

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

## Guide de l'étudiant

Robot Fanuc LR Mate 200iC

ROBOGUIDE

Langage KAREL

Par

Patrick McFridge et Luc Baron



## **TABLE DES MATIÈRES**

CHAPITRE 1 : Introduction .....	1
CHAPITRE 2 : Mesure sécurité .....	2
2.1. Précautions générales .....	2
2.2. Précaution lors de l'utilisation du robot en mode manuel.....	2
2.3. Précaution lors de l'utilisation du robot en mode automatique.....	2
CHAPITRE 3 : Robot réel – LR Mate 200iC .....	3
3.1. Schéma du LR MATE 200IC.....	3
3.2. Paramètres de Dénavit-Hartenberg .....	6
3.2.1. Cas particulier du LR-Mate 200IC.....	7
CHAPITRE 4 : Logiciel de simulation –ROBOGUIDE.....	9
4.1. Introduction à ROBOGUIDE .....	9
4.1.1. Organisation des fenêtres du programme.....	10
4.2. Tableau synthèse des boutons.....	11
4.2.1. Structure de la cellule de travail.....	12
4.3. Instruction des commandes.....	13
4.3.1. Outil de création.....	13
4.3.2. Déplacer le robot.....	16
4.3.3. Utilisation de l'outil de mesure .....	21
4.3.4. Comment exporter le programme au robot.....	22
CHAPITRE 5 : Langage KAREL.....	24
5.1. Création et utilisation d'un programme Karel dans ROBOGUIDE .....	24
5.2. Structure de programmation .....	25
5.2.1. Structure de base du programme .....	25
5.2.2. Type de variable prédéfinie.....	29
5.2.3. Structure de la commande de condition IF .....	30
5.2.4. Structure des boucles.....	31
5.3. Langage de programmation.....	32
5.3.1. Commande de programmation .....	32
5.3.2. Variables du système .....	35
5.4. Exemple de programmes .....	36
CHAPITRE 6 : ANNEXE .....	38
6.1. Tutoriel pour ROBOGUIDE.....	38
6.1.1. «Pick and Place».....	38
CHAPITRE 7 : Erreurs fréquentes .....	45
7.1. Problèmes de configuration de l'ordinateur.....	45
7.1.1. Changer le type de diviseur de décimale .....	45
7.1.2. Changer l'emplacement du fichier de sauvegarde .....	46
7.1.3. Appliquer les valeurs DCS .....	47
7.2. Déboguer le boîtier de commande virtuel .....	50

## **LISTE DES TABLEAUX**

TABLEAU No. 1:	Détails du robot FANUC LR Mate 200iC .....	6
TABLEAU No. 2:	Paramètres de Dénavit-Hartenberg du LR-Mate 200IC.....	7
TABLEAU No. 3:	Légende des différentes fenêtres du programme ROBOGUIDE.....	11
TABLEAU No. 4:	Description des différents boutons de ROBOGUIDE .....	11
TABLEAU No. 5:	Liste des directives du compilateur .....	27
TABLEAU No. 6:	Type de variables possible pour le langage KAREL.....	29
TABLEAU No. 7:	Liste de quelques commandes du langage KAREL.....	32
TABLEAU No. 8:	Liste des variables de quelque variable du système .....	35

## **LISTE DES FIGURES**

FIGURE No. 1:	Dimension du robot FANUC LR Mate 200iC .....	3
FIGURE No. 2:	Vue isométrique du robot FANUC LR MATE 200iC .....	4
FIGURE No. 3:	Dimension de la base du robot FANUC LR MATE 200iC.....	4
FIGURE No. 4:	Dimension du poignet du robot FANUC LR MATE 200iC.....	5
FIGURE No. 5:	Dimension du contrôleur du R-30iC .....	5
FIGURE No. 6:	Axes de référence des joints du robot à la position initiale .....	7
FIGURE No. 7:	Indépendance du joint 2 .....	8
FIGURE No. 8:	Présentation des différentes fenêtré du programme ROBOGUIDE.....	10
FIGURE No. 9:	Navigateur de procédé.....	14
FIGURE No. 10:	Arborescence .....	15
FIGURE No. 11:	Boitier de commande .....	16
FIGURE No. 12:	Boutons de déplacement sur le boitier de commande .....	17
FIGURE No. 13:	Option avancée pour le déplacement du robot sur ROBOGUIDE.....	18
FIGURE No. 14:	Contrôle avancé de la manipulation du robot sur ROBODUIDE.....	19
FIGURE No. 15:	Barre d'outils de coordonnées sur ROBOGUIDE.....	20
FIGURE No. 16:	Manipulation des joints du robot sur ROBOGUIDE.....	20
FIGURE No. 17:	Barred'outil «Move to» .....	21
FIGURE No. 18:	Fenêtré de l'outil de mesure.....	22
FIGURE No. 19:	Export d'un fichier de l'ordinateur vers le robot étape 1 .....	23
FIGURE No. 20:	Export d'un fichier de l'ordinateur vers le robot étape 2 .....	23
FIGURE No. 21:	Chemin pour ouvrir un nouveau fichier KAREL.....	24
FIGURE No. 22:	Fenêtré de l'éditeur de programme KAREL.....	25
FIGURE No. 23:	Outil de manipulation du point fixe du robot.....	41
FIGURE No. 24:	Bouton pour rouler un programme.....	43
FIGURE No. 25:	Fenêtré de «region et langue» du panneau de configuration .....	45
FIGURE No. 26:	Fenêtré des options avancés du format des nombres .....	46
FIGURE No. 27:	Fenêtré pour changer la trajectoire de sauvegarde des documents sur ROBOGUIDE.....	47
FIGURE No. 28:	Trajet pour appliquer un DCS – étape 1.....	48
FIGURE No. 29:	Trajet pour appliquer un DCS - étape 2 .....	48
FIGURE No. 30:	Appliquer un DCS.....	49
FIGURE No. 31:	Confirmation de l'application du DCS.....	50
FIGURE No. 32:	Procédure pour le déboguage du boitier de commande virtuel.....	50

## **LISTES DES EXEMPLES**

EXEMPLE No. 1:	Structure de base du programme.....	26
EXEMPLE No. 2:	Structure de la déclaration de constantes .....	28
EXEMPLE No. 3:	Structure de la déclaration de type .....	28
EXEMPLE No. 4:	Structure de la récupération des données d'une variable située à l'intérieur d'une structure .....	28
EXEMPLE No. 5:	Structure de la déclaration de variables.....	29
EXEMPLE No. 6:	Structure de la commande IF .....	31
EXEMPLE No. 7:	Première syntaxe d'une boucle FOR .....	31
EXEMPLE No. 8:	Deuxième syntaxe d'une boucle FOR.....	31
EXEMPLE No. 9:	Syntaxe d'une boucle WHILE .....	32
EXEMPLE No. 10:	Déplacer le robot vers une position prédéfinie.....	36
EXEMPLE No. 11:	Ajout de nœuds dans une variable de type PATH .....	37

# **CHAPITRE 1 : INTRODUCTION**

Les robots sont de plus en plus présents dans les industries afin d'augmenter leur production et parfois pour augmenter la précision de certaines manipulations. Ce qui fait du robot un outil très puissant et utile. Pour assurer une bonne répétabilité et une bonne précision des manipulations du robot, celui-ci doit être programmé par l'utilisateur.

Dans ce guide vous trouverez les informations nécessaires pour faire fonctionner un robot Fanuc LR Mate 200iC. Plus précisément, le guide est divisé en quatre parties. Il y aura tout d'abord une présentation des différentes mesures de sécurité. Ensuite, il y aura une description du robot ainsi que les indications nécessaires pour le démarrer et le manipuler. Par la suite, il y aura une brève description du logiciel de simulation ROBOGUIDE. Enfin, il y aura une explication des notions de base du langage de programmation KAREL.

De plus, en annexe se trouvent des notions supplémentaires sur le robot réel, ROBOGUIDE et le langage Karel. L'annexe contient aussi des tutoriels et un répertoire des erreurs fréquentes ainsi que leurs résolutions.

## **CHAPITRE 2 : MESURE SÉCURITÉ**

Avant d'utiliser le robot, il est bien important de connaître les mesures de sécurité afin d'éviter les blessures qui peuvent être mortelles.

### **2.1.Précautions générales**

- S'assurer que tous les dispositifs de sécurité sont présents et actifs
- Connaître où sont les arrêts d'urgence
- Ne jamais porter de cravate, de collier et d'autres bijoux ou vêtements pouvant se prendre dans le robot en mouvement, lorsque vous êtes dans l'enveloppe de travail du robot
- Vérifier que l'environnement de travail soit propre et sans débris indésirable. Si c'est le cas, contactez le technicien ou le professeur pour lui en aviser.
- Personne ne doit entrer dans l'enveloppe de travail du robot lorsqu'il est en marche, sauf pour enseigner des positions de travail. Dans ce cas, seule la personne avec le boîtier de commande a le droit d'entrer dans l'enveloppe de travail du robot.
- Ne jamais désactiver un dispositif de sécurité

### **2.2.Précaution lors de l'utilisation du robot en mode manuel**

- Déplacer le robot à vitesse réduite lors des manipulations en mode manuel
- S'assurer d'être à l'extérieur de la cage de sureté avant de déplacer le robot.

### **2.3.Précaution lors de l'utilisation du robot en mode automatique**

Pour utiliser le robot en mode automatique, vous devez obligatoirement fermer les portes de la cage du robot, sinon le robot restera immobile.

Lors de la première exécution d'un programme sur le robot réel, veuillez :

1. Effectuer la trajectoire étape par étape à vitesse réduite.
2. Effectuer la trajectoire au complet à vitesse réduite
3. Effectuer la trajectoire au complet à vitesse normale



## CHAPITRE 3 : ROBOT RÉEL – LR MATE 200iC

### 3.1.Schéma du LR MATE 200iC

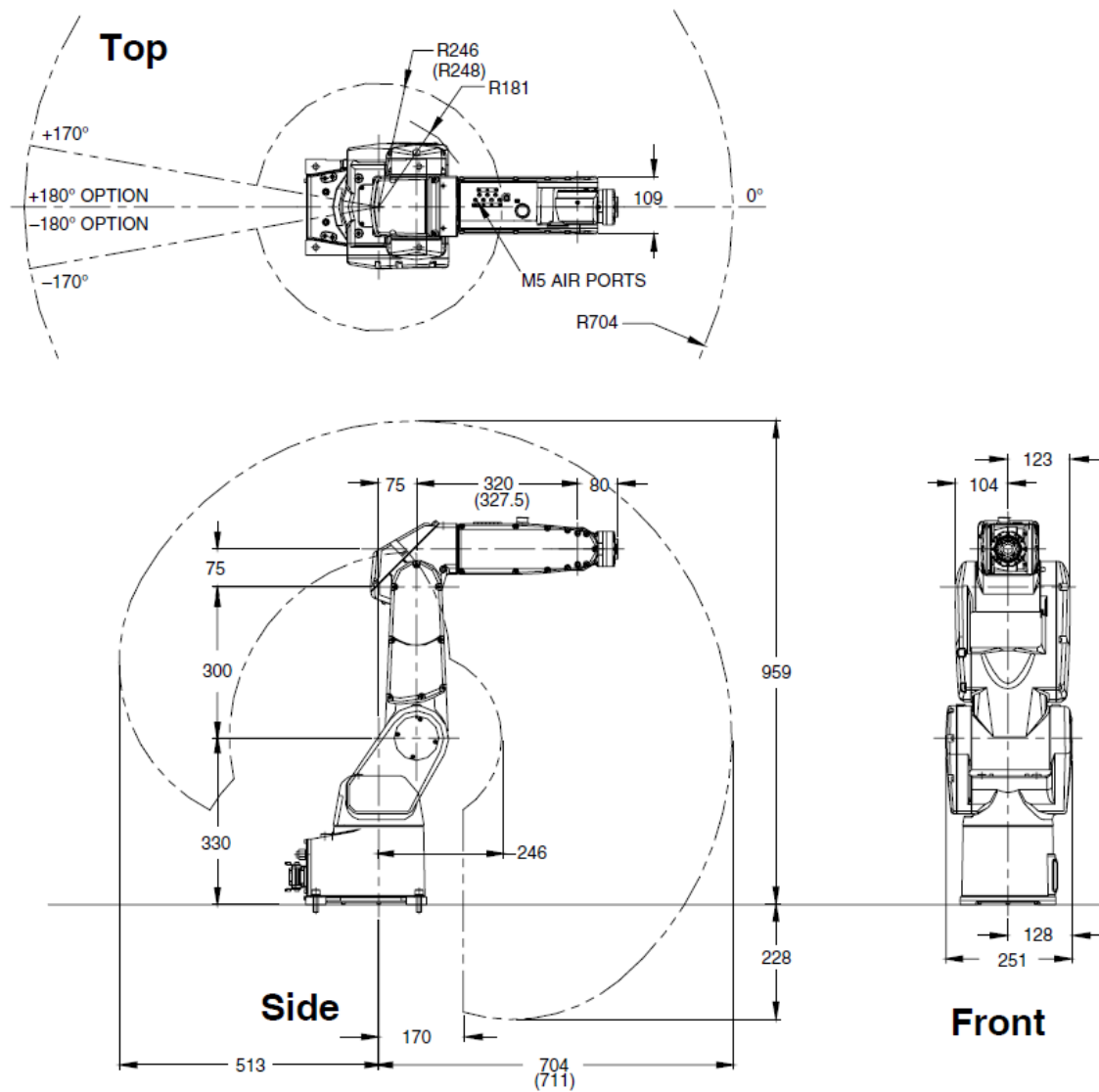
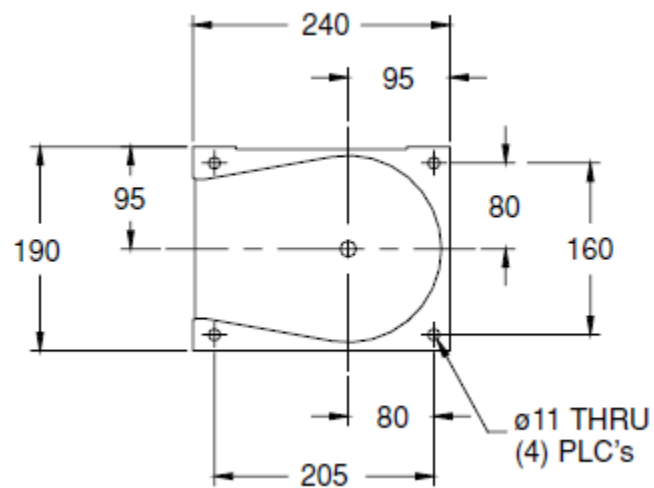


