



# Analyste Programmeur en Automatisation, Robotique et Informatique Industrielle TS ARII

## *Module MF 1.2*

*Analyser un traitement numérique*

*Appliquer les systèmes de codage à la Robotique*

**Patrick MONASSIER** – année 2019-2020

## Module 1.2

### Analyser un traitement numérique

#### Compétences

- **Analyser un traitement numérique.**

#### Objectifs

- *Appliquer les systèmes de codage aux automatismes analogiques et numériques.*
- **Appliquer les systèmes de codage à la robotique.**

#### Contenu

##### Codage des automatismes

- *Présentation des variables et objets utilisables en automatisation*
- *Elaboration d'un organigramme*
- *Construction d'un algorithme appliqué aux automatismes.*
- *Application sur les éditeurs dédiés (SCL pour Siemens et DFB pour Schneider)*
- *Développement des fonctions bibliothèques.*

##### Codage des robots

- **Introduction à la programmation d'un robot : architecture logicielle, contrôle des événements et des trajectoires**
- **Architecture d'une programmation appliquée aux robots, contrôle des trajectoires et repères**
- *Construction d'une programmation appliquée aux robots*
- *Elaboration d'une séquence robotique*
- *Exemples industriels*

## Présentation



### Présentation : Les topologies des robots industriels - Leurs domaines d'application - Les principales marques industrielles – Les matériels et logiciels constructeurs

Ce chapitre donne un aperçu des **topologies de robots**, de leurs **domaines d'utilisations** par rapport aux exigences et contraintes industrielles. Il permet en cela à l'auditeur de positionner les différentes architectures et leurs usages

Ce chapitre permet aussi référence aux **principales marques de robots** utilisées dans l'industrie, de façon à ce que l'auditeur puisse accéder aux **bases documentaires** des constructeurs, principalement par les sites internet, mais aussi par des approches directes

Ce chapitre présente aussi les **principaux éléments** constituant les robots industriels et leurs **logiciels associés**

## Topologies et domaines d'utilisations

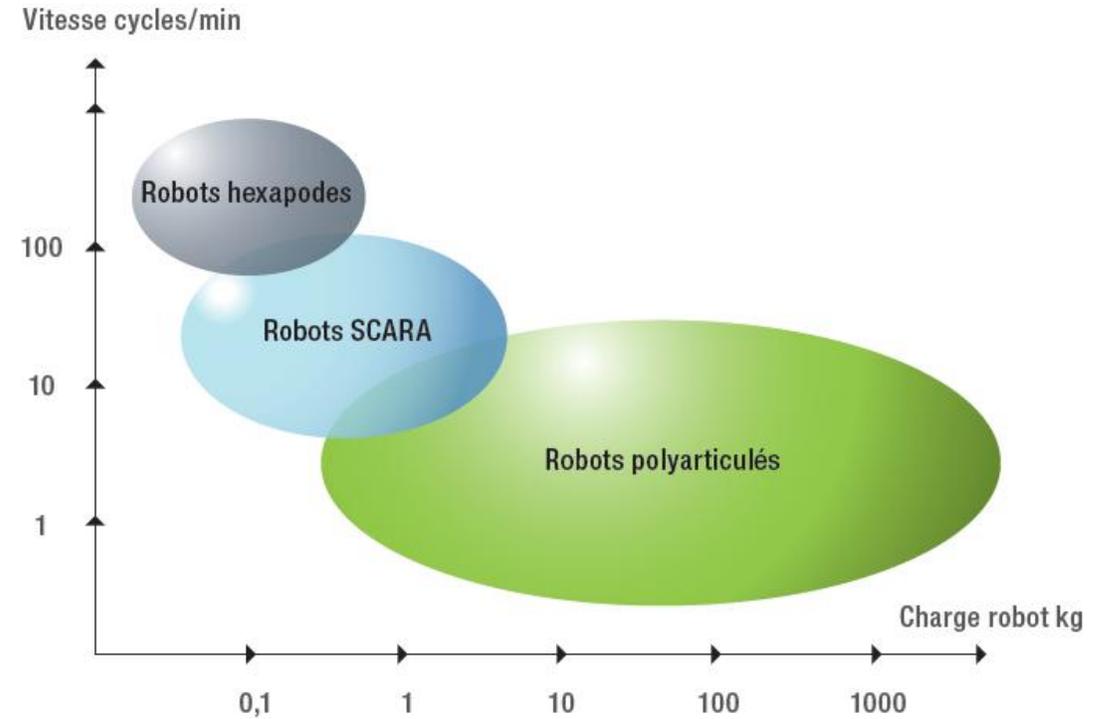


Robots hexapodes

Robots polyarticulés 6 axes  
*Possible de 4 à 7 axes*



Robots SCARA 3 ou 4 axes





**ABB Robotics** <https://new.abb.com/products/robotics>

ABB est une société multinationale dont le siège est à Zürich, en Suisse, et spécialisée dans les industries de l'énergie et de l'automatisation ainsi que la robotique. Avec plus de quatre décennies d'expérience, ABB Robotics est un leader mondial dans la fabrication de robots industriels. ABB Robotics compte environ 4600 employés dans 53 pays, et a déjà installé plus de 250 000 robots dans le monde entier.



**Fanuc** <https://www.fanuc.eu/fr/fr>

La société Fanuc Corporation, dont le siège se trouve dans la préfecture de Yamanashi au Japon, fournit des produits et services d'automatisation comme la robotique et des systèmes de commande numérique d'ordinateurs. C'est l'un des plus grands fabricants de robots industriels du monde. Les robots Fanuc sont utilisés dans l'aérospatiale, l'automobile, les biens de consommation et de nombreuses autres industries. Selon le site de l'entreprise, plus de 250 000 robots Fanuc ont été installés dans le monde.



### Kuka

<https://www.kuka.com/fr-fr>

Kuka, société dont le siège est à Augsburg, en Allemagne, est un important fabricant de robots industriels et de solutions pour l'automatisation industrielle. Elle a développé son premier robot industriel en 1973, et aujourd'hui, ses robots sont entre autres utilisés dans l'industrie automobile, les plastiques, la métallurgie, et l'industrie de l'électronique.



**Stäubli** <https://www.staubli.com/fr-fr/robotics/>

Stäubli est une société de mécatronique suisse, dont la spécialité est les machines textiles, les connecteurs et les produits de robotique. Sa division robotique a été fondée en 1982. La société produit des robots Scara, des robots à 4 axes et 6 axes et d'autres types de robots destinés à l'automatisation industrielle. Les robots Stäubli sont utilisés dans les matières plastiques, l'électronique, l'énergie photovoltaïque, les sciences de la vie et de nombreux autres secteurs. Plus de 1800 employés dans l'Usine de Faverges en Haute-Savoie

 **Kawasaki**  
Robotics



**Kawasaki** <https://robotics.kawasaki.com>

Kawasaki Heavy Industries est une société internationale dont le siège est au Japon, et qui produit des motos, des bateaux, des tracteurs, des moteurs, des équipements aéronautiques, ainsi que des robots industriels, parmi beaucoup d'autres. Avec plus de 45 ans d'expérience dans le développement des robots, la société a déjà installé plus de 120 000 robots dans le monde. Ses produits sont utilisés dans l'assemblage, la manipulation, la soudure, la peinture, l'étanchéité et de nombreux autres processus industriels.

 **YASKAWA**  
MOTOMAN ROBOTICS

**Yaskawa** <https://www.yaskawa.fr/>



La société Yaskawa Electric Corporation, fondée en 1915 et dont le siège est à Kitakyushu, dans la préfecture de Fukuoka, au Japon, est un fabricant japonais de servocommandes, de contrôleurs de mouvement, de commandes de moteur à courant alternatif, de commutateurs et de robots industriels. Depuis la première version d'un robot industriel tout-électrique appelé « Motoman » sorti au Japon en 1988, les robots Motoman fabriqués par l'entreprise sont largement utilisés dans le monde entier, ils sont aujourd'hui vendus sous la marque Yaskawa. Les robots sont utilisés dans le soudage à l'arc, le soudage par points, la manipulation, l'assemblage, la peinture et autres procédés industriels.



### Universal Robots

<https://www.universal-robots.com/fr>

Universal Robots est un fabricant danois de petits bras de robots collaboratifs industriels flexibles (cobots) basé à Odense, au Danemark. La société compte plus de 620 employés et 300 partenaires à travers le monde. Universal Robots a développé l'idée de rendre la technologie robotique accessible aux petites et moyennes entreprises. En 2008, les premiers cobots UR5 ont été lancés puis l'UR10 en 2012 et l'UR3 en 2015. En 2018, une nouvelle génération de cobots d'Universal Robots, baptisée e-Series, a été lancée, rehaussant le standard des robots collaboratifs.



### Epson <https://www.epson.fr/robots>

Epson Robots est la division de conception et de fabrication robotique de la société japonaise Seiko Epson, un des plus grands fabricants mondiaux d'imprimantes pour ordinateurs et d'appareils liés à l'image. Elle compte plus de 45 000 robots installés dans le monde. Ayant créé à l'origine des robots destinés à une utilisation dans ses usines d'horlogerie, la société japonaise a poursuivi par la création d'une gamme de robots compacts de haute précision et à grande vitesse.

