CHAPITRE 2 : ETUDE BIBLIOGRAPHIQUE

1. INTRODUCTION:

Ce chapitre est consacré à l'étude bibliographique du robot.

Au début, on présente les différents composants du robot. Ensuite, on étudie le système de coordonnées, les méthodes de calibrage et les étapes de mesures pour avoir un robot capable de faire un déplacement cartésien et d'accoster des positions programmées.

2. ETUDE BIBLIOGRAPHIQUE:

2.1. <u>Description générale du robot KUKA KRC2</u>:

Le robot industriel KUKA est composé d'un bras manipulateur, d'une unité de commande de type KR C2 de robot, d'un boîtier de programmation portatif (KCP), de câbles de liaison, de logiciel de programmation, d'options et accessoires.

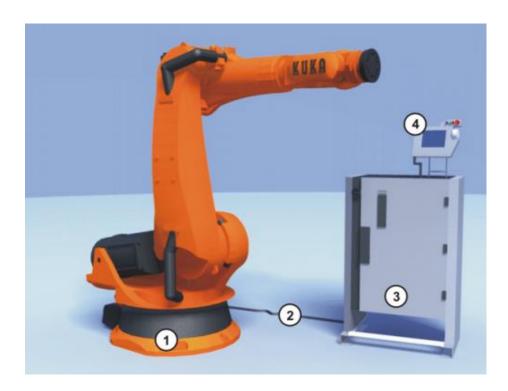


Figure 4: Architecture du robot KUKA KRC2

- 1. Manipulateur
- 2. Câbles de liaisons

- 3. Commande de robot (PC industriel)
- 4. Boitier de programmation portatif

2.1.1. <u>Architecture géométrique :</u>

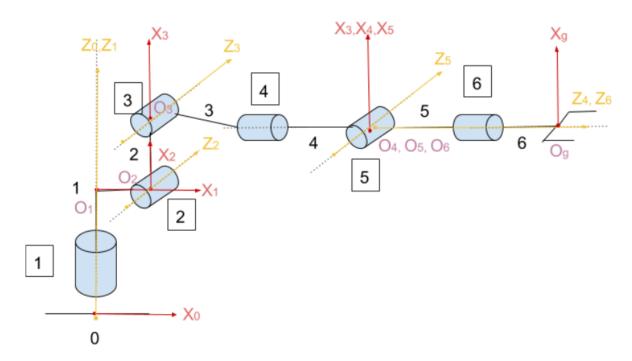


Figure 5: Architecture géométrique du robot KUKA KRC2

• Nombre d'axe : 6 axes

• Architecture : série

• Chaînage des articulations : RRRRRR

• Nombre de degrés de liberté : 6DDL

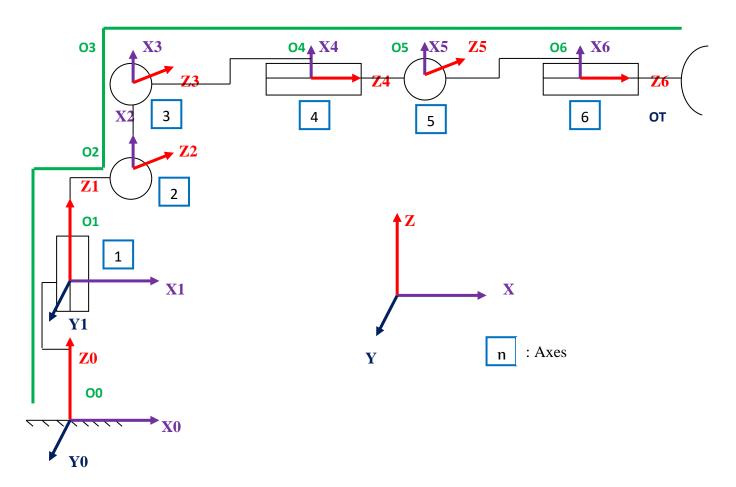
Le tableau 1 illustre la limite de rotation de chaque axe.

Axe 1	± 185°	Axe 4	±350°
Axe 2	0°/ -146°	Axe 5	±125°
Axe 3	+155°/-119°	Axe 6	±350°

Tableau 1: Limite de rotation de chaque axe

2.1.1. Modèle géométrique :

Voici un schéma qui met en évidence les 6 articulations (avec un référentiel pour la pince / effecteur terminal) et les liens associés (à partir de 0) entre les articulations.



2.1.2. Modèle cinématique direct :

La cinématique tente de décrire le mouvement des corps sans faire référence aux notions de forces ou d'énergie. Elle étudie les mouvements des corps à l'aide des notions d'espace et de temps en faisant abstraction de leurs causes. Elle utilise les concepts de position, déplacement, vitesse, accélération.

La cinématique directe permet de faire des mouvements d'arcs précis et bien définis, puisqu'on contrôle directement les rotations des articulations.

Afin d'être en mesure de positionner le robot pour chaque point de l'espace et quelle que soit l'orientation désirée, en plus du vecteur de translation, il faut spécifier la matrice de rotation.

$$R(z, \Theta) = \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R(Y, \Theta) = \begin{pmatrix} \cos(\Theta) & 0 & \sin(\Theta) \\ 0 & 1 & 0 \\ -\sin(\Theta) & 0 & \cos(\Theta) \end{pmatrix}$$

$$R(X, \Theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta) & -\sin(\Theta) \\ 0 & \sin(\Theta) & \cos(\Theta) \end{pmatrix}$$

On utilise la notation simplifiée :

 $S(\Theta)$ pour représenter $\sin(\Theta)$ et $C(\Theta)$ pour représenter $\cos(\Theta)$.

Les matrices de passages :

$$A_1^0 = \begin{pmatrix} \cos(\Theta 1) & -\sin(\Theta 1) & 0\\ \sin(\Theta 1) & \cos(\Theta 1) & 0\\ 0 & 0 & 1 \end{pmatrix}$$

$$A_2^1 = \begin{pmatrix} \cos(\Theta 2) & 0 & \sin(\Theta 2) \\ 0 & 1 & 0 \\ -\sin(\Theta 2) & 0 & \cos(\Theta 2) \end{pmatrix}$$

$$A_3^2 = \begin{pmatrix} \cos(\Theta 3) & 0 & \sin(\Theta 3) \\ 0 & 1 & 0 \\ -\sin(\Theta 3) & 0 & \cos(\Theta 3) \end{pmatrix}$$

$$A_4^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta 4) & -\sin(\Theta 4) \\ 0 & \sin(\Theta 4) & \cos(\Theta 4) \end{pmatrix}$$

$$A_5^4 = \begin{pmatrix} \cos(\Theta 5) & 0 & \sin(\Theta 5) \\ 0 & 1 & 0 \\ -\sin(\Theta 5) & 0 & \cos(\Theta 5) \end{pmatrix}$$

$$A_6^5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta 6) & -\sin(\Theta 6) \\ 0 & \sin(\Theta 6) & \cos(\Theta 6) \end{pmatrix}$$