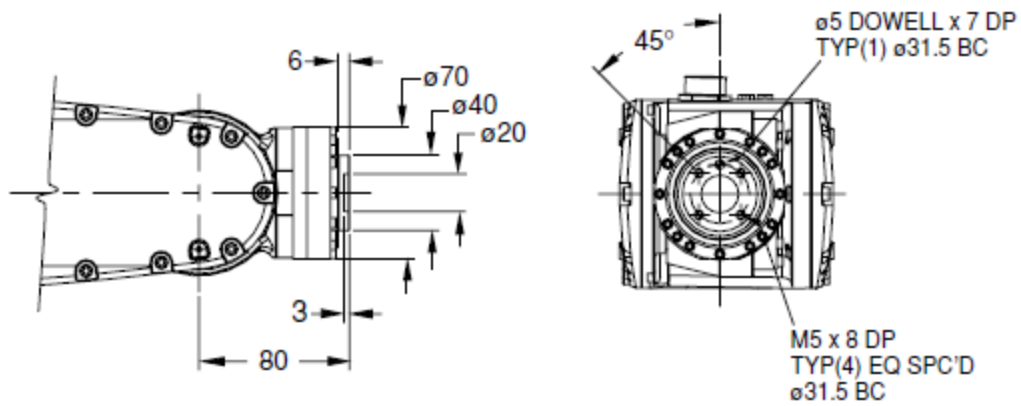
FIGURE No. 1: Dimension du robot FANUC LR Mate 200iC



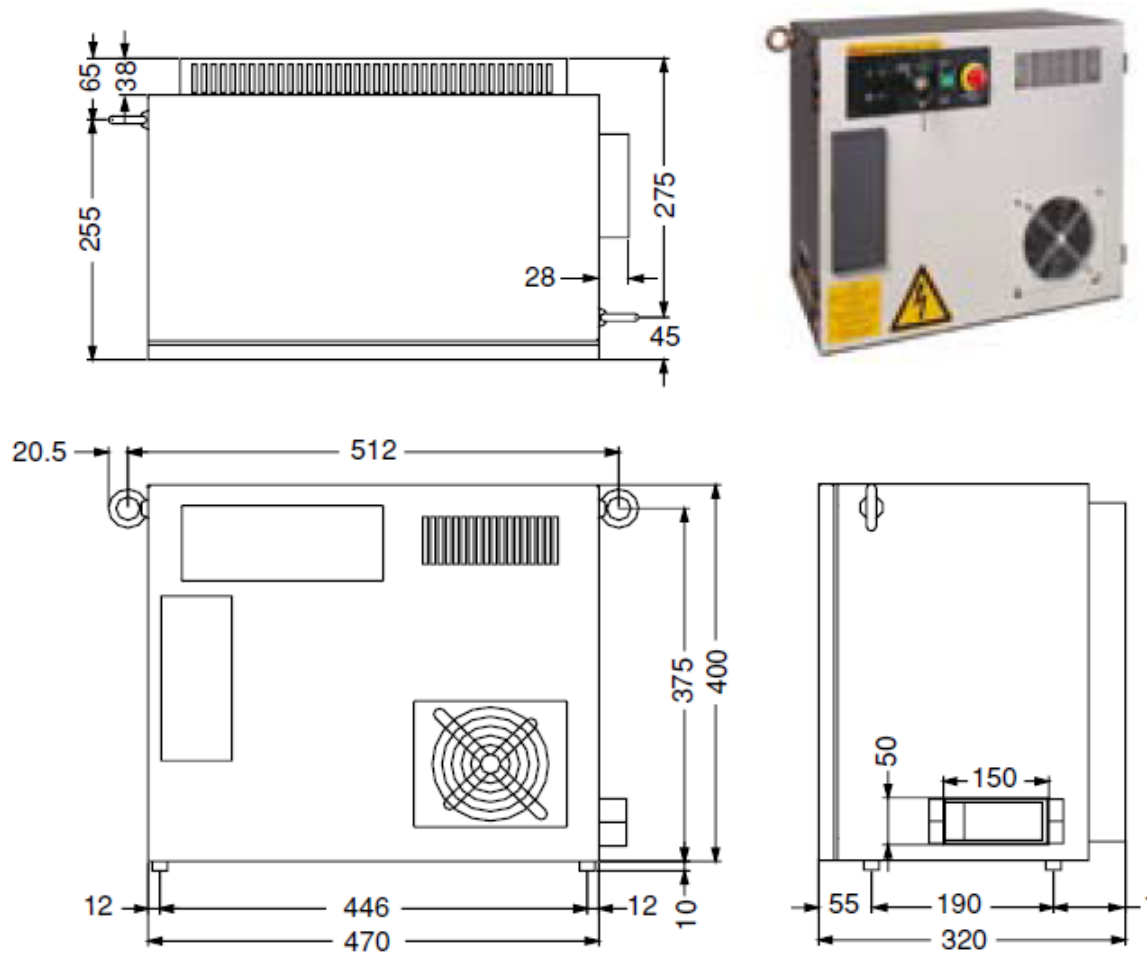
**FIGURE No. 2: Vue isométrique du robot FANUC LR MATE 200iC**



**FIGURE No. 3: Dimension de la base du robot FANUC LR MATE 200iC**



**FIGURE No. 4: Dimension du poignet du robot FANUC LR MATE 200iC**

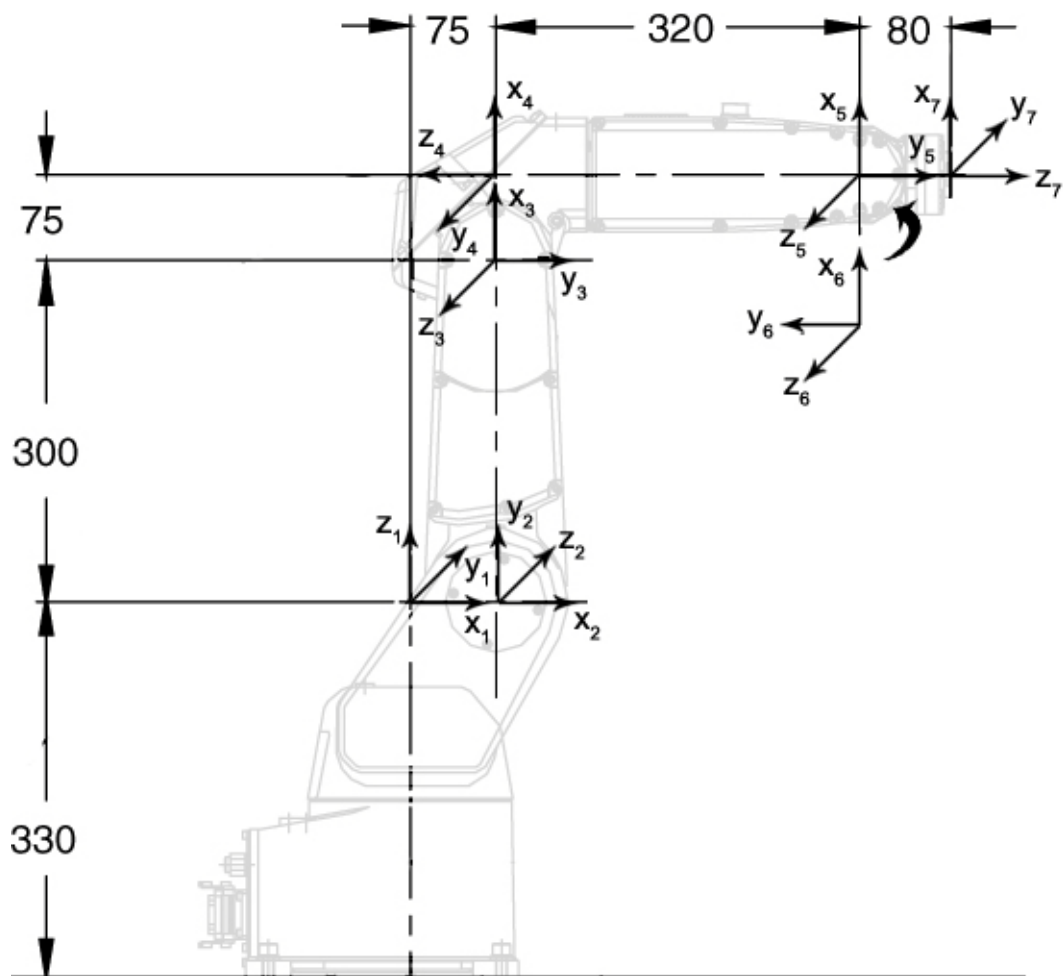


**FIGURE No. 5: Dimension du contrôleur du R-30iC**

**TABLEAU No. 1: Détails du robot FANUC LR Mate 200iC**

<b>Axes</b>		6
<b>Poids maximal de la charge (kg)</b>		5
<b>Rayon de l'enveloppe de travail sans outil (mm)</b>		704
<b>Répétabilité (mm)</b>		±0.02
<b>Rayon d'interférence (mm)</b>		181
<b>Degré de malléabilité des joints (degrés)</b>	<b>J1</b>	340
	<b>J2</b>	200
	<b>J3</b>	388
	<b>J4</b>	380
	<b>J5</b>	240
	<b>J6</b>	720
<b>Vitesse des joints (degrés/sec)</b>	<b>J1</b>	350
	<b>J2</b>	350
	<b>J3</b>	400
	<b>J4</b>	450
	<b>J5</b>	450
	<b>J6</b>	720
<b>Moment du poignet (N*m)</b>	<b>J4</b>	11.9
	<b>J5</b>	11.9
	<b>J6</b>	6.7
<b>Inertie de la charge du poignet (kg*m<sup>2</sup>)</b>	<b>J4</b>	0.3
	<b>J5</b>	0.3
	<b>J6</b>	0.1
<b>Poids du robot sans charge (kg)</b>		27

### 3.2.Paramètres de Dénavit-Hartenberg



**FIGURE No. 6: Axes de référence des joints du robot à la position initiale**

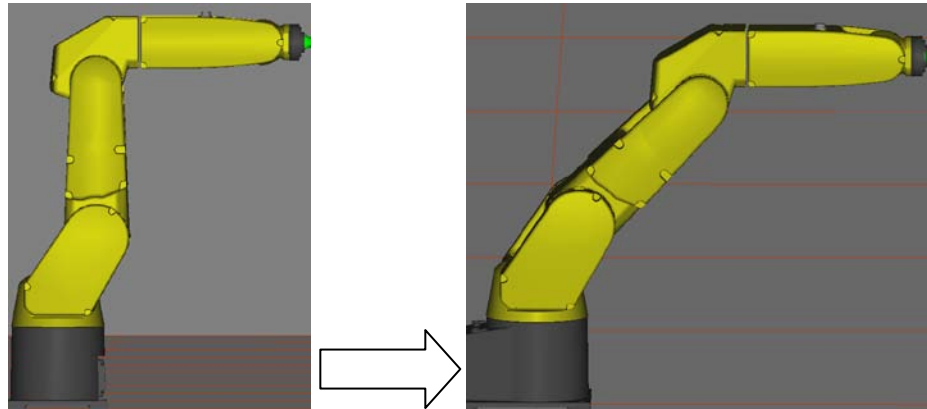
**TABEAU No. 2: Paramètres de Dénavit-Hartenberg du LR-Mate 200iC**

i	$a_i$	$b_i$	$\alpha_i$
1	75	0	-90
2	300	0	180
3	75	0	-90
4	0	-320	90
5	0	0	-90
6	0	-80	180

### 3.2.1. Cas particulier du LR-Mate 200iC

Le LR-MATE 200iC ne respecte pas complètement les conventions de Dénavit-Hartenberg. En effet, la figure ci-dessus montre la position initiale où tous les joints sont mis à zéro. Pourtant, l'angle  $\theta_2$ , soit l'angle entre  $x_2$  et  $x_3$  est de  $-90^\circ$ . Par conséquent, il y aura toujours un décalage de  $-90^\circ$  entre la valeur conventionnelle de  $\theta_2$  et la valeur apparente sur le boîtier de commande du robot réelle.