## Principaux éléments et logiciels constructeurs

Un robot industriel est un ensemble « autonome » constitué de **trois éléments** :

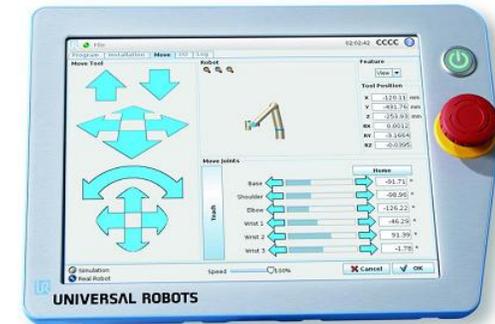
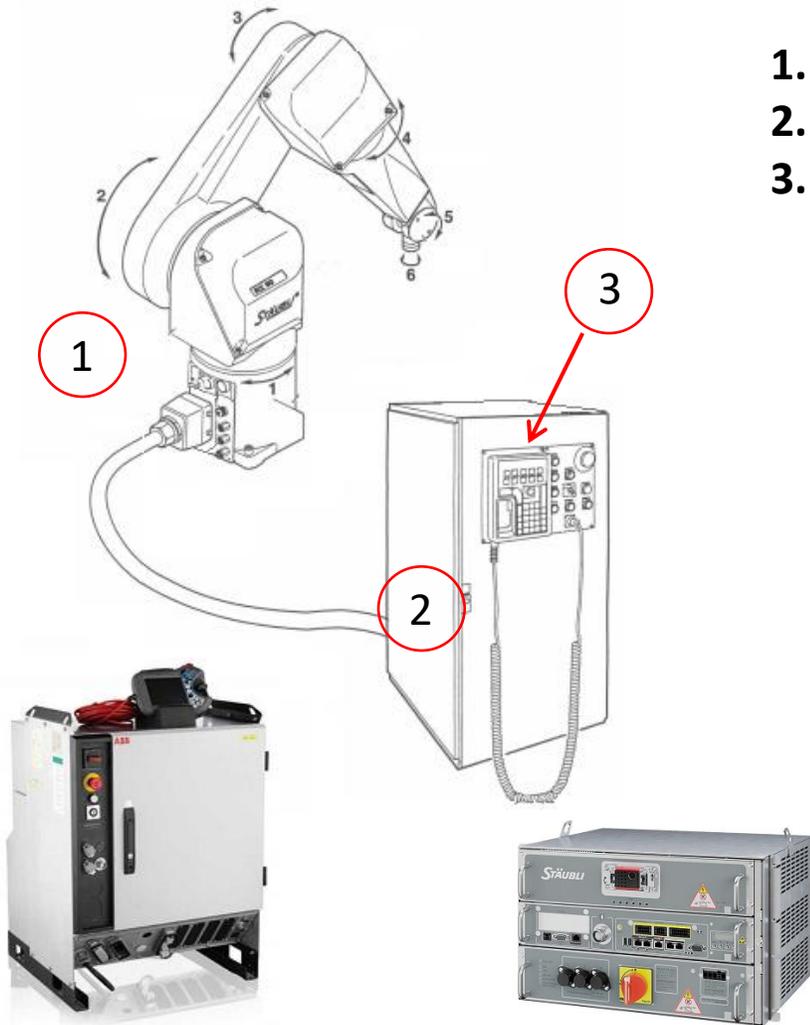
- ✓ **Un bras** : élément mécanique principal en mouvement, appelé ainsi par analogie avec le corps humain. Même si certains n'ont plus grand-chose à voir avec un bras humain, ils remplissent au final la même fonction : déplacement dans l'espace afin de répondre aux multiples applications pouvant être robotisées.
- ✓ **L'armoire de commande** : ou « **baie robot** », comporte l'électronique qui va piloter le robot dans ses tâches.
- ✓ **Pupitre de commande** : appelé aussi « **teach pendant** » ou boîtier opérateur, il s'agit du tableau de commande déporté du robot qui permet d'effectuer sa **programmation**.



## Principaux éléments et logiciels constructeurs

Positionnement des 3 trois éléments :

1. Bras
2. Armoire de commande : ou baie robot
3. Pupitre de commande : appelé aussi teach, teach pendant ou boîtier opérateur



## Principaux éléments et logiciels constructeurs



Programmation manuelle sur le **Teach**

### Principaux langages

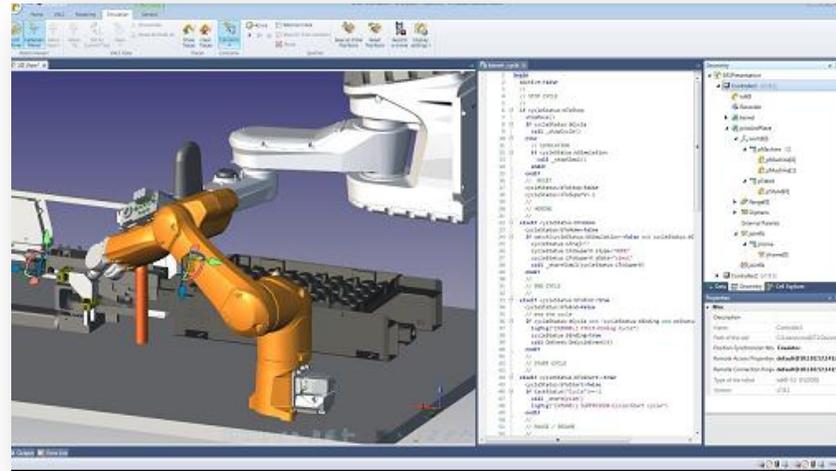
**FANUC** : TPE / KAREL

**ABB** : RAPID

**KUKA** : KRL

**STAÜBLI** : VAL3

...



Programmation hors ligne sur **logiciel de Simulation**

### Principaux simulateurs

**ABB** : RobotStudio

**KUKA** : KUKA.Sim

**FANUC** : ROBOGUIDE

**STAÜBLI** : S.R.S. *STAUBLI ROBOTICS SUITE*

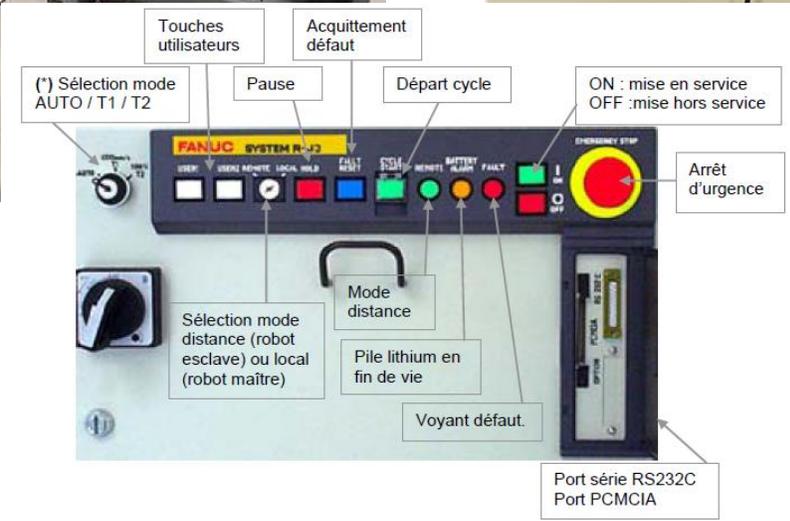
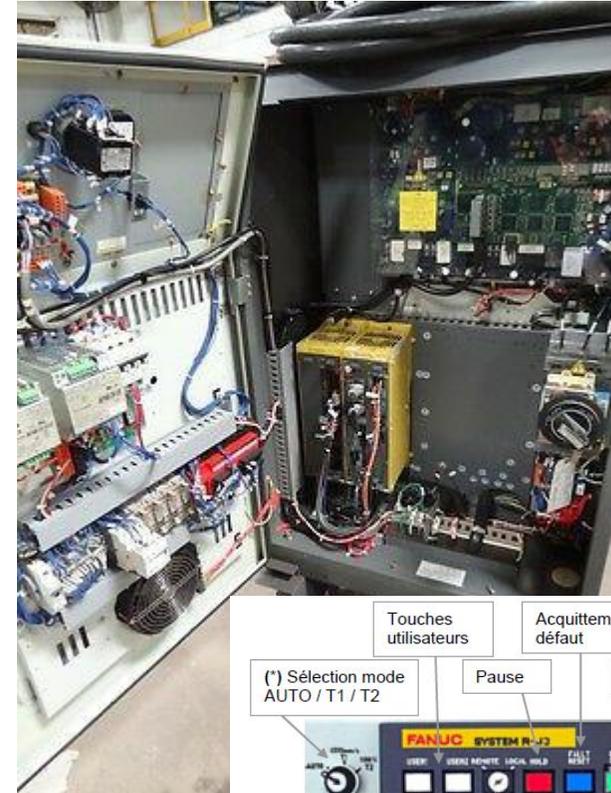
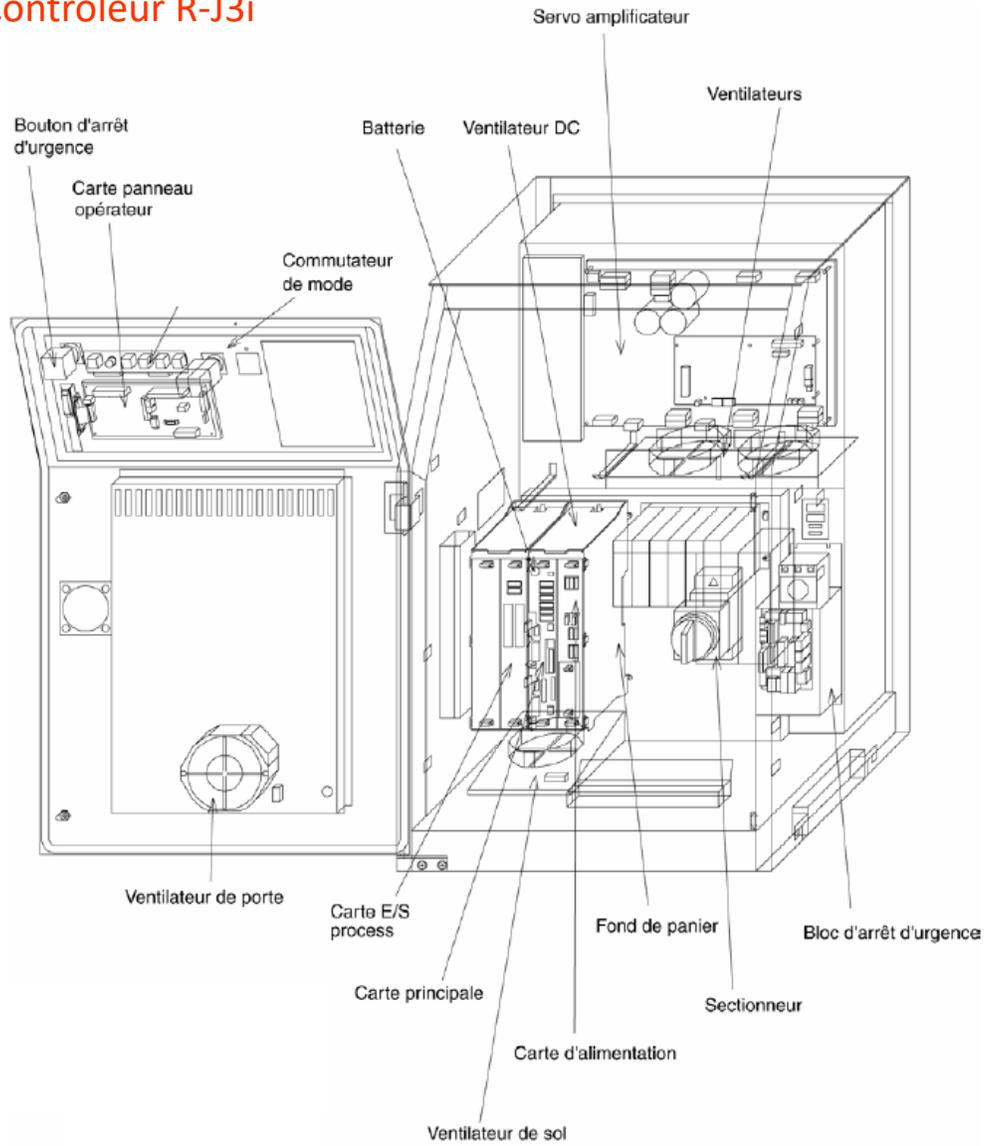
...

