



PROGRAMMATION DES ROBOTS FANUC ET DU COBOT ABB



I2S et 2I

3^{ème} Année

Philippe DUGRAIN
Nicolas BONZANI
Yves LORDONNÉ

Sommaire

I - CARACTERISATION DES ROBOTS	3
1.1 La charge maximale transportable.....	3
1.2 L'architecture du S.M.A.....	4
1-2-1 Anthropomorphe (RRR).....	4
1-2-2 Torique (RPR) ou Scara.....	4
1.3 Le volume de travail.....	4
1.4 Le positionnement absolu et la répétabilité.....	5
1.5 Autres paramètres :.....	5
II - DESCRIPTION DU SYSTEME	6
2-1 Unité mécanique A-600.....	6
2-2 LRMate 200 iD.....	7
2-3 Unité mécanique du robot ABB IRB 14000 –0,5/0,5 (YUMI).....	8
2-4 Capteurs de déplacements.....	9
2-5 Interrupteurs - Fin de course.....	9
2-6 Baie de commande KAREL (LR Mate 200iD).....	9
2-7 Boitier d'apprentissage robot ABB IRB14000 (YUMI).....	11
III - LANGAGES DE PROGRAMMATION (LR Mate 200iD)	11
3-1 Déplacements.....	11
3-1-1 Positions.....	11
3-1-2 Repères de référence.....	14
3-1-2-1 USER FRAME.....	14
3-1-2-2 Exemple d'application USER FRAME.....	14
3-1-2-3 TOOL FRAME.....	14
3-1-3 Systèmes de coordonnées de déplacement.....	16
3-1-4 Contrôle du déplacement.....	16
3-1-4-1 Trajectoire.....	17
3-1-4-2 Vitesse de déplacement.....	19
3-1-4-3 Terminaison du déplacement.....	19
3-2 Système de gestion des fichiers.....	20
3-3 Variables système.....	21
IV - LANGAGE TPE	21
4.1 Instructions de boucle.....	21
4.2 Instructions conditionnelles.....	22
4.4 Sous-programmes.....	22
4.4 Divers.....	22

I - CARACTERISATION DES ROBOTS

Principaux paramètres à prendre en compte dans le choix d'un robot :

1.1 La charge maximale transportable (de quelques kg à 2 tonnes)

Fonction de l'élongation et de la position du centre de gravité de la charge/ poignet.