De plus, le joint 2 du robot est indépendant des autres joints. C'est-à-dire que si le joint 2 change d'orientation, le joint 3, 4, 5, 6 vont rester parallèles à leur position de départ, comme le montre la figure ci-dessous



**FIGURE No. 7: Indépendance du joint 2**

## **CHAPITRE 4 : LOGICIEL DE SIMULATION –ROBOGUIDE**

### **4.1.Introduction à ROBOGUIDE**

Le logiciel ROBOGUIDE est spécialement conçu pour contrôler des robots de marque FANUC. À l'intérieur de ce logiciel, il est possible de simuler une cellule de travail et de programmer la trajectoire d'un robot. Les données compilées par le logiciel de simulation peuvent ensuite être transférées au robot réel afin d'effectuer une tâche réelle.

Avant de présenter les notions pour la conception d'une cellule de travail et les notions de la programmation, l'environnement et la structure de travail du logiciel ROBOGUIDE seront présentés.

De plus, pour approfondir la compréhension au logiciel, un tutoriel est disponible à la fin du document, ainsi qu'une rubrique «Erreurs fréquentes» pour vous dépanner lorsque certaines erreurs apparaissent lors de l'utilisation du programme.

#### 4.1.1. Organisation des fenêtres du programme

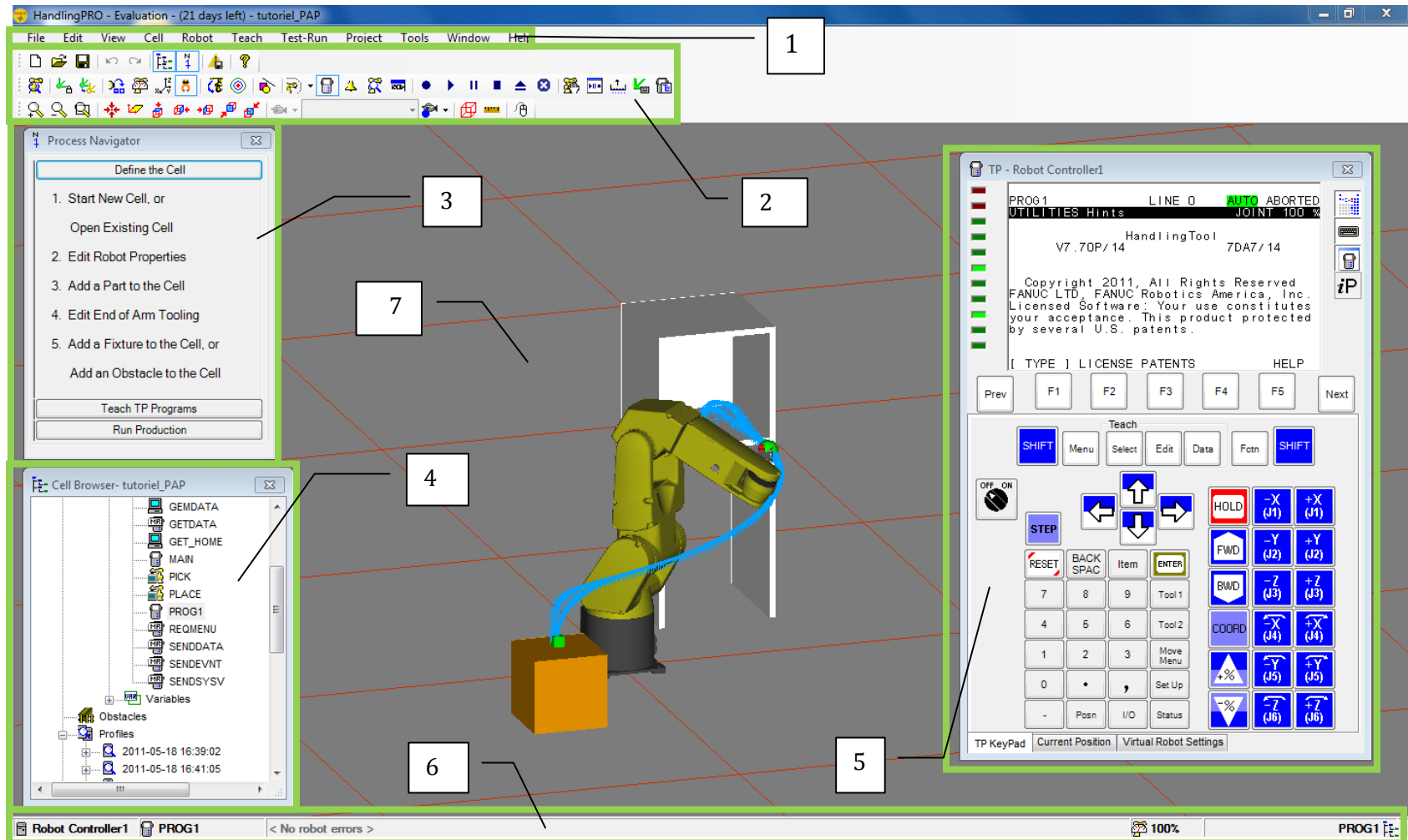


FIGURE No. 8: Présentation des différentes fenêtr du programme ROBOGUIDE

























**TABEAU No. 3: Légende des différentes fenêtres du programme ROBOGUIDE**

Numéro	Description
1	Barre du menu principal
2	Boutons des différentes fonctions du programme
3	«Progress Navigator», le navigateur de progrès
4	«Cell Browser», arborescence
5	«Teach Panel», simulation de la télécommande du robot.
6	Barre d'état, montrant les différents programmes et le contrôleur sélectionné
7	«workcell», cellule de travail, là où la représentation de l'environnement réel est présentée.

## 4.2. Tableau synthèse des boutons

**TABEAU No. 4: Description des différents boutons de ROBOGUIDE**

Icône	Description
	Créer un nouveau fichier
	Ouvrir un fichier déjà existant
	Retour à l'arrière (Undo)
	Retour à l'avant (Redo)
	Afficher / Masquer la fenêtre «Cell Browser»
	Afficher / Masquer la fenêtre «Process Navigator»
	Sauver les informations du système de ROBOGUIDE
	Afficher l'aide pour la fenêtre utilisée
	Exécuter le programme «MotionPRO»
	Verrouiller ou déverrouiller l'outil «UTOOL», ceci permet de bouger le point fixe du robot.
	Activer ou désactiver la répétition de la fonction «Move to»
	Afficher / Masquer la barre d'outils rapide pour modifier les coordonnées de référence du robot.
	Modifier la vitesse de la trajectoire du robot
	Afficher / Masquer la barre d'outils rapide pour enregistrer des coordonnées dans un programme de simulation.
	Ouvrir / Fermer l'outil au bout du bras du robot
	Afficher / Masquer la barre d'outils rapide pour la fonction «Move to»
	Afficher / Masquer la barre d'outils rapide pour les «Targets»
	Dessiner des nouvelles formes aux pièces
	Afficher / Masquer l'enveloppe de travail du robot
	Afficher / Masquer le «Teach Panel»
	Afficher / Masquer les alarmes du robot
	Afficher la page ayant les informations du robot

	Afficher l'invite de commande KAREL
	Enregistrer un fichier AVI de l'animation du robot.
	Rouler le programme sélectionné
	Mettre sur pause le programme
	Arrêter le programme
	Réinitialiser un programme
	Forcer à quitter le programme
	Manipuler les différents joints du robot
	Afficher / Masquer la fenêtre «Run Panel»
	Afficher / Masquer l'éditeur de position
	Afficher / Masquer la fenêtre d'option pour bouger et copier un objet
	Afficher les programmes qui sont appelés par le programme principale
	Zoom +
	Zoom -
	Zoom de la sélection
	Centrer la vue sur l'objet sélectionné
	Changer la direction de la caméra
	Vue de haut
	Vue du côté droit
	Vue de côté gauche
	Vue de face
	Vue de derrière
	Installer un point de vue
	Enregistrer le point de vue
	Afficher les objets en mode filaire
	Outil de mesure
	Afficher / masquer les options de la souris

#### 4.2.1. Structure de la cellule de travail

Le logiciel ROBOGUIDE organise les éléments selon différentes catégories :

- Le robot
- L'outil au bout du bras du robot (End of Arm Tooling)
- Les pièces (Part)
- Les fixations (Fixture)
- Les obstacles (Obstacle)

##### 4.2.1.1. *Le robot*

L'élément central de la cellule de travail (WorkCell) est le robot, car le but ultime du logiciel est de pouvoir contrôler un robot. À l'intérieur de la bibliothèque du logiciel, il est possible de choisir tous les robots FANUC mis sur le marché. Généralement, les robots sont fixe à la base et sont munis de bras pivotant afin d'effectuer des manipulations.

#### *4.2.1.2. L'outil au bout du bras du robot (End of Arm Tooling)*

Un autre élément essentiel est l'outil qui se trouve au bout du bras du robot (End of Arm Tooling), aussi appelé «EoAT». L'outil fait le lien entre le robot et les pièces (Part), il permet de prendre ou de relâcher une pièce.

Il y a plusieurs types d'outil, en voici quelques exemples :

- Pince (Grippers)
- Aspirateur (Vacuum)
- Torche à soudure (weldtorch)

#### *4.2.1.3. Les pièces (Part)*

Les pièces sont les éléments que l'on désire manipuler avec le robot. Les pièces peuvent être modélisées sur un logiciel 3D comme CATIA et être importées à l'intérieur de ROBOGUIDE.

#### *4.2.1.4. Les fixations (Fixture)*

Pour être en mesure de prendre ou de déposer une pièce dans le logiciel ROBOGUIDE, la pièce doit toujours être liée avec une fixation. Une fixation, aussi appelée «fixture» en anglais, est un objet sur lequel les pièces reposent.

#### *4.2.1.5. Les obstacles (Obstacle)*

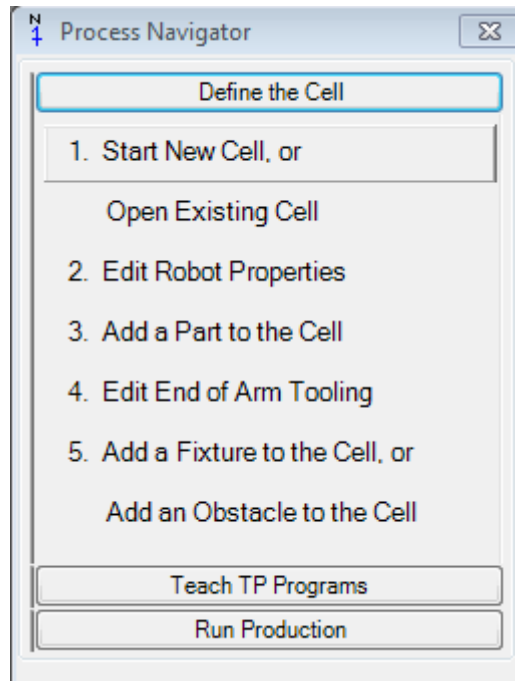
Les obstacles sont les éléments qui représentent les différents objets qui pourraient interférer avec la trajectoire du robot. Il est important d'incorporer les obstacles qui sont présents dans l'environnement de travail réel à l'intérieur du logiciel de simulations pour s'assurer que le robot n'entre pas en collision avec des objets.

### **4.3. Instruction des commandes**


#### **4.3.1. Outil de création**

##### *4.3.1.1. Navigateur de procédés (Process navigator)*

Le navigateur de procédés intitulé en anglais «Process navigator» est un menu où l'on retrouve les différentes étapes de conception d'une cellule de travail. Les étapes sont disposées dans l'ordre logique de création.



**FIGURE No. 9:   Navigateur de procédé**

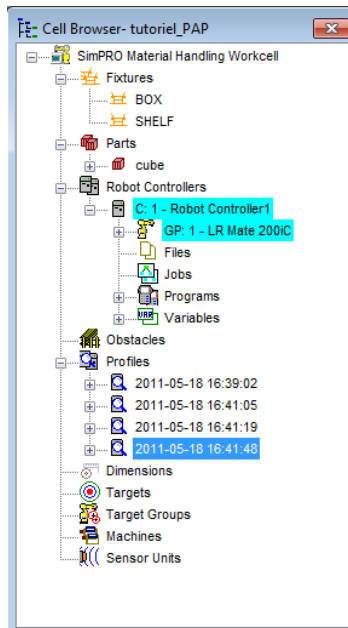
Pour accéder au navigateur de procédés, vous devez aller dans l'onglet «View» du menu principal et cliquez sur «Navigator». Vous pouvez aussi accéder au navigateur de procédés en cliquant directement sur le bouton .

Le navigateur de procédés est séparé en trois parties :


- «Define the Cell»,
  1. Création d'éléments pour garnir la cellule de travail, par exemple :
  2. Pièces
  3. Outils de robot
  4. Fixations
- «Teach TP Programs»
  1. Création d'un programme
- «Run Production»
  1. Rouler le programme
  2. Analyser son profil

#### **4.3.1.2. L'arborescence (*CellBrowser*)**

L'arborescence, appelée «Cell browser» en anglais est une fenêtre où l'on retrouve les différents éléments et les différents programmes de la cellule de travail. L'arborescence regroupe les éléments comme l'image ci-dessous.



