Departamento de Ingeniería de Sistemas y Automática

Universitat Politècnica de València

Simulation in robotics with *VirtualRobot*. Introduction *VRS*

Martin Mellado Arteche

etsinf

Escuela Técnica
Superior de Ingeniería
Informática

September, 2012

Contents

Introduction to the graphical simulation system for robotics *VRS*

1. Introduction

The risk of the *off-line* use of textual programming in robots is that the programmes developed may be tried out in the robot without there being a one hundred percent certainty that they are going to work properly. That is why there is a need for graphic simulation systems which permit the prior viewing of the results of the running of a programme in order to be able to perfect the latter. Although this process would greatly reduce the possibility of committing errors in programmes, as well as making it easier to design the latter, it is important to remember that what happens in a virtual environment on a computer does not always correspond exactly with what the robot will do in the real world, owing to the differences between the computer model of the robot and the robot itself, and also the surrounding elements.

The aim of this tutorial is to introduce the reader to a system of *off-line* robot simulation and programming, in order, in subsequent chapters, to show the reader how to programme the same for use in typical industrial applications. For this purpose, we shall use the programme, *VirtualRobot Simulator* (*VRS*), which can be programmed in C/C++. This chapter includes a *tutorial* for the learning of said programme, with a number of tasks to be carried out by the student.

2. VirtualRobot Simulator (*VRS*)

VirtualRobot Simulator (*VRS*) is a multipurpose simulator for the simulation of robots, designed by the robotics research group (http://robotica.ai2.upv.es) of the Systems and Automation Engineering Department *(Departamento de Ingeniería de Sistemas y Automática)* (DISA) of Valencia Polytechnic *(Universidad Politécnica de Valencia)* (UPV). It is designed for both industrial applications and research purposes.

VRS is basically a graphic modelling application with specific capabilities for the simulation of robots which includes the possibility of generating external application programmes (programmes runnable with Windows) for its control. In fact, many of its interface options are designed as external applications called *VRS Tools*. As well as the direct simulation possibilities offered by the programme and the *VRS Tools*, *VRS* permits the programming of the actions of virtual robots with programmes designed in C and/or C++. With *VRS* it is possible to give the robot an environment including items which the robot can manipulate using pincers, and which can be used as a reference for the movement of robots.

VRS was originally designed with the aim of being a tool for monitoring robots controlled by the Generis control system, developed by the Joint Research Centre (JRC), and can therefore be connected to such system in order to graphically represent the situation of a Generis controlled robot.

2.1. Installation of *VRS*

The **VirtualRobot** software installer is free and it is available at the official website: http://robotica.ai2.upv.es/virtualrobot. There are various installation options for its configuration.

To start up *VRS*, in the default version of the installer, a direct access is created to the same via **Start>>Programs>>Virtual Robot>>VRS**.

Before closing *VRS*, it is advisable to **close all external applications** which are in operation, including the *VRS Tools*. Otherwise, some of these applications may cause a *Timeout*[1] error in communication with *VRS*.

2.2. Structure of directories created for working with *VRS*

In the installation of the **VirtualRobot** software, the installation directory of the runnable *VRS* programme will have been specified. If we call this directory VRPATH, the installer will take it by default as "C:\Programme files\Virtual Robot\VRS". The runnable *VRS* programme is in the VRPATH directory. The remaining components of the VirtualRobot software can be found in files, some of the most important being:

- Applications: containing the different applications which are run over *VRS*, such as for example, *VRS Tools*, *VRM Tools*, *Demos*, etc., saved in different files. The file, Applications\Users, should be used to save the applications designed by the user.
- Models: containing the models of robots, crafts, environments and others which can be used in *VRS,* saved in separate files. The folder Models\Users (or any sub-folder with the name Users), should be used to save all models designed by the user.
- SourceCode: containing interface files and bookshop definitions to be used for designing new applications. The file SourceCode\Users, should be used to save the applications projects, with their source programmes, designed by the user.

Note: To avoid problems on shared computers, copy the folders Models and SourceCode to your own directories and use them to save your files. Following this protocol (saving applications, models and source projects in your folders), in order to *transport* the work of a user, it is enough just to copy the content of these files and transfer it to any other computer in the same place.

3. User Graphical Interface of *VRS*

3.1. Robot loading

On starting up the programme, after a presentation window, the empty application window and the *VRS Tool* window, called ***VRS Loader***[2]***,*** are displayed. This application is used to load robots. *VRS* is multi-robot; in other words, it allows for the simulation of a number of robots simultaneously.

In *VRS*, the robots are described by means of two files with the same name. In the first of these, with the extension RKF (Robot Kinematics File), certain general parameters are indicated along with a description of the kinematics. This is the file which should be loaded. The second file, with the extension RGF (Robot Geometric File), contains the geometric description of the robot and is loaded automatically along with the kinematics file. The description of the files is specified in the document, VRFFD.

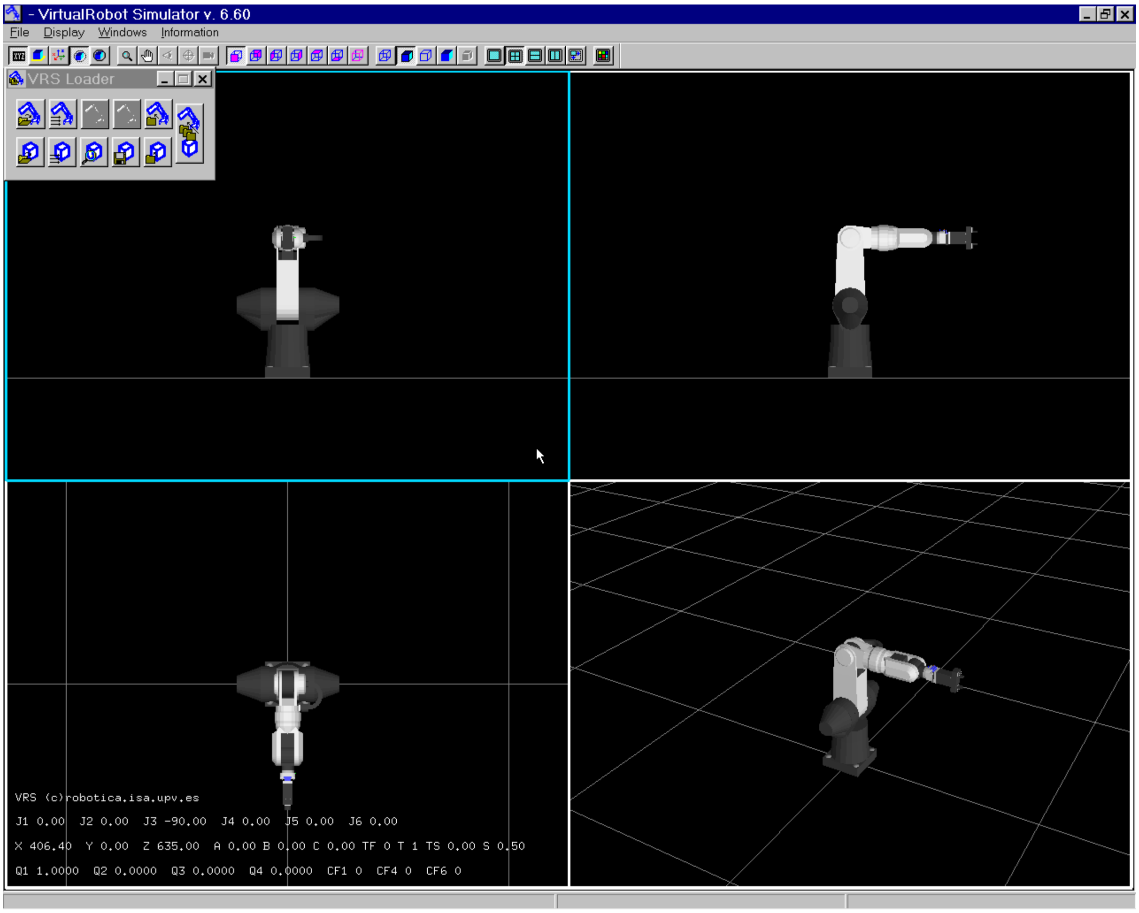
Task 3.1:
Load a robot in *VRS* following these steps:

- Select the button of the ***VRS Loader***
- In the file **Models\Tutorial,** select the robot file **CRS A465.rkf**.

The final result should be that shown in Figure 1.

[1] In appendix A you can see how to adjust values for the *Timeout*.