* Relations trigonométriques :

$$Cos(\Theta1 + \Theta2) = Cos(\Theta1) \times Cos(\Theta2) - Sin(\Theta1) \times Sin(\Theta2)$$

$$Cos(\Theta 1 - \Theta 2) = Sos(\Theta 1) \times Sos(\Theta 2) + Sin(\Theta 1) \times Sin(\Theta 2)$$

$$Sin (\Theta1+\Theta2)=Sin (\Theta1) \times Cos (\Theta2) + Cos (\Theta1) \times Sin (\Theta2)$$

$$Sin(\Theta 1-\Theta 2)=Sin(\Theta 1) \times Cos(\Theta 2) - Cos(\Theta 1) \times Sin(\Theta 2)$$

On obtient finalement la matrice de rotation :

$$A_6^0 = A_1^0 . A_2^1 . A_3^2 . A_4^3 . A_5^4 . A_6^5 =$$

$$C(\Theta 2 - \Theta 3 - \Theta 5) + C\Theta 1 \times C(\Theta 1 + \Theta 4) \qquad S(\Theta 1 + \Theta 4 + \Theta 5) - C(\Theta 5 + \Theta 2 + \Theta 3) \qquad C(\Theta 1 - \Theta 6 - \Theta 4) + C(\Theta 5\Theta 2 + \Theta 3) \times C\Theta 1$$

$$C(\Theta 2 - \Theta 3 - \Theta 5) \times S(\Theta 1 + \Theta 2) \times S\Theta 1 \qquad C(\Theta 1 + \Theta 4 + \Theta 6) + C(\Theta 2 + \Theta 3 + \Theta 5) \qquad S(\Theta 1 + \Theta 6 + \Theta 4) - S(\Theta 2 + \Theta 3)$$

$$S(\Theta 5 - \Theta 3 - \Theta 2) - C\Theta 4 \qquad S(\Theta 1 - \Theta 4 - \Theta 6) - C(\Theta 2 + \Theta 3 + \Theta 5) \qquad -C(\Theta 4 + \Theta 6) + C(\Theta 2 + \Theta 3) - C(\Theta 5 + \Theta 2 + \Theta 3)$$

2.2. Composition du robot KUKA KRC2:

2.2.1. <u>Le bras manipulateur :</u>

Le bras manipulateur KUKA existant est un robot industriel série à six axes cinématiques articulés pouvant être mis en œuvre pour les taches avec positionnement en continu et point par point. La charge maximale que peut porter le robot sans dégrader la répétabilité et les performances dynamiques est 210 kg. Ce robot est très rapide, léger et polyvalent et qui présente les caractéristiques de sécurité et de flexibilité.



Figure 6: Emplacement et sens de rotation des axes robotiques

2.2.2. La commande du robot :



Figure 7:PC industriel KUKA KRC2

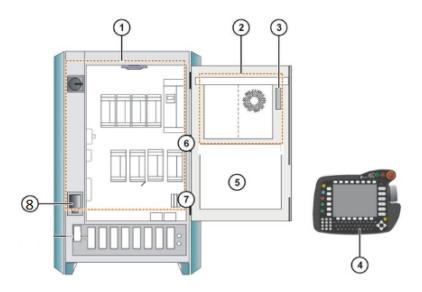


Figure 8: Aperçu de la commande de robot

- 1. Unité de puissance
- 2. PC de commande
- 3. Eléments de commande et d'affichage coupleur KCP
- 4. KCP
- 5. Emplacement installation client
- 6. Logique de sécurité (ESC)
- 7. Carte coupleur KCP (option)
- 8. Prise de service

2.2.2.1.Description de PC de commande :

Le PC assure toutes les fonctions de la commande du robot avec ses composants montés :

- Interface utilisateur Windows avec visualisation et entrée
- Création, correction, archivage, maintenance des programmes
- Commande du déroulement
- Gestion de trajectoire
- Transmission des données des axes au robot
- Surveillance
- Parties du circuit de sécurité
- Communication avec la périphérie externe (autres commandes, ordinateurs pilotes,
 PC, réseau)

2.2.3. <u>Boitier de programmation portatif :</u>

Le smartPAD ou (KCP: KUKA Control Panel) est le boîtier de programmation portatif pour le robot industriel. Il possède toutes les fonctions de commande et d'affichage indispensables à la commande et à la programmation du robot industriel. Le smartPAD dispose d'un écran tactile : l'interface smartHMI peut être utilisée avec un doigt ou un stylet. Une souris externe ou un clavier externe ne sont pas nécessaires.



Figure 9:face avant et arrière du smartPAD

1.	Sélecteur de mode
2.	Bouton ON
3.	bouton OFF
4.	Bouton d'arrêt d'urgence
5.	Souris
6.	Touche d'état de droite
7.	La touche entrée
8.	Touche directionnelle
9.	Clavier

10. clavier numérique
11. Touches programmables
12. Démarrer la touche arrière
13. Touche de démarrage
14. Touche d'arrêt
15. Touche de sélection fenêtre
16. Touche ESC
17. Touche d'état gauche
18. Touche de menu principale

s

2.2.3.1. <u>Sélection des modes :</u>

Le robot industriel peut être exploité dans les modes suivants :

Manuel à vitesse réduite (T1)	Manuel à vitesse élevée (T2)	Automatique (AUT)	Automatique Externe (AUT EXT)
Pour le mode Test de la programmation et Apprentissage	Pour le mode Test (L'apprentissage et la programmation ne sont pas autorisés dans ce mode.)	Pour les robots industriels sans commande prioritaire	Pour les robots industriels avec commande prioritaire (exp: API, capteur)

Tableau 2: Sélection des modes

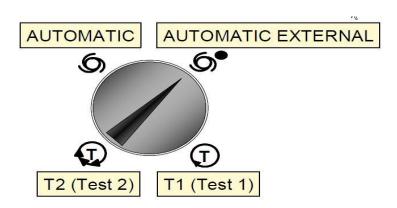


Figure 10: Sélection des modes

2.3. Déplacement manuel du robot :

Il existe deux modes de déplacement manuel :

Déplacement cartésien	Déplacement articulaire		
Le CDO est déplacé dans un sens positif ou négatif le long des axes d'un système de coordonnées.	Chaque axe de robot peut être déplacé individuellement dans le sens positif ou le sens négatif.		

Tableau 3: Déplacement manuel du robot

- ❖ L'espace cartésien : C'est l'espace physique du robot où on définit des coordonnées x et y de l'effecteur. Les positions à atteindre par le robot sont définies dans l'espace cartésien.
- ❖ L'espace articulaire : C'est l'espace qui a pour référence le repère lié à chaque articulation motorisée du robot. Les coordonnées associées sont appelées coordonnées articulaires.

2.4. Enveloppe d'évolution, zones de protection et de danger :

Les enveloppes d'évolution doivent être limitées à la taille minimum requise. Une enveloppe d'évolution est à protéger par des dispositifs de protection.

Les dispositifs de protection (par ex. portes de protection) doivent se trouver dans la zone de protection. Lors d'un stop, le manipulateur et les axes supplémentaires (option) freinent et s'arrêtent dans la zone de danger.

La zone de danger est formée de l'enveloppe d'évolution et des courses d'arrêt du manipulateur et des axes supplémentaires (option).