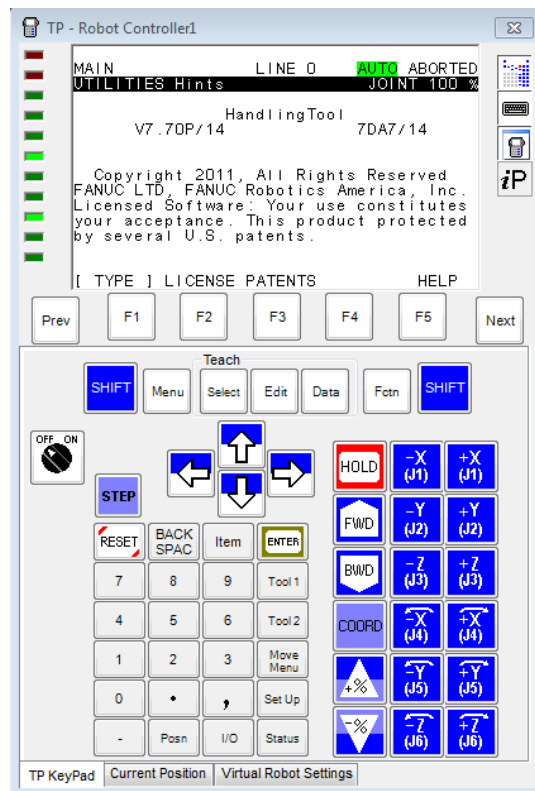
**FIGURE No. 10: Arborescence**

Pour accéder au navigateur de procédés, vous devez aller dans l'onglet «View» du menu principal et cliquez sur «Cell Browser». Vous pouvez aussi accéder au navigateur de procédés en cliquant directement sur le bouton .


De plus, vous pouvez ajouter ou supprimer un élément en choisissant une des options possibles lorsque vous cliquez droit sur un élément.

#### *4.3.1.3. Boitier de commande (Teach Pendant)*

Le boitier de commande appelée «Teach Pendant» en anglais est la reproduction à quelques détails près de la télécommande réelle du robot. Avec la télécommande, il est possible de déplacer le robot et de programmer une trajectoire.



**FIGURE No. 11: Boîtier de commande**

Pour accéder à la fenêtre de la télécommande virtuelle, vous devez cliquer sur le bouton .

L'utilisation de «Teach Pendant» sera expliquée plus loin dans le document.

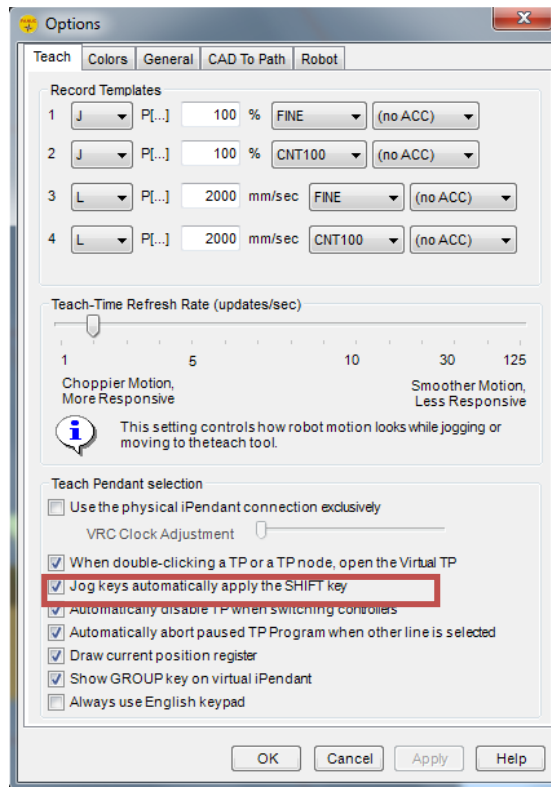
#### 4.3.2. Déplacer le robot

##### 4.3.2.1. *Déplacer le robot avec le boîtier de commande (Teach Pendant)*

Pour déplacer le robot à l'aide du boîtier de commande, vous devez premièrement l'activer en cliquant sur le bouton «ON». Deuxièmement, vous devez cliquer sur un des boutons encadrés ci-dessous tout en maintenant la touche «shift» enfoncée. Lorsque vous déplacez le robot à l'aide de ces commandes vous vous apercevrez que vous faites une rotation des différents joints du robot.

Pour changer le système de coordonnées, il faut cliquer sur «COORD». Les choix de système de coordonnées sont: JOINT, S/JGFRM, S/WORLD, S/TOOL, S/USER.

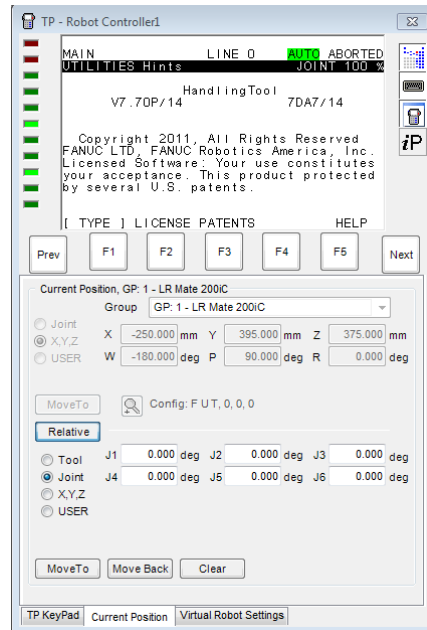




**FIGURE No. 13: Option avancée pour le déplacement du robot sur ROBOGUIDE**

Vous pouvez connaître l'orientation des différents joints lorsque vous cliquez sur l'onglet «Current Position» situé au bas de la télécommande. Si vous sélectionnez «X,Y,Z» au lieu de «Joint» vous obtiendrez les coordonnées cartésiennes du «TeachTool» par rapport à l'origine du robot. Si vous sélectionnez «USER» vous obtenez les coordonnées du «TeachTool» par rapport à un «UFrame» sélectionné, comme le montre la figure ci-dessous.







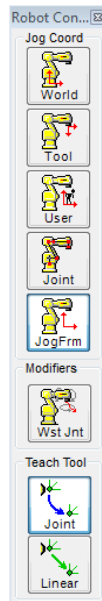
**FIGURE No. 14: Contrôle avancé de la manipulation du robot sur ROBODUIDE**

Vous pouvez aussi déplacer le robot relativement de la position actuelle en cliquant le bouton «Relative». Si vous choisissez l'option «Tool», vous modifiez l'emplacement du «TeachTool» relativement au positionnement actuel de l'outil. Si vous choisissez l'option «Joint», vous pouvez diminuer ou augmenter l'angle des joints. Si vous choisissez l'option «X,Y,Z», vous modifiez l'emplacement du «TeachTool» relativement à l'origine du robot. Finalement, si vous choisissez «USER», vous modifiez l'emplacement du «UTool» relativement à un «UFrame» sélectionné.

#### 4.3.2.2. Déplacer le robot avec l'outil «TeachTool»

Vous pouvez déplacer le robot en glissant manuellement le «TeachTool», représenté par une boule verte. Pour ce faire, vous devez cliquer sur le bouton  afin d'afficher les axes du «TeachTool». Ensuite, vous pouvez déplacer la boule verte en glissant la souris sur ses axes. Il est aussi possible de changer l'orientation du «TeachTool» en maintenant la touche «SHIFT» enfoncée et en glissant la souris sur ses axes. De plus, il est aussi possible de bouger librement la boule verte en maintenant la touche «Ctrl» enfoncée, tout en cliquant sur le «TeachTool» et tout en déplaçant la souris.


Vous pouvez changer le point de référence de déplacement en cliquant sur le bouton . Une fenêtre apparaîtra comme montrée ci-dessous. Ceci vous donnera l'option de choisir l'origine (World), l'outil (Tool), l'utilisateur (User), les joints (Joints) comme élément de référence pour le déplacement.

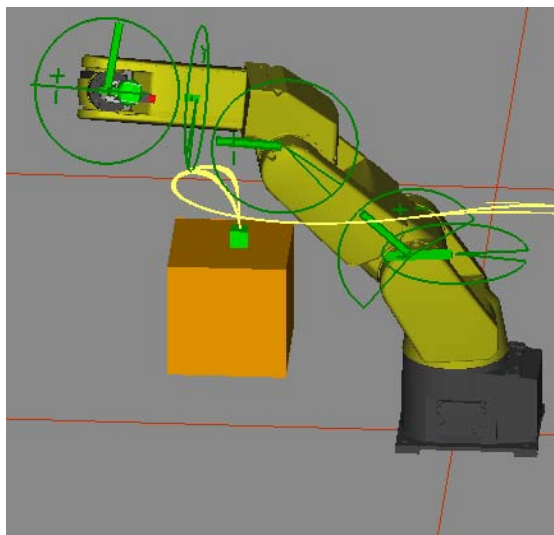


**FIGURE No. 15: Barre d'outils de coordonnées sur ROBOGUIDE**

ATTENTION. Si le robot ne se déplace pas à l'endroit où vous avez bougé le «TeachTool» c'est peut-être, parce que le point est hors de la portée du robot, ou bien que votre axe «Z» soit orienté dans la mauvaise direction.


#### *4.3.2.3. Déplacer le robot à l'aide des joints*

Vous pouvez déplacer le robot en modifiant manuellement l'orientation des joints. Pour ce faire, vous devez cliquer sur le bouton  et ensuite glisser votre souris sur les différents joints du robot comme montré ci-dessous.



**FIGURE No. 16: Manipulation des joints du robot sur ROBOGUIDE**

#### 4.3.2.4. Déplacer le robot automatiquement à l'aide de l'outil «Move to»


Il est possible de déplacer le robot automatiquement à l'aide de l'outil «Move to». Pour ce faire, vous devez cliquer sur le bouton . Une fenêtre, comme l'image ci-dessous, apparaîtra. Vous avez dans cette fenêtre l'option de diriger votre «TeachTool» sur une face (Face), un côté (Edge), un coin (Vertex), un arc de cercle (Arc Ctr) ou une pièce (Part).

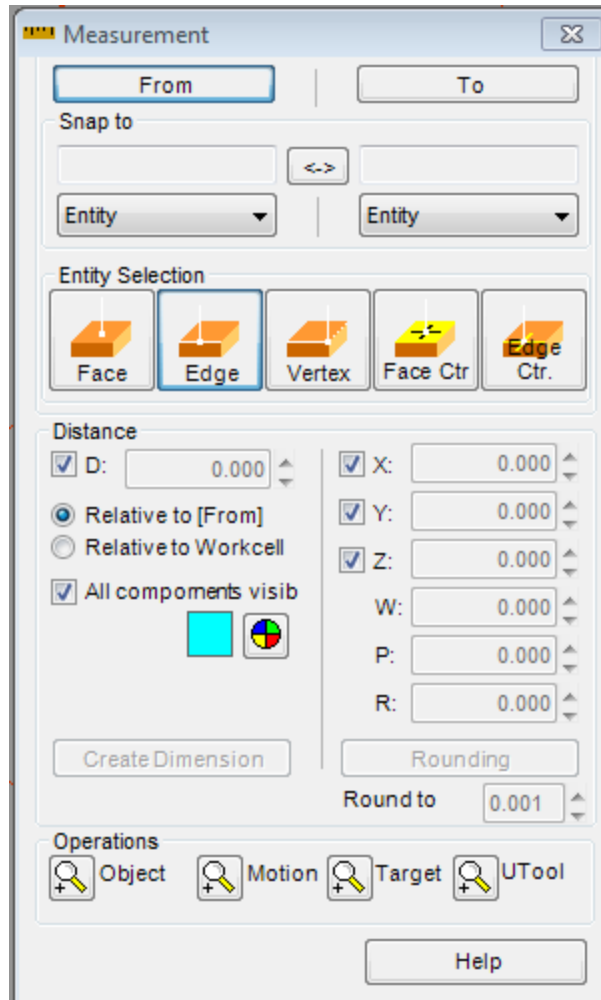


**FIGURE No. 17: Barred'outil «Move to»**

Pour utiliser cet outil, vous devez tout d'abord sélectionner une des options mentionnées ci-dessus, ensuite vous devez cliquer sur une région de votre cellule de travail qui correspond à l'endroit où vous voulez déplacer le bout du bras du robot.

#### 4.3.3. Utilisation de l'outil de mesure

Pour accéder à l'outil de mesure vous devez cliquer sur le bouton . Une fenêtre contenant les options de mesures comme montrées sur la figure ci-dessous s'ouvrira.



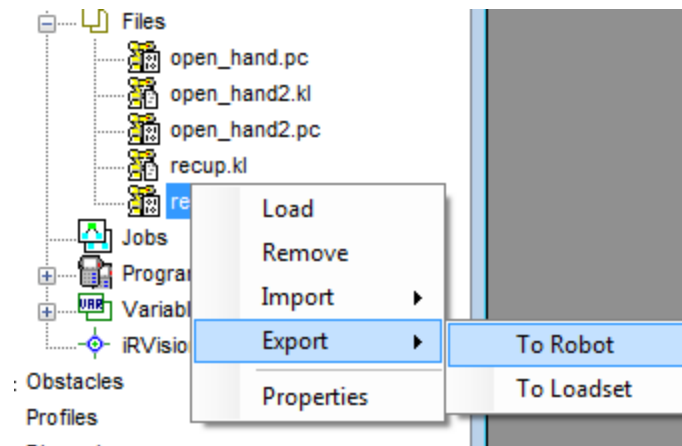
**FIGURE No. 18: Fenêtre de l'outil de mesure**

Pour mesurer la distance entre deux objets ou la longueur d'une pièce, il faut premièrement cliquer le bouton «From» et ensuite sélectionner le type de surface que vous désirez sélectionner : face, côté, arrête, centre d'une face ou le centre d'un côté. Après, vous devez sélectionner l'endroit où vous désirez positionner le point initial de la mesure.

Pour choisir le point final de la mesure, vous devez répéter les étapes mentionnées précédemment en choisissant les caractéristiques désirées pour le point final et en cliquant ensuite sur la position du point final.