[2] In appendix A it is shown that *VRS Tool* applications can be configured to auto-start or not, so it is posible that *VRS Loader* is nos started.



```
VRS (c)robotica.isa.upv.es
J1 0.00  J2 0.00  J3 -90.00  J4 0.00  J5 0.00  J6 0.00
X 406.40  Y 0.00  Z 635.00  A 0.00 B 0.00 C 0.00 TF 0 T 1 TS 0.00 S 0.50
Q1 1.0000  Q2 0.0000  Q3 0.0000  Q4 0.0000  CF1 0  CF4 0  CF6 0
```

Figure 1. Robot CRS A465 in VRS

The application includes:

- A list of menus at the top showing the options organized hierarchically.
- A **Toolbar**[3], also at the top for controlling the graphic window.
- A **Statusbar**, at the bottom to show status and error messages along with the name and status of the active robot.
- A graphic window in the centre which, in principle, consists of four graphic frames showing the plan, elevation, profile and perspective view.

The graphic window shows the loaded robot and at all times[4], on the bottom left-hand side, the dynamic information of the active robot:

- the present value of the articulation variables, shown as J1 value, J2 value, ...
- the location of the active tool system (*tool frame*) with regard to the original programming system (with orientation represented by type 2 Euler angles), shown as X value, Y value, Z value, A value, B value, C value
- the tool frame (TF value), the active tool (T value), the status of the active tool (TS value) and the status of robot speed (S value)
- depending on the configuration of the programme, the orientation can appear in quaternions (Q1 value, ... Q4 value) and the description of the configuration of the robot (CF1 value, CF4 value, CF6 value).

It should be noted that, depending on how the VRS is configured, the active coordinate system of the active tool may appear in the pincer of the robot in the form of three axes, one red, one green and one blue (RGB), which represent, respectively, the axes X, Y and Z of said system. As will be seen, this display may later be switched on and off.

[3] **Toolbar** and **Statusbar** can be hidden with the corresponding options in **Windows** menu. In addition, **Toolbar** can be dragged and used as a window.

[4] This can be activated or not as it will be seen son.

3.2. Visualization Options

VRS has a **Toolbar**, with controls to configure the way in which graphic information is displayed. These controls have their equivalent in the **Display** or **Windows** menu (moreover, some of them have function keys for access). The main options are the following:

Configuration of the graphic window

The graphic window is divided into four equal areas. This permits the display of four different viewpoints of the same scene, by which the complete visualisation of the different elements present in the simulation is achieved. Owing to diverse aspects of the simulation, it may be necessary to configure the graphic window in a different way. *VRS* offers four possible configurations: a single view occupying the whole graphic window, four views dividing the window into four equal parts, two views dividing the window horizontally into two equal parts, or two views dividing the window vertically into two equal parts. These options can be selected from the main menu, in the **Windows** menu, and from **Toolbar**:

Windows Menu	Toolbar
1 Window	□
4 Windows	⊞
2 Horizontal Windows	⊟
2 Vertical Windows	◫

Regardless of the configuration chosen, one of the views or areas will always be selected. The selection is shown by a white frame, and indicates which area will be subject to the changes in view we choose to effect, such as, for example, the option **Windows>>Maximize**. **Maximize** permits us to enlarge the selected window until it occupies the whole area of the graphic window. If the option is deactivated, the previous configuration is returned to.

Windows Menu	Toolbar
Maximize	⧉

Background colour

The background colour of the graphic window, which initially appears in black, can be altered, although it will always be the same for all the views or areas of the window. The command permitting this configuration is to be found in the **Windows** menu, and also in the **Toolbar:**

Windows Menu	Toolbar	Action
Background Color...	▦	Changes background colour of window

Changes background colour of window

The dynamic information of the active robot can be activated or deactivated using the option **Display>>Dynamic Info** or the equivalent button XYZ of the **Toolbar**.

Selection of the graphic window

Several of the graphic window options are applied to the selected window. To select a window it is necessary to press the right-hand button of the mouse (to cancel any active action) and then click with the left-hand button on the desired window.

Selection and alteration of viewpoints

On starting up the *VRS*, the graphic window appears divided into four areas. On loading any element in the simulator, it can be seen that the type of projection and the viewpoint vary from one view to another. Initially, the arrangement is the following:

Area	Projection	Viewpoint
Top Left	Orthogonal	Left
Top Right	Orthogonal	Frontal
Bottom Left	Orthogonal	Top
Bottom Right	Perspective	Free[5]

The projection and viewpoint can be altered from the **Display** menu and from the **Toolbar**. The projection change and predefined viewpoint change commands are enriched with options which permit the view to be adjusted according to the needs of the case in hand. These options (**Zoom**, **Scroll**, **Reference Point**, **Point of View**...), along with the projection change commands are summarised in the following table:

Display Menu	Toolbar	Action
Zoom	🔍	Brings the viewpoint nearer to the projection plane, moving the pointer from bottom to top while the left-hand button of the mouse is depressed.
Scroll	✋	Moves the picture being displayed. The picture can be moved up and down and from side to side by moving the mouse with the left-hand button depressed.
Point of View	◁	Alters the viewpoint in the selected window if this possesses a projection in perspective. Alterations in longitude are effected by moving the mouse, with the left-hand button depressed, laterally and alterations in latitude by moving the mouse vertically.
Reference Point	⊕	Alters the interest point in the selected window if this possesses a projection in perspective. The interest point is altered by moving the mouse, with the left-hand button depressed.
Display Parameters...	🎥	Permits the alteration of the viewing parameters of the selected window if this possesses a projection in perspective. By executing this option a dialogue is opened in which it is possible to alter the interest point, the viewpoint, the relation between these in polar (tecta,fi,ro) and the camera parameters (near plane, far plane and fov)
Projections>>		
Orthogonal>>Front	⧉	Shows the front view, with orthogonal projection
Orthogonal>>Back	⧉	Shows the back view, with orthogonal projection
Orthogonal>>Left	⧉	Shows the left side view, with orthogonal projection
Orthogonal>>Right	⧉	Shows the right side view, with orthogonal projection
Orthogonal>>Up	⧉	Shows the top view, with orthogonal projection
Orthogonal>>Down	⧉	Shows the bottom view, with orthogonal projection
Perspective	⧉	Shows a projection in perspective

VRS is started up by default with the solid shading mode. The simulator also has the option of displaying the loaded elements in wired mode, with hidden lines, or in rendered mode (only for robots or environmental elements containing rendered information in their definition files). The shading mode can be altered for each window independently. This is done by selecting the window for which the change of mode is desired and then selecting one of the following options, either in the Toolbar or in the **Windows** menu.

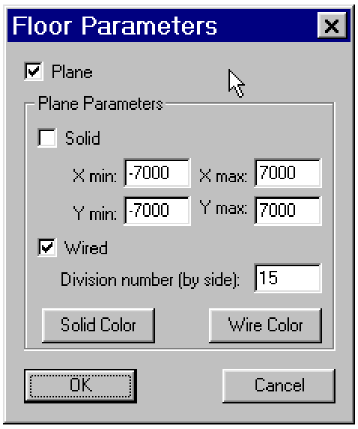
Display Menu	Toolbar	Shading Mode
Wired	⧉	Wired
Shaded	■	Shading
Hidden Lines	□	Hidden Lines
Rendered	■	Rendered

Floor display

[5] *VRS* can change the view point and reference point when a robot or environment is loaded.

The option **Display>>Floor Parameters...** (or the button of the **Toolbar**) permits the selection of the floor display mode. The control dialogue for this option is the one shown on the right, with the following options:

- With the option, **Plane,** activated, a floor will be displayed either in solid mode (**Solid** activated), wired mode (**Wired** activated) or both (both activated).
- The size of the floor is chosen with the fields, **X min**, **X max**, **Y min**, **Y max**. The number of wired divisions is chosen with the field, **Divisions number (by side):**
- With the buttons **Solid Color** and **Wire Color** the colour of the floor or of the lines in the wired mode can be chosen.



Floor Parameters

☑ Plane

Plane Parameters

□ Solid

X min: -7000 X max: 7000

Y min: -7000 Y max: 7000

☑ Wired

Division number (by side): 15

Display of coordinate systems

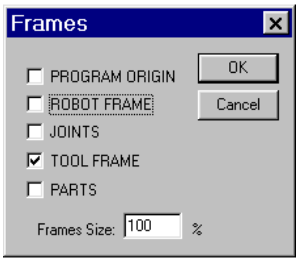
A robot possesses the following coordinate systems:

- Robot Coordinate System (**ROBOT FRAME**).
- Programme Origin Coordinate System (**PROGRAM ORIGIN**).
- A Coordinate System for each joint (**JOINTS**).
- Tool coordinate systems (**TOOL FRAME**). In the rkf robot description file there may be several different ones defined, but the one loaded as active by default is always the **Tool Frame 0**.