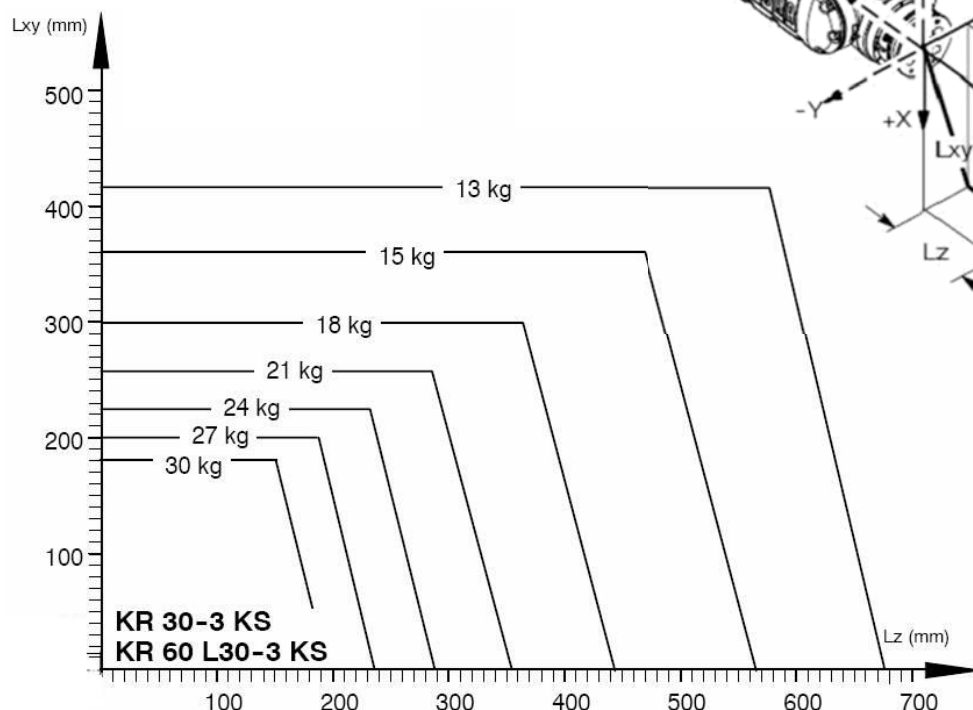
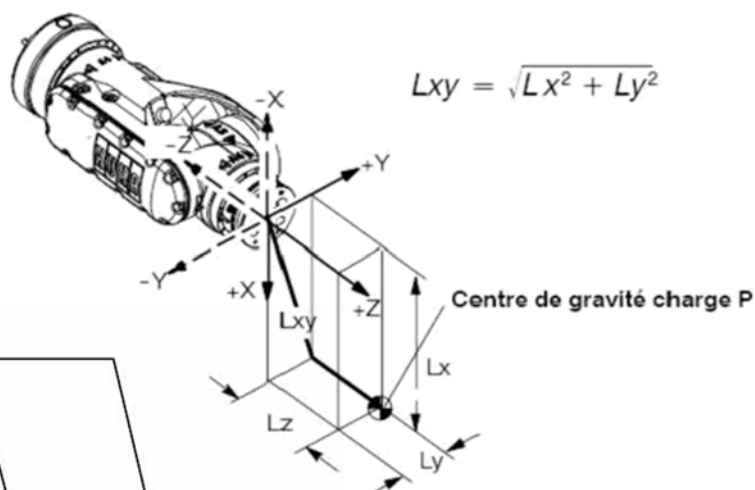
Exemple de charge maximum pour un robot KUKA KR30-3 KS



Nombre des axes	6
Répétabilité	< ±0,15 mm
Poids	aprox. 600 kg
Positions de montage	Sol, console

Charge admissible 30 kg

Portée max. 2233 mm



Nota : Sur le LRMate 200, le Payload setting permet d'optimiser l'asservissement en définissant la masse embarquée, la position de son centre de gravité et les moments d'inertie.

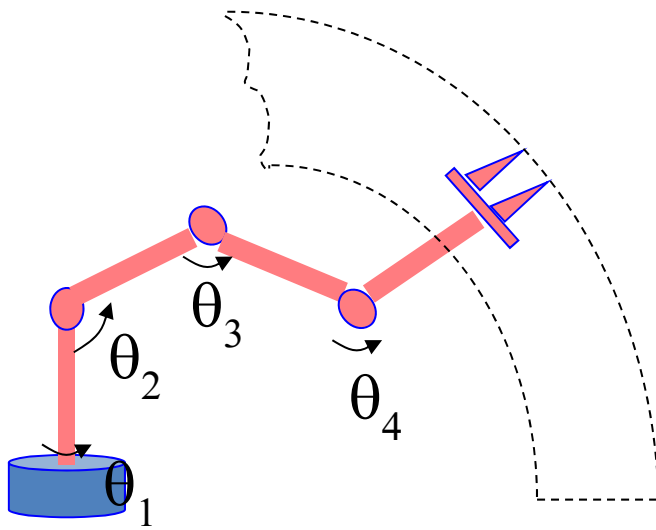
Actuellement les valeurs suivantes sont définies :

Payload [1] : Poignet équipé du système anticollision et du changeur d'outil à vide.

