

TP Stäubli Robotics Suite 2022 - SRS 2022 / robot TX60 / contrôleur Cs8

Initiation à travers la réalisation d'un premier programme

Jean-Louis Boimond
Université Angers

Stäubli Robotics Suite (SRS), ici dans sa version 2022.5.0, est un logiciel Stäubli fonctionnant sous Windows dédié au développement et au débogage de *systèmes robotisés* (on parlera plutôt de *cellules* par la suite) dans un environnement 3D. Un tel logiciel peut se connecter à un contrôleur de robot, voir la figure qui suit, par exemple, pour transférer un programme, une fois conçu et validé dans l'environnement de SRS, vers le contrôleur de robot.

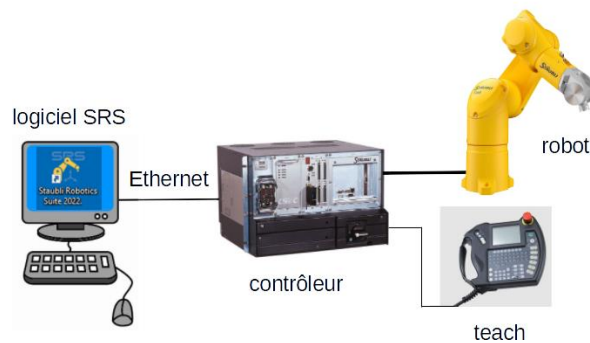


Figure 1 : Schéma simplifié du dispositif logiciel SRS/contrôleur/robot Stäubli.

Objectif visé

Après avoir brièvement introduit le logiciel SRS au §1, nous allons, à titre d'initiation, utiliser ce logiciel pour réaliser un programme en charge de déplacer une pièce au sein d'une cellule en 3D composée : d'un **robot TX60** muni d'une **ventouse** ; d'un **contrôleur Cs8** ; d'un **support** (représentant une plaque métallique) sur lequel vont reposer la **pièce à déplacer** (dans notre cas, un simple cube) et un **cylindre** sur lequel la pièce va être déposée (voir les figures 7 ou 14 à titre d'illustration). Il va s'agir pour cela :

- de modéliser la cellule en question à partir de composants mis à disposition par le logiciel, voir §2,
- d'écrire un programme (en VAL 3, le langage de programmation développé par Stäubli) permettant la réalisation de la trajectoire du robot ainsi que l'exécution des tâches de prise et de dépôt de la pièce en certains points de la trajectoire, voir §3. Une simulation du fonctionnement de la cellule en réponse au programme VAL 3 permettra sa validation, une vidéo montrant le résultat auquel vous devriez aboutir est accessible *via* le lien :

<http://perso-laris.univ-angers.fr/~boimond/Video-SRS-PremiereApplication.MOV>

Plan du document

1) Un premier contact avec le logiciel SRS 2022

- 1.1) Création d'une cellule composée d'un robot et de son contrôleur
- 1.2) Visualisation du robot dans la Vue 3D
- 1.3) Accès aux coordonnées articulaires et opérationnelles du robot
- 1.4) Mise en mouvement manuel des différents corps du robot

2) Conception de la cellule 3D

- 2.1) Positionnement d'un composant
- 2.2) Description des composants constituant la cellule
- 2.3) Ajout d'une ventouse au robot
 - 2.3.1) Création d'un cône pour représenter la ventouse
 - 2.3.2) Déclaration du cône en tant qu'outil (Tool) ventouse
 - 2.3.3) Rattachement de la ventouse au robot
- 2.4) Test sur le fonctionnement de la ventouse : prise et relâchement d'une pièce
 - 2.4.1) Capture du centre de la face supérieure de la pièce
 - 2.4.2) Prise ou relâchement de la pièce par la ventouse

3) Réalisation du programme VAL 3

3.1) Trajectoire et tâches à réaliser

3.2) Programme VAL 3 correspondant

3.3) Création et exécution du programme VAL 3 dans le logiciel SRS

3.3.1) Création du projet *Projet1*

3.3.2) Ecriture du programme


3.3.3) Exécution du programme


- Nous allons tout d'abord sélectionner le français comme étant la langue d'affichage utilisée par le logiciel SRS



. Une fois le logiciel ouvert, vous êtes, par défaut, dans la fenêtre du menu **Fichier**. Cliquez alors sur

« **Configuration ...** » (situé à gauche dans la liste) ; une fenêtre, intitulée **Configuration...**, apparaît laquelle permet de sélectionner (*via* le sous-menu **Environnement>Général**, sélectionné par défaut) la langue française. Cliquez sur le bouton **Ok** pour valider.

- Pensez à sauvegarder régulièrement votre travail *via* le menu **Fichier>Enregistrer tout**, ou, plus simplement, en cliquant sur  (icône située tout en haut à gauche dans la fenêtre principale du logiciel).

- Par la suite, si les fenêtres **Cellule**, **Données**, **Géométrie**, **Propriétés** ont été fermées de manière inopinée, il est possible de les rouvrir en allant dans le menu **Accueil>Général** (icône  située en haut vers la droite dans la fenêtre du menu **Accueil**).

1) Un premier contact avec le logiciel SRS 2022

Dans un premier temps, nous allons : créer une cellule composée uniquement d'un robot muni de son contrôleur (dans notre cas le robot TX60 avec le contrôleur Cs8) ; puis visualiser en 3D le robot et le mettre en mouvement de manière *manuelle*, c'est-à-dire, sans passer par l'exécution d'un programme VAL 3.

1.1) Création d'une cellule composée d'un robot et de son contrôleur

Si vous êtes dans la fenêtre du menu **Accueil** du logiciel (suite, par exemple, à la sélection de la langue française), cliquez sur le menu **Fichier** pour retourner dans la fenêtre du menu **Fichier** (fenêtre apparue lors de l'ouverture du logiciel), ce qui va permettre notamment de créer une cellule (ou d'en ouvrir une existante).

Dans la fenêtre du menu **Fichier**, cliquez sur **Nouvelle cellule** (disponible dans la liste située à gauche), puis sur



Nouvelle cellule vide.

Dans la fenêtre **Nouvelle cellule** qui apparaît, indiquez :

- dans le champ **Nom** l'intitulé de la cellule, à savoir : *PremiereApplication*,
- dans le champ **Chemin** un répertoire où l'application sera sauvegardée (faites en sorte de pouvoir la récupérer ceci malgré un arrêt intempestif de l'ordinateur !!),
- cliquez sur le bouton **Ok** pour finaliser l'étape de création de la cellule (qui est vide pour l'instant).

A l'issue de cette création, vous êtes dans la fenêtre du menu **Accueil** du logiciel.

Cliquez sur **Ajouter**  du menu **Accueil** afin d'ajouter un robot, sélectionnez **Nouveau robot** dans le menu

déroulé qui apparaît. Dans la fenêtre **Nouveau contrôleur** qui en résulte, sélectionnez le robot « *TX60 - Fixation sol - Horizontal base cable outlet - Butée(s) mécanique(s) externe(s) - Voyant puissance bras certifié UL* »



, puis cliquez sur le bouton **Suivant**.

La fenêtre qui s'ensuit concerne les options du contrôleur associé au robot TX60. Cliquez sur le bouton **Suivant** sans modifier ce qui est proposé, notamment le nom *Controller1* du contrôleur (Cs8). La fenêtre qui apparaît alors rappelle les caractéristiques du robot et du contrôleur, cliquez sur le bouton **Terminer** pour finaliser cette étape de sélection du robot et du contrôleur.

1.2) Visualisation du robot dans la Vue 3D

Cliquez sur **Afficher la vue 3D**  du menu **Accueil** pour visualiser le robot TX60 dans la *Vue 3D*.

Le repère situé en bas à gauche de la fenêtre **Vue 3D**, décrite dans la figure qui suit, est le repère associé à la **Vue 3D** avec une représentation des vecteurs x, y, z respectivement de couleur **rouge, vert, bleu**.

La rotation de la molette de la souris, dans un sens ou dans l'autre, permet de zoomer, ou dézoomer, la vue prise par la caméra en charge de visualiser la **Vue 3D**.

Le fait de faire un clic droit (n'importe où) dans la fenêtre **Vue 3D** permet, en sélectionnant **Best fit** dans le menu contextuel qui apparaît, d'obtenir un ajustement au mieux des composants présents dans la **Vue 3D** (ceci sans modifier l'orientation de la caméra), à l'image de ce qui est représentée dans la figure qui suit.

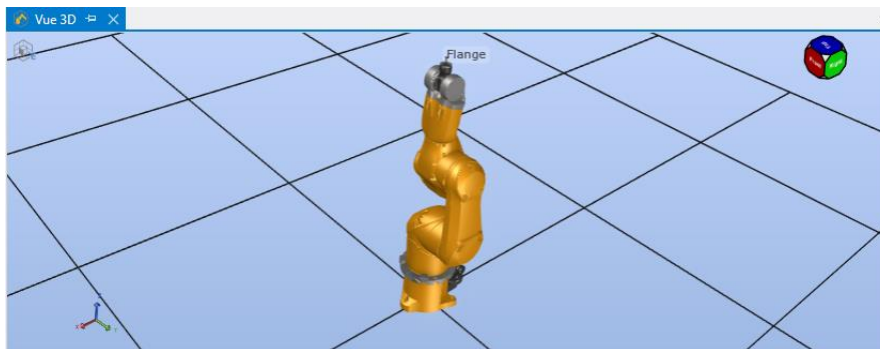



Figure 2 : Visualisation du robot TX60 dans la fenêtre **Vue 3D**.

