fichero rkf de un robot se pueden definir diferentes sistemas de coordenadas. Con el campo **Active Tool Frame** se permite cambiar entre los disponibles para el robot.

Para la realización de movimientos de robots de forma correcta, especialmente para movimientos en el espacio cartesiano, es fundamental conocer cómo se encuentren definidos los sistemas de coordenadas de herramienta. Los robots normalmente tienen el eje Z de la herramienta apuntando hacia “fuera” de la misma, pero el **CRS A465** no lo tiene así, siendo el eje X el que lleva esa dirección.

#### Tarea 5.1:

- Con el robot **CRS A465** cargado, ejecuta **VRS TeachPendant**.
- Cambia el estado de la herramienta entre los valores extremos.
- Deduce cuantos estadios hay modelados.

<sup>9</sup> La sutil diferencia entre estadio y estado (state y status en inglés respectivamente) viene dada por que el primer término se representa con un número entero (valor discreto) mientras que el segundo con un número real (valor continuo).

- Activa cada uno de los sistemas de coordenadas de herramienta disponibles para el robot y observa donde se encuentran. Observa que la localización mostrada en la información dinámica del robot cambia de posición en X al cambiar el sistema de herramienta.
- Deja activo el sistema de coordenadas de herramienta 1.

### Tarea 5.2:



- Carga el robot **ABB IRB L6** disponible en **Models\Tutorial** y localízalo con Y=1000.
- Realiza un **Refresh** en **VRS TeachPendant** y selecciona el robot **ABB IRB L6**.
- Deduce las herramientas existentes y cuáles son sus sistemas de coordenadas respectivos.
- Cierra el robot **ABB IRB L6** y realiza un **Refresh** en **VRS TeachPendant**.

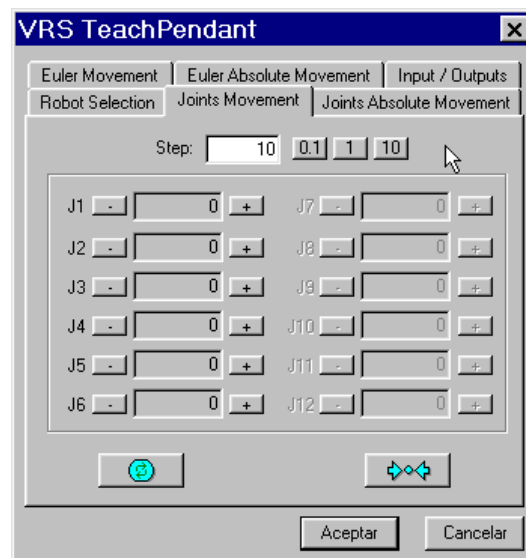
## 5.2. Diálogo Joint Motion

En el movimiento de robots por articulaciones se controla directamente el valor de las variables de articulación de los robots. Las variables de articulación del robot se muestran en todo momento en la información dinámica (parte inferior izquierda de la ventana gráfica) para el robot activo.

Al pulsar sobre la pestaña **Joint Motion** se abrirá un diálogo que permite mover el robot por articulaciones mediante pulsaciones del ratón en botones específicos. El diálogo se comenta a continuación.

En los campos **J1** a **J12** se mostrará el valor actual de las variables de articulación del robot, estando activos tantos botones como articulaciones tenga el robot. Para mover una articulación se debe pulsar en el botón **-** o **+** (variaciones en sentido negativo o positivo respectivamente). En dicha articulación se producirá el incremento especificado en el campo **Step**: que se puede modificar directamente o ajustar con los botones **0.1**, **1** ó **10**. Nótese que si los botones de movimiento se mantienen pulsados, el robot continuará moviéndose, lo que puede producir que, según la sensibilidad del ratón, se realicen más de un movimiento al pinchar en el botón.

Además se dispone de los botones  y  que llevan el robot a su configuración de *reset* o sincronismo respectivamente.



Si durante el movimiento de una articulación se sale fuera de su rango de trabajo, la pantalla parpadeará y el robot no podrá moverse más en ese sentido. Los rangos de las articulaciones están definidos en el fichero rkf.

Es posible desactivar el control de rangos de movimiento del robot, de forma que se permita moverlo a posiciones inalcanzables para el robot real e incluso imposibles (por colisiones). Para desactivar el control de rangos se dispone del campo **Check Range** del diálogo **Robot Selection** ya comentado anteriormente. El control de rangos debe estar siempre activo excepto para realizar alguna prueba de movimiento fuera del rango del robot. Si el robot se lleva fuera de rangos y entonces se activa el control de rangos, el robot puede quedar bloqueado y sólo será posible llevarlo a sincronismo o reposo.

Otro aspecto importante es la velocidad con que se realizan los movimientos del robot. Con la barra de desplazamiento **Speed** del diálogo **Robot Selection** se permite ajustar la velocidad de movimiento del robot. Cuando el valor es 1.0, el robot se moverá a la máxima velocidad que permita la simulación en el PC. Por el contrario, reduciendo este valor, se incluirán retardos en

la simulación de los movimientos<sup>10</sup>. La velocidad de los movimientos es apreciable principalmente en movimientos rectilíneos.