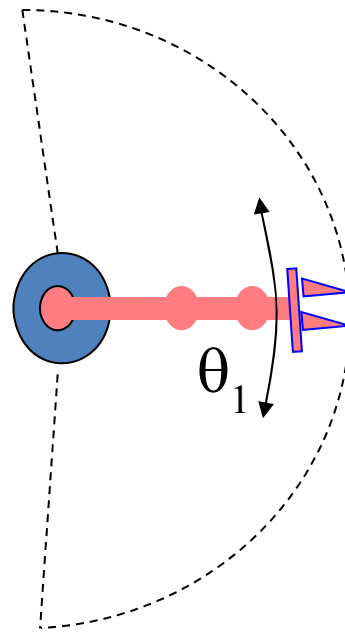
Payload [2] : Poignet équipé du système anticollision et du changeur d'outil monté avec la pince.

1.2 L'architecture du S.M.A.

1-2-1 Anthropomorphe (RRR) 4m/s

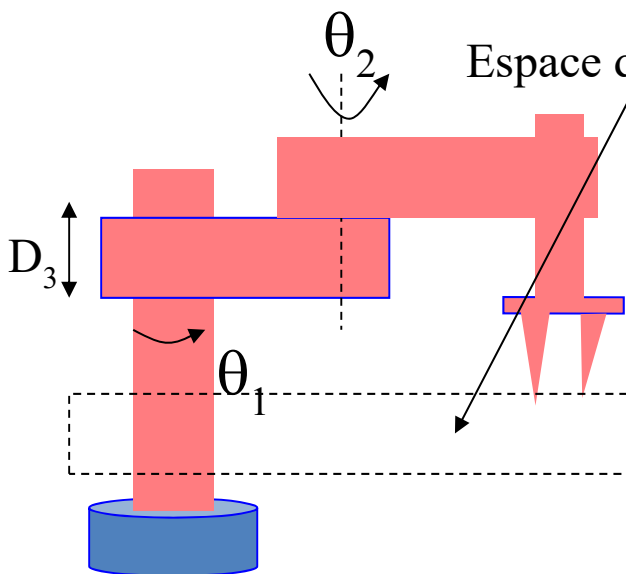


Vue latérale



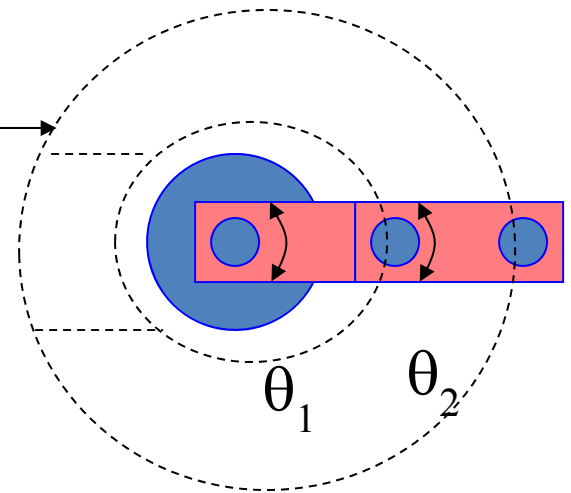
Vue dessus

1-2-2 Torique (RPR) ou Scara 9m/s



Vue latérale

Espace de travail



Vue dessus

1.3 Le volume de travail