Il existe plusieurs raccourcis qui permettent de *situer* (en position et orientation) la caméra afin de faciliter la visualisation de tout ou partie des composants présents dans la cellule :

- un clic droit accompagné d'un mouvement de la souris permet d'orienter la caméra (ceci sans changer sa position),
- un appui sur la molette de la souris accompagné d'un mouvement de la souris permet de déplacer la caméra (ceci sans changer son orientation),
- un clic sur l'une des faces du cube  (situé en haut à droite de la fenêtre **Vue 3D**) permet de sélectionner une vue prédéfinie de la caméra.

Le fait de faire un clic droit de la souris sur un composant préalablement sélectionné, ou sur l'un de ses constituants (par exemple, un des corps du robot), permet de :

- de zoomer sur le composant, ou sur l'un de ses constituants, en cliquant sur **Close up**,
- de positionner la caméra afin de centrer la vue (sans zoomer) sur le composant, ou sur l'un de ses constituants, préalablement sélectionné en cliquant sur **Center on model**.

✓ **A vous de tester ces différentes fonctionnalités !!**

1.3) Accès aux coordonnées articulaires et opérationnelles du robot

Pour une *situation* (en position et orientation) donnée du *Tool Center Point* (TCP), on dispose :

- des coordonnées articulaires, soit J_1, \dots, J_6 , fixant la posture correspondante du robot,
- des coordonnées opérationnelles, soit X, Y, Z, Rx, Ry, Rz , où : les valeurs de X, Y, Z , en mm, déterminent la position du TCP exprimée dans le *repère World* (correspondant au *repère de base*) du robot ; les valeurs de Rx, Ry, Rz , en degré, déterminent l'orientation du repère associé au TCP (cette orientation résultant de trois rotations appliquées **successivement** autour des axes x, y et z).

Ces coordonnées sont données dans la fenêtre **Déplacements manuels** laquelle est accessible en cliquant sur **Déplacements manuels** du menu **3D** (le menu **3D** étant apparu suite à l'appui sur **Afficher la vue 3D**, cf. début de §1.2).

Dans la figure qui suit, le robot TX60 est dans sa **posture initiale** (cliquer sur l'icône **Zéro** (située en haut de la fenêtre **Déplacements manuels**) pour mettre le robot dans cette posture). Notons que le robot n'est pas (encore) muni d'un outil aussi le TCP est positionné au centre de la flasque (*flange*) du robot.

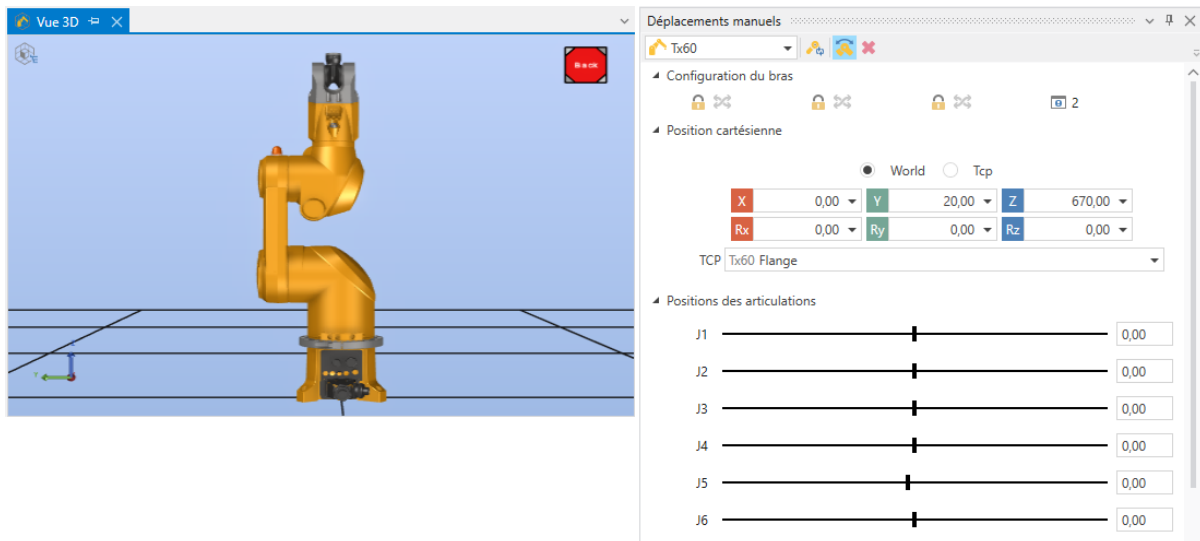


Figure 3 : Fenêtres **Vue 3D** (où le robot est dans sa posture initiale) et **Déplacements manuels** (avec un accès aux coordonnées articulaires et opérationnelles du robot).

N.B. : Il est possible d'afficher le repère associé au TCP en cliquant sur **Montrer Accroches/TCPs** **Show Handles/TCPs** du menu **3D**.

Question 1 : Retrouvez les coordonnées opérationnelles du TCP lorsque le robot est dans sa posture initiale, à savoir $X = 0, Y = 20, Z = 670, Rx = 0, Ry = 0, Rz = 0$, en vous référant aux caractéristiques du robot TX60 données dans le document accessible *via* le lien :

<http://perso-laris.univ-angers.fr/~boimond/TX2-60-6-axis-product-data-sheet-FR.pdf>

1.4) Mise en mouvement manuel des différents corps du robot

Il existe plusieurs possibilités pour mettre en mouvement, de manière manuelle, le robot à travers :

- la fenêtre **Déplacements manuels** :
 - soit dans l'espace articulaire en déplaçant le curseur d'une articulation (J_1, \dots, J_6),
 - soit dans l'espace opérationnel (encore appelé espace des tâches) en fixant une *situation* (atteignable) au TCP à travers les coordonnées X, Y, Z, Rx, Ry, Rz ,
- ou la fenêtre **Vue 3D** :
 - soit dans l'espace articulaire en cliquant sur un des corps du robot afin de sélectionner l'articulation correspondante, ce qui permet sa rotation *via* un clic gauche sur la souris,
 - soit dans l'espace des tâches en sélectionnant :
 - le point d'origine du repère associé au TCP (n'hésitez pas à bien zoomer au préalable sur le repère), ce qui permet de modifier sa *position* *via* un clic gauche sur la souris,
 - l'un des trois vecteurs du repère associé au TCP (x en rouge, y en vert, z en bleu), ce qui permet de modifier son *orientation* *via* un clic gauche sur la souris.

✓ **A vous de tester ces différentes fonctionnalités de mise en mouvement manuel du robot !!**

Question 2 : Après avoir mis le robot dans une posture quelconque (non initiale), cochez la case **Tcp** (plutôt que **World**), présente dans la fenêtre **Déplacements manuels**, il devrait en résulter des valeurs nulles pour X, Y, Z, Rx, Ry, Rz .

Effectuer alors un mouvement de translation (positif ou négatif) selon l'axe Z du repère associé au TCP (n'hésitez pas à zoomer pour mettre en évidence ce repère), expliquer les valeurs de X, Y, Z, Rx, Ry, Rz qui en résultent. Donnez un exemple montrant l'intérêt d'une telle action.

2) Conception de la cellule 3D

Le menu **3D** permet l'accès à certaines fonctionnalités pour concevoir une cellule dans un environnement 3D, notamment concernant :

- la modélisation (importation de modèle CAO, création d'outil), l'élaboration de points, voir la figure qui suit.

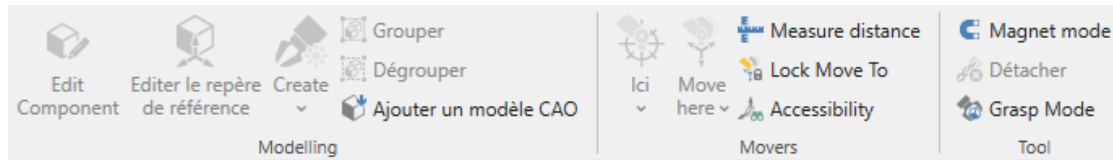


Figure 4 : Sections **Modelling**, **Movers** et **Tool** du menu **3D**.

- l'utilisation de composants propres à Stäubli (issus du **Catalog**, décrit au §2.2 ci-dessous), la mise en mouvement manuel du robot (**Déplacements manuels**, voir §1.3), le positionnement des composants dans la cellule (**Editer la position**, décrit au §2.1 ci-dessous), voir la figure qui suit.

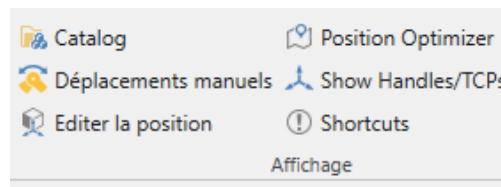




Figure 5 : Section **Affichage** du menu **3D**.

En dehors de la phase de programmation du robot traitée au §3, la conception d'une cellule consiste à importer les composants (dont vous avez besoin) en les *situant* (c'est-à-dire, en les positionnant et en les orientant) de manière adéquate les uns par rapport aux autres au sein de la cellule, ce que l'on va décrire ci-dessous.