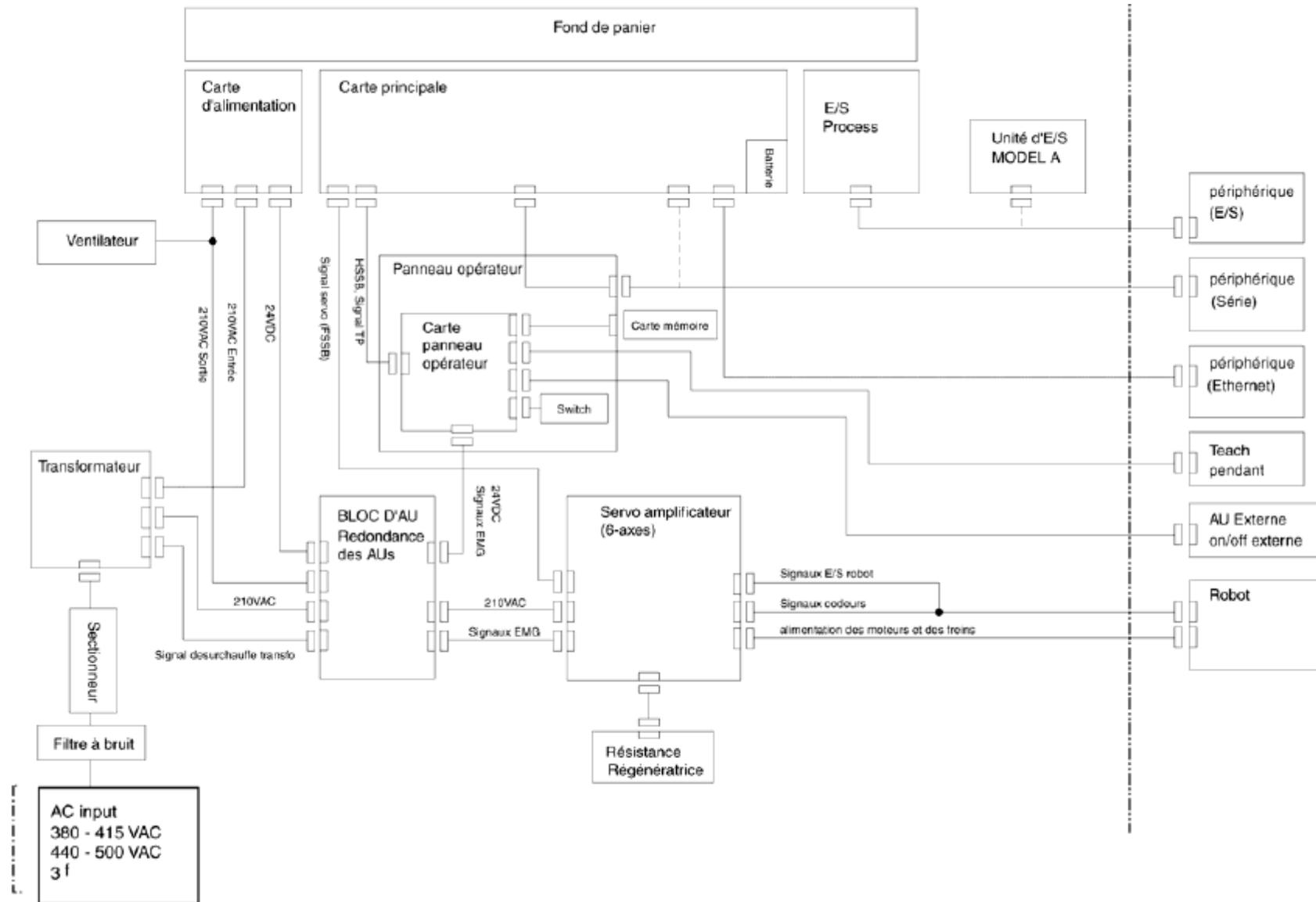
## Codage des robots

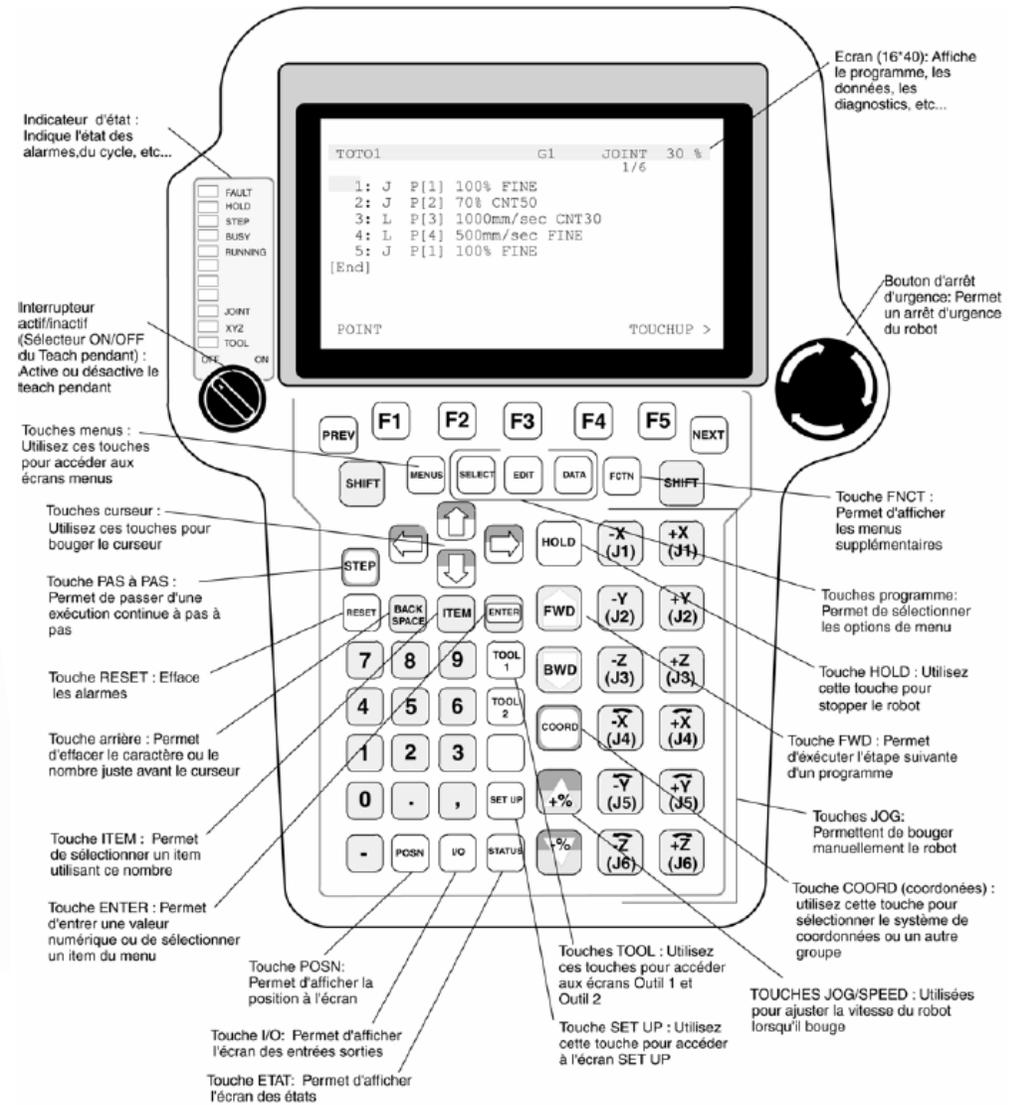
Ce chapitre présente l'architecture logicielle de programmation, le contrôle des évènements, des repères et des trajectoires pratiqués couramment dans le domaine de la robotique industrielle