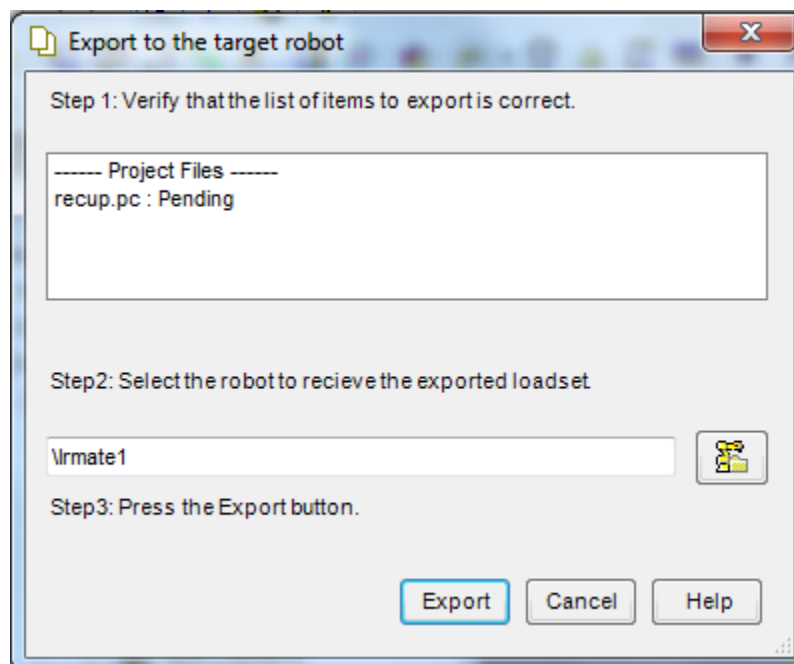
#### 4.3.4. Comment exporter le programme au robot

Pour exporter un fichier vers le robot réel, vous devez premièrement cliquer droit sur le fichier en question et ensuite cliquer sur «Export/To» Robot comme le montre la figure ci-dessous.



**FIGURE No. 19: Export d'un fichier de l'ordinateur vers le robot étape 1**

Une fenêtre, comme le montre la figure ci-dessous, apparaîtra. Il vous devrez, si ce n'est déjà pas fait, sélectionner le robot vers lequel vous voulez exporter le robot. Finalement, vous devez cliquer sur «Export» et le fichier devrait normalement être envoyé au robot.



**FIGURE No. 20: Export d'un fichier de l'ordinateur vers le robot etape 2**

## CHAPITRE 5 : LANGAGE KAREL

Le langage de programmation KAREL est un langage spécialement créé pour les robots FANUC. Si vous avez déjà programmé en PASCAL vous verrez plusieurs ressemblances dans la structure de programmation.

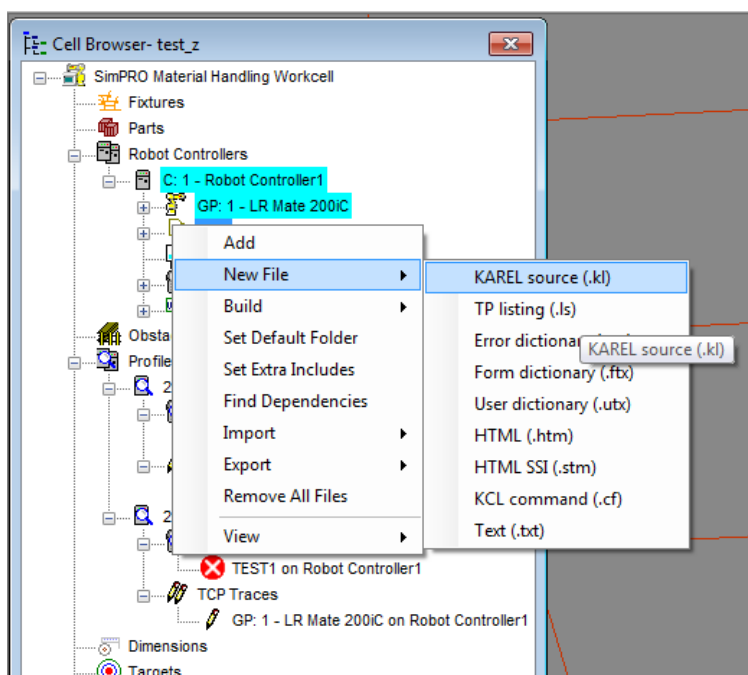
Afin de maîtriser le langage KAREL, ce chapitre comportera plusieurs sections :

- Création et utilisation d'un programme KAREL dans ROBOGUIDE
- Structure de programmation
- Commandes de programmation
- Exemple de programme

### 5.1.Création et utilisation d'un programme Karel dans ROBOGUIDE

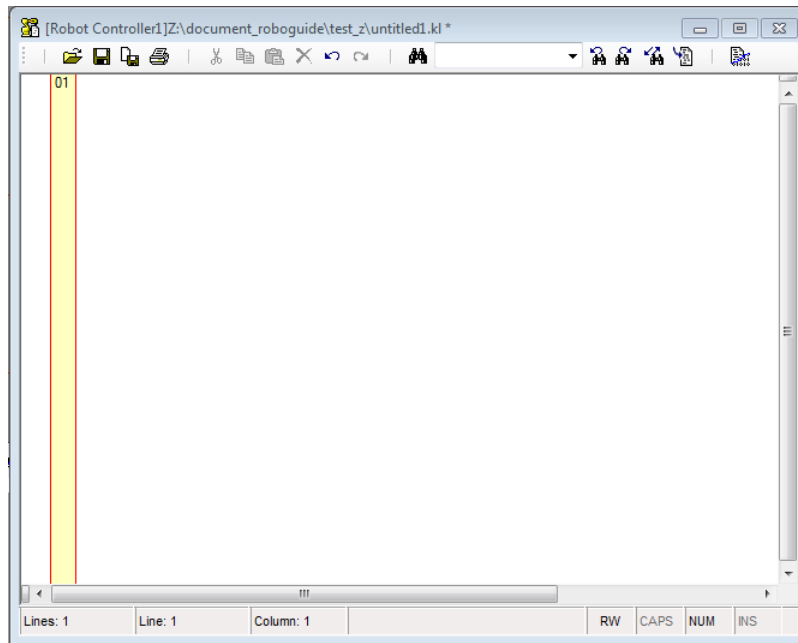
Le programme KAREL peut être écrit à l'intérieur du programme ROBOGUIDE. En effet, un éditeur pour KAREL est incorporé à l'intérieur du programme ROBOGUIDE.

Pour ouvrir un nouveau fichier KAREL il faut aller dans le «file browser» et cliquez droit sur «file» et sélectionnez «New file» et cliquez sur «KAREL source (.kl)» comme le montre la figure NO ci-dessous.





**FIGURE No. 21: Chemin pour ouvrir un nouveau fichier KAREL**

Ceci fait l'éditeur de programmation KAREL s'ouvrira, comme le montre la figure ci-dessous.



**FIGURE No. 22: Fenêtre de l'éditeur de programme KAREL**

Un fichier d'extension «.kl» est alors créé. Par défaut le nom du fichier est «untitled1.kl». Vous pouvez changer le nom du fichier en cliquant sur le bouton . Vous pouvez aussi compiler le programme que vous avez écrit en cliquant sur le bouton . Lorsque vous compilez le programme, un fichier d'extension «.pc» ayant le même nom que votre fichier d'extension «.kl» sera créé. Le fichier d'extension «.pc» est le fichier qui est lu par le contrôleur du robot.

Pour exécuter le programme dans le logiciel de simulation de ROBOGUIDE afin de voir le robot bougé sur l'écran, il faut passer par le boîtier de commande virtuelle. La procédure est la suivante :

- Allumer le boîtier de commande en cliquant sur «ON»
- Allez dans le répertoire des programmes en cliquant sur «Select»
- Choisir le programme désiré ayant le nom du programme désiré et appuyer sur la touche «ENTER». Le programme sera maintenant sélectionné.
- Maintenez appuyer la touche «shift» tout en cliquant sur le bouton «FWD» pour exécuter le programme.
- Le programme devrait maintenant démarrer s'il n'y a pas d'erreur.

## **5.2.Structure de programmation**

### **5.2.1. Structure de base du programme**

### EXEMPLE No. 1: Structure de base du programme

**PROGRAM** nom\_fichier

*Directive du compilateur*

**CONST**

Déclaration de constantes

**TYPE**

Déclaration de types

**VAR**

Déclaration de variables

**BEGIN**

*Énoncés exécutables*

**END** nom\_fichier

#### 5.2.1.1. Directive du compilateur

Les directives du compilateur servent à diriger la compilation du programme. Plus précisément ils servent à :

- Inclure d'autre fichier dans le programme lors de la compilation
- Spécifier le programme et l'attribut des tâches



**TABLEAU No. 5: Liste des directives du compilateur**

<b>Directive</b>	<b>Description</b>
%ALPHABETIZE	Specifies that variables will be created in alphabetical order when p-code is loaded.
%CMOSVARS	Specifies the default storage for KAREL variables is CMOS RAM.
%CMOS2SHADOW	Instructs the translator to put all CMOS variables in SHADOW memory.
%COMMENT = 'comment'	Specifies a comment of up to 16 characters. During load time, the comment is stored as a program attribute and can be displayed on the SELECT screen of the teach pendant or CRT/KB.
%CRTDEVICE	Specifies that the CRT/KB user window will be the default in the READ and WRITE statements instead of the TPDISPLAY window.
%DEFGROUP = n	Specifies the default motion group to be used by the translator.
%DELAY	Specifies the amount of time the program will be delayed out of every 250 milliseconds.
%ENVIRONMENT filename	Used by the off-line translator to specify that a particular environment file should be loaded.
%INCLUDE filename	Specifies files to insert into a program at translation time.
%LOCKGROUP =n,n	Specifies the motion group(s) locked by this task.
%NOABORT = option	Specifies a set of conditions which will be prevented from aborting the program.
%NOBUSYLAMP	Specifies that the busy lamp will be OFF during execution.
%NOLOCKGROUP	Specifies that no motion groups will be locked by this task.
%NOPAUSE = option	Specifies a set of conditions which will be prevented from pausing the program.
%NOPAUSESHFT	Specifies that the task is not paused if the teach pendant shift key is released.
%PRIORITY = n	Specifies the task priority.
%SHADOWVARS	Specifies that all variables by default are created in SHADOW.
%STACKSIZE = n	Specifies the stack size in long words.
%TIMESLICE = n	Supports round-robin type time slicing for tasks with the same priority.
%TPMOTION	Specifies that task motion is enabled only when the teach pendant is enabled.
%UNINITVARS	Specifies that all variables are by default uninitialized.

#### 5.2.1.2. Déclaration de constantes

La déclaration de constantes est utile si vous réutilisez souvent une même valeur dans un programme et que vous ne voulez pas la changer à tous les endroits où celle-ci est utilisée lorsque vous voulez la modifier. En effet, vous aurez seulement à changer la valeur de la constante dans la balise CONST et la valeur sera traduite à tous les endroits où vous avez appelé la constante. L'exemple ci-dessous présente la structure de la déclaration de constante.

##### **EXEMPLE No. 2: Structure de la déclaration de constantes**

```
CONST  
larg_table = 500  
long_table = 200  
haut_table = 100
```

#### 5.2.1.3. Déclaration de type

Les types servent à regrouper plusieurs variables ensemble afin de mieux les structurer. La structure de la déclaration de type est montrée dans l'exemple ci-dessous.

##### **EXEMPLE No. 3: Structure de la déclaration de type**

```
TYPE  
new_type_name = STRUCTURE  
field_name_1 : type_name_1  
field_name_2: type_name_2  
...  
ENDSTRUCTURE
```

Pour récupérer les données dans la structure, il suffit d'appeler la structure suivie de la variable désirée séparée par un point «.», comme montré dans l'exemple NO. 4.

##### **EXEMPLE No. 4: Structure de la récupération des données d'une variable située à l'intérieur d'une structure**

```
var_name = new_type_name.field_name_1
```

#### 5.2.1.4. Déclaration de variables

Pour utiliser des variables à l'intérieur d'un programme KAREL il faut préalablement les définir. L'exemple suivant montre la structure de la déclaration d'une variable.

**EXEMPLE No. 5: Structure de la déclaration de variables**

VAR

l, length : INTEGER

p1, p2, p3, : POSITION

c : CONFIG

**5.2.2. Type de variable prédéfinie**

Dans un programme il y a plusieurs types de variable. Elles ont chacune leur utilité et leur étendue, comme le montre le tableau ci-dessous.

**TABLEAU No. 6: Type de variables possible pour le langage KAREL**

Type de variable	Description	Étendue
BOOLEAN		True, false
INTEGER	Nombre entier	[-2147483648 ; +2147483646]
REAL	Nombre réel	$[-3,4028236E+38; -1,175494E-38] \cup [0,0] \cup [1,175494E-38; 3,4028236E+38] \in \mathbb{R}$
STRING	Chaine de caractère	0 à 254 caractères
VECTOR	Vecteur Composé de trois valeurs réelles  VECTOR = STRUCTURE X : REAL Y : REAL Z : REAL ENDSTRUCTURE	$X : Y : Z : [-3,4028236E+38; -1,175494E-38] \cup [0,0] \cup [1,175494E-38; 3,4028236E+38] \in \mathbb{R}$
ARRAY	Le type de variable que le tableau ARRAY contient est à spécifier Ex : JPOS : ARRAY[6] OF REAL	Min : ARRAY [1] Max : ARRAY[32767]
CONFIG	Configuration du robot  CFG_FLIP : BOOLEAN CFG_LEFT : BOOLEAN CFG_UP : BOOLEAN CFG_TURN_NO1 : INTEGER CFG_TURN_NO2 : INTEGER CFG_TURN_NO3 : INTEGER	CFG_FLIP : «R» ou «L» CFG_LEFT : «U» ou «D» CFG_UP : «N» ou «F» CFG_TURN_NO1 : [-8 ; 6] $\in \mathbb{E}$ CFG_TURN_NO2 : [-8 ; 6] $\in \mathbb{E}$ CFG_TURN_NO3 : [-8 ; 6] $\in \mathbb{E}$
XYZWPR	Spécifie la position cartésienne du UTOOL.	