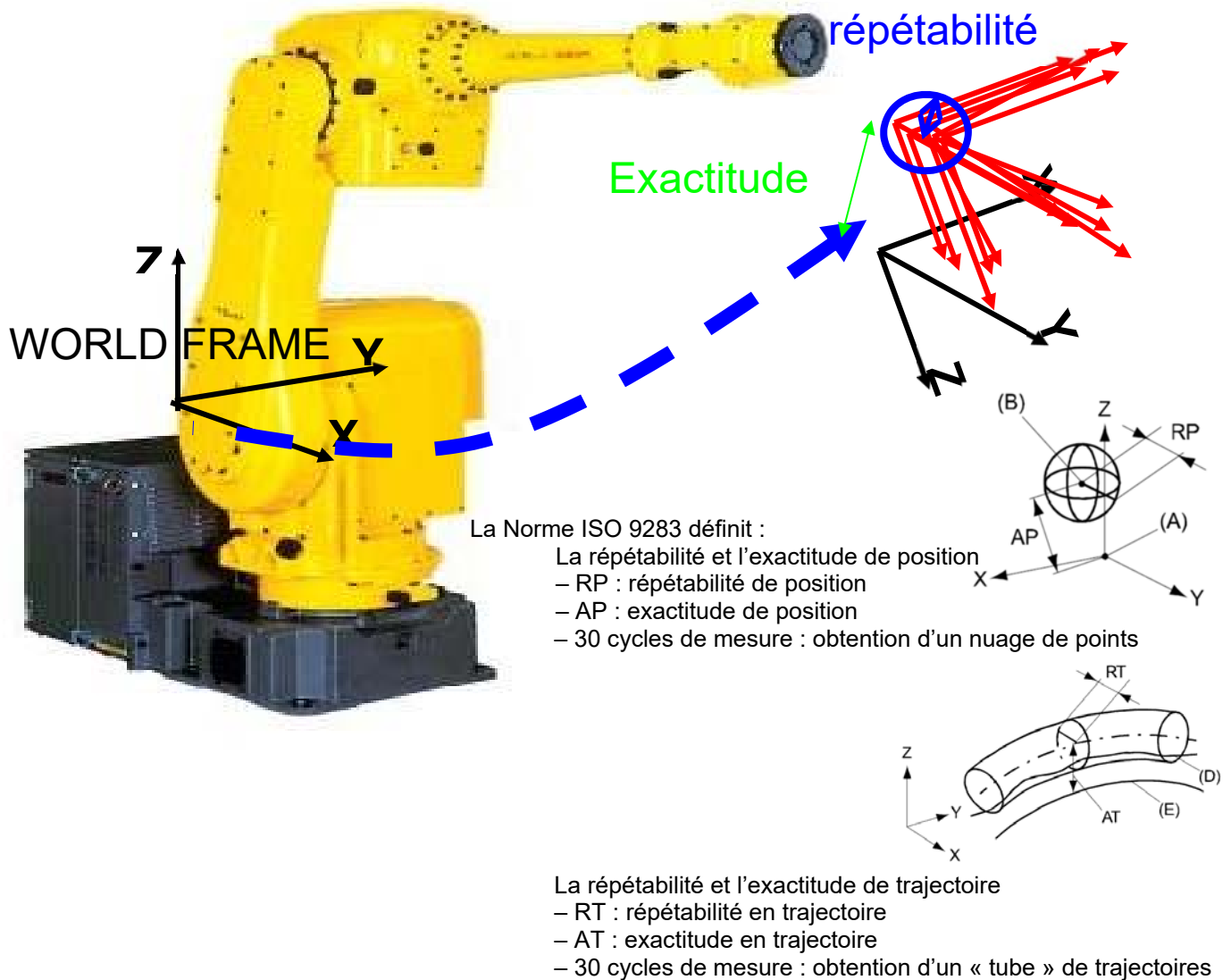
+ Paralleles : 12 m/s

Ensemble des points atteignables par l'organe terminal :

Espace de travail maxi (reachable workspace) : au moins une orientation.

Espace de travail dextre (dextrous workspace) : atteignable avec toutes les orientations possibles de l'effecteur (organe terminal). Sous-ensemble de l'espace de travail maximal et nettement plus réduit par rapport à celui-ci.