By default, only the tool coordinate system is displayed on screen. To see where other coordinate systems are, or to hide the tool coordinate system, there is the button of the **Toolbar** (option of the menu **Display>>Frames...**), which opens a dialogue for selecting which coordinate systems to display and changing the size of the same.

The dialogue is shown on the right. The coordinate systems are represented with the axes X,Y,Z in red, green and blue (RGB). Apart from the options listed earlier, the option **PARTS**, which will be mentioned later, permits the viewing of the operational coordinate systems for different parts of the environment.

The option **Frames Size: 100 %** is for adjusting the display size of the axes.



Frames

□ PROGRAM ORIGIN

□ ROBOT FRAME

□ JOINTS

☑ TOOL FRAME

□ PARTS

Frames Size: 100 %

Task 3.2:

- Click the right-hand button of the mouse to cancel any active option.
- Click the left-hand button on the perspective view (bottom right) to select it (the frame should be highlighted in blue).
- Maximize the perspective window ().
- Check the wired (), hidden lines () and shaded () displays.
- Activate zoom () and with the left-hand button depressed, move the mouse up and down.
- Activate viewpoint () and with the left-hand button depressed, move the mouse freely in all directions.
- Press the reference point button () and with the left-hand button depressed, move the mouse freely in all directions.
- Press the display parameters button () and introduce as reference point the coordinates where the tool coordinate system is situated (these are listed under X,Y,Z in the bottom left-hand part of the graphic window).
- Try out the zoom and viewpoint options to check that the camera is pointed at said coordinate system.
- Deactivate the floor display.
- Display the robot and tool coordinate systems at 500% and work out where they are located.
- Activate the floor display, testing how the plane is shown, changing colours and finally leaving the original configuration.

3.3. Running of an external application in *VRS*

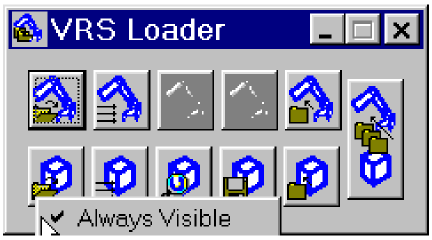
In *VRS* it is possible to run external applications generated in runnable files which communicate with *VRS* through the *VREAL* bookshop (see VREAL documents). To do so, it is necessary to run the programmes with the option **File>>Run Application**, which displays the 'open file' dialogue[6]. These applications are generated as shown in the section entitled "Programming of applications for VRS". There are basically four types of application:

- Demonstrations (**Demos**). These are applications for demonstrating industrial processes carried our in *VRS*. The runnable ones can be found in the file, **Applications\Demos**. They will normally be run from the option, **File>>Demos**.
- *VRS* Tools. These are applications which are useful for the running of *VRS* (programming unit, video recording, ...). The runnable ones can be found in the file, **Applications\VRSTools**. They will normally be run from the option, **File>>VRS Tools**.
- *VRM* Tools. These are applications which are useful for modelling of objects, parts and environments in *VRS*. The runnable ones can be found in the file, **Applications\VRMTools**. They will normally be run from the option, **File>> VRMTools.**
- User applications. These are applications designed by users, the runnable ones of which it is advisable to copy into the user's own personal folder within the file, **Applications\Users**. They should be run with the option, **File>>Run Application**.

4. *VRS Loader*

For managing the loading of robots, their spatial placing, closure or elimination from *VRS*, and for carrying out the same operations on the environment, there is a *VRS Tool* called ***VRS Loader***. ***VRS Loader*** is an external application used for loading, locating and closing both robots and the environment in which they move. As an external programme, its runnable version, **VRSLoader.exe,** can be run directly from the operative, and can be found in the folder, **Applications\VRSTools,** parting from the *VRS* directory. However, in the default *VRS* installation configuration, it is accessible through the option of the menu, **File>>VRS Tools>>VRS Loader,** and is run directly on starting up *VRS*. Once it has been run, it behaves like any other programme, with access from the Taskbar. If the ***VRS Loader*** has been run, it will open again automatically every time *VRS* is run again.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the right-hand button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.



4.1. Loading, placing and closing of robots

As mentioned previously, with the ***VRS Loader,*** robots can be loaded into the simulator, placed in space and closed. The buttons for each of these options are the following:

VRS Loader	**Action**
	Selection of an rkf file and loading of the corresponding robot
	Spatial placing of a robot
	Closure of one of the robots loaded in *VRS*

The loading of robots has been explained and used at the beginning of this section. Loaded robots are identified by their name[7] and a whole internally assigned identifier, the first robot

[6] Applications can also be started from operating system.

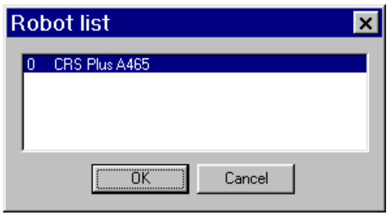
[7] This is the name as written in the field Name os section GENERAL of rkf and rgf files, not the file names.

being the number 0, and the following robots being 1,2,3… successively. When a robot is loaded and its name is already used by one of the previously loaded robots, *VRS* adds a number (1,2,3,...) to the name of the new robot. There is a limit to the number of robots which can be loaded in *VRS*, and if the limit is exceeded, an error message will be displayed.

VRS maintains a robot as the active one at all times. When a robot is loaded, this automatically becomes the active one. The name of the active robot is displayed on the Statusbar, and this robot is the one whose dynamic information is displayed.

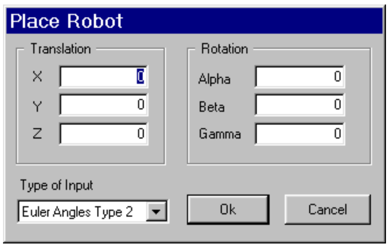
Robots are first loaded with their Robot Frame coinciding with the coordinate system of the *VRS* world. Nevertheless, this placing can be altered by using the corresponding button of the ***VRS Loader***, which, if there is any robot loaded in *VRS*, will display the following dialogues:

Robot List:
This shows a list of the robots loaded in *VRS*. The command can be cancelled with the **Cancel** button, or a robot can be selected from the list and the **OK** button clicked to locate it. If there is only one robot loaded in *VRS,* this dialogue will not be displayed, the following dialogue being displayed directly. The selected robot will be highlighted on the screen.



0 CRS Plus A465

Place Robot:
This requests a place using three transfer values (X,Y,Z) and three rotation values (alpha, beta and gamma angles). The type of input Euler angles can be selected from types 1, 2 and 3 with the "**Type of Input"** field. The placing command is made using "**OK"** or cancelled using **"Cancel"**.



X 0
Y 0
Z 0
Alpha 0
Beta 0
Gamma 0
Type of Input: Euler Angles Type 2

To close robots—that is to say, to eliminate them from *VRS*—, after selecting the corresponding button of ***VRS Loader***, the list of robots from which to select the robot to be closed will be displayed, even when there is only one robot loaded in *VRS*. To close a robot, the latter should be selected followed by "**OK",** or the command can be cancelled using "**Cancel"**.

Obviously, the robot placing or closing buttons have no effect if there is no robot loaded.

4.2. Loading, placing and closing of the environment

Robots are situated in work cells where, besides other robots (including any articulated element such as conveyer belts, turntables, etc…), there can be diverse objects which make up the environment of the robot. In *VRS* the environment of the robot is defined in an ENF file (Environment File, described in the document, VRFFD). These elements fulfil two functions: forming the environment of the robot to make the simulation more realistic, and defining the items destined to be manipulated or operated on by the robots. The inclusion of an environment model in a simulation makes *off-line* programming tasks easier.

As mentioned previously, with the ***VRS Loader,*** an environment can be loaded in the simulator, placed in space, searched for, saved and closed. The buttons for each of these options are:

VRS Loader	**Action**
	Selection of an enf file and loading of the corresponding environment
	Spatial placing of the environment
	Closure of the environment loaded in *VRS*
	Searching for an environment and loading of same in *VRS*
	Saving of environment loaded in *VRS*
	Closure of environment and all the robots loaded in *VRS*

To load an environment, the corresponding option of the ***VRS Loader*** must be called. It is only possible for one environment to be loaded in VRS, such that confirmation will be requested in the case that one already exists. From then on, the environment can be placed and closed in the same way as a robot, bearing in mind that no list of environments will be displayed, this being the only one. To place the environment, there is the dialogue, **Place Environment,** similar to **Place Robot**. To close the environment, confirmation is requested. For any of these options, all elements of the environment will be highlighted on the screen. Obviously, the environment placing and closing buttons will have no effect if no environment is loaded. The environment search button opens a new application, ***VRM Visor***, which, by looking through the directories with an explorer, permits the display of environment files to be loaded in *VRS*. The **Save Environment** button saves the environment in a file.