Existen dos botones para llevar el robot a su configuración de *reset* (todas las variables de articulación a cero) y a su configuración de *sincronismo* (definida en el fichero rkf). Nótese que la configuración *reset* puede estar fuera de rangos, mientras que la de sincronismo debe estar dentro de los rangos, ya que en otro caso producirá un error la carga del robot. Un robot puede estar definido con ambas configuraciones iguales o diferentes, es decir, la configuración de sincronismo no tiene porqué tener todas las variables a cero.

**Tarea 5.3:**



- Selecciona el robot **CRS A465**.
- Mueve todas las articulaciones del robot.
- Prueba el movimiento con velocidad 0.5 y velocidad 1.0, dejando esta velocidad.
- Prueba diferentes valores en el campo **Step**.
- Deduce los rangos de cada articulación llevándola sus extremos.
- Desactiva el control de rangos y mueve alguna articulación fuera de su rango.
- Activa el control de rangos y lleva el robot a su configuración de sincronismo.
- Prueba las configuraciones de sincronismo y reposo para determinar si son idénticas o no.
- Lleva el robot a su configuración de sincronismo.

**Tarea 5.4:**

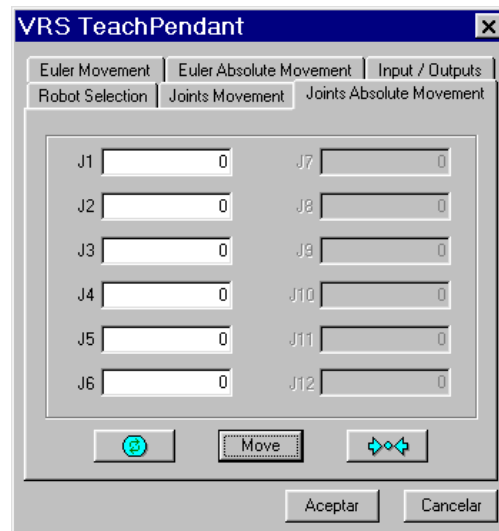
- Lee la sección correspondiente al **VRS Trace Control** para conocer cómo se controla la traza de los robots.
- Mueve el robot **CRS A465** visualizando las trazas y probando las opciones del **VRS Trace Control** para activar, ocultar y borrar las trazas (y sus inversas). En el resto de tareas se puede hacer uso de estas opciones para tener más claro los movimientos del robot.
- Lleva el robot a su configuración de sincronismo.

**5.3. Diálogo Joint Absolute Motion**

El robot también puede moverse a una configuración concreta mediante el diálogo **Joint Absolute Motion** que se muestra a la derecha.

Este diálogo tendrá tantos campos activos como articulaciones tenga el robot, numeradas desde **J1** a **J12**. Para mover el robot se introduce la configuración mediante el valor deseado de sus variables de articulación, pulsando intro o el botón **Move**. Los botones  y  tienen el mismo significado que en el diálogo anterior.

Si al teclear una configuración, una de las articulaciones está fuera de su rango, la pantalla parpadea y el robot no se mueve, siendo cancelada la orden.



**Tarea 5.5:**





- Mueve el robot **CRS A465** especificando diversas configuraciones.
- Lleva el robot a su configuración de sincronismo.

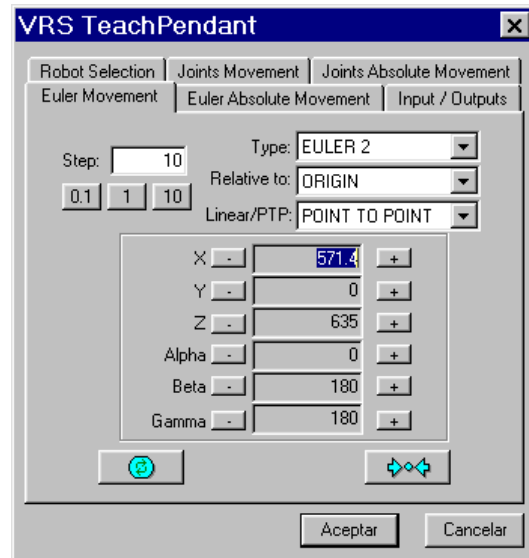
**5.4. Diálogo Cartesian Motion**

<sup>10</sup> Para velocidades muy lentas, un movimiento largo del robot puede tardar lo suficiente para disparar un error de *Timeout*, lo que produce que **VRS Teach Pendant** se cierre con un mensaje de error. En el apéndice A se muestra como ajustar los posibles valores de *Timeout*.

En el movimiento de robots en el Espacio Cartesiano se controla directamente la localización del sistema de coordenadas de herramienta activo respecto al sistema de coordenadas origen de programación del robot. La localización se representa con la posición X,Y,Z y tres ángulos de Euler de tipo 2 (Alpha,Beta,Gamma). Esta localización se muestra en todo momento en la información dinámica (parte inferior izquierda de la ventana gráfica) para el robot activo.

Para realizar estos movimientos se dispone del diálogo **Cartesian Motion** que se explica a continuación.