2.1) Positionnement d'un composant

Le fait de cliquer sur **Editer la position**  du menu **3D** fait apparaître la fenêtre **Editer la position** laquelle permet de disposer ou de renseigner la *situation* (en position et orientation) des différents composants présents dans la cellule (notamment le robot) relativement **au repère associé à la Vue 3D** (à ne pas confondre avec le repère World du robot).

Le fait de sélectionner dans la **Vue 3D** le robot (à travers la sélection de l'un de ses constituants), puis de cliquer sur l'icône  permet de disposer de la *situation* du **repère de référence** lié au robot, à savoir, $X = 0, Y = 0, Z = 0, Rx = 0, Ry = 0, Rz = 0$, de manière, soit *absolue* (correspondant à la position lors de son ajout dans la **Vue 3D**), soit *relative* (correspondant à la position obtenue suite à un éventuel déplacement du composant suite à son ajout dans la **Vue 3D**).

Notons que le *repère de référence* lié au robot apparaît après avoir cliqué sur le corps correspondant au socle (de couleur verte) du robot, comme illustré dans la figure qui suit.

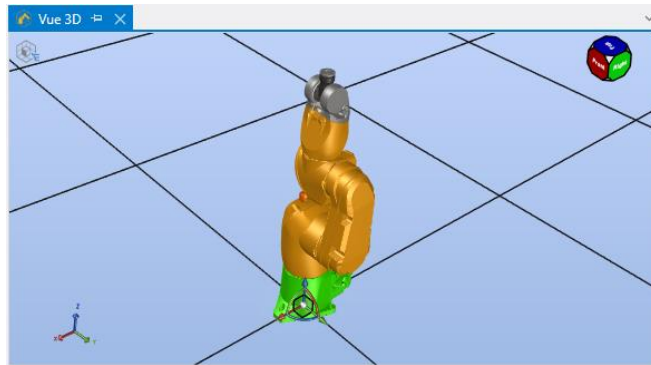
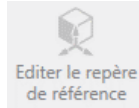


Figure 6 : Repère de référence lié au socle (de couleur verte) du robot.

NB : Il est également possible de faire apparaître le *repère de référence* associé à un composant constitué de différents corps (par exemple, un robot) en sélectionnant l'un de ses constituants (un des corps du robot), puis

en cliquant sur **Editer le repère de référence**



du menu **3D**.

2.2) Description des composants constituant la cellule

Intéressons-nous à la construction de la cellule sur laquelle nous allons travailler par la suite, à l'image de celle décrite ci-dessous (où le robot n'est pas encore muni d'une ventouse) :

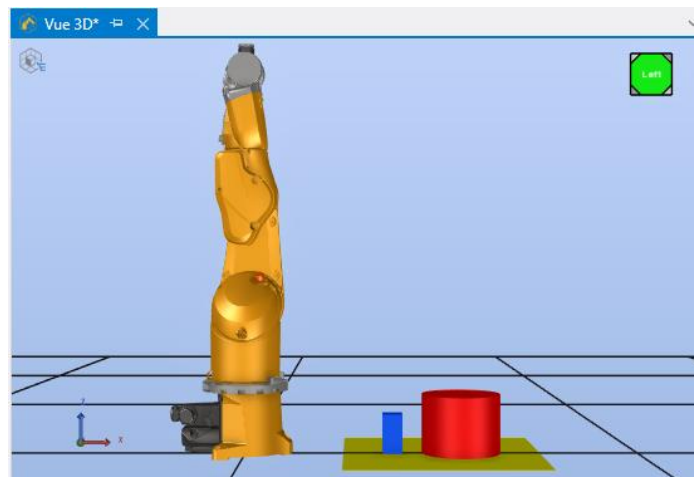


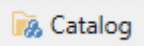
Figure 7 : Cellule à reproduire (pour l'instant sans la ventouse).

sachant que *les repères de référence* liés respectivement :

- au robot correspond à $X = 0, Y = 0, Z = 0, Rx = 0, Ry = 0, Rz = 0$,
- au support jaune (représentant une plaque métallique où reposent la pièce à déplacer et le cylindre sur lequel la pièce va être déposée) correspond à $X = 250, Y = -250, Z = 0, Rx = 0, Ry = 0, Rz = 0$ (voir §2.1 pour situer le composant à l'aide **Editer la position** en pensant à utiliser la *Position absolue* (et non *relative* !)).

Dans la section **Divers** de la fenêtre **Propriétés** (située au centre en bas) sont indiquées :

- les dimensions du support égales à : 500 mm selon X (champ **LengthX**), 500 mm selon Y (champ **LengthY**) et 1 mm selon Z (champ **LengthZ**),
- la couleur jaune du support (champ **Color**),
- au cube bleu (représentant la pièce posée sur le support) correspond à $X = 350, Y = -25, Z = 1, Rx = 0, Ry = 0, Rz = 0$. Les dimensions du cube sont égales à : 50 mm selon X , 50 mm selon Y et 100 mm selon Z ,
- au cylindre rouge (posé sur le support et sur lequel la pièce va être déposée) correspond à $X = 550, Y = 0, Z = 1, Rx = 0, Ry = 0, Rz = 0$. Les dimensions du cylindre sont égales à 100 mm de rayon (champs **BottomRadius** et **TopRadius**) avec une longueur de 150 mm (champ **Length**).

Afin de construire cette cellule, faites apparaître la fenêtre **Catalog**, en cliquant sur **Catalogue**  du menu **3D**, ce qui permet d'importer par *Drag and Drop* dans la **Vue 3D** des composants mis à disposition par

Stäubli. Les composants **box** et **cylinder**, rangés dans l'onglet **Basic geometries**, vont permettre de construire le support et la pièce (*via* le composant **box**) et le cylindre (*via* le composant **cylinder**).

Une fois un composant importé dans la **Vue 3D**, le fait de cliquer dessus permet d'afficher ses propriétés dans la fenêtre **Propriétés**, notamment son nom, sa couleur, ses dimensions.

✓ **A vous de jouer pour reproduire la cellule décrite précédemment !!**

2.3) Ajout d'une ventouse au robot

L'ajout d'un outil, dans notre cas une ventouse, au niveau de la flasque du robot se fait en trois étapes :

- création d'un composant représentant la ventouse,
- déclaration du composant (représentant la ventouse) en tant qu'outil (Tool) ventouse,
- rattachement de l'outil à la flasque du robot (le TCP du robot est situé à l'extrémité de l'outil, intitulé **Ventouse**).

2.3.1) Création d'un cône pour représenter la ventouse

Pour faciliter le travail de création, la ventouse est représentée par un simple cône tronqué (il aurait été également possible d'importer un composant réalisé par un logiciel de CAO, ce que nous n'allons pas faire ici). Le composant **cylinder** du **Catalog** est utilisé pour cela avec comme paramètres (accessibles dans la fenêtre **Propriétés**) : un rayon de sa base inférieure (champ **BottomRadius**) égal à 10 mm, un rayon de sa base supérieure (champ **TopRadius**) égal à 2 mm, une hauteur (champ **Length**) égale à 50 mm. De plus, la ventouse est de couleur noire (champ **Color**).

Le cône est placé à portée de la flasque du robot pour faciliter son rattachement au robot (pour cela, on choisit de situer son *repère de référence* au point : $X = -500, Y = 0, Z = 0, Rx = 0, Ry = 0, Rz = 0$, voir la figure ci-dessous).

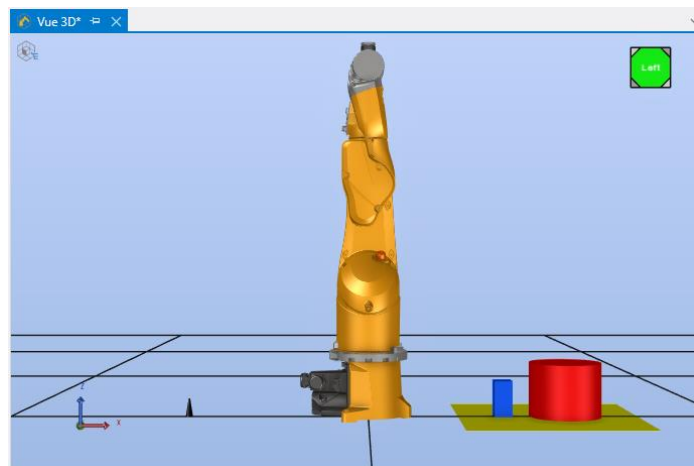

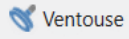


Figure 8 : Représentation de la ventouse (non encore liée à la flasque du robot).

Afin d'être plus explicite, remplacez le nom **cylinder1** par **Ventouse** à l'aide du champ **Nom** situé dans la fenêtre **Propriétés**.

2.3.2) Déclaration du cône en tant qu'outil (Tool) ventouse

Il s'agit à présent de déclarer le cône (représentant la ventouse), créé au §2.3.1, en tant qu'outil ventouse. Pour cela, une fois le cône sélectionné dans la **Vue 3D**, cliquez sur **Edition composant**  du menu **3D** ce qui provoque l'apparition d'une fenêtre intitulée du nom du composant (à savoir, **Ventouse**).