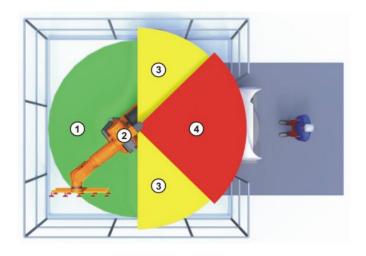


Figure 11: Exemple d'enveloppe d'évolution

- 1. Enveloppe d'évolution

2. Manipulateur

4. Zone de protection

3. Course d'arrêt

2.5. Système de coordonnées :

Les systèmes de coordonnées cartésiens suivants sont définis dans la commande du robot :



• **WORLD:** Le système de coordonnées WORLD est un système de coordonnées fixe et cartésien. Il fait office de système de coordonnées source pour les systèmes de coordonnées BASE et ROBROOT.



• **ROBROOT:** Le système de coordonnées ROBROOT est un système de coordonnées cartésien dont l'origine se trouve toujours au pied du robot. Il décrit la position du robot par rapport au système WORLD.



• **BASE:** Le système de coordonnées BASE est un système cartésien définissant la position de la pièce. Il se réfère au système de coordonnées WORLD.



• TOOL: Le système de coordonnées TOOL est un système de coordonnées cartésien se trouvant au point de travail de l'outil.

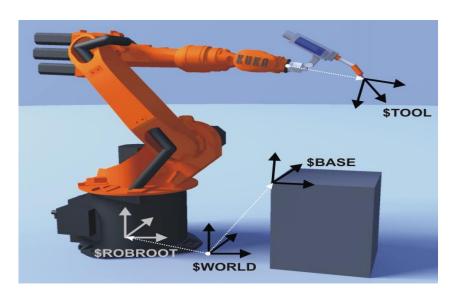


Figure 12: Système de coordonnées

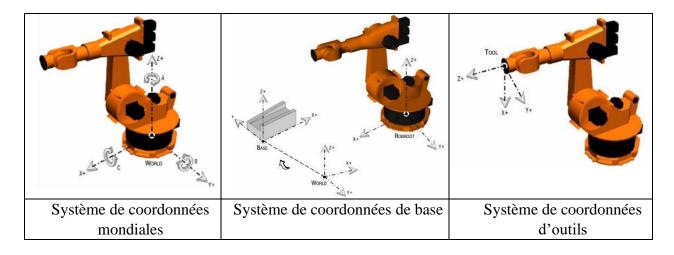


Figure 13: les différents systèmes de coordonnées

2.6. Méthode da calibration :

Chaque robot doit être calibré. Seul un robot calibré est en mesure de faire un déplacement cartésien et d'accoster des positions programmées. La calibration fait concorder la position mécanique et la position électronique du robot. A cette fin, le robot est amené dans une position mécanique définie, la position de calibration. Ensuite, la valeur du capteur est mémorisée pour chaque axe. La position de calibration est similaire pour tous les robots, mais elle n'est pas identique.

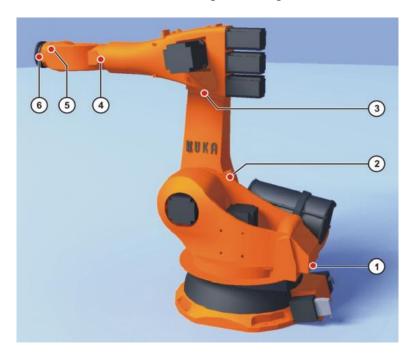


Figure 14: Les positions de calibration

Type de cartouche de mesure	Méthodes de calibration		
Cartouche de mesure pour EMD	Calibration avec l'EMD		
(Electronic Mastering Device)	Calibration avec le comparateur		
Capuchon de protection avec filet à pas fin	Calibration de référence		
M20			
Cartouche de mesure pour MEMD	Pour A1 à A5 : calibration avec le		
(Micro Electronic Mastering Device)	MEMD		
Capuchon de protection avec filet àpas fin M8	Pour A6 : calibration sur le trait de		
	repère		

Tableau 4: Méthodes de calibration

REMARQUE:

En se référant au catalogue du robot KUKA KRC2, la cartouche de mesure est de type EMD (ELECTRONIC MASTERING DEVICE) d'où on va appliquer le calibrage avec le comparateur.

2.6.1. Position de pré calibration :

Les axes doivent être amenés en position de pré calibration avant chaque calibration.

Pour ce faire, chaque axe est déplacé de façon à ce que les repères de calibration soient les uns au-dessus des autres.





Figure 15: Amener les axes en position de pré calibration

2.6.2. Calibration avec le comparateur :

Pour la calibration avec comparateur, la position de calibration est accostée manuellement par l'utilisateur. La calibration se fera toujours sous charge.

Condition préalables :

- La charge est montée sur le robot.
- Tous les axes sont en position de pré-calibration.
- Le mode de déplacement "Touches de déplacement" est actif et les axes sont sélectionnés en tant que système de coordonnées.
- Aucun programme n'est sélectionné.
- Mode T1



Figure 17: Les axes à calibrer



Figure 16: Le comparateur

REMARQUE:

Il est très important de tenir compte du déplacement de l'axe de "+" vers "-". Au point le plus profond de l'encoche de mesure, reconnaissable au changement de sens de l'aiguille, il faut régler le comparateur sur zéro. Cette étape est aussi délicate en ce qui suit.

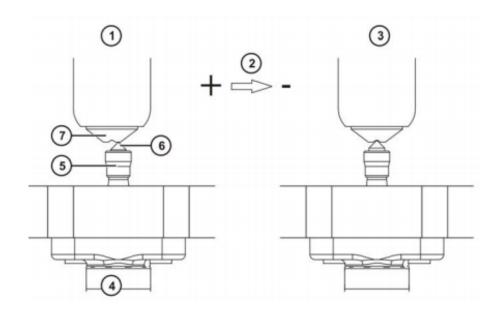


Figure 18: Déplacement de l'axe

- 1. Position de pré-calibration
- 2. Direction du mouvement de l'axe
- 3. Position zéro mécanique

- 4.cartouche de jauge
- 5. Goupille de jauge
- 6. Encoche de référence

2.7. Mesure :

L'objectif de cette étape est d'estimer à partir de mesures la pose réelle de l'effecteur final en fonction des variables articulaires. Une procédure de mesure typique consiste à déplacer le robot vers une position choisie de son espace de travail, à enregistrer les variables articulaires ayant permis d'atteindre cette position et à mesurer avec un système de mesure externe la pose totale (position et orientation) ou partielle (position ou orientation) de l'effecteur.

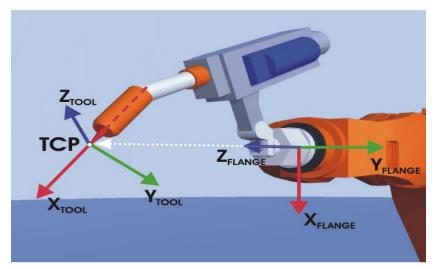


Figure 19: Principe de la mesure CDO

2.7.1. Mesure de l'outil :

Lors de la mesure de l'outil, l'utilisateur affecte un système de coordonnées cartésien (système TOOL) à l'outil monté à la bride de fixation. Le système TOOL a son origine à un point défini par l'utilisateur. Ce point s'appelle CDO (centre de l'outil). Le CDO est normalement positionné au point de travail de l'outil.