En este diálogo se muestra la localización del **Tool Frame** activo respecto al sistema origen de programación del robot. Esta localización puede cambiarse pulsando los botones  o  (variaciones en sentido negativo o positivo respectivamente) de cada una de las seis variables que especifican la localización (X,Y,Z,Alpha,Beta,Gamma). En dicha variable se producirá el incremento especificado en el campo **Step:** que se puede modificar directamente o ajustar con los botones **0.1**, **1** ó **10**. Nótese que si los botones de movimiento se mantienen pulsados, el robot no se mueve, esperando que se libere el botón. Por el contrario, si se pulsa varias veces el botón, los movimientos se irán realizando consecutivamente. Los botones  y  tienen el mismo significado que en los diálogos anteriores.



Si durante un movimiento, la localización a la que se quiere llevar el robot es inalcanzable o una articulación se sale fuera de su rango de trabajo, la pantalla parpadeará y el robot no se moverá.

Nótese que la localización mostrada y los movimientos son dependientes del sistema de coordenadas de herramienta que esté activo en el campo **Active Tool Frame**, como se explicó anteriormente en el diálogo **Robot Selection**. Hay otros campos de este diálogo que también tienen influencia en los movimientos del robot, como los ya comentados **Check Range**, que permitirá mover el robot a localizaciones que tengan sus articulaciones fuera de rango, y **Speed** que permite ajustar la velocidad de movimiento del robot.

En el campo **Type:** se permite seleccionar la representación de la localización en los tres tipos de ángulos de Euler existentes. El tipo 2 es el que se usa por defecto al ser el más común en robótica, si bien el tipo 3 es particularmente interesante para poder realizar rotaciones respecto a los ejes del sistema de coordenadas del **Tool Frame** activo. En la siguiente tabla se recuerda cuál es el significado de los tres tipos de representación de orientación mediante ángulos de Euler.

Ángulos de Euler	Alfa	Beta	Gama
<b>Tipo 1</b>	Eje Z (Origin Frame)	Eje U (Tool Frame)	Eje W (Tool Frame)
<b>Tipo 2</b>	Eje Z (Origin Frame)	Eje V (Tool Frame)	Eje W (Tool Frame)
<b>Tipo 3</b>	Eje X (Origin Frame)	Eje Y (Origin Frame)	Eje Z (Origin Frame)

#### **Tarea 5.6:**

- Mueve el robot **CRS A465** en posición y orientación con diferentes valores de **Step**.
- Mueve el robot en orientación con representación en diferentes tipos de ángulos de Euler y comprende su diferencia.
- Lleva el robot a su configuración de sincronismo.

#### **Movimientos punto a punto y rectilíneos**



Los movimientos del robot se pueden realizar con el tipo de movimiento punto a punto, donde el robot genera un movimiento interpolando en el espacio de articulaciones o con movimiento rectilíneo, de forma que el origen del sistema de coordenadas de herramienta activo siga una línea recta. El tipo de movimiento se puede seleccionar con las opciones **POINT TO POINT** y **LINEAR** accesibles en el campo **Linear/PTP**.

**Tarea 5.7:**

- Mueve el robot **CRS A465** en posición X,Y,Z con diferentes valores de **Step** probando los movimientos punto a punto y línea recta y comprende su diferencia.
- Lleva el robot a su configuración de sincronismo y deja seleccionado el modo punto a punto.

Los movimientos del robot se pueden especificar relativos a tres posibles sistemas de coordenadas, el sistema origen de programación (**ORIGIN**), el sistema de coordenadas de la herramienta (**TOOL FRAME**) y el sistema de coordenadas del mundo (**WORLD**). Estas opciones se pueden seleccionar en el campo **Relative to**.

**Tarea 5.8:**

- Localiza el robot **CRS A465** con el valor Gamma a 45°.
- Visualiza los sistemas de coordenadas origen del robot y del mundo.
- Mueve el robot de forma relativa a los diferentes sistemas de coordenadas y comprende su diferencia.
- Lleva el robot a su configuración de sincronismo y deja seleccionado los movimientos respecto al origen.
- Localiza el robot con el valor Gamma a 0°.

**Selección de la solución en la cinemática inversa**

Cuando el robot se mueve en el espacio cartesiano el simulador selecciona entre las posibles configuraciones del robot que resuelven la cinemática inversa. Normalmente interesa la configuración más cercana ya que producirá un ahorro de energía, pero se puede seleccionar otra configuración con el campo **IK Config...** del diálogo **Robot Selection** que abre el siguiente diálogo.

La opción **Closest Configuration** resuelve la cinemática inversa con la configuración más cercana a la anterior del robot. Por el contrario, se pueden seleccionar configuraciones diferentes (según el tipo de robot) seleccionando las opciones deseadas entre los posibles siguientes pares:

- Brazo derecho o izquierdo (**Left Arm/Right Arm**).
- Codo arriba o abajo (**Elbow Up/Elbow Down**).
- Muñeca positiva o negativa (**Positive Wrist/Negative Wrist**).

Seleccionando una configuración específica, el movimiento sólo será posible si esta configuración es correcta, descartándose el resto de posibles soluciones. Por el contrario, con la configuración más cercana, se considera la mejor solución de todas las alcanzables, por lo que normalmente se debe mantener esta opción.



**Tarea 5.9:**

- Con **IK Config...** selecciona la opción **Elbow Down** para el robot **CRS A465**.
- Mueve el robot en la dirección X. Observa la nueva configuración seleccionada.
- Selecciona la opción **Left Arm**.
- Mueve el robot en la dirección X. Observa la nueva configuración seleccionada.
- Selecciona la opción **Positive Wrist**.
- Mueve el robot en la dirección X. Observa la nueva configuración seleccionada.
- Selecciona el modo **Closest Configuration**.
- Mueve el robot en la dirección X. Observa la nueva configuración seleccionada.
- Lleva el robot a su configuración de sincronismo.