Pour pouvoir travailler sur des bases pratiques, ce chapitre s'appuie sur la présentation du langage TPE de FANUC et son contrôleur R-J3i

Les règles et principes de codages décrits dans ce chapitre et les chapitres suivants sont facilement portables sur d'autres marques de robots, dont les principaux leaders mondiaux









### Touches relatives aux menus

- F1 à F5** sélectionnent le menu de la dernière ligne d'écran
- NEXT** permet d'aller à la page suivante du menu fonction de l'écran
- PREV** permet d'aller à la page précédente du menu fonction de l'écran
- MENUS** affiche l'écran menu
- FCTN** affiche le menu fonction
- SELECT** affiche l'écran de sélection des programmes
- EDIT** affiche l'écran d'édition de programme
- DATA** affiche l'écran des données de programme
- STATUS** affiche l'écran des statuts du robot
- I/O** affiche l'écran des entrées/sorties
- POSN** affiche l'écran de la position courante

### Touches relatives à l'exécution

- FWD** et **BWD** (+ touche SHIFT) démarrent le programme. Quand la touche SHIFT n'est plus maintenue durant l'exécution, le robot s'arrête.
- HOLD** arrête le programme
- STEP** sélectionne le mode pas à pas ou le mode continu



## Touches relatives aux mouvements du robot

**SHIFT** est utilisée pour exécuter un mouvement avec le robot, apprendre une position et démarrer un programme

La touche **COORD** *cordonnée* permet de sélectionner le système de coordonnées en mouvement manuel (type JOG). Chaque fois que la touche COORD est pressée, cela sélectionne le type suivant : JOINT, JGFRM, World frame, TOOL, USER. Quand cette touche est pressée et que le SHIFT est maintenu, le menu pour changer de repère actif apparaît.

Les touches **OVERRIDE** ajustent la vitesse. *incrémente ou décrémente la vitesse robot de 5 en 5%*

Les touches de **mouvement** sont effectives tant que la touche SHIFT est pressée. Elles sont utilisées pour les mouvements robot.

## Touches relatives à l'édition

**ENTER** permet de valider des valeurs numériques ou de sélectionner un menu.

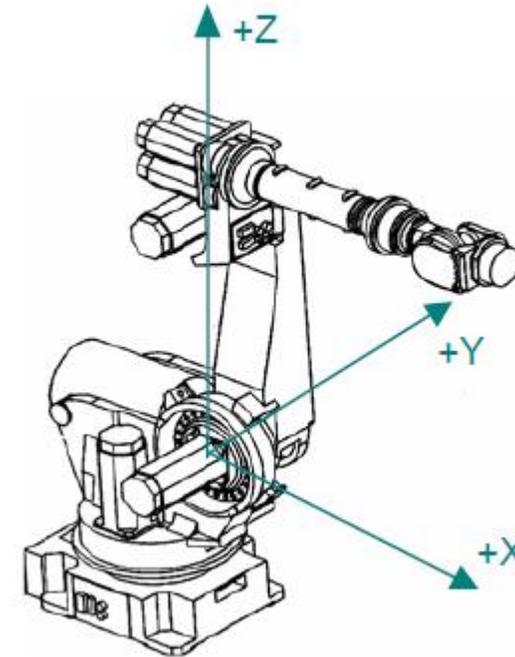
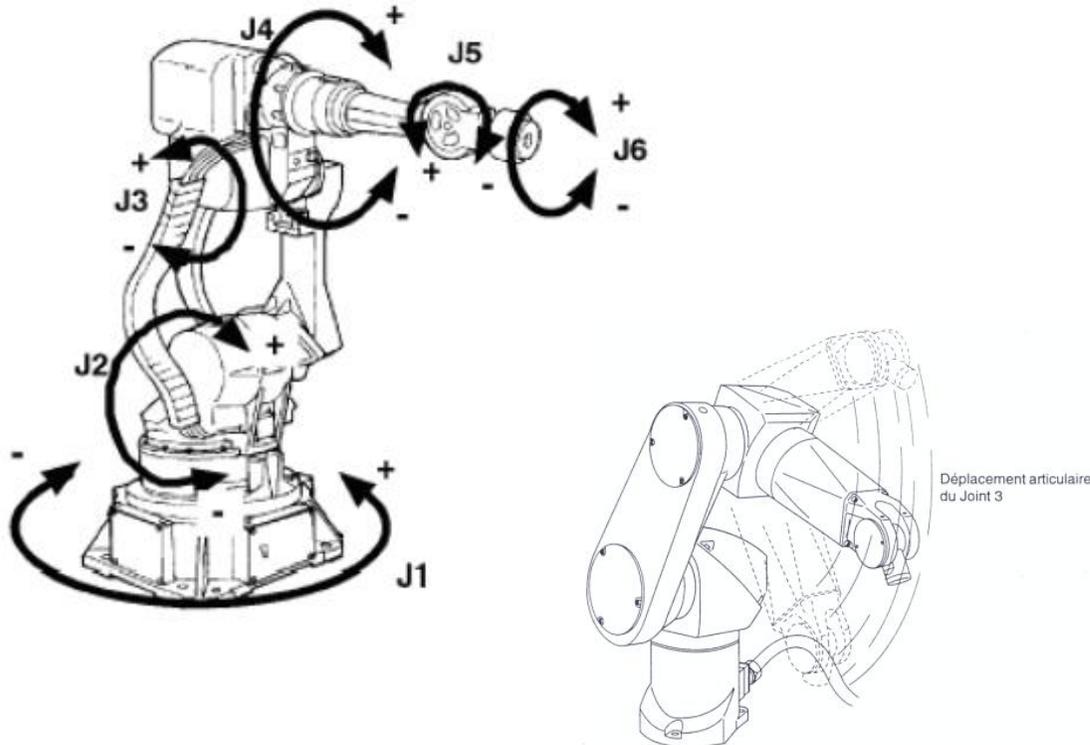
**BACK SPACE** efface le caractère ou le nombre immédiatement avant le curseur.

Les touches **curseurs** bougent le curseur.

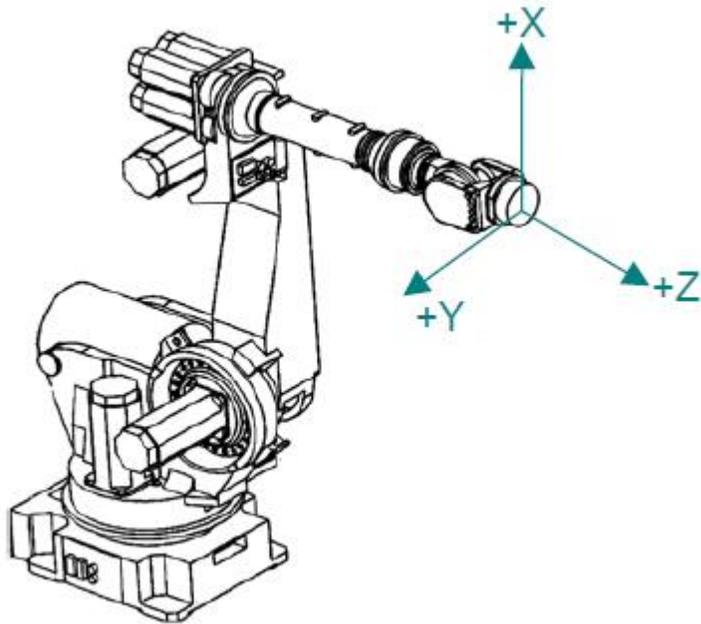
**ITEM** déplace le curseur à la ligne dont le numéro est spécifié