	XYZWPR = STRUCTURE X: REAL Y: REAL Z: REAL W: REAL P: REAL R: REAL CONFIG_DATA: CONFIG ENDSTRUCTURE	
POSITION	Matrice contenant le vecteur normal, d'orientation, d'approche et de localisation, ainsi que la configuration du robot.  POSITION = STRUCTURE NORMAL : VECTOR ORIENT : VECTOR APPROACH : VECTOR LOCATION : VECTOR CONFIG_DATA : CONFIG ENDSTRUCTURE	
JOINTPOS<n>	Représentation de la position en une matrice contenant les angles des axes du robot. Les angles sont exprimés en degré.	JOINTPOS1 : 1 degré de liberté JOINTPOS2 : 2 degrés de liberté JOINTPOS3 : 3 degrés de liberté JOINTPOS4 : 4 degrés de liberté JOINTPOS5 : 5 degrés de liberté JOINTPOS6 : 6 degrés de liberté JOINTPOS7 : 7 degrés de liberté JOINTPOS8 : 8 degrés de liberté JOINTPOS9 : 9 degrés de liberté
PATH	Trajet constitué de plusieurs nœuds (nodes)	

### 5.2.3. Structure de la commande de condition IF

La commande IF est une des commandes les plus utiles en programmation. Elle permet de faire des choix selon la condition énoncée. La condition peut-être rencontrée ou pas. Si elle est rencontrée, les énoncés rattachés à la commande THEN

seront exécutés, sinon, les énoncés rattachés à la commande ELSE seront exécutés. L'exemple qui suit montre la structure de la commande IF.

**EXEMPLE No. 6: Structure de la commande IF**

```
IF variable > 10
  THEN
    MOVE TO p1
  ELSE
    MOVE TO p2
ENDIF
```

Il n'est pas obligatoire d'utiliser la commande ELSE à l'intérieur de la commande IF. Si elle est absente de la commande If, les énoncés qui ne seront pas rencontrés par la condition seront tout simplement ignorés.

#### 5.2.4. Structure des boucles

##### 5.2.4.1. *Boucle FOR*

La boucle FOR est une répétition définie, c'est-à-dire que le nombre de répétitions est connu et fixe.

**EXEMPLE No. 7: Première syntaxe d'une boucle FOR**

```
FOR variable = a TO b DO
  Instructions
ENDFOR
```

**EXEMPLE No. 8: Deuxième syntaxe d'une boucle FOR**

```
FOR variable = b DOWNT0 a DO
  Instructions
ENDFOR
```

La variable indiquée dans la boucle FOR doit faire partie du type INTEGER.

L'exemple No. 7 montre une boucle FOR qui incrémentera la variable de 1 à chaque répétition. Par contre, l'exemple No. 8 présente plutôt une boucle FOR qui décrémente la variable de 1 à chaque répétition.

##### 5.2.4.2. *Boucle WHILE*

La boucle WHILE est une boucle indéfinie, car le nombre de répétitions dépend de la condition spécifiée par le programmeur.

**EXEMPLE No. 9: Syntaxe d'une boucle WHILE**

```

WHILE condition DO
  Instructions
ENDWHILE

```

**5.3.Langage de programmation****5.3.1. Commande de programmation****TABLEAU No. 7: Liste de quelques commandes du langage KAREL**

Nom de la commande	Description	Exemple
APPEND_NODE	Insert un nœud à la suite du dernier nœud d'une variable de type PATH	VAR length, I : INTEGER path1 : PATH BEGIN length = PATH_LEN(path1) FOR I = 1 TO longueur DO MOVE AXIS 4 BY 1 APPEND_NODE(path1, 6) path1[i].node_pos= CURPOS(1,5,1) ENDFOR END
CLOSE FILE	Permet de dissocier le fichier ouvert précédemment de la variable du type FILE	CLOSE FILE fichier1
CLOSE HAND	Permet de fermer les pinces du robot.  Le chiffre après la commande correspond au numéro du «Robot Output» auquel les pinces sont reliées.	CLOSE HAND 1

CURPOS	<p>Retourne la position actuelle du TCP</p> <p>Syntaxe :  CURPOS (axis_limit_mask, ovr_trv_mask, group_no)  Axis_limit_mask : [1 ; 6]*  Ovr_trv_mask : [1 ; 6]*</p> <p>*Dans cette version, ces commandes sont ignorées</p>	CURPOSE (1,5,1)
MOVE ABOUT	Permet une rotation du TCP autour d'un vecteur spécifier en restant aux mêmes coordonnées	VAR vector_1 : VECTOR pos_1 : POSITION BEGIN MOVE TO pos_1 vector_1 = ORIENT(pos_1) MOVE ABOUT pos_1 BY 30 END
MOVE AWAY	Permet d'éloigner le TCP d'un objet en reculant le TCP selon son axe des Z négatif	BEGIN MOVE AWAY 50 END
MOVE AXIS	Permet de changer l'orientation d'un des axes du robot	BEGIN MOVE AXIS 3 BY 45 END
MOVE NEAR	Permet de déplacer le TCP près de la position spécifiée, soit par une distance spécifiée selon l'axe des Z négatif du TCP	VAR pos_1 : POSITION BEGIN MOVE NEAR pos_1 BY 60 END
MOVE TO	Déplace le TCP à la coordonnée spécifiée. La variable utilisée par cette fonction peut être de type : POSITON, XYZWPR, XYZWPREXT, JOINTPOS	VAR cord : XYZWPR JPOS : JOINTPOS BEGIN MOVE TO cord MOVE TO JPOS END

OPEN FILE	<p>Permet de lier un fichier extérieur à une variable du type FILE</p> <p>Syntaxe : OPEN FILE variable ('attribut', 'nom_fichier')</p> <p>Attribut :</p> <ul style="list-style-type: none"> <li>• 'RO' : Lecture seulement</li> <li>• 'RW' : Lecture et écriture</li> <li>• 'AP' : Ajouter des données à un fichier déjà existant</li> <li>• 'UD' : Mise à jour de données</li> </ul>	OPEN FILE fichier1 ('RW','test.txt')
OPEN HAND	<p>Permet d'ouvrir les pinces du robot</p> <p>Le chiffre après la commande correspond au numéro du «Robot Output» auquel les pinces sont reliées.</p>	OPEN HAND 1
PATH_LEN	Détermine le nombre de nœuds dans une variable PATH	<pre>VAR p_len : INTEGER BEGIN p_len = PATHLEN(parcour) END</pre>
PATHPOS	Retourne la position d'un nœud	<pre>VAR p_len, l : INTEGER BEGIN p_len = PATHLEN(parcour) FOR l = 1 TO p_len DO      ENDFOR END</pre>



POS	Retourne les données cartésiennes et la configuration du bras du robot à une variable du type POSITION	VAR p1 : POSITION c : CONFIG BEGIN CNV_STR_CONF('NUT, 0,0,0',c, 0) p1 = POS(450, 0, 450, 0, 0, 0, c) MOVE TO p1 END
READ	Permet de lire des données	
WRITE	Permet d'écrire des données	

### 5.3.2. Variables du système

**TABLEAU No. 8: Liste des variables de quelque variable du système**

Nom de la variable	Description	Exemple
\$GROUP[n].\$termttype	Permet de changer le type de terminaison pour un groupe [n] Les valeurs possibles sont 1 = fine 2 = coarse 3 = nosset 4 = nodecel 5 = vardecel	\$GROUP[1].\$termttype = 2 \$GROUP[3].\$termttype = 1
\$GROUP[n].\$motype	Permet de changer le type de mouvement pour un groupe [n] Les valeurs possibles sont 6 = joint 7 = linear 8 = circular	\$GROUP[1].\$motype = 7
\$MOTYPE	Permet de changer le type de mouvement pour tous les groupes Les valeurs possibles sont 6 = joint 7 = linear 8 = circular	\$motype = 6

\$SPEED	Spécifie une vitesse cartésienne en mm/sec au robot Valeur varie entre 0 à 2000 mm/sec	\$speed = 1000
\$TERMTYPE	Permet de changer le type de terminaison pour tous les groupes Les valeurs possibles sont 1 = fine 2 = coarse 3 = nosset 4 = nodecel 5 = vardecel	\$TERMTYPE = 2
\$UFRAME	Transformation du référentiel de la cellule (UFRAME) par rapport à celui du robot.	
\$UTOOL	Transformation du référentiel de l'effecteur (TCP) par rapport à celui du poignet du robot.	

## 5.4.Exemple de programmes

### EXEMPLE No. 10: Déplacer le robot vers une position prédéfinie

Déplacer le robot vers de coordonnées prédéfinies	
PROGRAM dec_const	
CONST	--déclaration des constantes
larg_table = 500	
long_table = 200	
haut_table = 1	
VAR	--déclaration des variables
p1 : POSITION	
c : CONFIG	
BEGIN	
CNV_STR_CONF('N U T, 0, 0, 0', c, 0)	-- converti un variable
	--STRING en
	-- variable CONFIG
p1 = POS(larg_table, long_table, haut_table, 180, 0, 0, c)	-- assigne les
	--coordonnées du
	--point p1
MOVE TO p1	-- bouge le robot à la position

```
p1
END dec_const
```

**EXEMPLE No. 11:     Ajout de nœuds dans une variable de type PATH**

```
PROGRAM test_pht_2
VAR
  length, I : INTEGER
  p1 : XYZWPR
  c : CONFIG
  path1 : PATH
  longueur : INTEGER
  fichier1 : FILE
BEGIN
  CNV_STR_CONF('NUT, 0, 0,0', c, 0)           --initialisation de la variable 'c'
  p1 = POS(475,-190,375,-180,-90,0,c)         --initialisation de la position p1
  MOVE TO p1                                  --déplacement vers la position 'p1'
  length = PATH_LEN(path1)                   --détermine le nombre de noeuds
  longueur = 30                               --initialisation de la variable longueur
  FOR I= length DOWNTO 1 DO
    DELETE_NODE(path1, I, 6)                  --supprime tous les anciens nœuds
  ENDFOR
  FOR I = 1 TO longueur DO
    MOVE AXIS 4 BY 1                           --bouge l'axe 4 de 1 degré
    APPEND_NODE(path1, 6)                      --ajoute un nœud au PATH
    path1[i].node_pos= CURPOS(4,1,1)          --enregistre la position actuelle au
                                              --noeud
  ENDFOR
END test_pht_2
```

## CHAPITRE 6 : ANNEXE

### 6.1.Tutoriel pour ROBOGUIDE

#### 6.1.1. «Pick and Place»

Ce tutoriel vous guidera pas à pas pour créer un environnement virtuel qui simulera la trajectoire d'un robot lors d'une manipulation «Pick and place»

- Ouverture du fichier
  1. Ouvrez le logiciel ROBOGUIDE
  2. Allez dans «fichier» du menu supérieur du programme et cliquez sur «New Cell», une fenêtre apparaîtra
  3. Sélectionnez le logiciel HandlingPRO et cliquez sur Next
  4. Donner un nom au fichier, par exemple : tutoriel\_PAP. Ensuite, cliquez sur Next
  5. Cliquez Next encore et encore, jusqu'à ce que vous voyez la liste des différents robots
  6. Choisissez le robot LR Mate 200iC et cliquez Next
  7. Cliquez Next jusqu'à la fin et ensuite cliquez Finish
  8. Vous devriez alors avoir une cellule de travail avec le robot.
- Modifier les paramètres du robot
  1. Allez dans «View» du menu supérieur du programme et cliquez sur Navigator s'il n'est pas déjà sélectionné. Une petite fenêtre appelée «Process Navigator» s'ouvrira.
  2. À l'intérieur de la section «Define the Cell» du «Process Navigator» cliqué sur «Édit Robot Properties», une fenêtre avec les propriétés du robot s'ouvrira
  3. À l'intérieur de cette fenêtre, il est possible de modifier la localisation du robot, mais dans le cadre de ce tutoriel nous laisserons le robot à l'origine.
  4. Cocher «Lock All Location Values» pour fixer la position du robot et cliquez sur OK
- Ajouter une pièce (Part)
  1. À l'intérieur de la section «Define the Cell» du «Process Navigator» cliqué sur «Add a Part to the Cell», une fenêtre avec différente option concernant la nature de la pièce s'ouvrira.
  2. Dans le cadre de ce tutoriel, sélectionnez «Box Primitive Model» et cliquez sur OK. Une fenêtre avec les propriétés de la pièce apparaîtra.
  3. Donnez un nom à la pièce, par exemple : cube