**Control de orientación**

Cuando al robot se le manda a una localización, la alcanzará tanto en posición como en orientación. Para ciertas aplicaciones puede ser útil ordenar movimientos con cambios únicamente en posición, sin forzar al robot a seguir la orientación. Esto es posible desactivando el control de orientación con el campo **Check Orientation** del diálogo **Robot Selection**. El control de orientación debe estar siempre activo excepto para realizar alguna prueba de movimiento del robot. Este aspecto es especialmente importante para despreocuparse de la orientación en robots que tienen menos de seis grados de libertad. Si el robot se mueve sin control de orientación y entonces se activa el control de orientación, el robot puede quedar bloqueado y sólo será posible llevarlo a sincronismo o reposo.

#### Tarea 5.10:

- Desactiva el campo **Check Orientation** para el robot **CRS A465**.
- Mueve el robot en las direcciones Y y Z. Observa la orientación del sistema de herramienta.
- Intenta mover el robot en orientación cambiando los ángulos. Observa que no se produce movimiento.
- Activa el campo **Check Orientation**.
- Lleva el robot a su configuración de sincronismo.

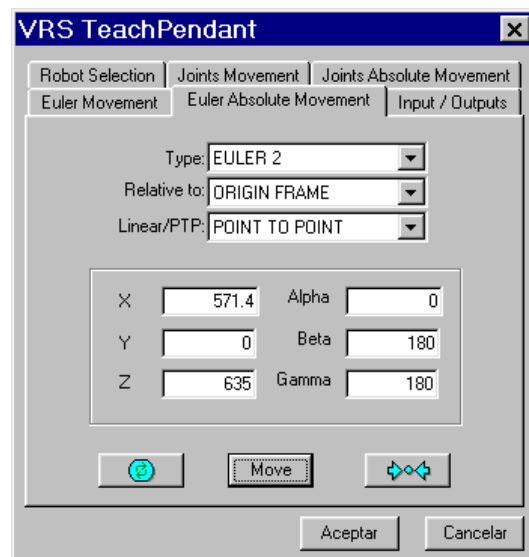
### 5.5. Diálogo *Cartesian Absolute Motion*

El robot también puede moverse a una localización concreta mediante el diálogo **Cartesian Absolute Motion** que se muestra a la derecha.

Este diálogo tiene los campos X,Y,Z para especificar la posición y los campos Alpha, Beta, Gamma para especificar la orientación. Para mover el robot se introduce la configuración mediante el valor deseado de posición y orientación, pulsando intro o el botón **Move**.

Los botones  y  tienen el mismo significado que en el diálogo anterior.

Los campos **Type**, **Relative to** y **Linear/PTP** tienen el mismo significado que en diálogo anterior.



Si al teclear una localización, la localización a la que se quiere llevar el robot es inalcanzable o una articulación se sale fuera de su rango de trabajo, la pantalla parpadeará y el robot no se moverá, siendo cancelada la orden.

#### Tarea 5.11:

- Mueve el robot **CRS A465** especificando diversas localizaciones.
- Lleva el robot a su configuración de sincronismo.

### 5.6. Diálogo *Input / Outputs*

En *VRS* cada robot cargado tiene asignadas 16 entradas digitales, 16 salidas digitales, 16 entradas analógicas y 16 salidas analógicas. Estas señales se pueden conectar de robot a robot mediante la *VRS Tool VRS IOConnection* que se verá posteriormente. Para tener acceso a estas señales (leer el valor de las entradas y modificar el valor de las salidas) se dispone del diálogo **Input/Outputs**.

Las entradas digitales están numeradas de 1 a 16 y agrupadas en el campo **Digital In**. Sobre estas entradas no se puede actuar directamente, pero el programa actualizará su estado continuamente (comprobando los valores cada medio segundo), existiendo una marca cuando están a nivel alto (1 lógico).

Las salidas digitales están numeradas de 1 a 16 y agrupadas en el campo **Digital Out**. Sobre estas salidas se puede actuar directamente, pulsando sobre el círculo correspondiente y el programa mostrará su estado, existiendo una marca cuando están a nivel alto (1 lógico).

Las entradas y salidas analógicas (ambas numeradas de 1 a 16) están agrupadas en el campo **Analogical**, separándose en el campo **In** las entradas y en el campo **Out** las salidas.



Seleccionando una entrada analógica de la lista de entradas, a su derecha mostrará el valor leído, siendo 0.0 si no está conectada. Seleccionando una salida analógica de la lista de salidas, a su derecha mostrará el valor actual, pudiéndose teclear otro valor que se enviará por la salida correspondiente al pulsar **Set**.






## 6. VRS TraceControl

Para la gestión de las trazas de los robots (activación, ocultación y eliminación) se dispone de la *VRS Tool* llamada **VRS TraceControl**. **VRS TraceControl** es una aplicación externa que sirve para controlar las trazas de los robots. Como programa externo, su ejecutable **VRSTraceControl.exe** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **ApplicationsVRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Trace Control** si bien no se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas. Si **VRS TraceControl** está en ejecución se volverá a abrir en caso de lanzarse a ejecución de nuevo.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *pop-up*, accesible al pulsar sobre la ventana (fuera de los botones) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*.