La mesure de l'outil se fait en deux phases :

Opération 1 : Définition de l'origine du système de coordonnées TOOL

- XYZ 4 POINTS

ou bien

XYZ Référence

Opération 2 : Définition de l'orientation du système de coordonnées TOOL

- ABC 2 points

ou bien

- ABC World

2.7.1.1. <u>Définition de l'origine : La méthode XYZ 4-point :</u>

Elle consiste à déplacer l'outil par rapport à une référence dans quatre directions différentes. Cette référence peut être un outil spécialement conçu ou libre. Le module de commande du robot calcule le TCP à partir des différentes positions de la base de montage.

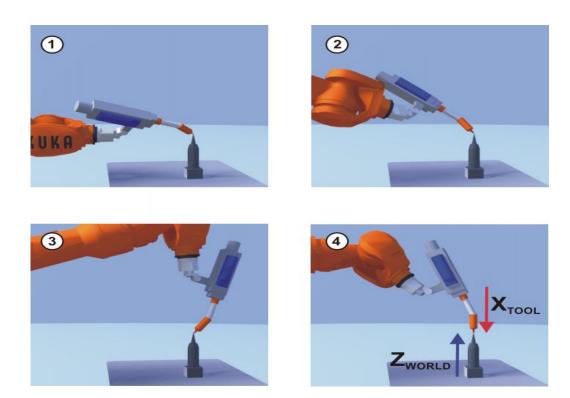


Figure 20: La méthode XYZ 4-point

2.7.1.2. <u>Définition de l'orientation : méthode ABC 2 points :</u>

Les axes du système de coordonnées TOOL sont communiqués à la commande de robot en accostant un point sur l'axe X et un point dans le plan XY. Cette méthode est utilisée s'il s'agit d'obtenir une haute précision dans l'alignement des axes.

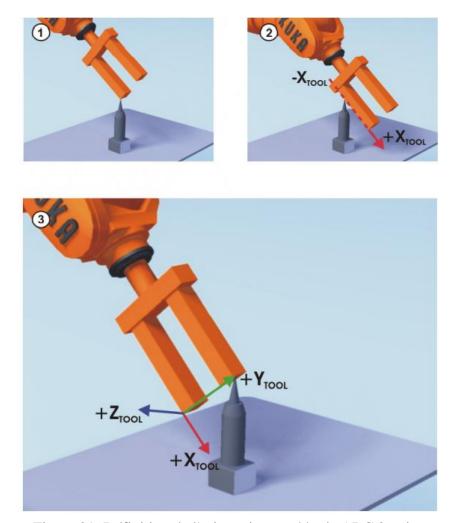


Figure 21: Définition de l'orientation : méthode ABC 2 points

2.7.2. Mesure de la base :

Dans le cas de la mesure de la base, l'utilisateur affecte un système de coordonnées cartésien (système de coordonnées BASE) à la pièce ou à la surface de travail. Le système de coordonnées BASE a son origine à un point défini par l'utilisateur.

2.7.2.1. Méthode 3 points :

L'origine et deux autres points de la base sont accostés. Ces 3 points définissent la nouvelle base.

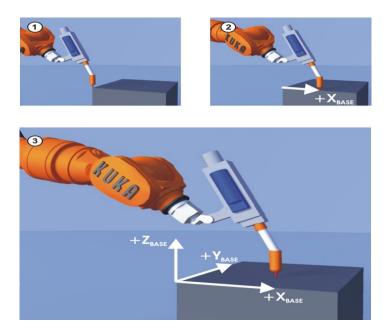


Figure 22: Méthode 3 points

2.7.3. <u>Mesure de la pièce :</u>

2.7.3.1. <u>Méthode directe</u>:

L'origine et deux autres points de la pièce sont communiqués à la commande de robot. Ces 3 points définissent la pièce sans équivoque.

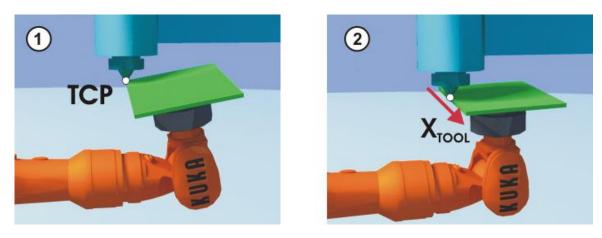


Figure 23: Mesure de la pièce- méthode directe

3. CONCLUSION:

Cette partie est utile pour la manipulation et l'utilisation du robot. Dans ce qui suit, on couvre tout ce qui concerne la programmation et la réalisation pratique.

CHAPITRE 3 : STRUCTURE DE PROGRAMMATION

1. INTRODUCTION:

Dans ce chapitre, on présente le langage de programmation adéquat et l'étude nécessaire pour ajouter un capteur externe avec lequel on va manipuler le robot.

2. LANGAGE DE PROGRAMMATION KRL (KUKA ROBOT LANGUAGE):

2.1. Généralité :

Le langage de robot KUKA, également appelé KRL, est un langage de programmation propriétaire similaire à Pascal et utilisé pour contrôler le bras manipulateur KUKA capable de créer et modifier un programme complexe.

Tous les fichiers de système sont invisibles à l'utilisateur, ils peuvent être vus et également édités par l'expert dans la fenêtre du programme.

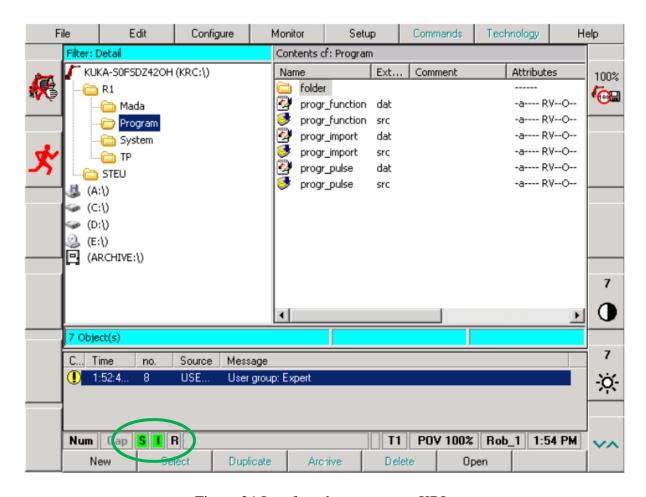


Figure 24:Interface de programme KRL

Activer / désactiver les entraînements :

L'état des entraînements est affiché dans la barre d'état. Les entraînements peuvent être activés ou désactivés :

Symbole	Description
	Les entraînements sont prêts
-	Les entraînements ne sont pas prêts

Affichage de l'état de l'interpréteur robot :

Symbole	Description
R	Aucun programme n'est sélectionné.
R	L'indicateur de bloc se trouve sur la première ligne du programme sélectionné.
R	Le programme est sélectionné et tourne.
R	Le programme sélectionné qui tourne a été arrêté.
R	L'indicateur de bloc se trouve à la fin du programme sélectionné.

Arrêter ou abandonner manuellement l'interpréteur Submit :

symbole	Description
S	L'interpréteur Submit a été arrêté.
S	L'interpréteur Submit est abandonné.
S	L'interpréteur Submit est sélectionné. L'indicateur de bloc se trouve sur la première ligne du programme SUB sélectionné.
S	L'interpréteur Submit tourne.