4. Vous pouvez modifier la couleur du cube en cliquant sur la palette de couleur
  5. Dans le champ «Mass» changer la masse pour 3 kg, car le robot peut supporter un maximum de 5 kg en incluant la masse de la pince.
  6. Dans les champs «Size» changez les dimensions pour 29 mm
  7. Cliquez sur OK, vous devriez avoir une plateforme (PartRack) surmontée de la pièce située à côté du robot.
- Ajouter un outil au bout du bras du robot (EoAT)
    1. À l'intérieur de la section «Define the Cell» du «Process Navigator» cliquez sur «Edit End of Arm Tooling», une fenêtre des propriétés de l'outil s'ouvrira.
    2. Donnez un nom à l'outil en changeant le champ «Name», par exemple : pince.
    3. Importer un outil déjà existant en cliquant sur le dossier à côté du champ «CAD File», une fenêtre pour parcourir le fichier s'ouvrira
    4. Aller dans EOATs/grippers et sélectionnez le fichier «36005f-200.IGS» et cliquez ouvrir
    5. Changez la masse de la pince pour 2 kg en modifiant le champ «Mass»
    6. Modifiez l'échelle de la pièce pour 0,2, en modifiant les champs «Scale», afin de donner une taille de pince proportionnelle au robot. Ensuite, cliquez sur OK
    7. Vous vous apercevrez que la pince n'est pas installée correctement. Pour corriger la situation, double-cliquez sur la pince pour faire apparaître les propriétés de l'outil et changer la valeur du champ «W» pour 270. Ceci effectuera une rotation autour de l'axe des «x» de 270°. Ensuite, cliquez sur Apply.
    8. Il faut maintenant modifier le point de contact du robot avec la pièce. Pour ce faire, cliquez sur l'onglet «UTOOL» des propriétés de la pince et cochez «Edit UTOOL».
    9. Vous pouvez maintenant entrer les valeurs de la position du point de contact par rapport au référentiel situé à l'extrémité du bras du robot. Dans le cadre de cet exemple, entrez la valeur 170 dans le champ «Z» et cliquez sur Apply. La sphère verte devrait maintenant se situer au point de contact des pinces
    10. Il faut maintenant déterminer l'emplacement de la pièce lorsque le robot prendra la pièce. Pour ce faire, aller dans l'onglet «Parts» des propriétés de la pince et sélectionné sur la pièce voulue (cube pour ce tutoriel) et cliquez Apply.
    11. Dans le même onglet, cocher «Edit Part Offset». Vous pouvez maintenant modifier la position de la pièce par rapport au référentiel situé à l'extrémité du bras du robot. Dans le cadre de

cet exemple, entrez les valeurs  $X=0$ ,  $Y=-187.06$ ,  $Z=0$ ,  $W=90$ ,  $P=R=0$ . Ensuite, cliquez sur Apply. Vous vous apercevrez que la pièce disparaîtra de la pince du robot, c'est normal.

12. Maintenant il faut déterminer l'interaction de la pince lors de la simulation. Pour ce faire, allez dans l'onglet «Simulation» des propriétés de la pince et sélectionnez la pièce désirée. Ensuite sélectionnez «Material Handling – Clamp» pour le champ «function» et choisissez le fichier 36005f-200-4.IGS dans trajectoire EOATs/grippers en cliquant sur le fichier à côté du champ «Actuated CAD». Cliquez OK.
- Ajouté une fixation (fixture) pour la position «pick»
    1. Pour rajouter une fixation, cliquez droit sur fixation dans le «Cell Browser», allez dans «AddFixture» et choisissez une option. Dans le cadre de cet exemple, choisissez «CAD Library». Une fenêtre avec la librairie des fixations s'ouvrira.
    2. Cliquez le petit «+» à gauche de «Fixture» pour avoir la liste de tous les types de fixation. Aux fins de ce tutoriel, choisissez le fichier «shelf04» et cliquez OK. Une fenêtre s'ouvrira avec les propriétés de la fixation.
    3. Donnez un nom à la fixation en modifiant le champ «Name», par exemple : SHELF.
    4. Changez l'échelle de la fixation en changeant les «Scales» pour 0.8 et changez la localisation de la pièce pour  $X=-600$ ,  $Y=200$ ,  $Z=W=P=R=0$ . Ensuite, cliquez sur Apply.
    5. Configurez la pièce sur la fixation en allant dans l'onglet «Parts» des propriétés de la fixation, sélectionnez la pièce désirée, soit «cube» et cliquez sur Apply.
    6. Dans le même onglet, cocher «Edit Part Offset» pour modifier la localisation de la pièce sur la fixation. Dans le cadre de ce tutoriel, entrez les valeurs  $X=160$ ,  $Y=160$ ,  $Z=510$ ,  $W=-90$ ,  $P=0$ ,  $R=90$ . Ensuite, cliquez sur Apply
    7. Modifier les paramètres de simulation en allant dans l'onglet «Simulation» des propriétés de la fixation et cliquez sur la pièce désirée, soit «cube» pour ce tutoriel. Ensuite, cochez le champ «Allow part to be picked» pour autoriser le programme à prendre une pièce sur la fixation. Ensuite, cliquez sur Apply et ensuite OK.
  - Ajouté une fixation (fixture) pour la position «place»
    1. Les étapes sont sensiblement les mêmes que la création de la fixation précédente. Par contre, cette fixation sera spécialement configurée pour permettre le dépôt d'une pièce.
    2. Créez une nouvelle fixation, mais au lieu choisir l'option «CAD Library» cliquez plutôt sur «Box»

3. Renommez la fixation pour «BOX», changez les dimensions pour  $X=Y=Z=200$  et changez la localisation pour  $X=400, Y=-300, Z=200, W=P=R=0$ . Ensuite, cliquez sur Apply.
  4. Configurez les paramètres de la pièce en allant dans l'onglet «Parts» et en sélectionnant la pièce «Cube». Ensuite, cliquez Apply. Cocher les champs «Edit Part Offset» et entrer  $X=Y=0, Z=14.58, W=-90, P=0, R=-90$ . Ensuite, cliquez sur Apply.
  5. Configurez les paramètres de simulation en allant dans l'onglet «Simulation». Sélectionnée la pièce «cube» et cochez le champ «Allow part to be placed». Ensuite, cliquez sur Apply et OK.
- Programme de simulation« Pick»
    1. Pour créer un nouveau programme de simulation, cliquez droit sur «programs» dans le «Cell Browser» et cliquez sur «Add simulation program». Une fenêtre de propriété pour le programme apparaîtra.
    2. Modifiez le nom pour «PICK» et cliquez Apply ensuite OK. Vous tomberez alors sur la fenêtre d'édition pour le programme de simulation.
    3. Bouger le point fixe du robot (la balle verte) près de la pièce sur la fixation «Shelf». Pour bouger le point fixe en glissant la souris sur ses axes après avoir cliqué le bouton des options principales du programme ROBOGUIDE.



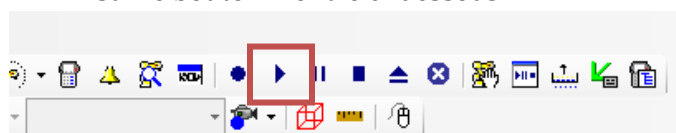
**FIGURE No. 23: Outil de manipulation du point fixe du robot**

4. Dans la fenêtre du programme de simulation, cliquez sur «Record» pour enregistrer les coordonnées du robot. Une ligne de programme s'ajoutera.
5. Dans la ligne de programme, changer l'option «L» (linear) pour «J» (joint), le robot suivra alors un trajet circulaire pour atteindre ce point.
6. Changer l'option «Fine» pour «CNT100» si ce n'est déjà pas fait. Ceci indiquera au robot passé par ce point sans y arrêter.
7. Bouger ensuite le robot sur la pièce. Pour que les pinces soit exactement à l'endroit de la pièce, double cliquez sur la fixation «SHELF» allez dans l'onglet «Parts» et cliquez sur «Move to». La pince du robot se déplacera automatiquement sur la pièce.
8. Cliquez sur «Record» dans menu du programme de simulation. Ensuite, mettez un mouvement linéaire «L» et mettez un mouvement «Fine» à la nouvelle ligne de programme créé.

9. Déplacez les pinces hors du champ de la pièce et cliqué encore sur record.
  10. Ensuite, changer les paramètres de la ligne de programme pour «J» et «CNT100»
  11. Sélectionné la ligne juste après l'action où la pièce est prise, soit la ligne 3 dans ce cas. Ensuite, cliqué la flèche du bouton «Inst» du menu du programme de simulation et cliqué sur «Pickup». Une nouvelle ligne de programme se créera en dessous de la ligne 2.
  12. Pour cette ligne de programme, choisissez la pièce prise par les pinces, soit «cube» et sélectionnez la fixation et l'outil utilisé.
  13. Sélectionnée la ligne de programme après la commande «Pickup» et cliquez sur la flèche du bouton «Inst» puis sélectionner sur «PAYLOAD[1]». Une ligne de programme apparaîtra après la commande «Pickup». La commande «PAYLOAD» servira à indiquer le poids supplémentaire que le robot doit soulever lorsqu'il ramasse l'objet.
  14. Cliquez sur X pour fermer la fenêtre du programme de simulation.
- Programme de simulation «Place»
    1. Le programme de simulation «Place» se construira sensiblement de la même manière que le programme «Pick».
    2. Cliquez droit sur «programs» dans le «Cell Browser» et cliquez sur «Add simulation program». Une fenêtre de propriété pour le programme apparaîtra.
    3. Modifiez le nom pour «PLACE» et cliquez Apply ensuite OK.
    4. Bouger le point fixe du robot près de la pièce sur la fixation «BOX».
    5. Cliquez sur «Record» pour enregistrer les coordonnées du robot.
    6. Changer les paramètres de la ligne de programme pour «J» et «CNT100»
    7. Déplacez les pinces du robot sur la pièce de la fixation «BOX» en utilisant la commande «Move to»
    8. Cliquez su «Record» et insérez les paramètres «L» et «Fine».
    9. Bouger les pinces hors de la portée de la pièce.
    10. Cliquez sur Record et insérez les paramètres «J» et «CNT100».
    11. Sélectionnez la ligne de programme après avoir déposé la pièce.
    12. Insérez la commande «Drop» en cliquant sur la flèche du bouton «Inst»
    13. Dans la ligne de programme de la commande «Drop», sélectionnez la pièce, l'outil et la fixation appropriés
    14. Sélectionnez la ligne de programme après la commande «DROP»
    15. Insérez la commande «PAYLOAD [1]» en cliquant sur la flèche du bouton «Inst»



16. Sur la ligne de commande «PAYLAOD» changer le «1» pour un «2».
  17. Cliquez sur X pour fermer la fenêtre du programme de simulation.
- Programme TP «Main»
    1. Pour créer un nouveau programme TP, cliquez droit sur «programs» dans le «Cell Browser» et cliquez sur «Add TP program». La télécommande «Teach Panel» s'ouvrira.
    2. Cliquez droit sur le programme TP dans le «Cell Browser» et renommez le nom du programme pour «MAIN».
    3. Cliquez sur «ON» pour autoriser l'écriture sur la manette.
    4. Cliquez sur «EDIT» pour accéder à la fenêtre de commande.
    5. Cliquez sur «NEXT» pour accéder à plus d'options et cliquez sur F1 [INST]. Une liste de commande apparaîtra. Sélectionné «Next Page» [8] et sélectionnez «Payload» [2]. Cliquez sur ENTER et entrez le nombre du «Payload», soit 2, car c'est le nombre du «Payload» lorsque le robot est à vide.
    6. Cliquez sur F1 [INST] et sélectionnez la commande «CALL» [6].
    7. Sélectionnez «Call program» et choisissez le programme «PICK». Cliquez ENTER deux fois de sorte que la balise «[END]» soit sélectionnée.
    8. Cliquez sur F1 [INST] et sélectionnez la commande «CALL» [6].
    9. Sélectionnez «Call program» et choisissez le programme «DROP». Cliquez ENTER deux fois de sorte que la balise «[END]» soit sélectionnée.
    10. Cliquez sur F1 [INST] et sélectionnez la commande «JMP/LBL» [6]. Sélectionnez «JMP LBL[ ]»
    11. Entrez le nombre du «JMP LBL», soit «1».
    12. Sélectionnez la ligne après le «PAY LOAD» et cliquez sur F5 [EDCMD]. Sélectionnez «Insert» et cliquez ENTER. Une ligne vide apparaîtra à la suite de la commande «PAYLOAD».
    13. Cliquez sur F1 [INST] et sélectionnez la commande «JMP/LBL» [6]. Sélectionnez «LBL[ ]»
    14. Entrez le nombre du «LBL», soit «1». Vous avez maintenant une boucle dans votre programme borné par les commandes «LBL» (Label) et «JMP LBL» (Jump Label)
    15. Cliquez sur X pour fermer la fenêtre du programme TP.
  - Roulez le programme
    1. Assurez-vous que le programme «MAIN» est sélectionné et cliquez sur le bouton montré ci-dessous.



**FIGURE No. 24: Bouton pour rouler un programme**

2. Votre robot devrait alors prendre la pièce de la fixation «SHELF» et la déposer sur la fixation «BOX»

## CHAPITRE 7 : ERREURS FRÉQUENTES

### 7.1.Problèmes de configuration de l'ordinateur

Pour pouvoir utiliser le logiciel ROBOGUIDE sur l'ordinateur, celui-ci doit avoir certaines configurations précises.