Como ya se ha comentado, con **VRS TraceControl** se pueden controlar las trazas de los robots en el simulador. La traza de un robot es el rastro que deja el origen del sistema de coordenadas de herramienta activo del robot al moverse. Las trazas se deben activar para dibujarse, se pueden ocultar para no estar visibles y se pueden eliminar. Los botones de cada una de estas opciones son los siguientes:

<b>VRS TraceControl</b>	<b>Acción</b>
	Conmuta entre activar / desactivar la traza de todos los robots cargados en <i>VRS</i> .
	Conmuta entre ocultar /mostrar la traza de todos los robots cargados en <i>VRS</i> .
	Elimina la traza de todos los robots cargados en <i>VRS</i> .
	Permite elegir el color de traza de todos los robots cargados en <i>VRS</i> .
	Salva la traza de los robots (opción no disponible).

Nótese que cualquiera de los botones afecta a todos los robots cargados en *VRS*, pero sólo tiene sentido para los robots que en ese momento estén cargados en *VRS*, no para robots que se carguen posteriormente. Por otra parte, si otra aplicación realiza también el manejo de la traza de un robot, puede ocurrir que la información entre ambas aplicaciones no sea consistente.

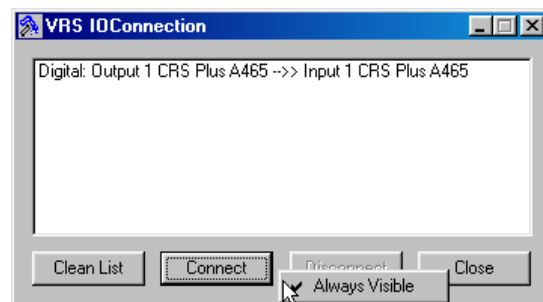
Conviene aclarar algunos aspectos respecto a estas opciones:

- Al desactivar la traza de un robot se cierra un segmento de traza, mientras que al activar la traza se abre un nuevo segmento de traza. Por tanto, la traza será la unión de segmentos generados al conmutar traza activada y desactivada.
- Si con la traza activa se cambia el sistema de coordenadas activo del robot se comienza un nuevo segmento de traza.
- La traza se puede ir activando y desactivando para que sólo se represente los movimientos hechos mientras que la traza esté activa.
- El ocultar la traza no la elimina ni la desactiva, sólo hace que no se muestre en pantalla. Por ejemplo, si la traza está activa y oculta (ambos botones pulsados) guardará los puntos por los que pase el origen del sistema de coordenadas de herramienta, pese a que no los muestre. En el momento que se desactive la opción de ocultar traza, aparecerá toda la traza generada mientras la traza estaba activa.
- Al eliminar la traza de un robot se desactiva la misma.

## 7. *VRS IOConnection*

Para la conexión de salidas de un robot a entradas de otro robot en *VRS* se dispone de la *VRS Tool* llamada ***VRS IOConnection***. ***VRS IOConnection*** es una aplicación externa que sirve para conectar salidas (digitales y analógicas) de un robot a entradas (digitales y analógicas respectivamente) de otro robot. Como programa externo, su ejecutable ***VRIOConnection.exe*** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **Applications\VRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Input/Output Connection** si bien no se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas. Si ***VRS IOConnection*** está en ejecución se volverá a abrir en caso de lanzarse a ejecución de nuevo.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *popup*, accesible al pulsar sobre la ventana (fuera de los botones) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*.



Como ya se ha comentado, con ***VRS IOConnection*** se pueden conectar salidas de un robot a las entradas de otro robot, de forma que las salidas se transfieran a las entradas correspondientes. Evidentemente, la conexión siempre será salida digital a entrada digital o bien salida analógica a entrada analógica.

Una salida de un robot se puede conectar a varias entradas de diferentes robots, incluso entradas de sí mismo, pero a una entrada sólo se le puede conectar una salida. Cualquier nueva conexión en una entrada sustituye la conexión anterior a esa entrada.

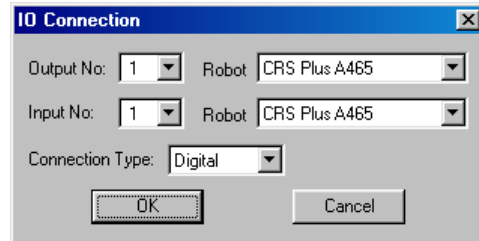
Los campos para permitir las conexiones son los siguientes:

<b>VRS IOConnection</b>	<b>Acción</b>
<b>Clean List</b>	Vacía la lista de conexiones, si bien las conexiones permanecen.
<b>Connect</b>	Conecta señales según el diálogo <b>IO Connection</b>
<b>Disconnect</b>	Desconecta señales (no disponible)
<b>Close</b>	Cierra la ventana

Es importante resaltar que la primera opción sólo vacía la lista de conexiones, siendo necesario cerrar un robot para que desaparezcan todas sus conexiones en *VRS*. Con el botón **Connect** se entra en el diálogo que permite establecer conexiones, que se muestra a continuación.