2.2. Changement de groupe d'utilisateurs :

Il en existe deux modes:

Mode Opérateur	Mode Expert		
Pour tous les travaux devant être exécutés sur le robot industriel pour pouvoir commencer le mode automatique. - Mode pas à pas - Apprentissage - Programmation - Vérification de programme	Groupe pour le programmeur, ce groupe est protégé par un mot de passe.		

Tableau 5: Changement de groupe d'utilisateur

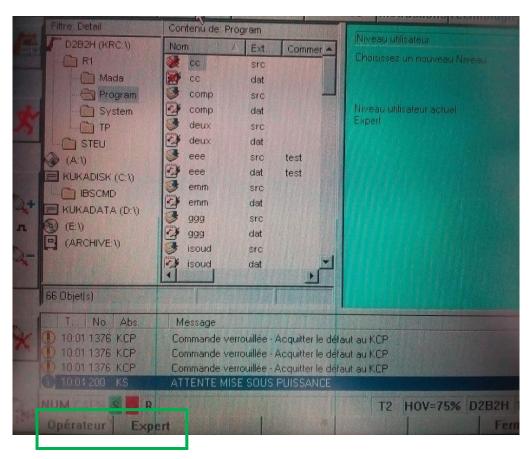


Figure 25: Changement de groupe d'utilisateurs

2.3. Composition d'un programme KRL:

Un programme en KRL (KUKA ROBOT LANGUAGE) peut être constitué d'un ou de plusieurs fichiers. Un programme est souvent composé :

- D'un fichier « .SRC » qui contient le code du programme,
- D'un fichier « .DAT » qui contient une Data List (liste de données) contient les données permanentes et les coordonnées des points.
- → Le fichier SRC et le fichier DAT correspondant se nomment ensemble module

2.3.1. Fichier SRC :

Le fichier source SRC implanté dans le répertoire du robot KUKA permet d'introduire les instructions à faire par le robot en fonction des variables envoyées et reçues.

Chaque fichier SRC commence par la déclaration « DEF » suivie par le nom du fichier et se termine par « END ».

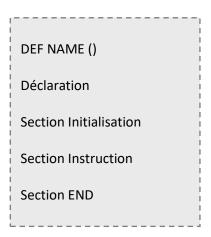


Figure 26: Structure du fichier SRC

2.3.2. Fichier DAT :

Une DATA LIST (liste de données) contient des données spécifiques de programme. Lorsque la liste de données et le fichier de codes ont le même nom, les variables déclarées dans la liste de données peuvent être utilisées comme si ces variables avaient été déclarées dans le fichier SRC.

Un fichier DAT commence par la déclaration « DEFDAT » et se termine par « ENDDAT ». 1 élection modèle Contenu de: Program Commentaire Nom Ext commentaire Propriétés Cell Commutateur Automatique. . 3 solution1 dat -a-- RV--O--Expert Module Expert solution2 STC -a-- RV--O--Expert Submit Submit Expert solution2 dat -a-- RV--O--Function Fonction Modul Module SSS -a-- RV-P--src Submit Submit Utilisateur SSS dat -a-- RV-P--test src -a-- RV--OEtest dat -a-- RV--OEteste2 src -a-- RV--O-teste2 -a-- RV--O-ttt src -a-- RV--OEttt dat -a-- RV--OEun src -a-- RV-P--un dat -a-- RV-P--verin src -a-- RV--O--D282H (KRC1) Ext Commentaire Propriétés -a-- RV-P---SSS src -a-- RV-P---SSS dat Mada test src -a -- RV -- OE-Program test dat -a -- RV -- OE-System teste2 src -a-- RV--O-teste2 -a-- RV--O-dat STEU src -a-- RV--OE-(A:1) ttt -a-- RV--OEdat KUKADISK (C:1) un -a-- RV-P--src BSCMD un dat -a-- RV-P---KUKADATA (D:1) verin src -a-- RV--O-verin dat -a-- RV--O--(ARCHIVE:\) program

1. Module : ensemble des deux fichiers SRC et DAT

Figure 27: Composition d'un programme

- 2. Nom du programme
- 3. Fichier SRC Fichier DAT

2.4. Structure d'un programme KRL:

```
1 DEF my_program()
2 INI
3
4 PTP HOME Vel= 100 % DEFAULT
...
8 LIN point_5 CONT Vel= 2 m/s CPDAT1 Tool[3] Base[4]
...
14 PTP point_1 CONT Vel= 100 % PDAT1 Tool[3] Base[4]
...
20 PTP HOME Vel= 100 % DEFAULT
21
22 END
```

Figure 28: structure d'un programme KRL

- La ligne DEF: affiche le nom du programme. Si le programme est une fonction, la ligne DEF commence avec "DEFFCT" et contient encore d'autres informations. La ligne DEF peut être affichée ou supprimée.

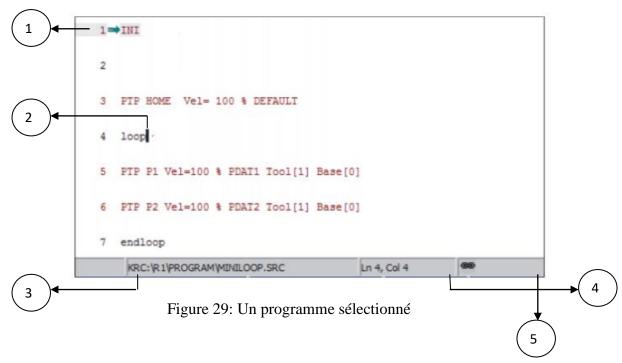
La ligne DEF est supprimée par défaut. Dans un programme, les déclarations ne sont possibles que si la ligne DEF est affichée.

- La ligne INI : contient les initialisations pour les variables et paramètres internes.
- **Position HOME**: La position HOME est une position prioritaire valable pour tous les programmes. Elle est normalement utilisée comme première et dernière position.
- → Par défaut, la commande de robot mémorise la position HOME avec les valeurs suivantes :

Axe	A1	A2	A3	A4	A5	A6
Position	0°	-90°	+90°	0°	0°	0°

Tableau 6: Position des angles - position home

- Déplacement Point To Point.
- Déplacement Linéaire.
- **La ligne END :** c'est la dernière ligne dans chaque programme. Si le programme est une fonction, la ligne END s'appelle "ENDFCT".



- 1. Indicateur de bloc
- 2. Curseur
- 3. Chemin d'accès du programme et nom du fichier
- 4. Position du curseur dans le programme
- 5. Le symbole indique que le programme est sélectionné

2.5. Programmation de mouvement :

Une particularité d'un langage de programmation de robot est la possibilité de programmer des points entre lesquels se déplace le robot TCP (TOOL Center Point). Il y a trois parcours de base modes:

2.5.1. <u>Le Mode de déplacement PTP (Point To Point) :</u>

C'est la fonction la plus simple disponible dans le générateur de trajectoire, ce mode de déplacement apporte un temps de mouvement optimisé du point de départ au point final programmé. Pour ce faire, le contrôleur calcule les différences d'angle nécessaires pour chaque axe.