Afin que l'outil se comporte comme une ventouse, déclarez le composant **Ventouse** en tant que ventouse en sélectionnant **Ventouse**  du menu **Editeur de composants** (apparu à droite du menu **3D** en même temps que la fenêtre **Ventouse**).

Deux repères, appelés **Handle** et **Tcp**, sont associés au cône (déclaré en tant qu'outil (Tool)) :

- Le repère **Handle** va permettre d'attacher l'outil à la flasque du robot, ce qui sera le cas lorsque ce repère coïncidera avec le repère **Handler** du robot (à savoir, le repère associé à la flasque du robot), voir la figure qui suit.

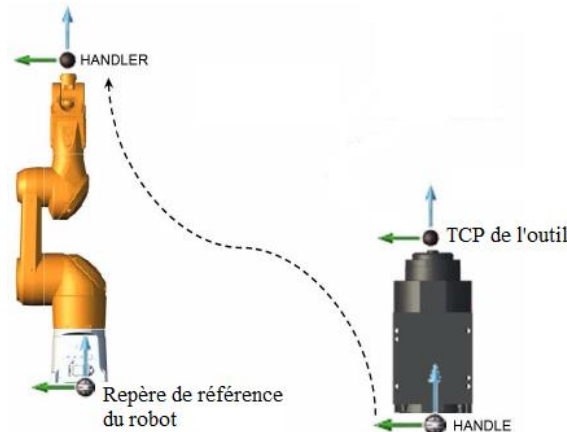
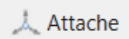
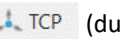


Figure 9 : **Handler** du robot et **Handle** de l'outil.

- Le repère **Tcp** est associé au TCP, point situé à l'extrémité de l'outil. Ce repère correspondra au TCP du robot une fois le rattachement de l'outil à la flasque du robot effectué.

Cliquez sur **Attache**  du menu **Editeur de composants** ce qui provoque l'apparition du **Handle** rattaché à l'outil **Ventouse** (le repère associé au **Handle** apparaît dans la fenêtre **Ventouse**).

Cliquez sur **TCP**  (du menu **Editeur de composants**) ce qui provoque l'apparition du **Tcp** rattaché à l'outil **Ventouse** (le repère associé au **Tcp** apparaît dans la fenêtre **Ventouse**).

Le fait de sélectionner dans la fenêtre **Ventouse** le **Handle** (de l'outil) permet de visualiser (en vert) sa *situation*, de même pour le **Tcp** (lié au TCP de l'outil), comme illustré dans la figure qui suit.

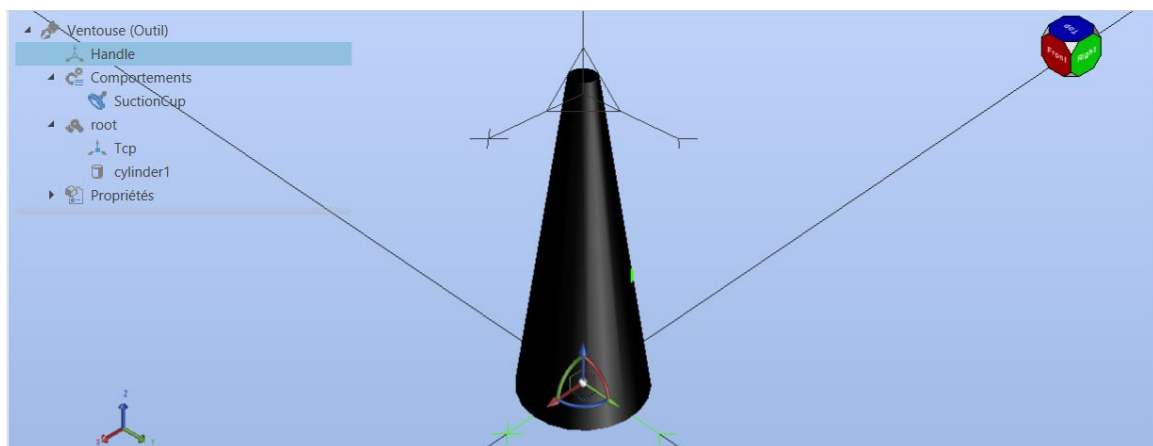


Figure 11 : Représentation en vert du repère **Handle** de l'outil **Ventouse**.

Le repère **Handle** (que l'on sélectionne en cliquant sur **Handle** dans la fenêtre **Ventouse**) est situé par défaut au centre de la base inférieure du cône, au point : $X = 0, Y = 0, Z = 0, Rx = 0, Ry = 0, Rz = 0$ relativement au repère associé à la fenêtre **Ventouse** (ces données étant accessibles dans la fenêtre **Editer la position**).

Le repère **Tcp** (que l'on sélectionne en cliquant sur **Root>Tcp** dans la fenêtre **Ventouse**) est situé par défaut sur un point situé au bord de la base supérieure du cône. Il s'agit de modifier cette position afin de se positionner au

centre de la base supérieure du cône, c'est-à-dire, au point : $X = 0, Y = 0, Z = 50, Rx = 0, Ry = 0, Rz = 0$ (ceci en étant en *Position absolue*).

Enfin, sélectionnez **SuctionCup** dans la fenêtre **Ventouse** afin de configurer dans la fenêtre **Propriétés** les champs **TCP**, **Enable Trigger**, **Disable Trigger**, **Événement à la saisie**, **Distance de saisie** comme indiqués dans la figure qui suit.

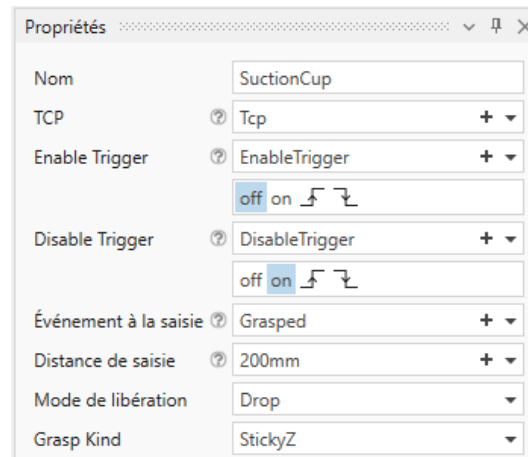




Figure 12 : Configuration de la ventouse.

Le fait d'avoir sélectionné l'outil en tant que *Suction cup* fait apparaître dans la fenêtre **Propriétés** des caractéristiques propres à une ventouse, notamment la *Distance de saisie (GraspDistance)*, par défaut égale à 200 mm, qui correspond à la distance qu'il doit y avoir *au maximum* entre le repère de référence de la pièce et le TCP du robot au moment de l'activation de la ventouse pour que l'aspiration soit effective.

La fenêtre **Ventouse** peut alors être fermée en enregistrant  votre travail au préalable afin de prendre en compte les caractéristiques de l'outil **Ventouse**.

2.3.3) Rattachement de la ventouse au robot

Afin de lier l'outil à la flasque du robot, cliquez sur **Mode magnétique**  du menu **3D**, sélectionnez **le point d'origine du repère de référence** de l'outil et approchez le vers la flasque du robot *via un Drag and Drop*. Il devrait en résulter ce qui est décrit dans la figure qui suit.

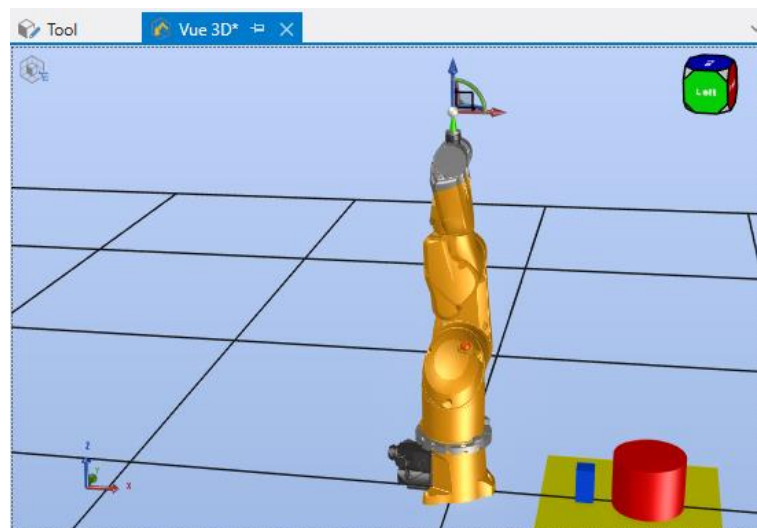


Figure 13 : Outil rattaché à la flasque du robot.

✓ Vérifiez en bougeant manuellement le robot que l'outil est effectivement lié à la flasque du robot.

2.4) Test sur le fonctionnement de la ventouse : prise et relâchement d'une pièce

Il est possible de tester la prise, ou le relâchement, de la pièce (pour nous le cube bleu) par la ventouse à travers deux étapes : on « capture » dans un premier temps le centre de la face supérieure de la pièce (voir §2.4.1), puis on actionne la ventouse afin de prendre, ou relâcher, la pièce (voir §2.4.2).