Con los campos **Output No** e **Input No** se seleccionan los números de salida y entrada, siendo aplicados sobre el robot seleccionado en los campos **Robot** de su derecha.

Con el campo **Connection Type** se puede seleccionar el tipo de conexión (digital o analógica). Con **OK** o **Cancel** se valida la conexión o se cancela.



Al introducir una conexión, ésta se mostrará en la lista de conexiones realizadas del diálogo **VRS IOConnection**. Debe recordarse que si se repite una conexión en una entrada, la nueva conexión sustituye a la anterior. La lista se mantiene ordenada según se han introducido, por lo que la conexión situada más abajo es la que permanece al repetirse una entrada. Por otra parte, en la lista sólo se muestran las conexiones que se han realizado con esta aplicación, no las realizadas por otras aplicaciones.

**Tarea 7.1:**

- Conecta la salida digital 1 del robot **CRS A465** a su entrada digital 1.
- Comprueba con el **VRS TeachPendant** la conexión.
- Conecta la salida analógica 1 del robot **CRS A465** a su entrada analógica 1.
- Comprueba con el **VRS TeachPendant** la conexión.

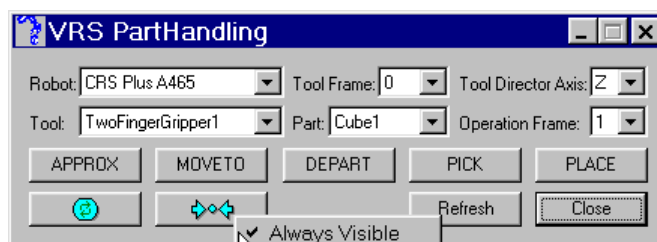
**Tarea 7.2:**

- Con dos robots cargados en *VRS*, conecta señales entre ellos y verifica las conexiones.



## 8. *VRS PartHandling*

Para la manipulación de piezas del entorno con robots en *VRS* se dispone de la *VRS Tool* llamada **VRS PartHandling**. **VRS PartHandling** es una aplicación externa que sirve manipular piezas del entorno con robots. Como programa externo, su ejecutable **VRSPartHandling.exe** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **Applications\VRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Part Handling** si bien no se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas. Si **VRS PartHandling** está en ejecución se volverá a abrir en caso de lanzarse a ejecución de nuevo.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *pop-up*, accesible al pulsar sobre la ventana (fuera de los botones) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*.



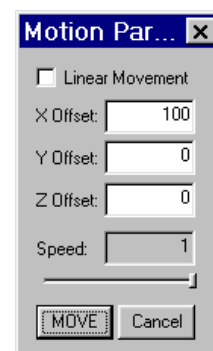
Como ya se ha comentado, con **VRS PartHandling** se pueden manipular piezas del entorno con los robots. Para ello dispone de las siguientes opciones:

<b>VRS PartHandling</b>	<b>Acción</b>
<b>Robot</b>	Permite seleccionar el robot.
<b>Tool Frame</b>	Permite seleccionar el sistema de coordenadas de herramienta activo.
<b>Tool Director Axis</b>	Permite seleccionar el eje director de la herramienta para los movimientos de aproximación y retirada.
<b>Tool</b>	Permite seleccionar la herramienta activa.
<b>Part</b>	Permite seleccionar la pieza a manipular.
<b>Operation Frame</b>	Permite seleccionar el sistema de operación de la pieza para el agarre y los movimientos relativos.
<b>APPROX</b>	Permite realizar la aproximación a un sistema de operación de una pieza.
<b>MOVETO</b>	Permite realizar el movimiento a un sistema de operación de una pieza.
<b>DEPART</b>	Permite realizar un movimiento de retirada.
<b>PICK</b>	Permite realizar el agarre de una pieza.
<b>PLACE</b>	Permite realizar la dejada de una pieza.
	Mueve el robot a su configuración de <i>reset</i> .
	Mueve el robot a su configuración de sincronismo.
<b>Refresh</b>	Actualiza la información de robots, herramientas y piezas.
<b>Close</b>	Cierra la ventana.

Los campos **Robot**, **Tool Frame** y **Tool** y los botones de *reset* y sincronismo son similares a los ya explicados para otros diálogos. Igualmente, los dos últimos botones tienen significado obvio. Con el campo **Tool Director Axis** se selecciona cuál es el eje del sistema de coordenadas de herramienta activo que se toma como director para los movimientos de aproximación y retirada del robot. Con el campo **Part** se especifica la pieza a coger, dejar o sobre la que se realizarán los movimientos relativos, en función del sistema de operación seleccionado en el campo **Operation Frame**.

Los botones **APPROX**, **MOVETO** y **DEPART** abren el diálogo **Motion Parameters** que se muestra a continuación. El significado de las acciones y los campos de este nuevo diálogo son los siguientes:

- Con el botón **APPROX** se realiza una aproximación al sistema de operación de una pieza. El movimiento del robot se puede especificar que sea en línea recta con el campo **Linear Movement** (por defecto no lo es). Igualmente se puede especificar la velocidad de movimiento del robot con la barra de desplazamiento del campo **Speed** (por defecto es 1.0). Con los valores **X Offset**, **Y Offset**, **Z Offset** se pueden modificar las distancias de aproximación (por defecto será 100 en el eje director de la herramienta y cero el resto). Nótese que estos valores indican distancias en los sentidos **NEGATIVOS** de los ejes del sistema de operación de pieza usado como base de los movimientos. Con **MOVE** se realiza el movimiento mientras que se cancela con **Cancel**.