Finally, the last button, after requesting confirmation, closes all the robots and the environment which are loaded in *VRS*. All elements to be erased will be highlighted on the screen.

<u>Task 4.1:</u>

- Load the environment, **CRS Table.enf,** from the folder, **Models\Tutorial**.
- Move the robot, **CRS A465,** 800 millimetres in Z such that it is positioned over the table.
- By activating the necessary coordinate system displays, observe that the original programming coordinate system is still the same in relation to the robot, but that it is situated elsewhere with regard to the universal *VRS* system.
- Observe the parts operation coordinate systems by activating the option in question.

<u>Task 4.2:</u>

- Close everything loaded in *VRS*.
- Load robot, **CRS A465**.
- Load the same robot again.
- Place one of the robots so that it is facing the other but without overlapping the same.

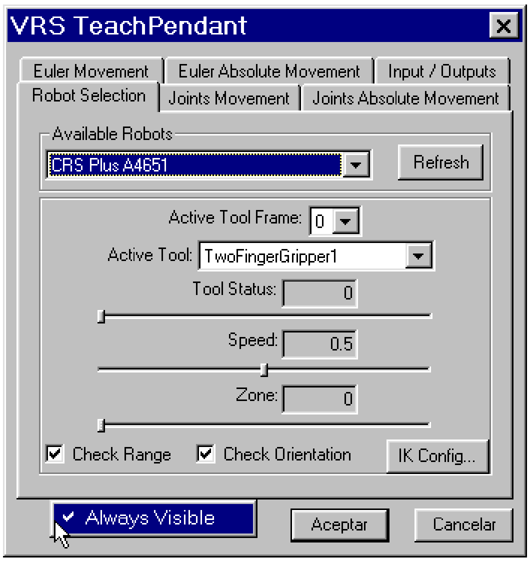
5. *VRS TeachPendant*

For managing a robot (its movement, its tool control, input and output management,…), there is a *VRS Tool* called ***VRS TeachPendant***. ***VRS TeachPendant*** is an external application used for the management of robots. As an external programme, its runnable version, **VRSTeachPendant.exe,** can be run directly from the operative, being located in the folder, **Applications\VRSTools,** parting from the *VRS* directory. Nevertheless, in the *VRS* default installation configuration, it is accessible using the option from the menu **File>>VRS Tools>>*VRS Teach Pendant***, although it does not run automatically on starting up *VRS*. Once it has been run, it behaves like any other programme, with access from the Taskbar.

Note: this application cannot be run if no robot is loaded.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the <u>right-hand</u> button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.

The application has six dialogues organized by tabs, an “accept” button and a “cancel” button. To move from one dialogue to another, it is necessary to click on the tab at the top with the name of the desired dialogue.



Available Robots: CRS Plus A4651

Active Tool Frame: 0

Active Tool: TwoFingerGripper1

Tool Status: 0

Speed: 0.5

Zone: 0

☑ Check Range ☑ Check Orientation

Unlike the other *VRS Tools*, ***VRS TeachPendant*** can be opened more than once in order to control several different robots simultaneously.

5.1. *Robot Selection* Dialogue

With the ***Robot Selection*** dialogue, it is possible to select one of the available robots and configure its status. The available fields are listed below:

- **Available Robots**: shows the robot which will be managed by ***VRS TeachPendant***, allowing for the selection of another from the list of robots loaded in *VRS*.
- **Refresh**: updates the list of robots. If new robots are loaded with another application, for example, ***VRS Loader***, it is necessary to update the list of available robots in ***VRS TeachPendant*** with the **Refresh** button. The updating of the list takes place automatically on passing to the **Robot Selection** dialogue from another dialogue, it being unnecessary in this case to use the **Refresh** button.
- **Active Tool Frame**: shows the active tool coordinate system, allowing for the selection of another from the list of Tool Frames defined for the robot.
- **Active Tool**: shows the active tool, allowing for the selection of another from the list of tools defined for the robot.
- **Tool Status**: shows the tool status value, allowing this to be changed between 0.0 y 1.0 with a horizontal movement bar (0 on the left and 1 on the right).
- **Speed**: shows the nominal speed value for robot movements, allowing this to be changed between 0.0 y 1.0 with a horizontal movement bar (0 on the left and 1 on the right).
- **Zone**: shows the zone parameter value (also called “fly” or “precision”) of robot movements, allowing this to be changed between 0.0 y 1.0 with a horizontal movement bar (0 on the left and 1 on the right). This field will not affect robot movements with the ***VRS Teach Pendant***, only having effect in the case of consecutive movements.
- **Check Range**: for activating or deactivating the range check of the robot. When it is activated, the robot will be unable to move outside the joint ranges defined in its rkf file. When it is deactivated, the robot will be able to move outside the joint ranges defined in its rkf file.
- **Check Orientation**: for activating or deactivating the orientation check of the robot. When it is activated, the robot will be unable to move in places with unreachable orientation. When it is deactivated, the robot will be able to move in places with unreachable orientation, taking notice only of the position of the placement.
- **IK Config...**: Opens a new dialogue to configure the desired configuration in inverse kinematics resolution.

The last five fields are useful when it comes to performing robot movements with the ***VRS TeachPendant***, because they are movement configuration options, for which reason they will be mentioned in more detail in later sections.

Tool control

Robots have tools attached with which to operate. The tools can have their own movements. In *VRS* it is necessary to model the different phases with which one wishes to represent the status[8] of the tool in the rgf file. After that, it will be possible to control the status of the tool by means of a numeric value from 0.0 to 1.0, in such a way that it interpolates between the possible phases of the tool. That is to say, if the tool has been modelled with two phases, when the status of the tool is less than 0.5, it will be represented with the first phase, whereas when it is equal to or more than 0.5, it will be represented with the second phase. The same concept is applied when there are more phases modelled.

A robot can have more than one tool, attached to an adapter. To select the active tool, there is the **Active Tool** field, with which it is possible to select one of the available tools (the option is not available if the robot has no tool). To change the tool status, there is the **Tool Status** field, with a movement bar which permits the introduction of values between 0.0 and 1.0. The active

[8] The difference between state and status comes because state is a describe value represented with an integer number while status represents a continuous value with a float number.

tool and the tool status are displayed in the dynamic information (bottom left of the graphic window) with the identifiers **T** and **TS** respectively.

Another important aspect to control is the Tool Frame, which is used to command movements or obtain the configuration of the robot. The place in which a robot configuration is found is defined by the placing of a system of coordinates associated with the tool with regard to the original robot programming system. In a robot rkf file, different coordinate systems can be defined. The **Active Tool Frame** field is used to change between those available for the robot.

In order to design robot movements correctly, especially movements in Cartesian space, it is essential to know how the tool frames are defined. Robots normally have the Z axis of the tool pointing “away” from the same, but in the **CRS A465** this is not the case, the X axis being the one which points this way.

Task 5.1:

- With robot **CRS A465** loaded, run ***VRS TeachPendant***.
- Change the tool status between the upper and lower values.
- Work out how many phases there are modelled.
- Activate each of the tool frames available for the robot and observe their location. Observe that the place shown in the dynamic information of the robot changes position in X on changing the tool frame.
- Leave tool frame 1 activated.

Task 5.2:

- Load robot **ABB IRB L6,** available in **Models\Tutorial** and locate it with Y=1000.
- Perform a **Refresh** in ***VRS TeachPendant*** and select robot **ABB IRB L6**.
- Work out the existing tools and their respective tool frames.
- Close robot **ABB IRB L6** and perform a **Refresh** in ***VRS TeachPendant***.

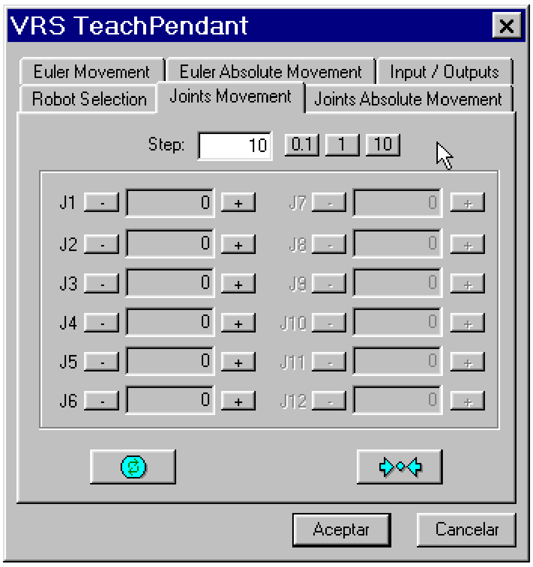
5.2. *Joint Motion* Dialogue

In robot joint movement, the value of robot joint variables is controlled directly. The robot’s joint variables are shown at all times in the dynamic information (bottom left of the graphic window) for the active robot.