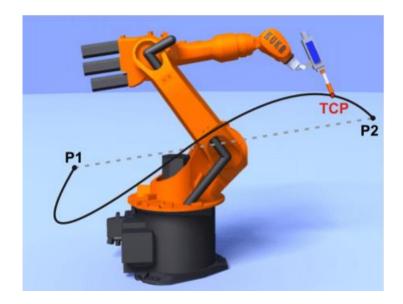


Figure 30: Programmation de mouvement PTP

2.5.2. Le Mode de déplacement LIN (linéaire) :

C'est le plus couramment utilisé. La commande calcule une ligne droite à partir de la position actuelle (le dernier point programmé dans le programme) à une position spécifique.

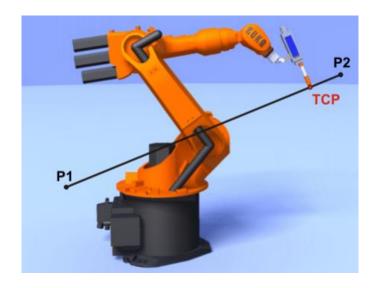


Figure 31: Programmation de mouvement LIN

2.5.3. <u>Le Mode de déplacement CIRC (circulaire) :</u>

Il est utilisé pour définir un cercle ou un arc dans l'espace. Pour programmer un mouvement circulaire, il faut définir le point de départ, le point auxiliaire et le point de destination.

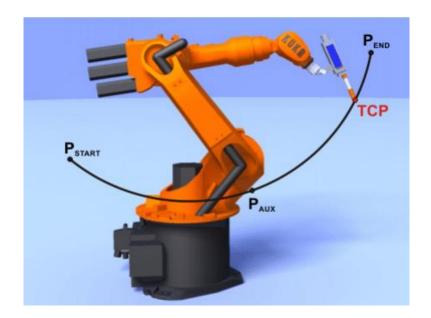


Figure 32: Programmation de mouvement CIRC

2.6. Programmation de déplacement PTP, LIN, CIRC:

2.6.1. Programmation d'un déplacement PTP :

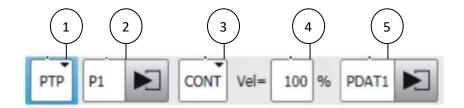


Figure 33: Programmation d'un déplacement PTP

- 1. Type de déplacement PTP
- 2. Nom du point de destination
- 3. CONT : le point de destination est lissé

[vide] : le point de destination est accosté avec précision.

- 4. Vitesse: [1....100%]
- 5. Nom du bloc de déplacement

2.6.2. Programmation d'un déplacement LIN :

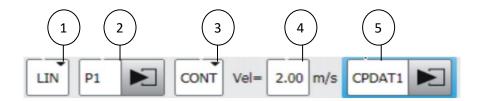


Figure 34: Programmation d'un déplacement LIN

- 1. Type de déplacement LIN
- 2. Nom du point de destination
- 3. CONT : le point de destination est lissé.

[vide] : le point de destination est accosté avec précision.

- 4. Vitesse [0.001... 2m/s]
- 5. Nom du bloc de déplacement

2.6.3. Programmation d'un déplacement CIRC :

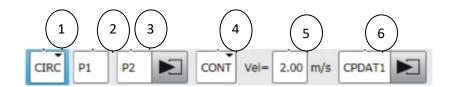


Figure 35:Programmation d'un déplacement CIRC

- 1. Type de déplacement CIRC
- 2. Nom du point auxiliaire
- 3. Nom du point de destination
- 4. CONT : le point de destination est lissé.

[vide] : le point de destination est accosté avec précision.

- 5. Vitesse [0.001...2 m/s]
- 6. Nom du bloc de déplacemenllt

2.7.PROGRAMMATION EXTERNE:

S'il est nécessaire de piloter des processus de robot à partir d'une commande prioritaire (par ex. un API, un capteur, interrupteurs de fin de course, capteurs photoélectriques, blocs de vannes, démarreurs de moteur, capteurs de processus, lecteurs de code, variateurs de fréquence, panneaux d'affichage et interfaces opérateur...), cela se fait par une interface Automatique Externe.

L'interface Automatique Externe permet à la commande prioritaire de transmettre les signaux pour les processus de robot (par ex. autorisation de déplacement, acquittement de défaut, lancement de programme, etc.) à la commande de robot. La commande de robot transmet à la commande prioritaire des informations sur l'état du service et des défauts.

2.7.1. Câblage des E/S du robot avec l'environnement extérieur :

L'opération du soudage avec le robot KUKA KRC2 va être tout simplement avec un capteur externe pour assurer la sécurité de l'opérateur et de l'environnement. Pour cela le responsable doit programmer le robot, ensuite, l'opérateur n'a qu'appuyer sur le capteur et le robot fait son cycle.

Le capteur peut être configuré avec DeviceNet (Master) via MFC.

2.7.1.1. DeviceNet :

DeviceNet est un protocole de communication de couche applicative (couche 7 du modèle OSI) utilisé dans l'industrie pour connecter et administrer à distance une large gamme d'appareils tels que des senseurs en utilisant la technologie CAN (Controller Area Network). Le réseau DeviceNet permet de connecter des dispositifs de niveau bas directement aux automates de l'unité de contrôle. Ce réseau vous permet d'utiliser un seul câble pour interfacer des dispositifs sur une longueur de 500 m maximum avec l'automate programmable. Il peut câbler chaque dispositif à un châssis d'E/S. Ceci apporte une réduction des coûts de câblage et une installation plus rapide.

Le réseau DeviceNet est une solution économique pour la connexion de dispositif de niveau bas, qui fournit un accès à l'intelligence de ces dispositifs.

Câble DeviceNet :

Le câble DeviceNet distant contient quatre fils chacun torsadé avec un seul blindage commun.



Figure 36: Câble DeviceNet

Connecteur DeviceNet :

Avec DeviceNet, le bus inter station se raccorde par l'intermédiaire de 2 x connecteur TWINCOMBICON à 5 broches du connecteur TWIN-COMBICON :

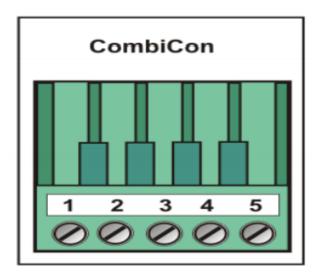


Figure 37: Connecteur DeviceNet

- 1. Alimentation 0V
- 2. CAN Low
- 3. Bouclier
- 4. CAN High
- 5. Alimentation

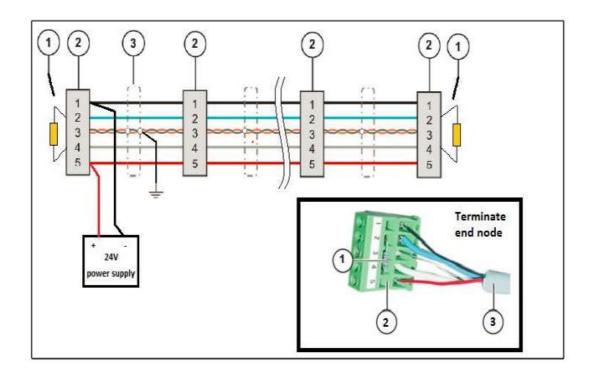


Figure 38: Connexion de Câble DeviceNet

- 1. Une résistance de terminaison (121 Ω / 0,25 W avec 1% de tolérance) doit être montée à chaque extrémité de la ligne principale.
- 2. Connexion.
- 3. Câble DeviceNet.