Déplacement axe par axe → JOINT

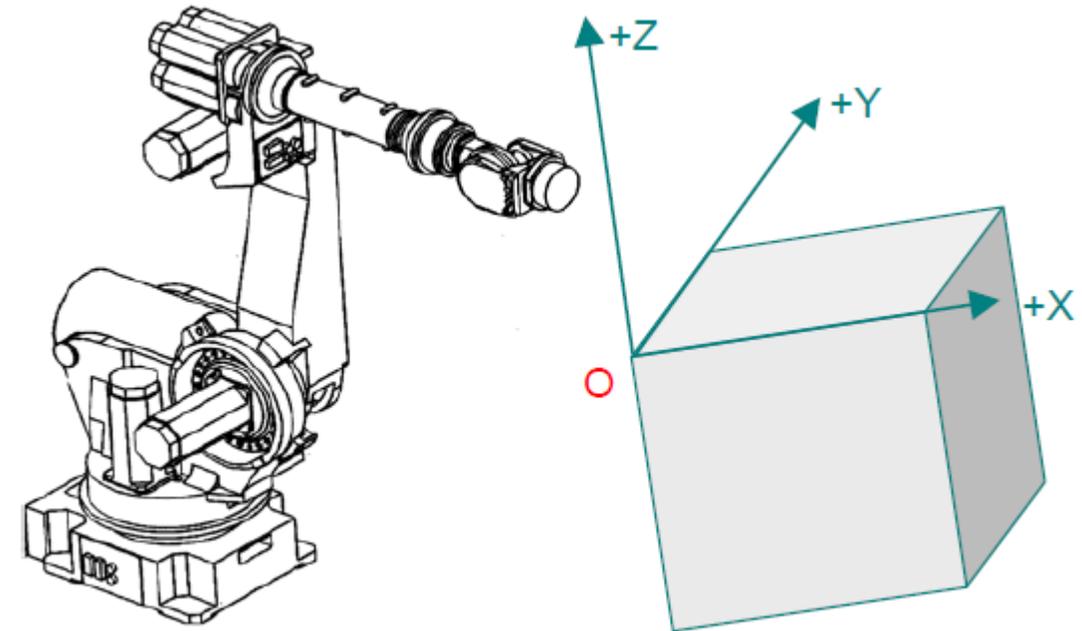
Déplacement dans le repère universel → WORLD



Déplacement dans un repère outil → TOOL



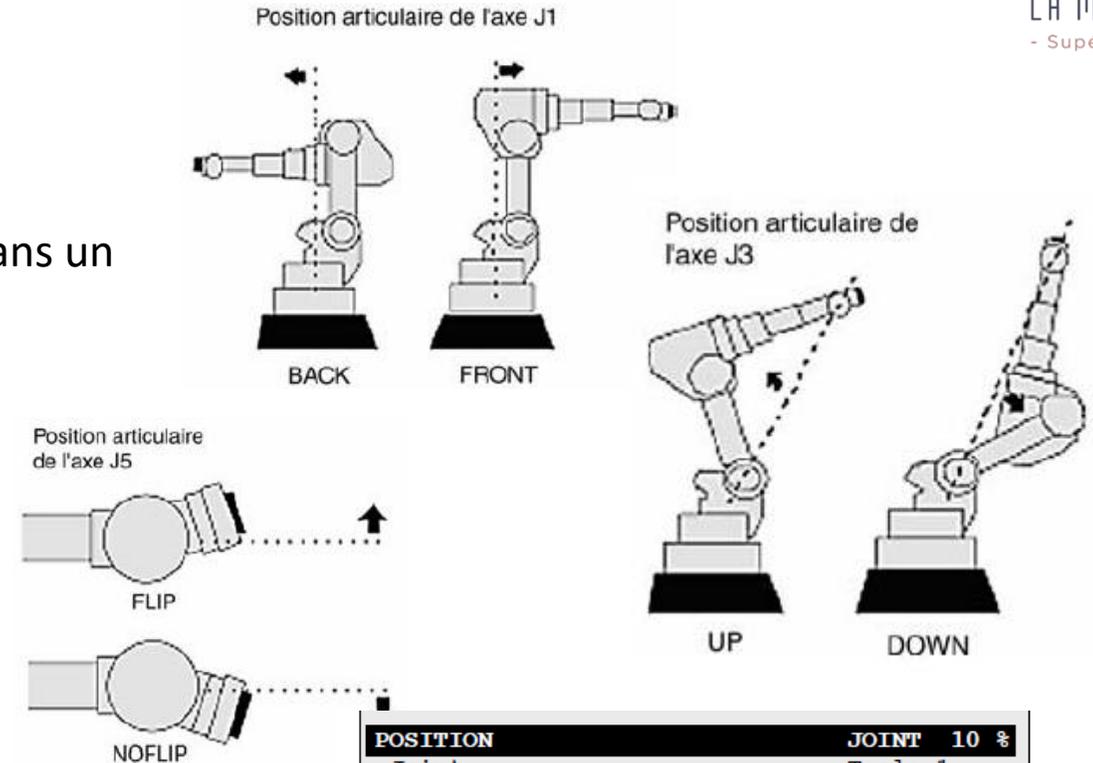
Déplacement dans un repère utilisateur → USER  
Repère de déplacement → JOG FRAME



## Position du robot dans l'espace

Une position peut être exprimée de 2 façons

- en **JOINT** : Valeurs angulaires sur chaque axe (en degrés)
- en **Cartésien** : Coordonnées cartésiennes du repère outil dans un repère utilisateur
  - UT est le numéro du repère outil utilisé pour ce point
  - UF est le numéro du repère utilisateur utilisé pour ce point
  - X, Y et Z sont la localisation du centre outil dans le repère
  - W, P et R sont l'orientation de l'outil dans le repère
  - CONF est la configuration du robot



<b>F : FLIP</b> <b>N : NOFLIP</b>	Poignet « cassé »(positionné vers le haut) Poignet « non cassé »(positionné vers le bas)
<b>L : LEFT</b> <b>R : RIGHT</b>	Bras positionné à gauche Bras positionné à droite <i>(uniquement sur modèle SCARA)</i>
<b>U : UP</b> <b>D : DOWN</b>	Avant bras positionné vers le haut Avant bras positionné vers le bas
<b>T : TOWARD</b> <b>B : BACKWARD</b>	Bras positionné vers l'avant Bras positionné vers l'arrière

POSITION			JOINT 10 %		
			Tool: 1		
J1:	21.505	J2:	-48.755	J3:	-12.381
J4:	9.322	J5:	20.388	J6:	-74.872

POSITION			JOINT 10 %		
User		Frame: 1	Tool: 1		
Configuration: N U T, 0, 0, 0					
x:	734.030	y:	1356.688	z:	751.781
w:	179.995	p:	.001	r:	-7.426

Un **nom** de programme (1) est spécifié pour identifier les programmes stockés dans la mémoire de l'unité de contrôle. un **commentaire** (3) peut être ajouté au nom de programme. La **protection** contre l'écriture (5) spécifie si le programme peut être modifié.



Le **type de sub** (2) est utilisé pour configurer un programme.

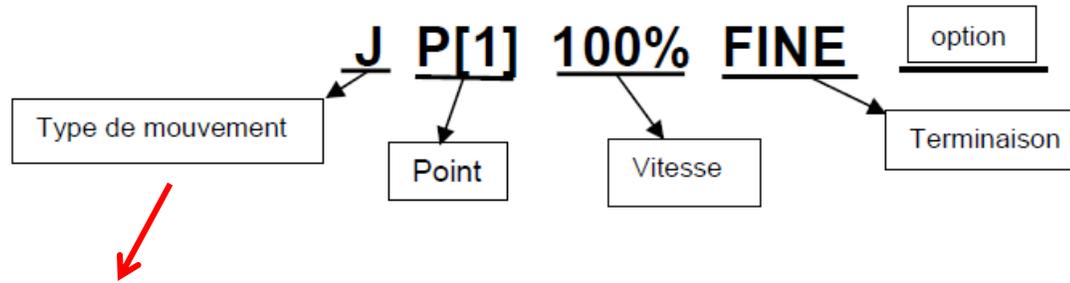
None : représente un programme qui peut être démarré en utilisant un équipement comme le Teach Pendant.

Macro (MR): permet l'exécution d'une macro.

Cond (CH) : programme qui permet de gérer une condition « handler

Un **groupe de masque** (4) détermine quel groupe d'axes est géré par un programme. Cela représente un groupe de différents axes (moteurs) utilisés pour des robots indépendants, un positionneur et autres balancelles.

La désactivation **interruption** (ignore pause) permet à un programme en cours d'exécution (sans groupe de mouvement) de ne pas être interrompu par une alarme type arrêt d'urgence ou arrêt temporaire. (sur 2<sup>e</sup> écran)