1.4 Le positionnement absolu et la répétabilité



Source des erreurs :

- ❑ Positionnement absolu imprécis (≈ 1 mm)
 - erreurs de modèle géométrique
 - erreurs de quantification de la mesure de position
 - flexibilités
 - dilatation
- ❑ Répétabilité : *erreur maximale de positionnement répété de l'outil en tout point de son espace de travail.*
En général répétabilité < 0.1 mm
 - erreurs de quantification de la mesure de position
 - flexibilités
 - dilatation

1.5 Autres paramètres :

- La vitesse de déplacement (vitesse maximum en élongation maximum), accélération.
- La masse du robot.
- Le coût du robot.
- La maintenance
- La sécurité : voir les normes NF EN ISO 102018-1, NF EN ISO 102018-2 et NF EN ISO 13849-1 disponibles sur l'ENT

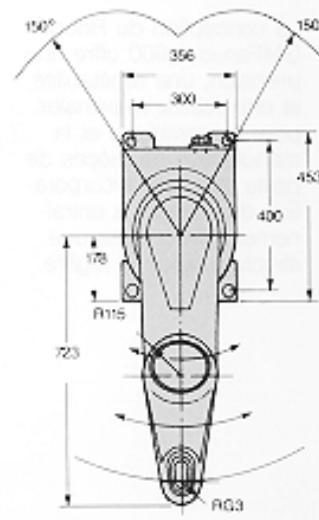
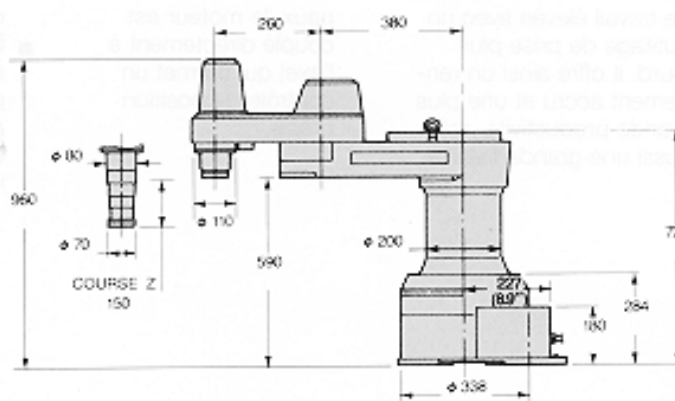
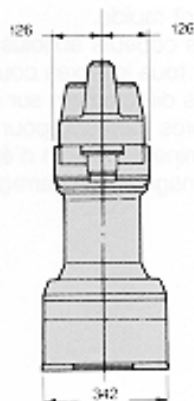
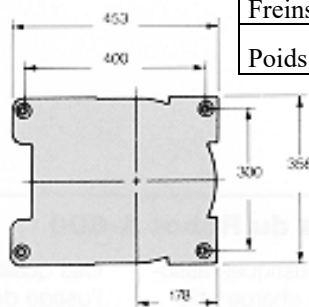
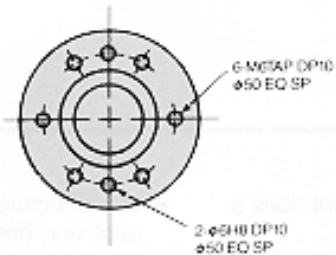
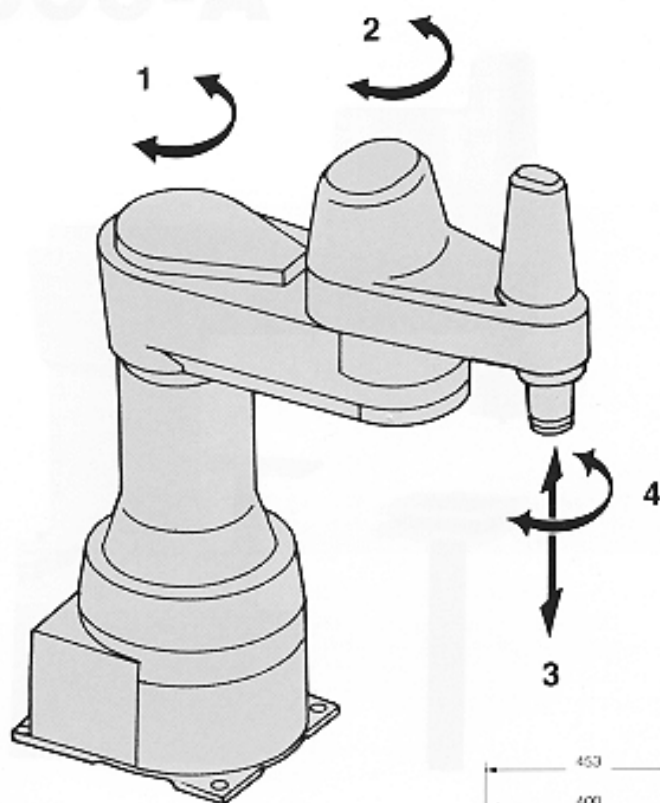
II - DESCRIPTION DU SYSTEME