- Con el botón **MOVETO** se realiza un movimiento al sistema de operación de una pieza. El movimiento del robot se puede especificar que sea en línea recta con el campo **Linear Movement** (por defecto sí lo es). Igualmente se puede especificar la velocidad de movimiento del robot con la barra de desplazamiento del campo **Speed** (por defecto es 0.5). Los campos **X Offset**, **Y Offset**, **Z Offset** están deshabilitados porque deben tener siempre valor 0 (en otro caso se trata de una aproximación). Con **MOVE** se realiza el movimiento mientras que se cancela con **Cancel**.
- Con el botón **DEPART** se realiza una retirada desde la localización en que se encuentra el robot. El movimiento del robot se puede especificar que sea en línea recta con el campo **Linear Movement** (por defecto sí lo es). Igualmente se puede especificar la velocidad de movimiento del robot con la barra de desplazamiento del campo **Speed** (por defecto es 0.5). Con los valores **X Offset**, **Y Offset**, **Z Offset** se pueden modificar las distancias de retirada (por defecto será 100 en el eje director de la herramienta y cero el resto). Nótese

que estos valores indican distancias en los sentidos **NEGATIVOS** de los ejes del sistema de herramienta activo usado como base de los movimientos. Con **MOVE** se realiza el movimiento mientras que se cancela con **Cancel**.

Los botones **PICK**, y **PLACE** abren el diálogo **Pick&Place Parameters** que se muestra a continuación. El significado de las acciones y los campos de este nuevo diálogo son los siguientes:

- Con el botón **PICK** se realiza el agarre de una pieza. Tras realizar el agarre, la herramienta cambiará al estado de herramienta al valor especificado en el campo **Tool Status** que se puede modificar con su barra de desplazamiento (por defecto es 1). La pieza quedará agarrada a la pinza activa del robot y la transportará consigo en sucesivos movimientos. Con **OK** se realiza el agarre mientras que se cancela con **Cancel**.



Con el campo **Check Coincidence** se puede especificar que realice, antes de realizar el agarre, una comprobación de coincidencia entre las localizaciones de los sistemas de coordenadas de herramienta activo y de operación de pieza, de forma que sólo agarre la pieza si son coincidentes (por defecto se realiza la comprobación).

- Con el botón **PLACE** se realiza la dejada de una pieza. Tras realizar la dejada, la herramienta cambiará al estado de herramienta al valor especificado en el campo **Tool Status** que se puede modificar con su barra de desplazamiento (por defecto es 0). La pieza quedará desligada de la pinza activa del robot y ya no la transportará consigo en sucesivos movimientos. Con **OK** se realiza la dejada mientras que se cancela con **Cancel**. El campo **Check Coincidence** está deshabilitado porque no tiene sentido en este caso.

Es importante resaltar que esta acción tiene efecto sobre la pieza que esté seleccionada en el campo **Part**, por lo que se producirá un error si el robot no la tiene agarrada.

#### **Tarea 8.1:**

- Carga el robot **CRS A465** y su mesa como en la Tarea 4.1 y abre **VRS PartHandling**.
- Selecciona el sistema de coordenadas de herramienta 1 y como eje director el eje X.
- Selecciona la pieza Cubo1 y realiza un movimiento de aproximación a la pieza con **APPROX** y un movimiento hasta la pieza con **MOVETO** (con los valores por defecto).
- Realiza el agarre de la pieza y la retirada del robot (usa 0.4 para el **Tool Status**).
- Selecciona como pieza el pallet y su sistema de operación 2.
- Comprueba lo que realizan ahora los botones **APPROX** y **MOVETO**.
- Selecciona la pieza Cubo1, déjala (con **Tool Status** a 0) y retira el robot con **DEPART**.

#### **Tarea 8.2:**

- Mueve las piezas Cubo2 y Cubo3 a las localizaciones especificadas con los sistemas de operación 1 y 3 del pallet respectivamente.
- No olvides alternar el campo **Part** entre el pallet y el cubo correspondiente al hacer movimientos y dejadas de pieza.
- Sigue la lógica de **APPROX-MOVETO-PICK/PLACE-DEPART**.

## **9. VRS SpeedControl**

Para el ajuste de la velocidad de simulación de robots en *VRS* se dispone de la *VRS Tool* llamada **VRS SpeedControl**. **VRS SpeedControl** es una aplicación externa que sirve ajustar la velocidad de simulación de robots. Como programa externo, su ejecutable **VRSSpeedControl.exe** se puede ejecutar directamente desde el operativo, estando localizado en la carpeta **ApplicationsVRSTools** a partir del directorio de *VRS*. Sin embargo, en la configuración de instalación por defecto de *VRS* se encuentra accesible con la opción del menú **File>>VRS Tools>>VRS Speed Control** si bien no se ejecuta directamente al arrancar *VRS*. Una vez ejecutado, es un programa como otro cualquiera, con acceso desde la barra de tareas. Si **VRS SpeedControl** está en ejecución se volverá a abrir en caso de lanzarse a ejecución de nuevo.