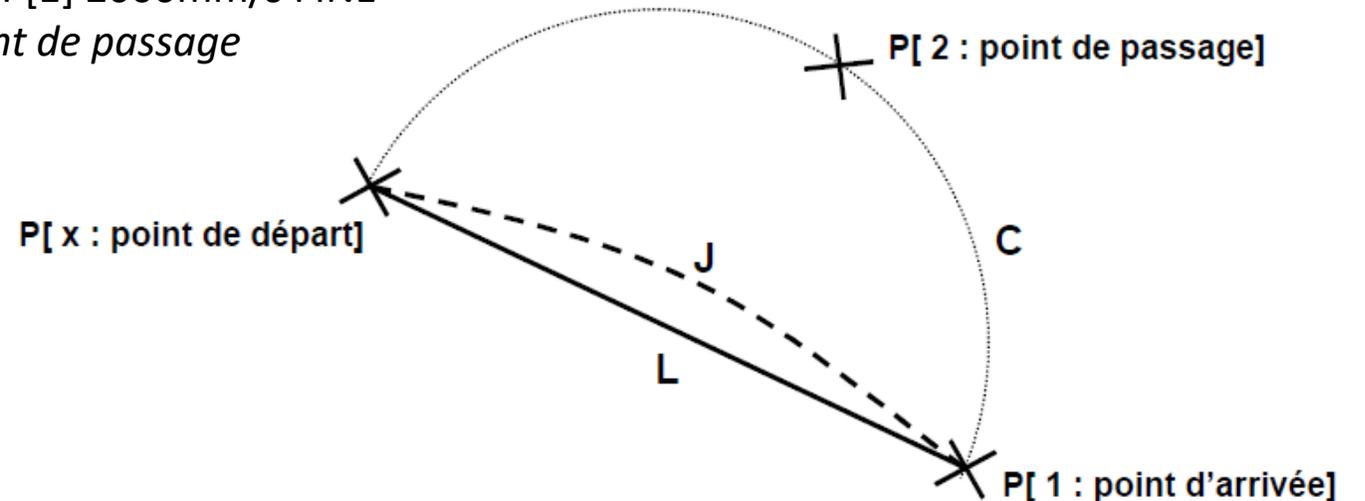


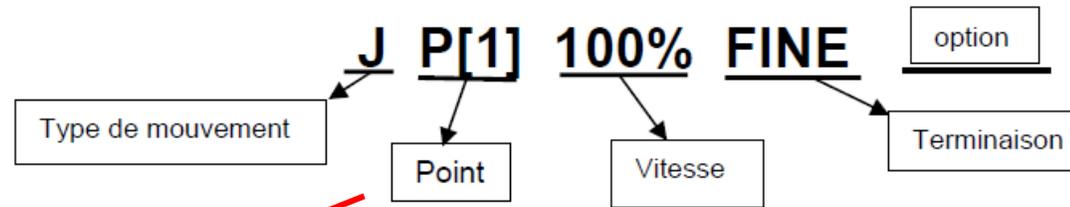
### Type de mouvement vers un point

- J (Joint) : mouvement angulaire → J P[1] 100% FINE
- L (Linear) : mouvement linéaire → L P[1] 2000mm/s FINE
- C (Circular) : mouvement circulaire → C P[2] P[1] 2000mm/s FINE  
où P[1] est le point d'arrivée et P[2] un point de passage

```

FANUC          LINE 0
Default Motion JOINT 10 %
                1/4
1:J P[] 100% FINE
2:J P[] 100% CNT100
3:L P[] 100mm/sec FINE
4:L P[] 100mm/sec CNT100
    
```





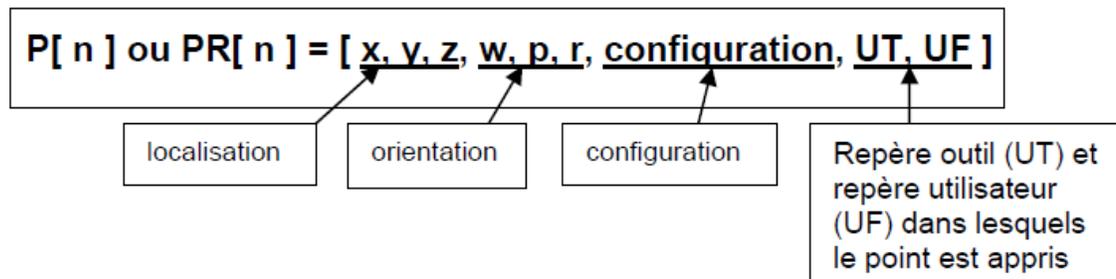
### Type de point

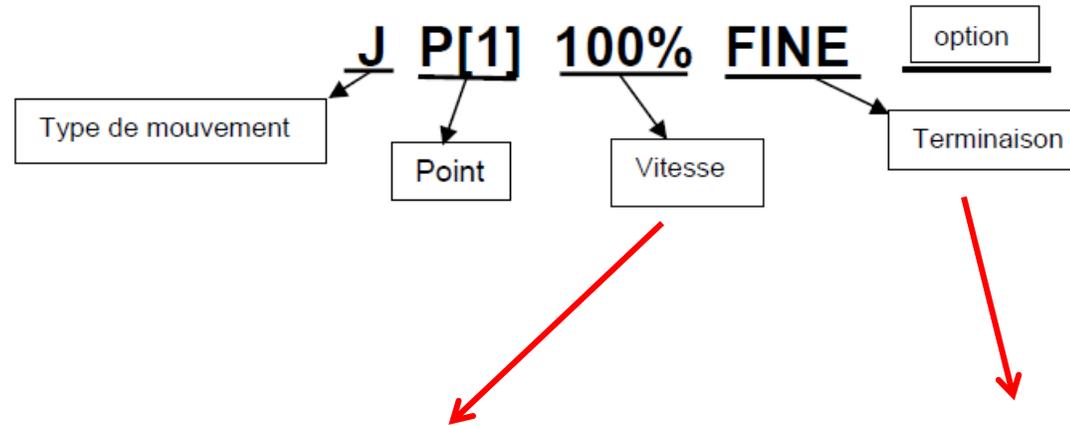
Il existe deux types de point :

Les POSITIONS → **P[ n ]** : position du repère outil par rapport à un repère utilisateur

Les REGISTRES DE POSITION → **PR[ n ]** : position qui dépend du repère utilisateur et du repère outil sélectionné

Leur format est le suivant :





La **vitesse** peut être exprimée de plusieurs façons selon le type de déplacement choisi :

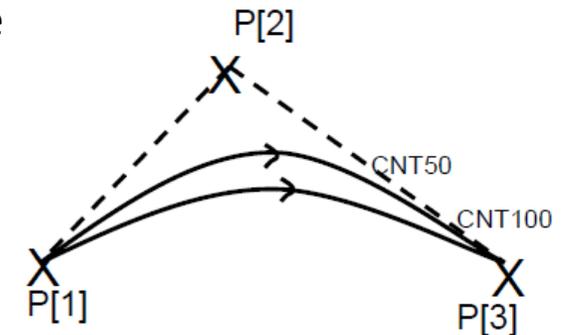
- JOINT : **valeur en %** de la vitesse maximale
- valeur en **secondes** (un temps de parcours est imposé)

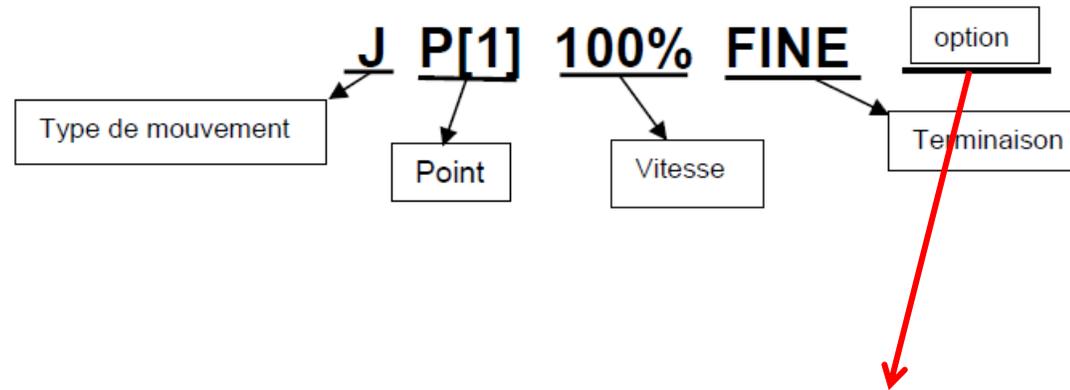
L ou C :

- valeur en **mm/s**
- valeur en **cm/min**
- valeur en **secondes** et **millisecondes** (un temps de parcours est imposé)
- valeur en **degré/seconde**

La **terminaison** définit comment le robot termine le mouvement.

- Terminaison fine (**FINE**) : arrêt du robot avec une précision maximale en position
- Terminaison continue (**CNT**) : pas d'arrêt sur le point programmé





**WJNT : Instruction de mouvement poignet** - l'attitude du poignet change durant le mouvement

**ACC : Réglage de l'accélération** - spécifie le pourcentage du taux d'accélération/décélération durant le mouvement

**SKIP,LBL[n] : Instruction de saut** - permet un saut vers un label spécifié si la condition n'est pas satisfaite

**OFFSET : Instruction OFFSET** - modifie l'information de position programmée par un décalage spécifié dans un registre

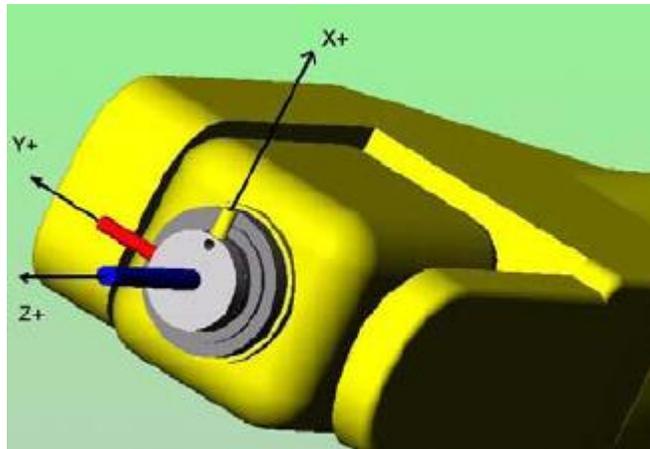
**TOOL OFFSET PR[n]: Instruction condition de décalage outil direct** - le robot bouge conformément au décalage stocké

**INC : Instruction d'incrémentatation** - Le robot bouge à la position de destination plus la valeur d'incrémentatation

**PTH : Instruction trajectoire** - améliorer la performance d'un mouvement continu

### Définition du repère outil

Lorsqu'un point est enregistré dans un système de coordonnées cartésien, les coordonnées relevées sont celles du repère outil par rapport au repère utilisateur choisi (WORLD par défaut).