2.4.1) Capture du centre de la face supérieure de la pièce

Il s'agit de faire en sorte que le TCP (situé à l'extrémité de la ventouse) pointe au centre de la face supérieure de la pièce lors de la saisie ou du relâchement de la pièce. Pour « capturer » ce point, on procède en deux temps :

- i) Dans un premier temps, il s'agit de (simplement) placer le TCP à proximité de la face supérieure de la pièce/du cube, à l'image de ce qui est illustré dans la figure suivante. Pour cela, on choisit de *situer* le TCP par rapport **au repère World du robot** (cf. fenêtre **Déplacements manuels**) au point : $X = 365, Y = 10, Z = -150, Rx = 0, Ry = -180, Rz = 0$.

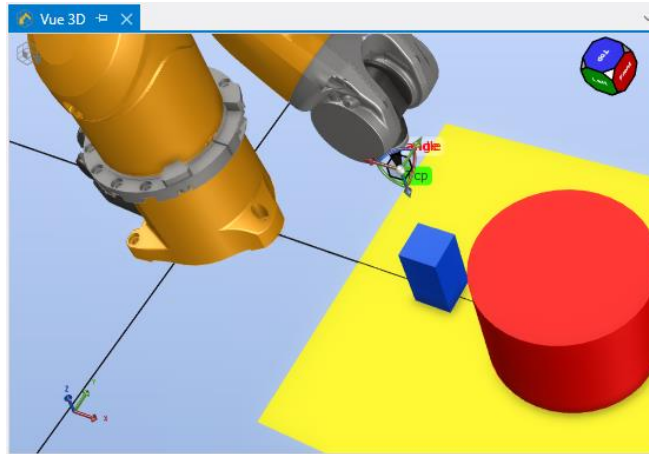


Figure 14 : Rapprochement du TCP par rapport à la pièce.

- ii) A présent, il s'agit de faire en sorte que le TCP pointe (exactement) au centre de la face supérieure de la pièce. Pour faciliter la « capture » de ce point, on sélectionne la fonction **Milieu de segment** (située en bas de la fenêtre **Déplacements manuels** (pour rappel, cette fenêtre apparaît en cliquant sur l'icône **Déplacements manuels** du menu **3D**)). Une fenêtre (transparente), à l'image de celle décrite dans la figure ci-dessous, apparaît alors en bas à droite de la fenêtre **Vue 3D** ; on prendra soin à ce que le champ **Centre** soit sélectionné/grisé pour permettre de positionner le TCP (exactement) au centre de la face supérieure de la pièce/du cube (ce que l'on cherche à faire).

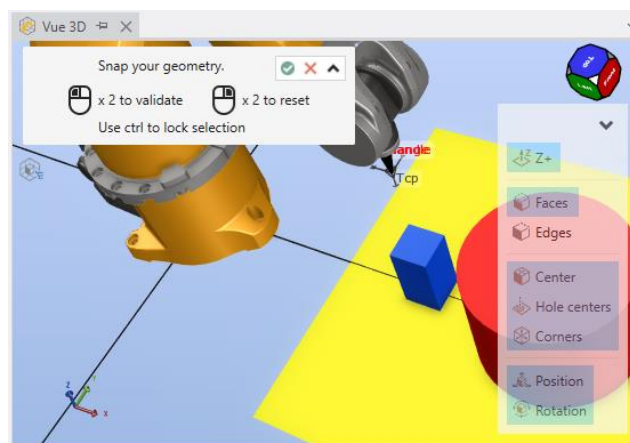


Figure 15 : Utilisation de la fonction **Milieu de segment (Snap)** pour *situer* (précisément) le TCP par rapport à la pièce.

Le fait de déplacer la souris afin que le curseur pointe au centre de la face supérieure de la pièce devrait donner lieu à ce qui est décrit dans la figure ci-dessous. Il reste à double-cliquer à gauche sur la souris, comme cela est indiqué en haut à gauche dans la fenêtre **Vue 3D**, pour valider l'opération de positionnement du TCP.

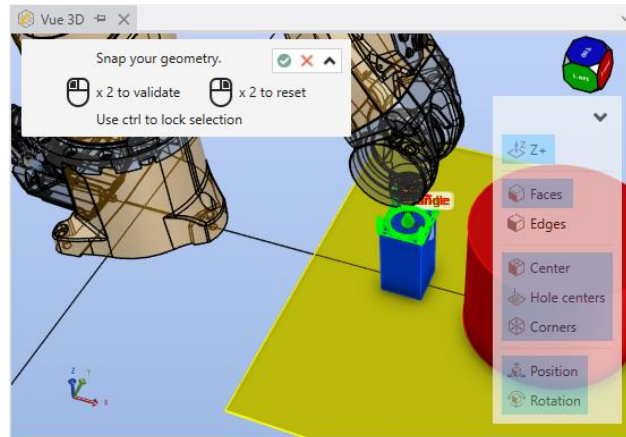





Figure 16 : Positionnement du TCP au centre de la face supérieure de la pièce.

Remarque : Les contours des composants décrits dans la figure précédente sont mis en évidence du fait de la sélection de l'icône **Affichage des faces**  **Display wire frame** accessible *via* le menu déroulant **Réglages** du menu **3D**.

Question 3 : A l'issue de la phase de saisie, retrouvez les valeurs des coordonnées du TCP exprimées dans le repère World du robot (valeurs affichées dans la fenêtre **Déplacements manuels**) en vous référant aux caractéristiques du robot TX60 données dans le document accessible via le lien :

<http://perso-laris.univ-angers.fr/~boimond/TX2-60-6-axis-product-data-sheet-FR.pdf>

2.4.2) Prise ou relâchement de la pièce par la ventouse

Pour actionner la ventouse, cliquez sur **Mode de prise**  **Grasp Mode** du menu **3D**, puis (comme demandé dans une fenêtre apparue en haut de la fenêtre **Vue 3D**, voir la figure ci-dessous), sélectionnez le robot ou la ventouse, puis (si nécessaire) la pièce. Il reste alors à cliquer sur l'icône  de la fenêtre pour actionner la ventouse et donc pour que la pièce soit saisie par le robot.

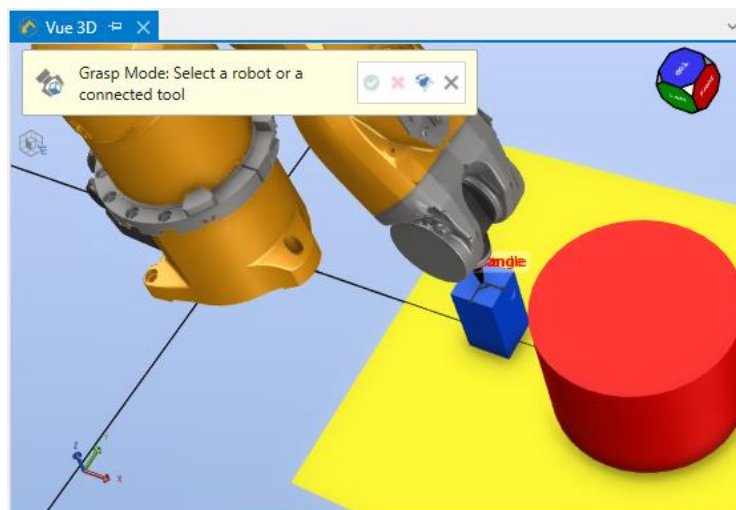



Figure 17 : La pièce est saisie par le robot.

✓ **Bouger le robot afin de confirmer que le robot a bien saisi la pièce.**

Cliquer sur l'icône  (**Release**) pour que la ventouse relâche la pièce. Deux cas de figure se produisent alors suite au relâchement de la pièce :

- un objet 3D est présent sous la pièce (c'est-à-dire, dans la direction négative le long de l'axe Z de la cellule/du repère associé à la **Vue 3D**) auquel cas la pièce tombe jusqu'à ce qu'elle entre en collision avec l'objet 3D (en ce qui nous concerne la pièce sera relâchée au-dessus du cylindre),
- dans le cas contraire, la pièce sera libérée et restera dans sa position actuelle (plutôt que de tomber sans que rien ne puisse arrêter sa chute !!).

✓ **Testez si la pièce relâchée alors qu'elle est située au-dessus du cylindre provoque son dépôt sur le cylindre.**

3) Réalisation du programme VAL 3

Le programme qu'il s'agit de réaliser, en utilisant le langage VAL 3, va interagir avec la cellule 3D construite précédemment et ainsi permettre sa validation (avant un transfert vers le contrôleur (réel) du robot, étape que nous n'allons pas réaliser ici).

3.1) Trajectoire et tâches à réaliser

La trajectoire du robot est calculée dans l'*espace articulaire*. Ainsi sa vitesse d'exécution est optimisée, par contre en dehors des points mentionnés dans les instructions de mouvement par lesquels le TCP doit passer, la trajectoire du robot n'est pas contrainte (en ligne droite ou circulaire).

Décrivons la trajectoire à réaliser par le robot ainsi que les tâches (de saisie et de relâchement de la pièce) à réaliser en certains de ses points.

Au départ, le robot est dans sa *posture initiale* (il est tendu à la verticale, toutes les valeurs articulaires J_1, \dots, J_6 sont nulles), on note $jj1$ le point du TCP (c'est-à-dire, le point situé à l'extrémité de la ventouse) correspondant à cette posture.