Su interfaz es el que se muestra a la derecha, disponiendo, como todas las aplicaciones de tipo *VRS Tool*, de un menú propio de tipo *pop-up*, accesible al pulsar sobre la ventana (fuera de los botones) con el botón derecho del ratón, que permite, si se activa la opción **Always Visible** (como aparece por defecto), tener esta aplicación siempre visible por encima de cualquier otra ventana, incluida la de *VRS*.



Como ya se ha comentado, con **VRS SpeedControl** se puede ajustar la velocidad de simulación de los robots en *VRS*. Los robots se moverán en cada comando de movimiento según su parámetro específico de velocidad, pero existe una escala de velocidades que puede reducir esta velocidad. Esta escala de velocidades, que se controla en esta *VRS Tool* mediante la barra de desplazamiento del campo **Speed Scale** permitirá reducir esta velocidad, ya que para el valor de 1.0 no aplica ninguna reducción mientras que para valores más pequeños reduce la velocidad de simulación. La ventana actualiza el valor de la escala de velocidades periódicamente, por lo que si otra aplicación la modifica, se detectará con **VRS SpeedControl**.

## 10. Aplicaciones de demostración

El programa dispone de varias aplicaciones de demostración que se pueden ejecutar con la opción **File>>Demos>>Industrial Robots**. Estas aplicaciones se han desarrollado mediante programas externos tal como se muestra en el guión titulado "Programación de aplicaciones para *VRS*". En la carpeta **Videos\Demos** se encuentran disponibles vídeos de las demostraciones (en caso de haberse instalado).

### **Tarea 10.1:**

- Ejecuta las demostraciones de aplicaciones con robots industriales disponibles en la opción **File>>Demos>>Industrial Robots**.



## A. Configuración de VRS

### Configuración del *Timeout*

El fichero VREAL.INI instalado en el directorio del operativo permite configurar el *Timeout* de la comunicación entre VRS y las aplicaciones externas. El contenido del fichero es el siguiente (si bien los valores de los campos podrían ser diferentes):

```
[GENERAL]
TimeOut=5
LoadTimeOut=40
;Communication Timeout in seconds
```

El campo **TimeOut** es el *timeout* en segundos para la comunicación con la aplicación externa. El campo **LoadTimeOut** es el *timeout* en segundos para la comunicación con la aplicación externa para las instrucciones de carga de ficheros. Normalmente, este *timeout* será mayor que el anterior.

### Configuración del Menú *VRS Tools*

Las herramientas *VRS Tools* se pueden configurar mediante el fichero **VRS.INI** que se instala en el operativo.

El contenido del fichero **VRS.INI** es del estilo siguiente (si bien el número de campos y los valores de los campos pueden ser diferentes):

```
[VERSION]
Version      =      UPV

[IKLIB]
DllName      =      iklib5.8.dll

[ROBOT_MENU]
Enabled = NO

[ENVELOPS]
ComputeEnvelops = NO
DrawEnvelops    = NO

[TOOL_MENU]
Name = VRS Tools

[TOOL_MENU_ENTRY1]
Name = VRS &Loader
Path = ./Applications/VRSTools/VRSLoader.exe
InitRun = YES

[TOOL_MENU_ENTRY2]
Name = VRS &Teach Pendant
Path = ./Applications/VRSTools/VRSTeachPendant.exe
InitRun = NO

[TOOL_MENU_ENTRY3]
Name = VRS T&race Control
Path = ./Applications/VRSTools/VRSTraceControl.exe
InitRun = NO

[TOOL_MENU_ENTRY4]
Name = VRS &I/O Connection
```

```
Path = ./Applications/VRSTools/VRSlOConnection.exe  
InitRun = NO
```

```
; Max number of tool menu entries is 10
```

- El campo **Version** de la sección **VERSION** será el seleccionado en el proceso de instalación. **No se debe alterar el contenido de este campo.**
- El campo **DIName** de la sección **IKLIB** es el fichero de la *dll* que se utiliza por defecto en la cinemática del robot. **No se debe alterar el contenido de este campo** a no ser que se haya desarrollado una nueva librería de cinemática.
- El campo **Enabled** de la sección **ROBOTMENU** se debe poner a **YES** si el programa realiza funciones de servidor de un robot con comunicación con el mismo. **No se debe alterar el contenido de este campo** en otro caso.
- Los campos **ComputeEnvelops** y **DrawEnvelops** de la sección **ENVELOPS** se debe poner a **YES** si en el programa se utilizan aplicaciones avanzadas de detección de colisiones o simulación de sensores de distancias. **No se debe alterar el contenido de este campo** en otro caso.
- El campo **Name** de la sección **TOOL\_MENU** contendrá el nombre de la opción del menú File donde se encuentran las herramientas *VRS Tools*.
- Para cada sección **TOOL\_MENU\_ENTRYi** (donde *i* es un entero ordenado) se puede especificar para cada *VRS Tool* que se instale los siguientes valores:
  - **Name**: nombre que aparece en el menú.
  - **Path**: nombre del ejecutable con el *path* a partir del *VRS Path*.
  - **InitRun**: puesto a **YES** se ejecutará directamente al inicializarse *VRS*, puesto a **NO**, no se ejecutará, debiendo lanzarlo el usuario.