By clicking on the ***Joint Motion*** tab, a dialogue is opened which allows the robot to be moved by joints by clicking the mouse on specific buttons. The dialogue is described below.

In fields **J1** to **J12** the present value of the robot’s joint variables is shown, there being as many active buttons as the robot has joints. To move a joint, it is necessary to click on the button - or + (variations in a negative or positive direction respectively). In said articulation, the increase specified in the field, **Step:,** will take place, this being alterable directly or by using the buttons **0.1**, **1** or **10**. Note that if the movement buttons are held down, the robot will continue to move, which may mean, depending on the sensitivity of the mouse, that more than one movement occurs when the button is clicked on.

There are also the buttons ⟳ and ⇔, which take the robot to its *reset* or synchronism configuration respectively.



Step: 10

J1 0 J2 0 J3 0 J4 0 J5 0 J6 0

J7 0 J8 0 J9 0 J10 0 J11 0 J12 0

If, during the movement of a joint, it goes beyond its work range, the screen will flash and the robot will be unable to move further in that direction. The joint ranges are defined in the rkf file.

It is possible to deactivate the movement range control of the robot, so that it can be moved to positions which are unreachable or even impossible (due to collisions) for the real robot. To deactivate the range control, there is the **Check Range** field of the ***Robot Selection*** dialogue, which was mentioned earlier. The range control should always be activated except for carrying out a test of movement beyond the range of the robot. If the robot is taken out of range and the range control activated, the robot may be blocked, in which case it will only be possible to take it to reset or synchronism.

Another important aspect is the speed at which the movements of the robot are performed. With the movement bar, **Speed,** of the dialogue, ***Robot Selection,*** it is possible to adjust the speed of robot movement. When the value is 1.0, the robot will move at the maximum speed permitted by the simulation on the PC. If this value is reduced, on the other hand, the movement speed of the simulation will be slowed down[9]. Movement speed is mainly noticeable in rectilinear movements.

There are two buttons for taking the robot to its *reset* configuration (all joint variables set to zero) and to its synchronism configuration (defined in the rkf file). Note that the *reset* configuration can be out of range, whereas the synchronism configuration must be within range, because otherwise an error will be caused by the loading of the robot. A robot can be defined with both configurations the same or different; that is, all the variables of the synchronism configuration do not have to be at zero.

Task 5.3:

- Select robot **CRS A465**.
- Move all the joints of the robot.
- Try out the movement at speed 0.5 and speed 1.0, leaving it at the latter.
- Try out different values in the **Step** field.
- Work out the ranges of each joint by taking it to its extremes.
- Deactivate the range control and move a joint beyond its range.
- Activate the range control and take the robot to its synchronism configuration.
- Try out the synchronism and reset configurations to see whether or not they are identical.
- Take the robot to its synchronism configuration.

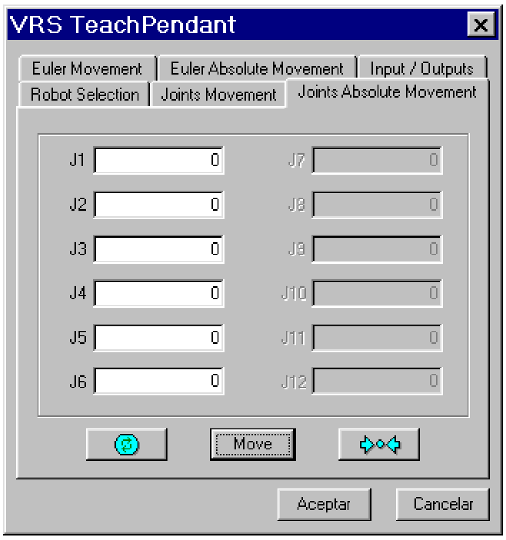
Task 5.4:

- Read the section corresponding to the ***VRS Trace Control*** to find out how the tracing of robots is controlled.
- Move robot **CRS A465,** displaying the traces and trying out the ***VRS Trace Control*** options for activating, hiding and deleting the traces (and the reverse). In the remaining tasks, these options can be used to be more clear about the movements of the robot.
- Take the robot to its synchronism configuration.

[9] For very slow velocities, a large motion can be so long that a Timeout error is shot, giving that ***VRS Teach Pendant*** is closed with an error message. Please, modify the value of *Timeout* as explained in Appendix A.

5.3. *Joint Absolute Motion* Dialogue

The robot can also be moved to a specific configuration by means of the ***Joint Absolute Motion*** dialogue, which is shown on the right.



J1: 0
J2: 0
J3: 0
J4: 0
J5: 0
J6: 0
J7: 0
J8: 0
J9: 0
J10: 0
J11: 0
J12: 0

This dialogue will have the same number of active fields as the robot has joints, numbered from **J1** to **J12**. to move the robot, the configuration is introduced by means of the desired value of its joint variables, by using "enter" or the **Move** button. The buttons ⊙ and ⇨◇⇦ have the same meaning as in the previous dialogue.

If one of the joints is out of range when a configuration is entered, the screen will flash and the robot will fail to move, the order being cancelled.

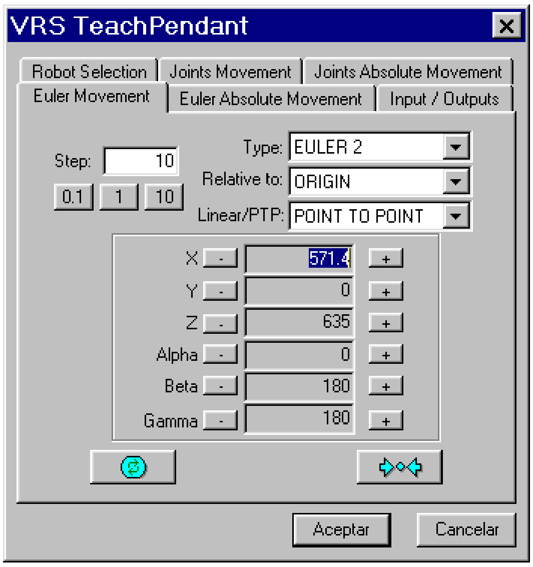
Task 5.5:
- Move robot **CRS A465,** specifying various configurations.
- Take the robot to its synchronism configuration.

5.4. *Cartesian Motion* Dialogue

In the movement of robots in Cartesian Space, active tool frame placement with regard to the original robot programming coordinate system is controlled directly. The placement is represented by position X,Y,Z and three type 2 Euler angles (Alpha,Beta,Gamma). This placement is shown at all times in the dynamic information (bottom left of the graphic window) for the active robot.

To perform these movements, there is the ***Cartesian Motion*** dialogue, which is explained below.

In this dialogue, the placement of the active **Tool Frame** is shown with regard to the original robot programming system. This placement can be changed using the buttons - or + (variations in a negative or positive direction respectively) for each of the six variables specified by the placement (X,Y,Z,Alpha,Beta,Gamma). In said variable, the increase specified in the field, **Step:,** will take place, this being alterable directly or by using the buttons **0.1**, **1** or **10**. Note that if the movement buttons are held down, the robot will not move until the button is released. On the other hand, if the button is pressed several times, the movements will take place consecutively. The buttons ⊙ and ⇨◇⇦ have the same meaning as in previous dialogues.



Step: 10
Type: EULER 2
Relative to: ORIGIN
Linear/PTP: POINT TO POINT
X 571.4
Y 0
Z 635
Alpha 0
Beta 180
Gamma 180

If, during a movement, the place to which one wishes to take the robot is unreachable or a joint moves beyond its work range, the screen will flash and the robot will fail to move.

Note that the place shown and the movements are dependent on the the tool which is active in the **Active Tool Frame** field, as explained previously in the ***Robot Selection*** dialogue. There are other fields in this dialogue which also influence the movements of the robot, such as the aforementioned **Check Range**, which allows the robot to be moved to positions in which its joints are out of range, and **Speed,** which permits the adjustment of the movement speed of the robot.

In the **Type** field**:** it is possible to select the representation of placement in the three existing types of Euler angles. Type 2 is the one which is used by default, being the most common type in robotics, although type 3 is particularly useful for carrying out rotations with regard to the axes of the active **Tool Frame**. In the following table, the meaning of the three types of orientation by means of Euler angles are set out.

Euler angles	Alpha	Beta	Gamma
Type 1	Axis Z (Origin Frame)	Axis U (Tool Frame)	Axis W (Tool Frame)
Type 2	Axis Z (Origin Frame)	Axis V (Tool Frame)	Axis W (Tool Frame)
Type 3	Axis X (Origin Frame)	Axis Y (Origin Frame)	Axis Z (Origin Frame)

Task 5.6:

- Move the position and orientation of robot **CRS A465** with different values of **Step**.
- Move the orientation of the robot with representation in different types of Euler angle, and understand the difference.
- Take the robot to its synchronism configuration.