Le **TCP** (Tool Center Point) est **l'origine du repère outil**.

Par défaut, le TCP se trouve au centre du flasque, sur le dernier axe du robot  
Le **repère outil par défaut** est orienté comme décrit sur la figure ci-contre  
Lors de la **création** d'un repère outil, le TCP est déplacé à l'extrémité de l'outil utilisé et le repère outil peut être réorienté suivant l'axe d'attaque du nouvel outil. Nous distinguons deux types d'outil :

- Outil simple
- Outil complexe

Il est possible de définir 9 repères outils en RJ3 et RJ3i. Trois **méthodes d'apprentissages** sont proposées :

- Méthode d'entrée directe des valeurs
- méthode des 3 points
- méthode des 6 points

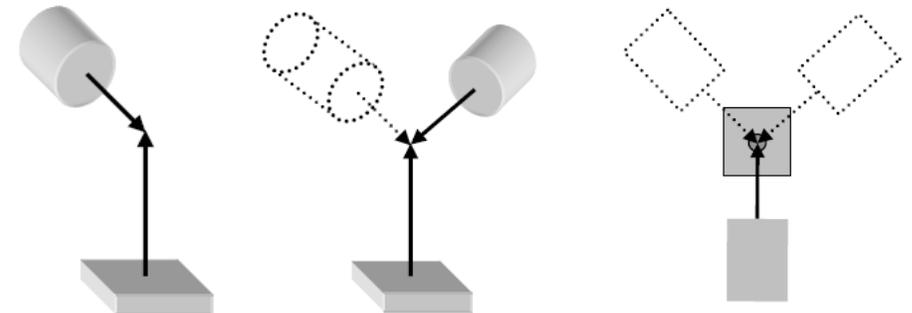
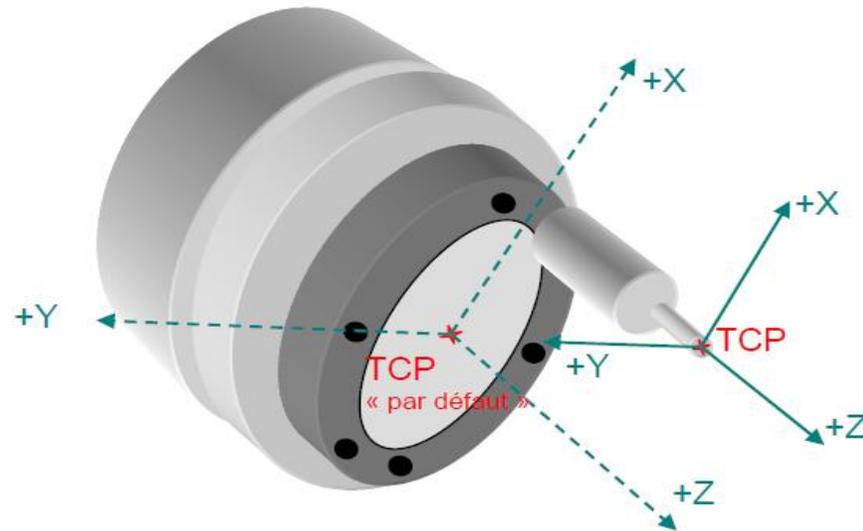
```

SETUP Frames JOINT 10 %
Tool Frame Setup/ Direct Entry 1/7
Frame Number: 1
1 Comment: *****
2 X: 0.000
3 Y: 0.000
4 Z: 0.000
5 W: 0.000
6 P: 0.000
7 R: 0.000
Configuration: N D B, 0, 0, 0

Active TOOL $MNUTOOLNUM[1] = 1
[ TYPE ] [METHOD] FRAME
    
```

**Méthode d'entrée directe des valeurs :** Dans cette méthode, les coordonnées et orientations de l'outil à définir dans le repère par défaut sont parfaitement connues.

**Méthode des 3 points :** Le but de cette méthode est de déplacer le TCP à l'extrémité de l'outil utilisé. Pour cela, nous allons pointer un même point avec trois orientations différentes et enregistrer ses positions.



```

SETUP Frames JOINT 10 %
Tool Frame Setup/ Three Point 4/4
Frame Number: 1
X: 0.0 Y: 0.0 Z: 0.0
W: 0.0 P: 0.0 R: 0.0
Comment:*****

Approach point 1: RECORDED
Approach point 2: RECORDED
Approach point 3: UNINIT

Active TOOL $MNUTOOLNUM[1] = 1
[ TYPE ] [METHOD] FRAME MOVE_TO RECORD
    
```

## Méthode des 6 points :

Le but de cette méthode est de déplacer le TCP à l'extrémité de l'outil utilisé, et de réorienter le repère dans l'axe d'attaque de ce nouvel outil. Les trois premières étapes sont **identiques** aux trois étapes de la méthode des trois points. Le TCP est défini et nous allons maintenant réorienter le repère en enregistrant trois points supplémentaires

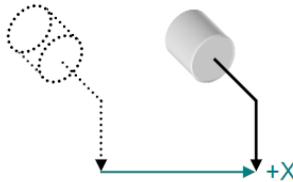
SETUP Frames	JOINT 10 %
Tool Frame Setup/ Six Point 5/7	
Frame Number: 1	
X: 0.0 Y: 0.0 Z: 0.0	
W: 0.0 P: 0.0 R: 0.0	
Comment:*****	
Approach point 1:	RECORDED
Approach point 2:	RECORDED
Approach point 3:	RECORDED
<b>Orient Origin Point:</b>	UNINIT
X Direction Point:	UNINIT
Z Direction Point:	UNINIT



### Etape 4 : Orient Origin Point

Pour enregistrer le point d'origine de l'orientation, l'axe OZ de l'outil doit être placé verticalement.

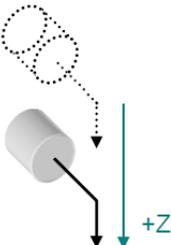
SETUP Frames	JOINT 10 %
Tool Frame Setup/ Six Point 6/7	
Frame Number: 1	
X: 0.0 Y: 0.0 Z: 0.0	
W: 0.0 P: 0.0 R: 0.0	
Comment:*****	
Approach point 1:	RECORDED
Approach point 2:	RECORDED
Approach point 3:	RECORDED
Orient Origin Point:	RECORDED
<b>X Direction Point:</b>	UNINIT
Z Direction Point:	UNINIT



### Etape 5 : X Direction Point

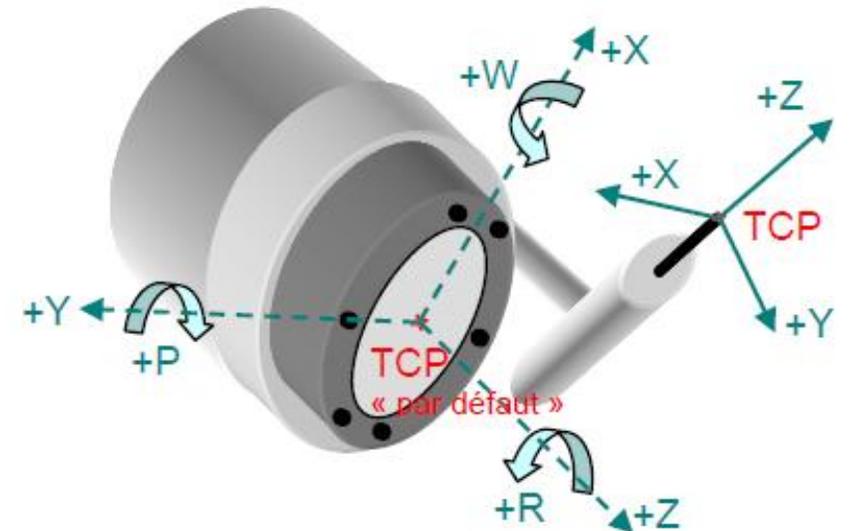
Définissons maintenant la direction et le sens de l'axe X

SETUP Frames	JOINT 10 %
Tool Frame Setup/ Six Point 7/7	
Frame Number: 1	
X: 0.0 Y: 0.0 Z: 0.0	
W: 0.0 P: 0.0 R: 0.0	
Comment:*****	
Approach point 1:	RECORDED
Approach point 2:	RECORDED
Approach point 3:	RECORDED
Orient Origin Point:	RECORDED
X Direction Point:	RECORDED
<b>Z Direction Point:</b>	UNINIT