1. Le diviseur de décimal doit être le «.» au lieu de la «,»
2. La sauvegarde des fichiers doit se faire sur un disque physique et non un disque réseau

#### 7.1.1. Changer le type de diviseur de décimale

Pour changer le type de diviseur de décimal vous devez :

1. allez dans panneau de configuration
2. cliquez sur la section «Horloge, langue et région»
3. cliquez sur «Modifier le format de la date, de l'heure ou des nombres»
4. une fenêtre apparaîtra comme celle-ci

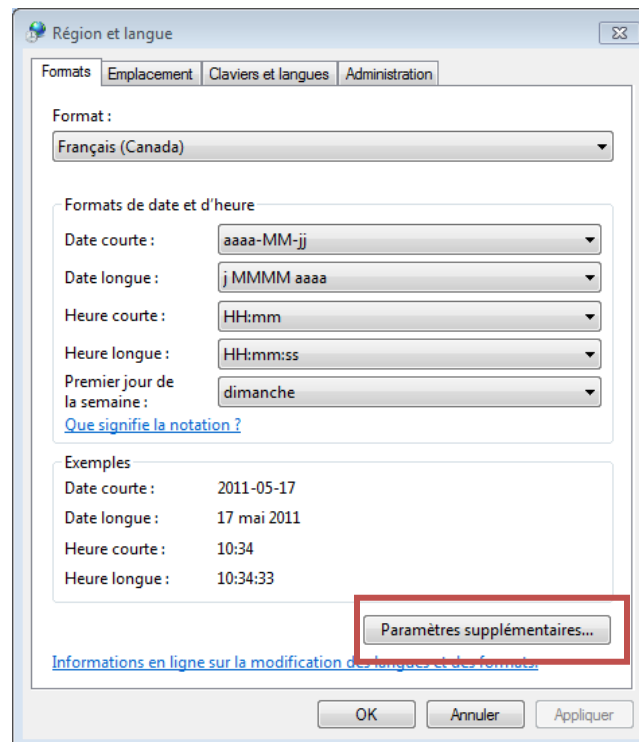
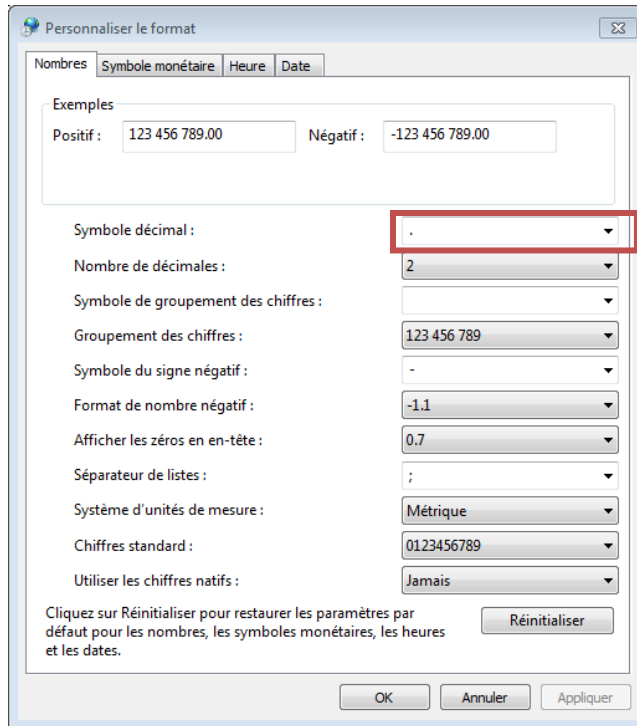


FIGURE No. 25: Fenêtre de «région et langue» du panneau de configuration

5. cliquez sur paramètre supplémentaire, une fenêtre apparaîtra comme ci-dessous



**FIGURE No. 26: Fenêtre des options avancées du format des nombres**

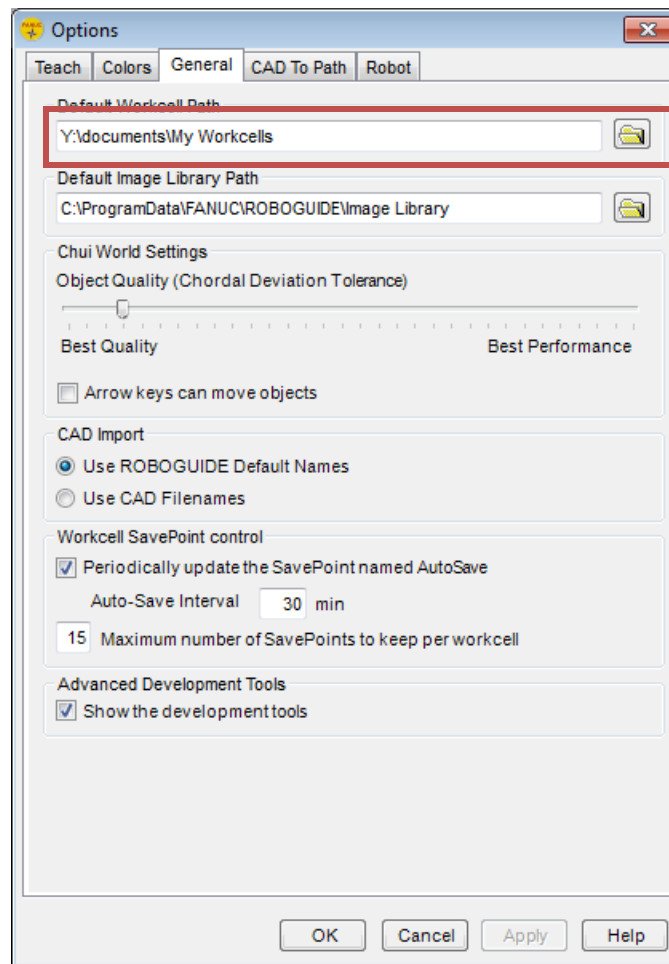
6. Changez dans la section «Symbole décimal» la virgule par un point si besoin
7. Cliquer OK

### 7.1.2. Changer l'emplacement du fichier de sauvegarde

Si votre programme signale une erreur lorsque vous voulez créer un fichier, il est fort probable que l'emplacement du fichier de sauvegarde soit sur un disque réseau au lieu d'un disque local.

Pour changer l'emplacement du fichier de sauvegarde, vous devez :

1. Ouvrir ROBOGUIDE
2. Cliquez sur l'onglet «Tools» de la barre d'outils du programme
3. Allez dans la section «General» telle que montrée ci-dessous



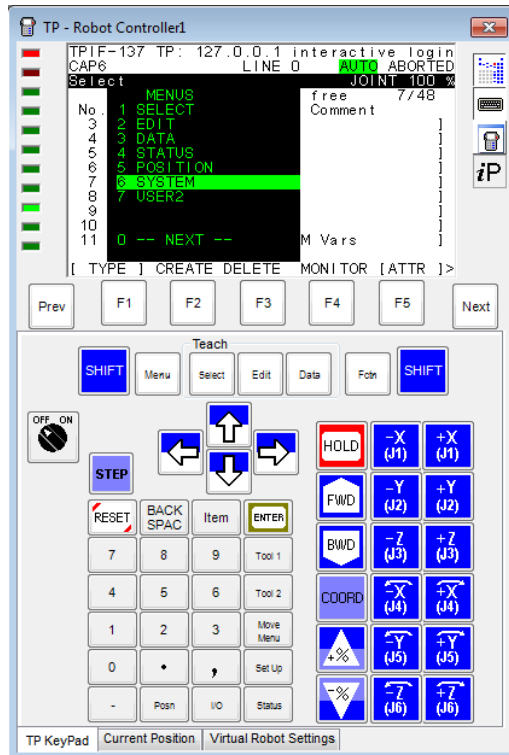
**FIGURE No. 27: Fenêtre pour changer la trajectoire de sauvegarde des documents sur ROBOGUIDE**

4. Changer le champ «Default Workcell Path» pour un emplacement sur un disque local
5. Cliquez sur OK

### 7.1.3. Appliquer les valeurs DCS

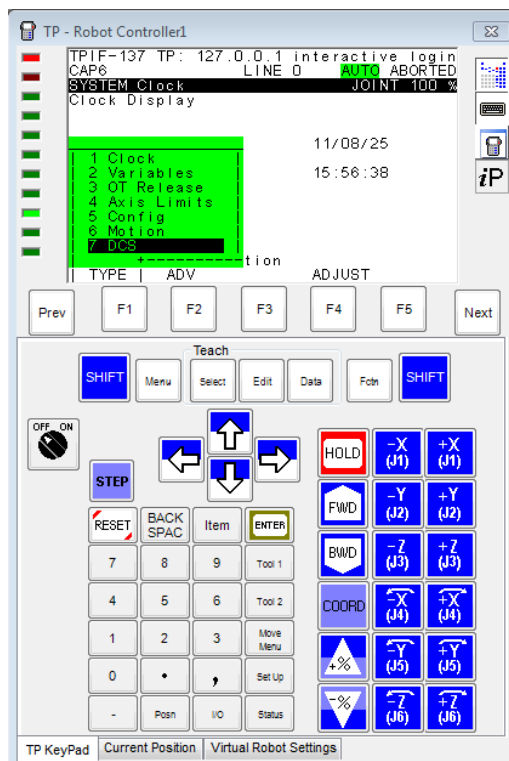
Si l'erreur «Need to apply DCS parameter» apparaît sur votre boîtier de commande virtuel, vous devez suivre les instructions suivantes pour vous défaire de l'impasse.

1. Ouvrez le boîtier de commande virtuelle dans ROBOGUIDE
2. Cliquez sur menu
3. Dans le menu, sélectionnez «NEXT» pour voir d'autre onglet
4. Sélectionner «System», la fenêtre des options du système apparaîtra



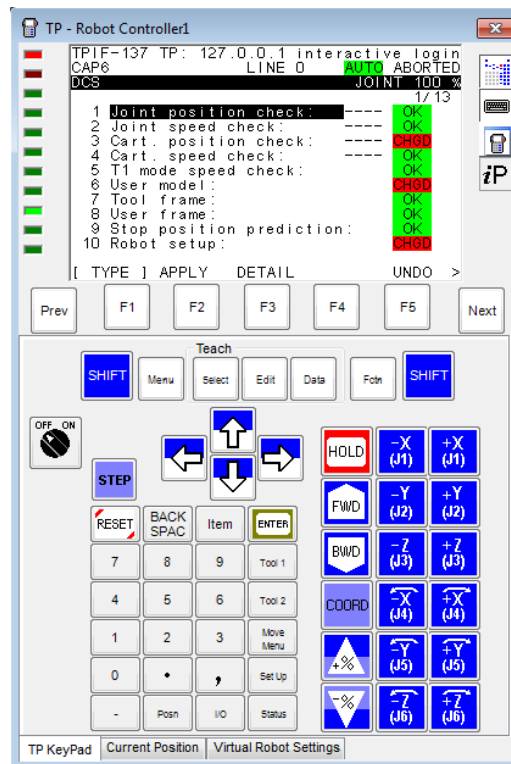
**FIGURE No. 28: Trajet pour appliquer un DCS – étape 1**

5. Cliquez F1 pour «TYPE» et choisissez «DCS»



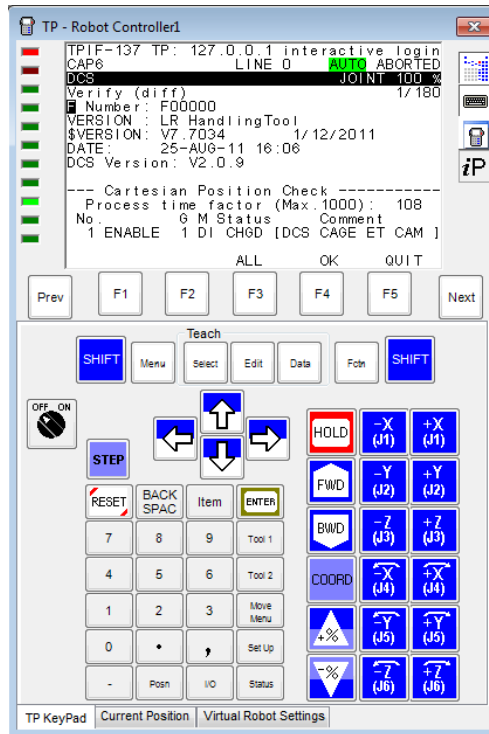
**FIGURE No. 29: Trajet pour appliquer un DCS - étape 2**

6. Cliquez sur F2 pour «APPLY»



**FIGURE No. 30: Appliquer un DCS**

7. Un mot de passe vous sera demandé, alors demandez au responsable de cours ou de laboratoire pour qu'il inscrive le mot de passe.
8. Vous devez ensuite confirmer les modifications en pesant sur F4 pour «OK»

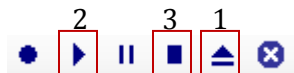


**FIGURE No. 31: Confirmation de l'application du DCS**

9. Redémarrez le contrôleur en allant dans «Robot/restart controller/cold start» du menu principal de ROBOGUIDE.

## 7.2. Déboguer le boîtier de commande virtuel

Parfois, lors de l'exécution de certains programmes, le boîtier de commande virtuel signal des erreurs. La plupart du temps pour enlever le message d'erreur il faut seulement peser sur «shift» et «reset». Par contre, parfois ceci ne suffit pas. Dans ce cas il faut appuyer sur l'icône «reset»<sup>1</sup> du menu de bouton et ensuite peser sur «play»<sup>2</sup> et finalement peser du «stop»<sup>3</sup> comme le montre la figure ci-dessous.



**FIGURE No. 32: Procédure pour résoudre les messages d'erreurs du boîtier de commande virtuel**

Si le message d'erreur ne disparaît toujours pas alors il faut redémarrer le contrôleur virtuel en cliquant sur «Robot/Restart Controller/Cold Start».