Point to point and linear movements

Robots can be moved with a point to point type movement, in which the robot generates a movement by interpolating points in the joint space, or with linear movement, in which the origin of the active tool frame follows a straight line. The movement type can be selected with the **POINT TO POINT** and **LINEAR** options accessible in the **Linear/PTP** field.

Task 5.7:

- Move robot **CRS A465,** in X,Y,Z position, with different values of **Step,** trying out point to point and linear movements, and understand their difference.
- Take the robot to its synchronism configuration and leave the point to point mode selected.

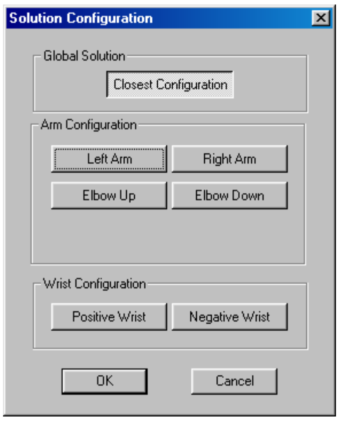
The movements of the robot can be specified in relation to three possible coordinate systems: the original programming system (**ORIGIN**), the tool frame (**TOOL FRAME**) and the world coordinate system (**WORLD**). These options can be selected in the "**Relative to"** field.

Task 5.8:

- Place robot **CRS A465** with Gamma value at 45º.
- Display the robot origin and world coordinate systems.
- Move the robot in relation to the different coordinate systems and understand the difference.
- Take the robot to the synchronism configuration and leave the movements with regard to origin selected.
- Place the robot with Gamma value at 0º.

Selection of the solution in inverse kinematics

When the robot moves in Cartesian space, the simulator selects from among the possible configurations of the robot which resolve inverse kinematics. The closest configuration will normally be chosen, because it saves energy, but a different configuration can be selected with the **IK Config** field**...** of the ***Robot Selection*** dialogue, which opens the following dialogue.



The **Closest Configuration** option resolves inverse kinematics with the closest configuration to the robot's previous configuration. On the other hand, different configurations can be selected (depending on the type of robot), by selecting the desired options from among the following pairs:

- **Left Arm/Right Arm**.
- **Elbow Up/Elbow Down**.
- **Positive Wrist/Negative Wrist**.

When selecting a specific configuration, movement will only be possible if said configuration is correct, other possible solutions being rejected. On the other hand, with the closest configuration, the best possible solution is considered, for which reason this option should normally be maintained.

Task 5.9:

- With **IK Config...** select the **Elbow Down** option for robot **CRS A465**.
- Move the robot in direction X. Observe the new configuration selected.
- Select the **Left Arm** option.
- Move the robot in direction X. Observe the new configuration selected.
- Select the **Positive Wrist** option.
- Move the robot in direction X. Observe the new configuration selected.
- Select the **Closest Configuration** mode.
- Move the robot in direction X. Observe the new configuration selected.
- Take the robot to its synchronism configuration.

Orientation control

When a robot is sent to a place, it will reach there in both position and orientation. For certain applications, it may be useful to order movements only in position, without forcing the robot to follow the orientation. This can be done by deactivating the orientation control with the **Check Orientation** field of the ***Robot Selection*** dialogue. The orientation control should always be activated except to perform a robot movement test. This aspect is especially useful in order to forget about orientation in robots with less than six degrees of freedom. If the robot moves without orientation control and the latter is then activated, the robot may become blocked, in which case it will only be possible to take it to synchronism or reset.

Taks 5.10:

- Deactivate the **Check Orientation** field for robot **CRS A465**.
- Move the robot in directions Y and Z. Observe the orientation of the tool system.
- Try to move the orientation of the robot by changing the angles. Observe how no movement takes place
- Activate the **Check Orientation** field.
- Take the robot to its synchronism configuration.

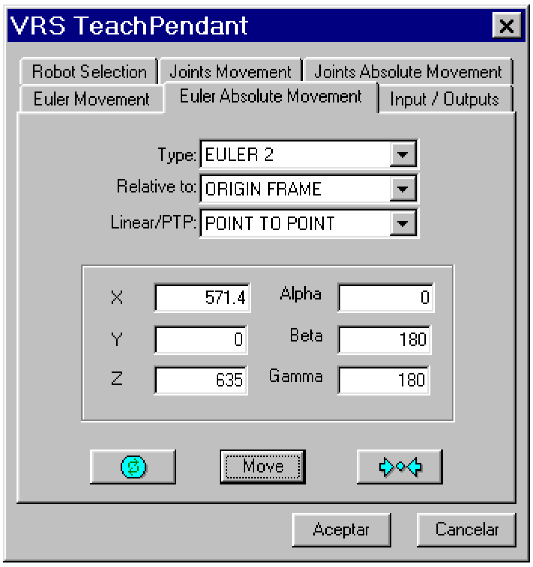
5.5. *Cartesian Absolute Motion* Dialogue

The robot can also be moved to a specific place by means of the ***Cartesian Absolute Motion*** dialogue, which is shown on the right.

This dialogue has the X,Y,Z fields to specify position and the Alpha, Beta, Gamma fields o specify orientation. To move the robot, the configuration is introduced by means of the desired position and orientation values, using “enter” or the **Move** button.

The ◎ and ⇨•⇦ buttons have the same meaning as in the previous dialogue.

The **Type**, **Relative to** and **Linear/PTP** fields have the same meanings as in the previous dialogue.



Type: EULER 2
Relative to: ORIGIN FRAME
Linear/PTP: POINT TO POINT

X	571.4	Alpha	0
Y	0	Beta	180
Z	635	Gamma	180

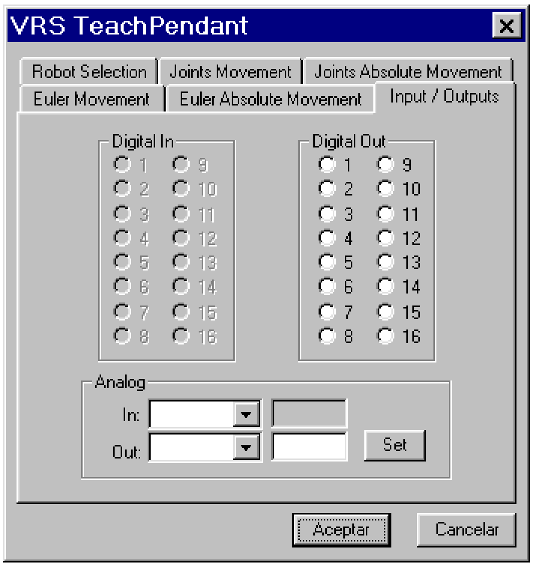
If, on selecting a place, the place to which one wishes to take the robot is unreachable or a joint moves beyond its work range, the screen will flash and the robot will fail to move, the order being cancelled.

Task 5.11:
- Move robot **CRS A465,** specifying different places.
- Take the robot to its synchronism configuration.

5.6. *Input / Outputs* Dialogue

In *VRS,* each robot loaded is assigned 16 digital inputs, 16 digital outputs, 16 analogical inputs and 16 analogical outputs. These signals can be connected from robot to robot by means of the *VRS Tool,* ***VRS IOConnection,*** which will be seen later. Access to these signals (reading the inputs and changing the value of the outputs) is attained through the ***Inputs/Outputs*** dialogue.

The digital inputs are numbered from 1 to 16 and grouped in the **Digital In** field. These inputs cannot be acted on directly, but the programme will continuously update their status (checking their values every half second), a mark being shown when they are at a high level (logical 1).
The digital outputs are numbered from 1 to 16 and grouped in the **Digital Out** field. These outputs can be acted on directly, by clicking on the corresponding circle, at which the programme will display their status, a mark being shown when they are at a high level (logical 1).
The analogical inputs and outputs (both nimbered from 1 to 16) are grouped in the **Analogical** field, the inputs in the **In** field and the outputs in the **Out** field.

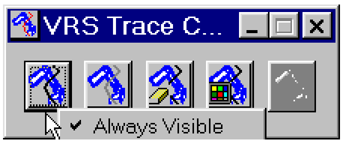


On selecting an analogical input from the inputs list, the value read will be displayed to its right, being 0.0 if it is not connected. On selecting an analogical output from the outputs list, the present valued will be displayed to its right, it being possible to enter a different value, which will be sent via the corresponding output on pressing **Set**.