Puis le TCP doit se *situer* successivement :

- au point, noté $pP2$, correspondant au centre de la face supérieure de la pièce (bleue) afin que la ventouse puisse la saisir (pour cela, l'axe Z de $pP2$ sera parallèle à l'axe Z_0 du repère World du robot),
- à un point intermédiaire, noté $pP3$, suffisamment éloigné du cylindre (rouge) pour que la pièce ne rentre pas en collision avec le cylindre lors du déplacement du robot vers le dessus du cylindre. Pour cela, on choisit de prendre un point $pP3$ situé à la verticale (à une distance de 174 mm dans le sens positif de l'axe Z du repère associé à la **Vue 3D**) du point $pP2$, soit : $pP3_z = pP2_z + 174$. Nous verrons que le programme réalisé par la suite sera tel que le TCP passera en fait au voisinage du point $pP3$,
- au point, noté $pP4$, correspondant au centre de la face supérieure du cylindre surélevé d'une hauteur égale à celle de la pièce afin que la ventouse puisse déposer sans heurt la pièce sur le cylindre.

Pour finir, le TCP retourne vers le point de départ ($jj1$) afin que le robot retourne à sa position initiale.

Question 4 : Exprimer la *situation* du point $jj1$ dans l'*espace articulaire* du robot, ainsi que celles des points $pP2$, $pP3$ et $pP4$ dans l'*espace opérationnel* (ces informations étant utilisées par la suite dans le programme VAL 3).

3.2) Programme VAL 3 correspondant

Avant de décrire le programme, donnons quelques informations de base sur le langage VAL 3.

Un programme VAL 3 est composé au minimum :

- d'une fonction *start*, appelée automatiquement au démarrage du programme, qui contient le programme principal et l'initialisation des variables globales,
- d'une fonction *stop*, appelée automatiquement lors de l'arrêt du programme, qui contient les instructions permettant de quitter proprement le programme,

sachant qu'il est bien sûr possible d'adjointre à ces deux fonctions ses propres fonctions pour le besoin du programme.

VAL 3 est un langage de programmation proche du C/C++. Outre les types de données standards tels que *bool* (pour les booléens), *num* (pour les nombres entiers ou à virgules), *string* (pour les chaînes de caractères), VAL 3 possède des types propres à la robotique. Une liste des types de données est décrite dans la figure qui suit.

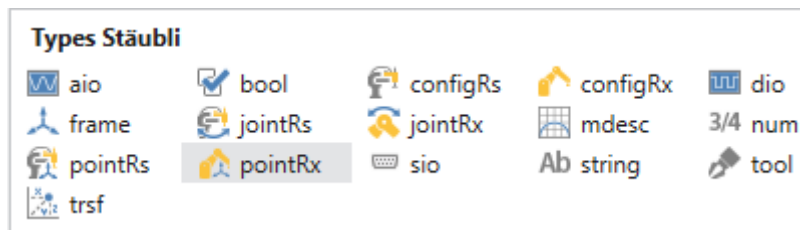


Figure 18 : Liste des types de données disponibles dans VAL 3.

Les types que nous allons utiliser par la suite sont les suivants :

- *jointRx* pour définir un point dans l'*espace articulaire* à travers les valeurs articulaires J_1, \dots, J_6 ,
- *pointRx* pour définir un point dans l'*espace opérationnel* à travers les valeurs X, Y, Z, Rx, Ry, Rz ,
- *tool* pour définir un outil en vue de sa connexion à la flasque du robot,
- *mdesc* pour définir les paramètres d'un mouvement (notamment la vitesse, l'accélération, la décélération maximales du robot).

Le programme VAL 3 qui suit permet la réalisation des actions à mener :

```
begin
  movej(jj1, Ventouse, vitesse)
  open(Ventouse)
  movej(pP2, Ventouse, vitesse)
  close(Ventouse)
  movej(pP3, Ventouse, vitesse)
  movej(pP4, Ventouse, vitesse)
  open(Ventouse)
  movej(jj1, Ventouse, vitesse)
  waitEndMove()
end
```

sachant que :

- une instruction du type `movej(pPoint, Ventouse, vitesse)` déclenche un mouvement du robot, calculé dans l'*espace articulaire*, de sorte qu'à l'issue du mouvement le TCP (point situé à l'extrémité de la ventouse, dont le nom est *Ventouse* (voir §2.3.3)) coïncide avec le point de destination (*pPoint*).

Des raisons pratiques font que le point *pPoint* sera défini tantôt dans l'*espace opérationnel*, tantôt dans l'*espace articulaire* (voir la question 4), ce qui revient à le déclarer en VAL 3 comme étant de type *pointRx* ou *jointRx*.

Il se produit un phénomène de « lissage » lorsque plusieurs instructions `movej()` se suivent, ce qui est le cas pour les instructions de mouvement associés aux points *pP3* et *pP4*. En effet, la trajectoire du TCP allant du point *pP2* vers le point *pP4* est « lissée » au niveau du point *pP3*, au sens où la trajectoire va passer, sans s'arrêter, au voisinage du point *pP3*. Dans le cas présent, cela ne pose pas de problème car le point *pP3* sert seulement à faire en sorte que la pièce prenne suffisamment de hauteur pour qu'elle ne rentre pas en collision avec le cylindre durant son déplacement vers le dessus du cylindre. Un tel phénomène de « lissage » permet ainsi un mouvement plus fluide (*smooth*) et rapide. Ce n'est pas le cas lorsqu'entre deux instructions de mouvement, il y a une instruction qui force la fin complète du mouvement avant d'entamer le suivant, telle que `open()`, `close()` ou `waitEndMove()` décrite ci-dessous.

Le paramètre *vitesse* est de type *mdesc*, il permet de fixer notamment la vitesse, l'accélération, la décélération maximales lors de mouvement.


Remarque : L'instruction `movej()`, resp. `movec()`, permet la réalisation de trajectoires en ligne droite, resp. circulaire.

- l'instruction `open(Ventouse)` déclenche l'aspiration de l'outil *Ventouse* lorsque sa sortie digitale (intitulée *Controller1:valve1* comme nous le verrons plus tard) est à *true* ; l'instruction `close(Ventouse)` provoque l'arrêt de l'aspiration lorsque cette sortie est à *false*.
- l'instruction `waitEndMove()` attend la fin du mouvement en cours avant d'exécuter la prochaine instruction. Cette instruction placée à la fin du programme permet l'exécution de tous les mouvements avant l'arrêt du programme.

3.3) Création et exécution du programme VAL 3 dans le logiciel SRS

3.3.1) Création du projet *Projet1*

On va créer un projet, lequel contiendra le programme à implémenter, au sein du contrôleur relié au TX60, à savoir *Controller1* (il pourrait y en avoir plusieurs s'il y avait plusieurs robots dans la cellule).

Pour cela, sélectionner le contrôleur *Controller1* dans la fenêtre **Cellule**, puis cliquez sur l'icône  du menu **VAL3**.

Dans la fenêtre **Nouvelle application** qui apparaît, indiquez l'intitulé du projet, on prendra par exemple *Projet1*, dans le champ **Nom**. Sans modifier les autres champs de la fenêtre, cliquez sur le bouton **Ok** pour valider.

3.3.2) Ecriture du programme

Le fait de déplier, dans la fenêtre **Cellule**, le Projet *Projet1* fait apparaître les fonctions *start* et *stop*, comme cela est décrit dans la figure qui suit.

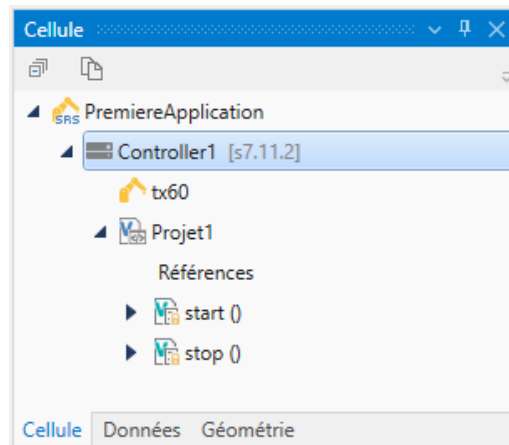


Figure 19 : Contenu de la fenêtre **Cellule** à l'issue de la création d'un projet.

Le programme va être placé au sein de la fonction *start* (en effet, la simplicité du programme fait qu'il n'est pas nécessaire d'utiliser d'autres fonctions). Le fait de double-cliquer sur la fonction *start* (listée dans la fenêtre **Cellule**) fait apparaître une fenêtre **Projet1-start** dans laquelle les balises *begin* et *end*, mises par défaut, délimitent l'endroit où insérer les instructions du programme.

Une méthode standard pour écrire un programme consiste, préalablement à l'écriture des instructions composant le programme, à déclarer et à définir l'ensemble des variables utilisées dans le programme. Ici, on choisit une méthode, plus pratique pour ce que l'on souhaite faire, qui consiste à déclarer et à définir les variables au fur et à mesure de leurs utilisations dans les instructions.

Tapez la première instruction (située sous la balise *begin*), à savoir :

```
movej(jj1, Ventouse, vitesse)
```

puis, intéressons-nous aux trois variables *jj1*, *Ventouse*, *vitesse*, associées à cette instruction.

▪ **Concernant le point *jj1***

Une fois la variable *jj1* créée, on va lui attribuer un *type* (à savoir *jointRx*, voir §3.2), puis lui assigner des valeurs.