2.7.1.2. Le module d'entrées/sorties ILB DN 24 DI16 DO16 :

Les dispositifs de conception de blocs préconçus permettent l'intégration d'un certain nombre d'E/S dans un réseau ou un système de bus, ce qui est avantageux pour le traitement d'un petit nombre de signaux d'E/S.

Ces modules IO Inline Block peuvent être directement connectés au bus de terrain en tant qu'esclaves. Le module d'entrée/sortie ILB DN 24 DI16 DO16 contient :

- 16 entrées numériques
- 16 sorties numériques

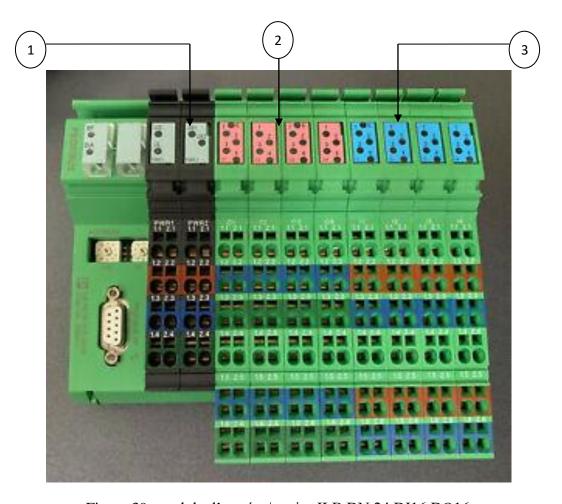


Figure 39: module d'entrées/sorties ILB DN 24 DI16 DO16

- 1. Bloc d'alimentation
- 2. Bloc des sorties numériques
- 3. Bloc des entrées numériques

2.7.2. <u>Configuration de la carte d'entrées/sorties :</u>

La configuration de la carte de bus DEVNET se fait en deux étapes :

- Configurer le fichier DEVNET.INI.
- Configurer le fichier IOSYS.INI.Les deux fichiers DEVNET.INI et IOSYS.INI se trouvent dans le dossier C:\ KRC \ Roboter \ IN \.

2.7.2.1. Configuration du fichier DEVNET.INI:

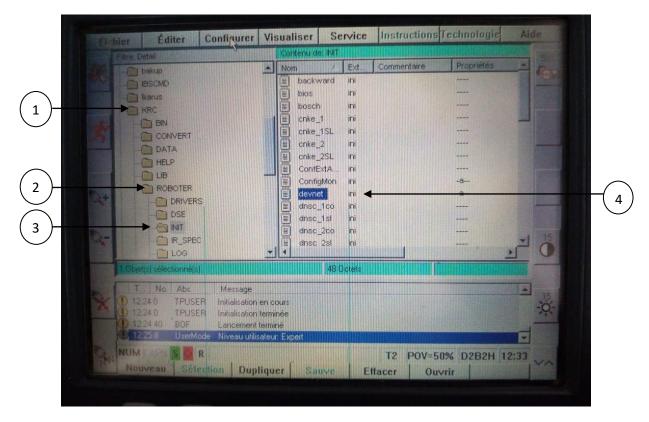


Figure 40: Etape de configuration

Le fichier DEVNET.INI est le fichier de configuration du pilote DeviceNet.

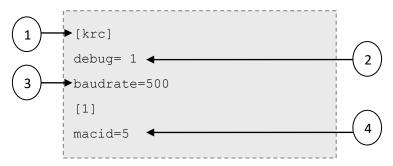


Figure 41: Fichier DIVNET.INI

- 1. **[KRC]:** Affiche les paramètres d'affichage Telnet, vitesse de transmission et de génération d'enregistrement de fichiers.
- 2. **Debug :** informations de diagnostic avancé (en option) 0 = off, 1 = Réglage par défaut 0.

- 3. **BAUDRATE**: Vitesse de transmission de mise en kBaud Valeurs admises : 125, 250, 500 Réglage par défaut : 500.
- 4. **MACID**: doivent être spécifiés dans l'ordre croissant. Valeurs admises : 0 ... 63 Réglage par défaut : 5.

2.7.2.2. Configuration du fichier IOSYS.INI:

Le fichier est IOSYS.INI est le fichier de configuration du système d'E/S. C'est là que les conducteurs de bus sont activés. Les entrées/sorties des dispositifs de bus sont affectées.

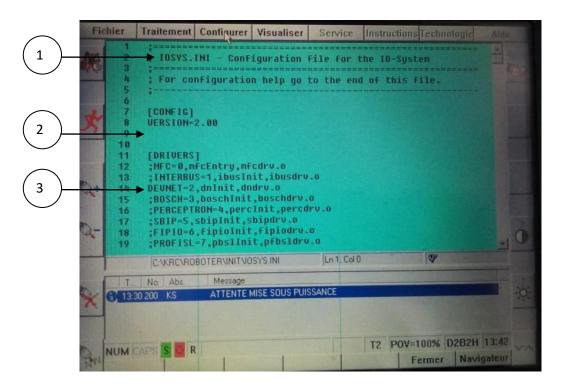
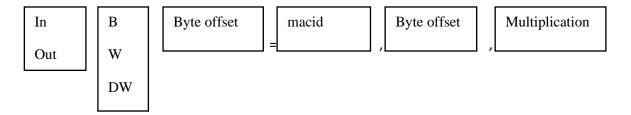


Figure 42: Configuration du fichier IOSYS.INI

- 1. [CONFIG] : Indication du numéro de version.
- 2. [PILOTES]: Les lignes pour l'activation des conducteurs de bus.
- 3. DEVNET = 2, dnInit, dn2drv.o : conducteur DeviceNet.

2.7.2.3. Affectation des entrées et des sorties numériques :



- In/out : entrée ou sortie du contrôleur de robot
- B : Octet = 8 bits (signaux)
- W : Mot = 2 octets = 16 bits (signaux)
- BW : Double mot = 2 mots = 4 octets = 32 bits (signaux)
- Byte offset : décalage d'octet

L'affectation des entrées et des sorties des dispositifs :

```
[DEVNET]
inB0=5, 0, x2
outB0=5, 0, x2
```

Figure 43: Affectation des entrées et des sorties des dispositifs

REMARQUE:

La configuration du deviceNET nous a pris beaucoup du temps, ce n'était pas du tout facile en tant qu'un premier essai. Parmi les échecs que nous avons rencontrés lors de la configuration, une perte de donnée due à une panne de disque dur du PC industriel.

Après une large discussion avec l'encadreur de la société et heureusement qu'il y avait un backup du disque dur que nous avons pu mettre à la place de l'autre, on a décidé de recourir à une autre solution; il s'agit d'un appel de fonction de soudage existante dans le smartPAD. On a tenu compte que l'utilisation de cette solution est limitée contrairement au deviceNET qui est valable pour tous les robots industriels.