6. *VRS TraceControl*

For managing robot traces (activation, hiding and elimination) there is the *VRS Tool* called ***VRS TraceControl***. ***VRS TraceControl*** is an external application used to control robot traces. As an external programme, its runnable version, **VRSTraceControl.exe,** can be run directly from the operative, being located in the folder, **Applications\VRSTools,** parting from the *VRS* directory. Nevertheless, in the *VRS* default installation configuration, it is accessible using the option from the menu **File>>VRS Tools>>*VRS* Trace Control**, although it does not run automatically on starting up *VRS*. Once it has been run, it behaves like any other programme, with access from the Taskbar. If the ***VRS TraceControl*** has been run, it will open again automatically every time *VRS* is run again.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the right-hand button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.



✓ Always Visible

As mentioned earlier, with ***VRS TraceControl*** robot traces in the simulator can be controlled. A robot's trace is the trace left by the robot's active tool frame origin when it moves. Traces must be activated to be drawn, and they can be hidden or eliminated. The buttons for each of these options are the following:

VRS TraceControl	**Action**
	Changes the traces of all robots loaded in *VRS* activate to deactivate.
	Changes the traces of all robots loaded in *VRS* hide to display.
	Eliminates the traces of all robots loaded in *VRS*.
	Permits the choice of colour of the traces of all robots loaded in *VRS*.
	Saves the traces of robots (option unavailable).

Note that any of the buttons affects all the robots loaded in *VRS*, but only makes sense for those robots already loaded, not for those loaded subsequently. On the other hand, if another application also handles the trace of a robot, it is possible that the information is not consistent between the two applications.

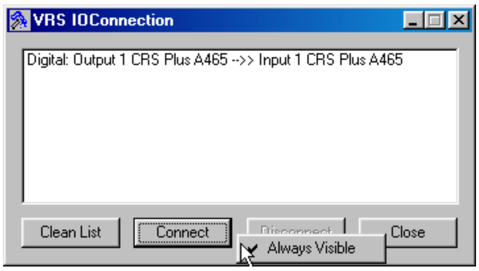
It is important to clarify certain aspects regarding these options:

- On deactivating the trace of a robot, a trace segment is closed, whereas on activating the trace, a new trace segment is opened. Therefore, the trace will be the union of segments generated on changing activated and deactivated trace.
- If the active robot coordinate system is changed with the active trace, a new trace segment is started.
- The trace can be activated and deactivated so that only the movements made while the trace is active are represented.
- Hiding the trace neither eliminates it nor deactivates it, it only makes it invisible on the screen. In other words, if the trace is active and hidden (both buttons activated) the points through which the the tool frame origin pass will be saved despite their not being displayed. As soon as the hide trace option is deactivated, the whole trace generated while the trace was active will be displayed.
- Eliminating the trace of a robot deactivates the same.

7. *VRS IOConnection*

For connecting the outputs of one robot to the inputs of another robot in *VRS,* there is the *VRS Tool* called ***VRS IOConnection***. ***VRS IOConnection*** is an external application used for connecting the (digital and analogical) outputs of one robot to the inputs (digital and analogical respectively) of another robot. As an external programme, its runnable version, **VRSIOConnection.exe,** can be run directly from the operative, being located in the folder, **Applications\VRSTools,** parting from the *VRS* directory. Nevertheless, in the *VRS* default installation configuration, it is accessible using the option from the menu **File>>VRS Tools>> VRS Input/Output Connection**, although it does not run automatically on starting up *VRS*. Once it has been run, it behaves like any other programme, with access from the Taskbar. If the ***VRS IOConnection*** has been run, it will open again automatically every time *VRS* is run again.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the right-hand button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.



Digital: Output 1 CRS Plus A465 -->> Input 1 CRS Plus A465

As mentioned before, with ***VRS IOConnection*** the outputs of one robot can be connected to the inputs of another robot, in such a way that the ouputs are attached to the corresponding inputs. Obviously, the connection will always be digital output to digital input or analogical output to analogical input.

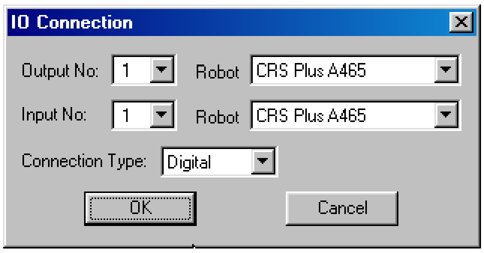
A single output of a robot can be connected to several inputs of different robots, or even to its own inputs; but a single input can only be connected to a single output. Any new connection to an input substitutes the previous connection to said input.

The fields permitting connections are the following:

VRS IOConnection	**Action**
Clean List	Clears the list of connections, although the connections themselves remain.
Connect	Connects signals according to the **IO Connection** dialogue.
Disconnect	Disconnects signals (not available)
Close	Closes the window

It is important to emphasize that the first option only clears the *list* of connections, it being necessary to close a robot in *VRS* in order for all its actual connections to disappear. Witht the **Connect** button, the dialogue permitting the making of connections is entered, as shown below.

With the **Output No.** and **Input No.** fields, the output and input numbers are selected, to be applied to the robot selected in the **Robot** fields to their right.
With the **Connection Type** field, the type of connection can be selected (digital or analogical).
With **OK** or **Cancel** the connection is validated or cancelled.



Output No: 1 Robot CRS Plus A465
Input No: 1 Robot CRS Plus A465
Connection Type: Digital

When a connection is made, this will be displayed in the list of established connections in the **VRS IOConnection** dialogue. It should be remembered that if more than one connection is made to the same input, the new connection replaces the previous one. The list is ordered chronologically, such that when there is more than one connection to the same input, the lowest one on the list is that which remains in place. However, the list only shows the connections made in the present application, not those made in previous applications.

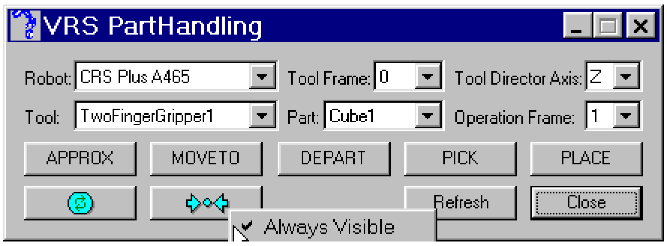
Task 7.1:
- Connect digital output 1 of robot **CRS A465** to its digital input 1.
- Check the connection using the ***VRS TeachPendant.***
- Connect analogical output 1 of robot **CRS A465** to its analogical input 1.
- Check the connection using the ***VRS TeachPendant.***

Task 7.2:
- Connect signals between two different robots loaded in *VRS*, and check the connections.

8. *VRS PartHandling*

For connecting the outputs of one robot to the inputs of another robot in *VRS,* there is the *VRS Tool* called ***VRS IOConnection***. ***VRS IOConnection*** is an external application used for connecting the (digital and analogical) outputs of one robot to the inputs (digital and analogical respectively) of another robot. As an external programme, its runnable version, **VRSIOConnection.exe,** can be run directly from the operative, being located in the folder, **Applications\VRSTools,** parting from the *VRS* directory. Nevertheless, in the *VRS* default installation configuration, it is accessible using the option from the menu **File>>VRS Tools>> VRS Input/Output Connection**, although it does not run automatically on starting up *VRS.* Once it has been run, it behaves like any other programme, with access from the Taskbar. If the ***VRS IOConnection*** has been run, it will open again automatically every time *VRS* is run again.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the <u>right-hand</u> button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.



Robot: CRS Plus A465
Tool Frame: 0
Tool Director Axis: Z
Tool: TwoFingerGripper1
Part: Cube1
Operation Frame: 1

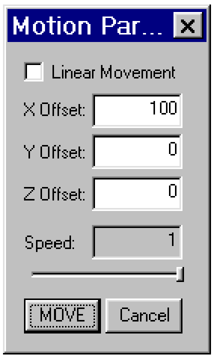
As mentioned before, with ***VRS PartHandling*** it is possible to handle parts from the environment with robots. For this purpose there are the following options:

VRS PartHandling	**Action**
Robot	Selection of robots.
Tool Frame	Selection of active tool frame.
Tool Director Axis	Selection of tool director axis for movements of approach and withdrawal.
Tool	Selection of active tool.
Part	Selection of part to be handled.
Operation Frame	Selection of the operation frame of the part for grasping and relative movements.
APPROX	For approaching the operation frame of a part.
MOVETO	For moving to the operation frame of a part.
DEPART	For making a movement of withdrawal.
PICK	For picking a part up.
PLACE	For putting a part down.
⟳	Moves a robot to its reset configuration.
⇨○⇦	Moves a robot to its synchronism configuration.
Refresh	Updates information about robots, tools and parts.
Close	Closes the window.