- **Création de la variable *jj1* et attribution de son *type*** : Le fait de cliquer droit sur le point *jj1* (présent au sein de l'instruction) permet de sélectionner le menu **Ajouter>Nouvelle donnée**, comme indiqué dans la figure qui suit.

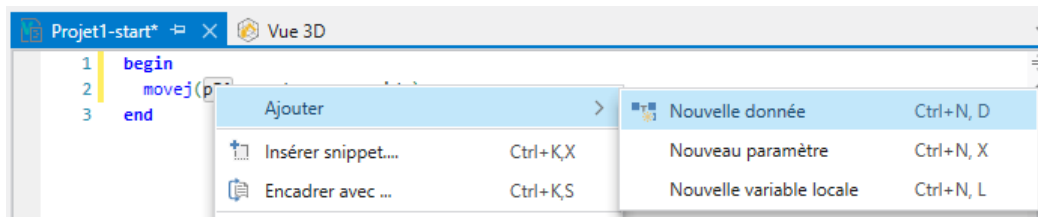


Figure 20 : Menu **Ajouter>Nouvelle donnée** utilisé pour créer la variable $pP1$.

Il en résulte l'apparition d'une fenêtre **Ajouter une nouvelle donnée** laquelle permet de sélectionner le type, à savoir *jointRx*, de la variable $jj1$, voir la figure ci-dessous. Sans modifier les autres champs de la fenêtre (notamment le champ **Conteneur** étant un *Tableau* fera que la variable $jj1$ sera en fait l'élément d'indice zéro du tableau $jj1$, c'est-à-dire, $jj1[0]$), cliquez sur le bouton **Ok** pour valider.

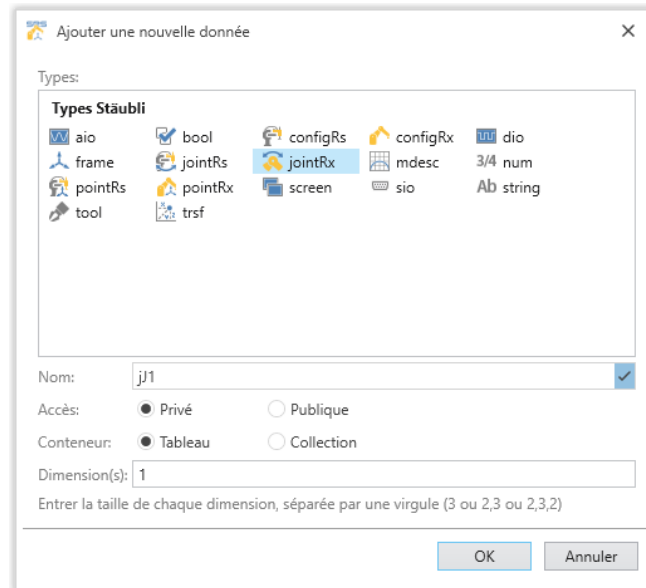


Figure 21 : Fenêtre **Ajouter une nouvelle donnée** utilisée pour *typer* la variable $jj1$.

En cliquant sur l'onglet **Données**, situé en bas de la fenêtre **Cellule**, la variable $jj1[0]$ apparaît alors dans la fenêtre **Données** (cette fenêtre s'affichant à la place de la fenêtre **Cellule**), voir la figure ci-dessous.

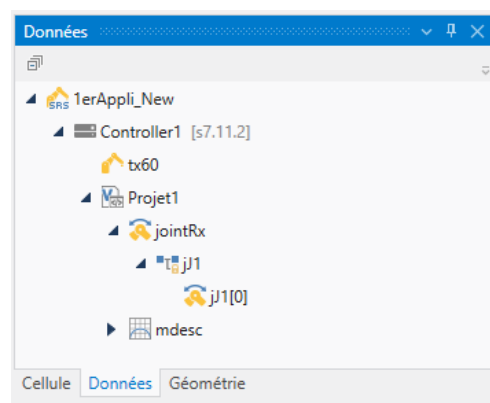


Figure 22 : Fenêtre **Données** relative à la variable $jj1$ (de type *jointRx*).

- **Assignment de valeurs à la variable $jj1$** : Le fait de double-cliquer sur la variable $jj1[0]$ a pour effet d'ouvrir une fenêtre **Données 'Projet1-jj1'** laquelle permet (du fait du type *jointRx* de la variable) d'assigner des valeurs (mises à 0 par défaut) aux articulations J_1, \dots, J_6 , voir la figure qui suit.

| Index | J1 | J2 | J3 | J4 | J5 | J6 |
|-------|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 23 : Fenêtre **Données 'Projet1-jj1'**.

✓ **A vous de modifier, s'il y a lieu, les valeurs de J_1, \dots, J_6 correspondant au point $jj1$.**

▪ **Concernant l'outil *Ventouse***


Il faut indiquer au projet *Projet1* que le composant construit au §2.3.1 est un outil, pour cela, la variable *Ventouse*, une fois créée, sera déclarée comme étant de type *tool*. Puis, des valeurs seront assignées à la variable.

La démarche de création/typage/assignation est comparable à celle faite pour le point *jj1*. Le fait de cliquer droit sur la variable *Ventouse* (présente au sein de l'instruction) permet de sélectionner le menu **Ajouter>Nouvelle donnée**.

Il en résulte l'apparition d'une fenêtre **Ajouter une nouvelle donnée** laquelle permet de sélectionner le type, à savoir *tool*, de la variable *Ventouse*. Sans modifier les autres champs de la fenêtre (ce qui fera que la variable *Ventouse* sera en fait l'élément d'indice zéro du tableau *Ventouse*, c'est-à-dire, *Ventouse[0]*), cliquez sur le bouton « **Ok** » pour valider.

Le fait de double-cliquer sur la variable *Ventouse[0]* a pour effet d'ouvrir une fenêtre **Données 'Projet1-Ventouse'** dans laquelle on va indiquer la *situation* du TCP de la ventouse par rapport à son repère **Handle**, voir Figure 11. Pour cela, on assigne à l'axe *Z* (de la variable *Ventouse[0]*) la valeur 50 mm (au lieu de la valeur 0 mise par défaut) en laissant inchangé les autres valeurs.

Se rajoute le fait de devoir rendre opérationnel les instructions `open()` et `close()` (utilisées dans le programme) en spécifiant explicitement la sortie digitale du contrôleur *Controller1* en charge d'actionner l'outil *Ventouse*.

Pour cela on commence par déplier l'*arbre de scène* 3D  **Scene** (icône située en haut à gauche dans la fenêtre **Vue 3D**) afin de sélectionner l'outil *Ventouse*, voir la figure qui suit :

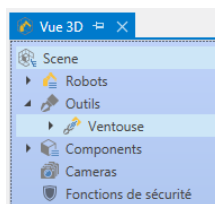


Figure 24 : *Arbre de scène* 3D déplié de manière à sélectionner l'outil *Ventouse*.

Il en résulte dans la fenêtre **Propriétés** une liste de propriétés (de l'outil *Ventouse*) relatives à la section **Signals**, à l'image de ce qui est indiqué dans la figure qui suit.



| ▼ Signals | |
|------------------|---|
| EnableTrigger | (Sélectionner une... ⋮  X) |
| DisableTrigger | (Sélectionner une... ⋮  X) |

Figure 25 : Section **Signal** de la fenêtre **Propriétés** relative à l'outil *Ventouse*.

Cette section de la fenêtre va permettre : de choisir une sortie digitale (on laissera la sortie *valve1*, proposée par défaut) ; puis de confirmer le lien entre cette sortie et l'outil *Ventouse* en cliquant sur les trois points situés sur la ligne du champ **Enable Trigger** (en lien avec l'instruction `close(Ventouse)`), ce qui donne lieu à l'ouverture de la fenêtre **Digitals** qui suit.

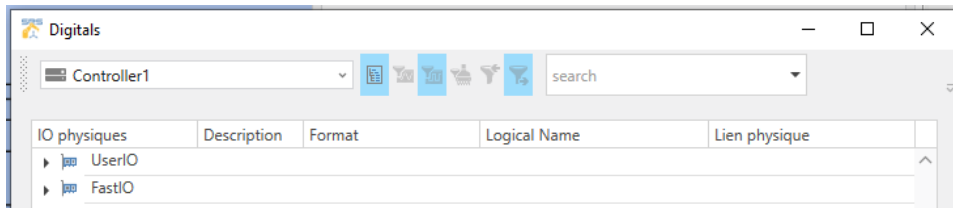


Figure 26 : Fenêtre **Digitals** du *Controller1*.

Le fait de déplier l’item *UserIO* permet d’obtenir la fenêtre, décrite dans la figure qui suit, dans laquelle sont listées les deux sorties digitales du contrôleur *Controller1*: *valve1* (reliée par défaut à l’outil *Ventouse*) et *valve2*.