2.8. Programmes:

2.8.1. Programme 1 :

```
INI

PTP HOME Vel= 100 % DEFAULT

LIN P1 Vel = 2 m/s CPDAT4 Tool [1] Base [0]

PTP P2 Vel = 100% PDAT1 Tool [1] Base [0]

PTP P10 Vel = 100% PDAT4 Tool [1] Base [0]

LIN P5 Vel = 2 m/s CPDAT5 Tool [1] Base [0]

LIN P6 Vel = 2 m/s CPDAT6 Tool [1] Base [0]

LIN P7 Vel = 2 m/s CPDAT7 Tool [1] Base [0]

LIN P8 Vel = 2 m/s CPDAT8 Tool [1] Base [0]

PTP HOME Vel = 100% DEFAULT
```

2.8.2. Programme 2 :

```
INI

PTP HOME Vel= 100 % DEFAULT

CIRC P1 P3 CONT Vel = 2 m/s CPDAT10 Tool [1] Base [0]

LIN P4 Vel = 2 m/s CPDAT2 Tool [1] Base [0]

PTP P5 CONT Vel = 100% PDAT1 Tool [1] Base [0]

WAIT Time= 2 sec

LIN P6 Vel = 2 m/s CPDAT3 Tool [1] Base [0]

PTP HOME Vel = 100% DEFAULT
```

2.8.3. Programme 3:

```
TNT
PTP HOME Vel= 100 % DEFAULT
LIN P1 CONT Vel = 2 m/s CPDAT7 Tool [1] Base [0]
CIRC P2 P13 CONT Vel = 2 m/s CPDAT8 Tool [1] Base [0]
PTP P14 CONT Vel = 100% PDAT1 Tool [1] Base [0]
LIN P15 CONT Vel = 2 m/s CPDAT9 Tool [1] Base [0]
CIRC P16 P17 CONT Vel = 2 m/s CPDAT10 Tool [1] Base [0]
LIN P18 CONT Vel = 2 m/s CPDAT11 Tool [1] Base [0]
WAIT Time= 5 sec
CIRC P19 P20 CONT Vel = 2 m/s CPDAT12 Tool [1] Base [0]
LIN P21 CONT Vel = 2 m/s CPDAT13 Tool [1] Base [0]
PTP P22 CONT Vel = 100% PDAT2 Tool [1] Base [0]
LIN P23 CONT Vel = 2 m/s CPDAT14 Tool [1] Base [0]
CIRC P24 P25 CONT Vel = 2 m/s CPDAT15 Tool [1] Base [0]
LIN P26 CONT Vel = 2 m/s CPDAT16 Tool [1] Base [0]
PTP HOME Vel = 100% DEFAULT
```

REMARQUE:

Les trois programmes ci-dessus étaient nos premiers essais ; des programmes simples pour bien comprendre les différents modes de déplacements du bras manipulateur du robot.

3. CONCLUSION:

On ne peut nier qu'il s'agissait de la partie la plus importante dans notre projet. Au cours duquel nous avons eu la chance de collecter une quantité décente d'informations sur la robotique industrielle.

```
INI
PTP HOME Vel= 100 % DEFAULT
PTP P1 Vel = 100% PDAT1 Tool [1]:tableeur Base [0]
LIN P2 CONT Vel = 2 m/s CPDAT1 Tool [1]:tableeur Base [0]
PTP P3 Vel = 100% PDAT2 Tool [1]:tableeur Base [0]
PTP P4 Vel = 100% PDAT3 Tool [1]:tableeur Base [0]
LIN P5 CONT Vel = 2 m/s CPDAT2 Tool [1]:tableeur Base [0]
PTP P6 Vel = 100% PDAT4 Tool [1]:tableeur Base [0]
PTP P7 Vel = 100% PDAT5 Tool [1]:tableeur Base [0]
LIN P8 CONT Vel = 2 m/s CPDAT3 Tool [1]:tableeur Base [0]
PTP P9 Vel = 100% PDAT6 Tool [1]:tableeur Base [0]
PTP P10 Vel = 100% PDAT7 Tool [1]:tableeur Base [0]
LIN P11 CONT Vel = 2 m/s CPDAT4 Tool [1]:tableeur Base [0]
LIN P12 CONT Vel = 2 m/s CPDAT5 Tool [1]:tableeur Base [0]
LIN P13 CONT Vel = 2 m/s CPDAT6 Tool [1]:tableeur Base [0]
LIN P14 CONT Vel = 2 m/s CPDAT7 Tool [1]:tableeur Base [0]
LIN P15 CONT Vel = 2 m/s CPDAT8 Tool [1]:tableeur Base [0]
PTP P16 Vel = 100% PDAT8 Tool [1]:tableeur Base [0]
PTP P17 Vel = 100% PDAT9 Tool [1]:tableeur Base [0]
LIN P18 CONT Vel = 2 m/s CPDAT9 Tool [1]:tableeur Base [0]
LIN P19 CONT Vel = 2 m/s CPDAT10 Tool [1]:tableeur Base [0]
LIN P20 CONT Vel = 2 m/s CPDAT11 Tool [1]:tableeur Base [0]
PTP P21 Vel = 100% PDAT10 Tool [1]:tableeur Base [0]
PTP P22 Vel = 100% PDAT11 Tool [1]:tableeur Base [0]
PTP P23 Vel = 100% PDAT12 Tool [1]:tableeur Base [0]
LIN P24 CONT Vel = 2 m/s CPDAT12 Tool [1]:tableeur Base [0]
PTP P25 Vel = 100% PDAT13 Tool [1]:tableeur Base [0]
PTP P26 Vel = 100% PDAT14 Tool [1]:tableeur Base [0]
LIN P27 CONT Vel = 2 m/s CPDAT13 Tool [1]:tableeur Base [0]
PTP P28 Vel = 100% PDAT15 Tool [1]:tableeur Base [0]
PTP P29 Vel = 100% PDAT16 Tool [1]:tableeur Base [0]
LIN P30 CONT Vel = 2 m/s CPDAT14 Tool [1]:tableeur Base [0]
PTP HOME Vel = 100% DEFAULT
```

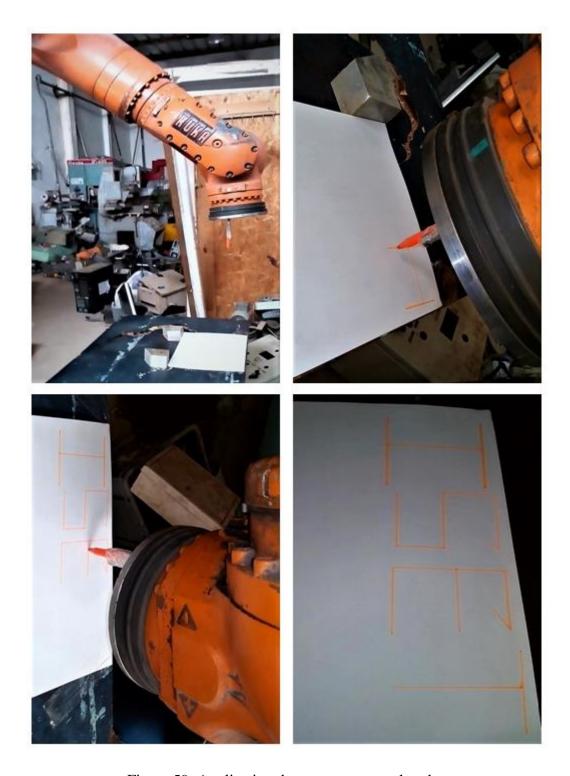


Figure 59: Application du programme par le robot