The **Robot**, **Tool Frame** and **Tool** fields, and the *reset* and synchronism buttons, are similar to those already described for other dialogues. And the meaning of the last two buttons is obvious. The **Tool Director Axis** field is used to select which active tool frame axis is to be used as the director for the movements of approach and withdrawal of the robot. The **Part** field is used to specify which part is to be picked up or put down, or over which relative movements are to be performed, depending on the operation frame selected in the **Operation Frame** field.

The **APPROX**, **MOVETO** and **DEPART** buttons open the **Motion Parameters** dialogue shown below. The significance of the actions and fields of this new dialogue are the following:

- The **APPROX** button is used to make an approach movement towards the operation frame of a part. The **Linear Movement** field can be used to specify that the robot's movement should be in a straight line (this is not the case by default). The speed of movement of the robot can also be specified using the movement bar of the **Speed** field (the default speed is 1.0). The **X Offset**, **Y Offset**, and **Z Offset** values can be used to alter the approach distances (by default these will be 100 in the tool director axis, and 0 in the rest). Note: these values indicate distances in the <u>**NEGATIVE**</u> directions of the axes of the part operation frame used as the basis for movements. **MOVE** is used to make the movement and **Cancel** to cancel it.


Motion Par...

☐ Linear Movement

X Offset: 100

Y Offset: 0

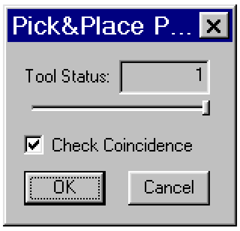
Z Offset: 0

Speed: 1

- The **MOVETO** button is used to make a movement *to* the operation frame of a part. The **Linear Movement** field can be used to specify that the robot's movement should be in a straight line (this *is* the case by default). The speed of movement of the robot can also be specified using the movement bar of the **Speed** field (the default speed is 0.5). The **X Offset**, **Y Offset** and **Z Offset** fields are deactivated because they must always have a value of 0 (otherwise it would be an approach). **MOVE** is used to make the movement and **Cancel** to cancel it.
- The **DEPART** button is used to make a withdrawal from the place in which the robot is situated. The **Linear Movement** field can be used to specify that the robot's movement should be in a straight line (this *is* the case by default). The speed of movement of the robot can also be specified using the movement bar of the **Speed** field (the default speed is 0.5). The **X Offset**, **Y Offset**, and **Z Offset** values can be used to alter the withdrawal distances (by default these will be 100 in the tool director axis, and 0 in the rest). Note: these values indicate distances in the <u>**NEGATIVE**</u> directions of the axes of the part operation frame used as the basis for movements. **MOVE** is used to make the movement and **Cancel** to cancel it.

The **PICK** and **PLACE** buttons open the **Pick&Place Parameters** dialogue shown below. The significance of the actions and fields of this new dialogue are the following:

- The **PICK** button is used to pick up a part. After picking up the part, the tool will change its status to the value specified in the **Tool Status** field. This value can be altered using the movement bar (the default value is 1). The robot will hold the part in its active pincer and carry it in successive movements. **OK** is used to make the pick-up and **Cancel** to cancel it.


Pick&Place P...

Tool Status: 1

☑ Check Coincidence

  The **Check Coincidence** field can be used to specify that, before the part is picked up, the coincidence should be checked between the placement of the active tool frame and that of the operation frame of the part, such that the part is only picked up if the coincidence exists (by default the check is made).
- The **PLACE** button is used to put a part down. After putting down the part, the tool will change its status to the value specified in the **Tool Status** field. This value can be altered using the movement bar (the default value is 0). The part will become detached from the active pincer of the robot, which will no longer carry it with it in successive movements. **OK** is used to put the part down and **Cancel** to cancel the operation. The **Check Coincidence** field is deactivated because it is of no sense in this case.
  It is important to point out that this action affects the part which is selected in the **Part** field, such that if the robot is not holding the part in question, this will cause an error to occur.

<u>**Task 8.1:**</u>

- Load robot **CRS A465** and its table as in Task 4.1 and open **VRS PartHandling**.
- Select the tool frame for tool 1 and axis X as the director axis.
- Select part, Cube1, and make an approach movement towards the part using **APPROX** and a movement *to* the piece using **MOVETO** (with the default values).
- Pick up the part and withdraw the robot (use 0.4 for the **Tool Status**).
- Select the pallet as the part and its operation system 2.
- Find out what the **APPROX** and **MOVETO** buttons do now.

- Select the part, Cube1, put it down (with **Tool Status** at 0) and withdraw the robot using **DEPART**.

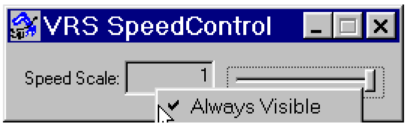
Task 8.2:

- Move the parts, Cube2 and Cube3, to the places specified in pallet operation systems 1 and 3 respectively.
- Do not forget to alternate the **Part** field between the pallet and the corresponding cube when making movements and putting down parts.
- Follow the logical sequence: **APPROX-MOVETO-PICK/PLACE-DEPART**.

9. *VRS SpeedControl*

For adjusting robot simulation speed in *VRS,* there is the *VRS Tool* called ***VRS SpeedControl***. ***VRS SpeedControl*** is an application used to adjust robot simulation speed.
As an external programme, its runnable version, **VRSSpeedControl.exe,** can be run directly from the operative, being located in the folder, **Applications\VRSTools,** parting from the *VRS* directory. Nevertheless, in the *VRS* default installation configuration, it is accessible using the option from the menu **File>>VRS Tools>> VRS Speed Control**, although it does not run automatically on starting up *VRS*. Once it has been run, it behaves like any other programme, with access from the Taskbar. If the ***VRS SpeedControl*** has been run, it will open again automatically every time *VRS* is run again.

Its interface is shown on the right, and like all *VRS Tool* type applications, it possesses its own *pop-up* menu, accessible by clicking the right-hand button of the mouse over the window (outside the buttons). If the option, **Always Visible** (which appears by default)**,** is activated, the application will always be visible over any other window, including the *VRS* window.



As mentioned before, ***VRS SpeedControl*** can be used to adjust robot simulation speed in *VRS*. The robots will move at each movement command according to their specific speed parameter. However, there is a speed scale which can be used to reduce this speed. This speed scale, which is controlled in this VRS Tool using the movement bar of the **Speed Scale** field, can be used to reduce the speed: for the value 1.0 there is no reduction, but for lower values the simulation speed is reduced. The window updates the speed scale value periodically, such that if it is altered by another application, this will be detected by ***VRS SpeedControl***.

10. Demonstration Applications

The programme has several demonstration applications which can be run using the option, **File>>Demos>>Industrial Robots**. These applications have been designed using external programmes, as shown in the section entitled, “Programming of applications for *VRS*”. In the **Videos\Demos** folder, there are videos of the demonstrations (if these have been installed).

Task 11.1:

- Run the demonstrations of applications with industrial robots available in the option, **File>>Demos>>Industrial Robots**.

A. *VRS* configuration

Configuration of the *Timeout*

The VREAL.INI file, installed in the operative directory, is used to configure the Timeout of the communication between *VRS* and external applications. The content of the file is the following (although the field values may be different):

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

The **TimeOut** field is the *timeout* in seconds for communication with the external application.
The LoadTimeOut field is the *timeout* in seconds for communication with the external application for the file loading instructions. This *timeout* will normally be greater than the previous one.

Configuration of the *VRS Tools* Menu

The *VRS Tools* can be configured using the **VRS.INI** file which is installed in the operative.
The content of the **VRS.INI** file is of the following style (although the number and value of fields can be different):

```
[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/VRSIOConnection.exe
InitRun = NO
; Max number of tool menu entries is 10
```

- The **Version** field of the **VERSION** section will be the one selected in the installation process. **The content of this field should not be altered.**
- The **DllName** field of the **IKLIB** section is the *dll* file used by default in the kinematics of the robot. **The content of this field should not be altered** unless a new kinematics bookshop has been compiled.
- The **Enabled** field of the **ROBOTMENU** section should be set to **YES** if the programme carries out server functions for a robot which is communicated with the programme itself. Otherwise, **the content of this field should not be altered**.
- The **ComputeEnvelopes** and **DrawEnvelopes** fields of the **ENVELOPES** section should be set to **YES** if advanced collision detection or distance sensor simulation applications are used in the programme. Otherwise, **the content of this field should not be altered**.
- The **Name** field of the **TOOL_MENU** section will contain the name of the File menu option containing the *VRS Tools*.
- For each **TOOL_MENU_ENTRYi** section (where i is an ordered integer) the following values can be specified for each *VRS Tool* installed:
  - **Name**: the name which appears on the menu.
  - **Path**: name of the runnable with the *path* parting from the *VRS Path*.
  - **InitRun**: set to **YES,** it will run automatically on starting up *VRS*, set to **NO**, it will not run automatically, the user having to launch it him or herself.