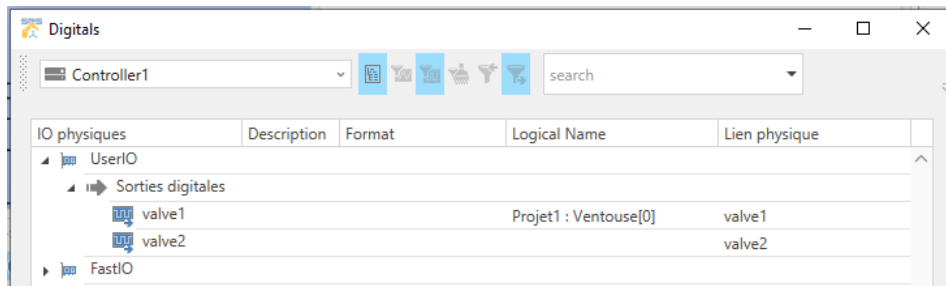


Figure 27 : Fenêtre **Digitals** dépliée de manière à faire apparaître les sorties *valve1* et *valve2*.

Le fait de double cliquer sur *valve1* permet de valider sa liaison avec l’outil *Ventouse*. Il en résulte la fenêtre **Propriétés** (relative à l’outil *Ventouse*), décrite dans la figure qui suit, où apparaît explicitement (au niveau du champ *ActionSignal*) le fait que *valve1* est bien la sortie digitale (du contrôleur *Controller1*) qui va actionner l’outil *Ventouse*.



Figure 28 : Configuration de **Enable Trigger**.

Ce qui été fait concernant le signal **Enable Trigger** doit être également fait pour le signal **Disable Trigger** (en lien avec l’instruction *open(Ventouse)*). Il en résulte la fenêtre **Propriétés** suivante :



Figure 28 (bis) : Configuration de **Enable Trigger** et **Disable Trigger**.

▪ **Concernant la variable *vitesse***

Sélectionnez, dans un premier temps, la fenêtre **Données** puis déployez *Projet1* jusqu’à obtenir ce qui est décrit dans la figure qui suit concernant la variable *mNomSpeed* de type *mdesc*.

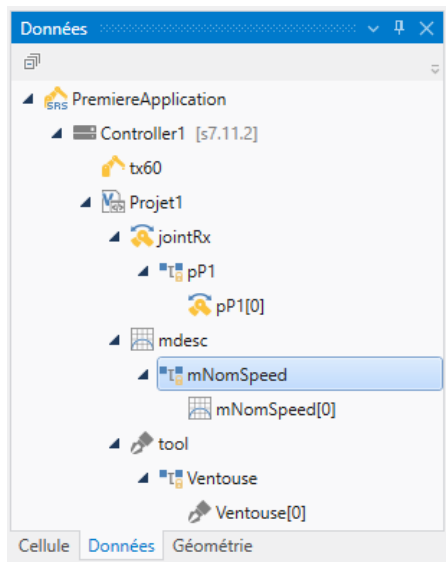


Figure 29 : Fenêtre **Données** relative à la variable *mNomSpeed* (de type *mdesc*).

Cliquez droit sur *mNomSpeed* (variable proposée par défaut), ce qui permet dans la fenêtre **Propriétés** de remplacer dans le champ **Nom** *mNomSpeed* par *vitesse*, ce qui donne lieu à la fenêtre **Données** décrite dans la figure qui suit. En fait, comme pour les variables *jj1* et *Ventouse*, la variable *vitesse* est considérée comme l'indice zéro du tableau *vitesse*, c'est-à-dire, *vitesse[0]*.

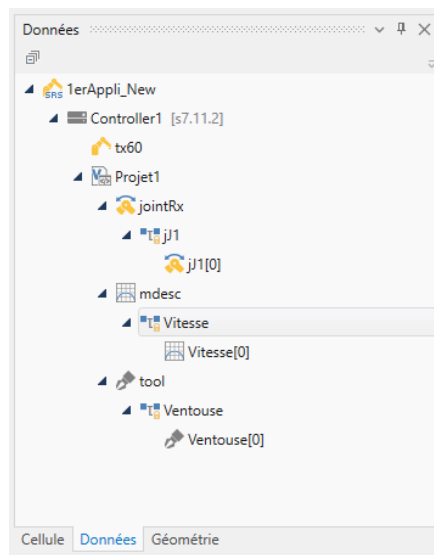


Figure 30 : Fenêtre **Données** relative à la variable *vitesse[0]* (de type *mdesc*).

⚠ Pensez à faire une sauvegarde ces changements !!

Une fois l'instruction :


`movej(jj1, Ventouse, vitesse)`

écrite avec ses trois paramètres correctement déclarés et assignés, il est possible de vérifier sa syntaxe en cliquant sur l'icône **Vérifier la syntaxe** du menu **VAL3**, ce qui provoque un listage des erreurs/avertissements/messages éventuels dans la fenêtre **Liste des erreurs** (située en bas à gauche).

✓ **A vous d'écriture le reste du programme (donné au §3.2) en vérifiant que les instructions que vous tapez au fur et à mesure ne contiennent pas d'erreur de syntaxe !!**

3.3.3) Exécution du programme

L'exécution du programme réclame d'intervenir : dans le menu **Panneau de contrôle (Control Pad)**, puis dans les fenêtres **Cellule** et **Controller1[s7.11.2]:tx60**.

- Dans le menu **Control Pad**, cliquez sur l'icône  (Démarrer) (situé en haut à gauche) ce qui a pour effet de faire apparaître une fenêtre **Control pad Status [Exécution en cours]** (dans laquelle vous ne devriez pas avoir à intervenir) qui renseigne l'utilisateur sur l'état des différents composants de la cellule durant la simulation.
- Dans la fenêtre **Cellule** (ou de manière équivalente, **Données** ou **Géométrie**), faire un clic droit sur *Projet1* pour sélectionner **Démarrer l'application**. Une fenêtre **Controller1[s7.11.2]:tx60** apparaît alors, à l'image de la figure qui suit.

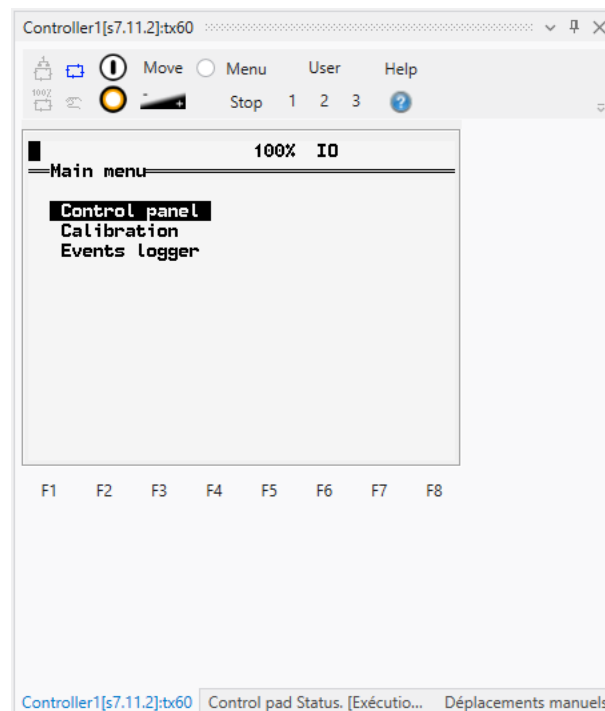


Figure 31 : Fenêtre **Controller1[s7.11.2]:tx60**.


- Dans la fenêtre **Controller1[s7.11.2]:tx60**



Le mode de travail (*Working mode*) permet de spécifier les conditions de fonctionnement du robot. Le mode sélectionné, parmi quatre options, est indiqué en haut à gauche de la fenêtre. Dans le cas présent, le mode doit être *Automatique* afin que le robot puisse être contrôlé de manière autonome *via* un programme ; ce mode est représenté graphiquement par un rectangle bleu muni de flèches, comme illustré dans la figure qui suit.



Figure 32 : Sélection du mode de travail *Automatique*.


Cliquez autant de fois que nécessaire sur l'icône  (situé en haut à gauche de la fenêtre) pour atteindre le mode **Automatique**.

Remarque : Il existe également un mode de travail dit *Manuel*  qui permet de contrôler le robot à l'aide du *Teach* (pour notamment enregistrer des *points*) ou d'exécuter un programme pour autant que l'opérateur maintienne la touche *Move/Hold* appuyée (les déplacements sont inférieurs à 250 mm/sec).

Une fois le mode *Automatique* sélectionné, on peut mettre la puissance sur le robot. Pour cela, cliquez sur l'icône  (situé en haut à gauche de la fenêtre) lequel se met à clignoter pour se transformer au bout de quelques secondes en  ce qui indique que la puissance est effectivement mise sur le robot.

N.B. : Si vous n'arrivez pas, en suivant cette procédure, à mettre la puissance sur le robot, alors, dans le menu **Cellule**, cliquez droit sur *Controller1*, puis sélectionnez sur **Redémarrer l'émulateur**.

On peut alors appuyer sur **Move** (bouton situé à droite de ) pour exécuter le programme et ainsi mettre le robot en mouvement dans la fenêtre **Vue 3D**.

Avant de relancer le programme, il faut penser à stopper la simulation afin de remettre la cellule dans son état initial, notamment la pièce. Pour cela, cliquez sur l'icône  (Stop) (située en haut à gauche) dans le menu **Control Pad**.

✓ **A vous d'exécuter le programme *Projet1* que vous avez écrit sachant que le résultat attendu devrait être comparable à celui donné dans la vidéo mentionnée à la page 1 du document !!**