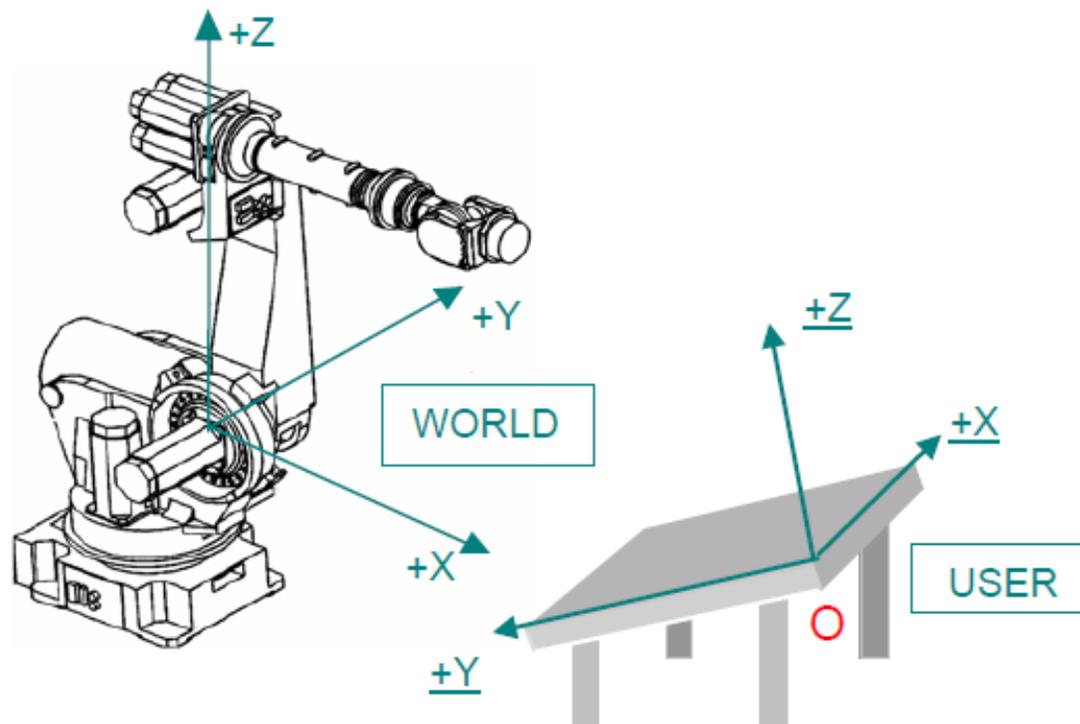
### Etape 6 : Z Direction Point

Pour donner la direction en Z, il faut au préalable se repositionner sur le point d'origine de l'orientation



### Le référentiel utilisateur :

Le repère utilisateur est le **repère de référence** pour toutes les **positions enregistrées** dans un programme. Le **TCP** évolue dans ce repère. Si aucun repère utilisateur n'est défini, c'est le **WORLD** qui sert de référentiel.



Il est possible de définir 9 repères utilisateurs en R-J3 et R-J3i.

Trois **méthodes d'apprentissages** sont proposées :

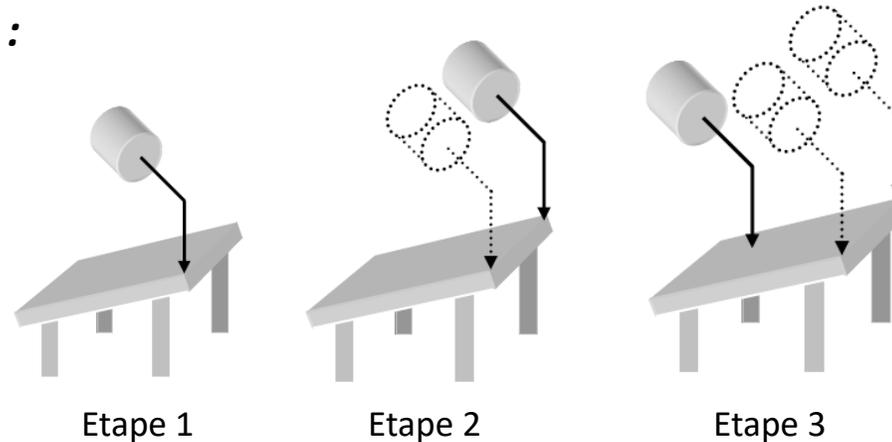
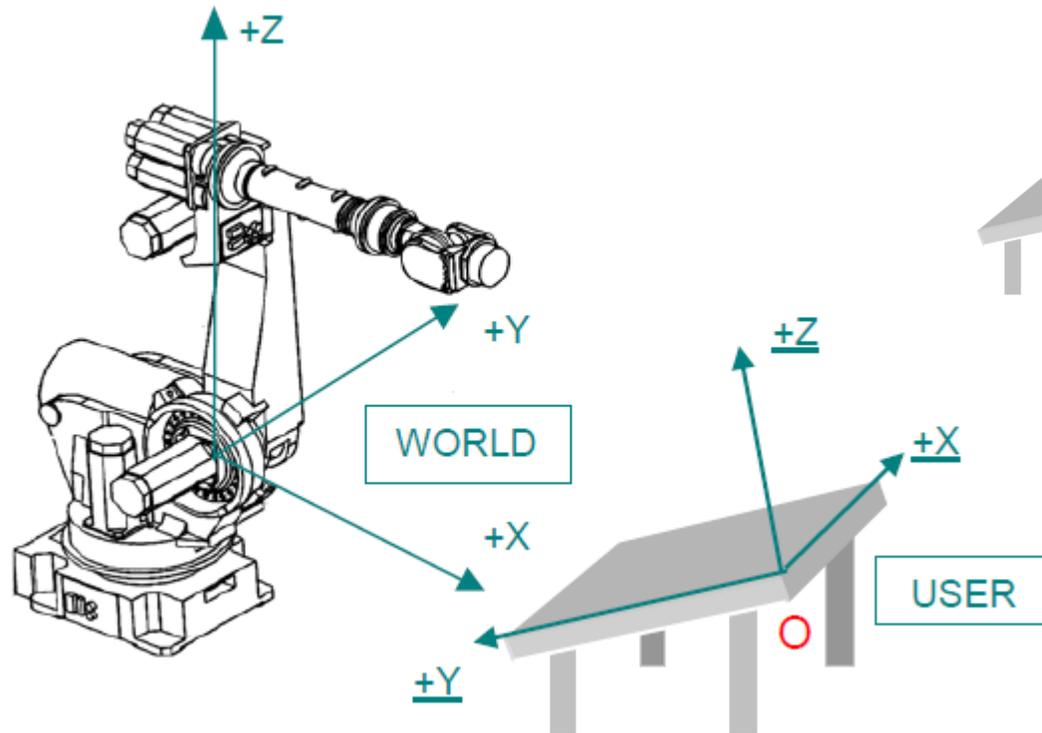
- Méthode d'entrée directe des valeurs
- méthode des 3 points
- méthode des 4 points

```

SETUP Frames JOINT 10 %
User Frame Setup/ Direct Entry 1/7
Frame Number: 5
1 Comment: *****
2 X: 0.000
3 Y: 0.000
4 Z: 0.000
5 W: 0.000
6 P: 0.000
7 R: 0.000
Configuration: N D B, 0, 0, 0
    
```

**Méthode d'entrée directe des valeurs :** Dans cette méthode, les coordonnées et orientations du repère utilisateur dans le repère WORLD sont parfaitement connues.

**Méthode des 3 points :**



Le point enregistré est un point du plan (+X,O,+Y).

- Etape 1 : Orient Origine Point** - enregistrons l'origine du repère
- Etape 2 : X Direction Point** - indiquons ensuite au robot la direction et le sens de l'axe X
- Etape 3 : Y Direction Point** - Indiquons la direction et le sens de l'axe Y et de l'axe Z.

### Méthode des 4 points :

Cette méthode est utilisée lorsque l'endroit désiré pour la définition du repère n'est pas accessible ou peu commode. Par exemple, pour définir l'origine d'un repère au centre d'une table, il est plus aisé de le définir sur les bords et de le déplacer ensuite au centre.

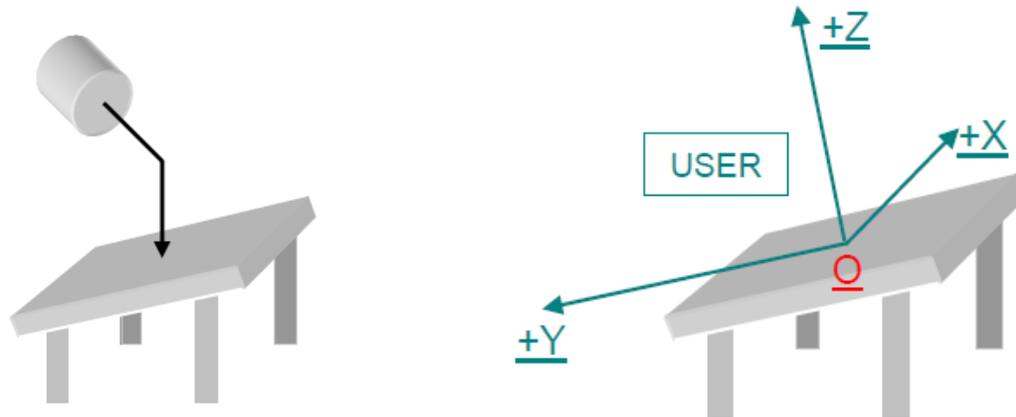
#### Etape 1, 2, 3 :

Les trois premières étapes sont identiques aux trois étapes de la méthode des trois points

#### Etape 4 :

Pour placer l'origine du repère ainsi défini, positionner le robot sur le point désiré

*Ces coordonnées sont données dans le WORLD.*



SETUP Frames	JOINT	10 %
User Frame Setup/ Four Point		1/5
Frame Number: 1		
X: 933.6	Y: 309.4	Z: 1035.1
W: -0.9	P: 0.5	R: 89.9
Comment: *****		
Orient Origin Point:		USED
X Direction Point:		USED
Y Direction Point:		USED
System Origin:		USED

Analyste Programmeur en Automatismes, Robotique et  
Informatique Industrielle  
TS ARII

**Module MF 1.2**

***Analyser un traitement numérique***

***Appliquer les systèmes de codage à la Robotique***

**Fin de Présentation**