2-1 Unité mécanique A-600

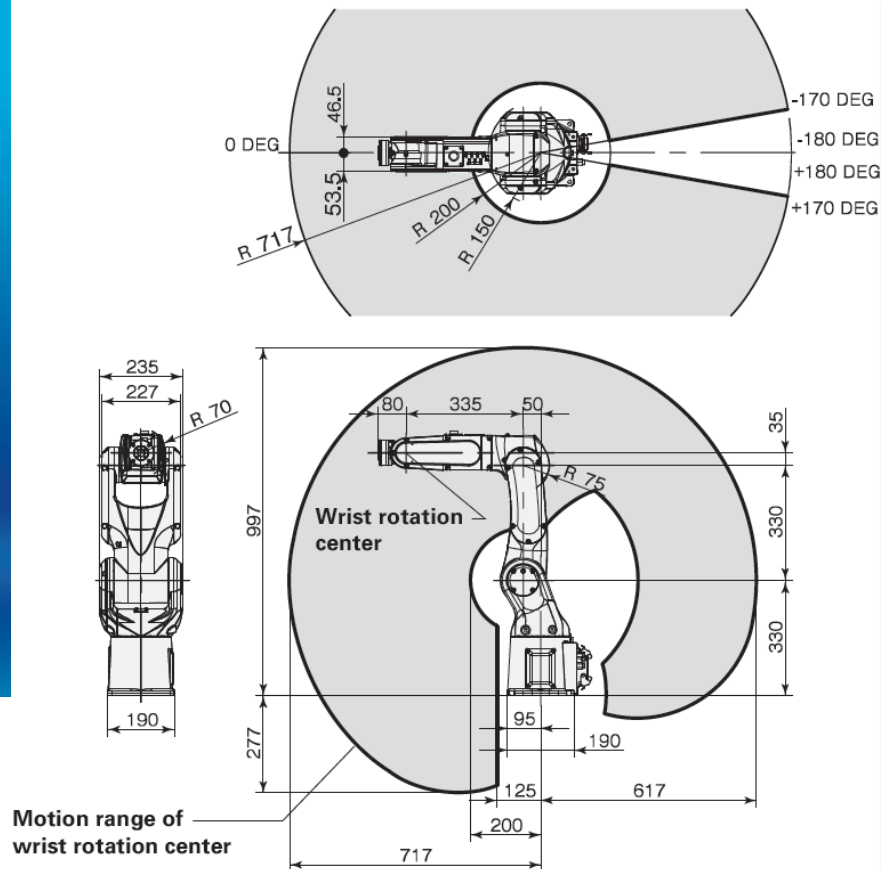
L'unité mécanique A-600 est un robot industriel de type SCARA à 4 axes commandés par des servomoteurs

Performances

Items		Spécification A-600
Axes contrôlés		4
Portée et vitesse	axe 1	300° - 720°/sec
	axe 2	270° - 720°/sec
	axe 3	150 mm 650 mm/sec
	axe 4	1080° 1080°/sec
Charge utile au poignet		6 kg
Répétabilité		± 0,013 mm
Montage		Au sol
Entrainement	axe 1	Direct
	axe 2	Direct
	axe 3	AC servo
	axe 4	AC servo
Détection de position		Codeur absolu
Freins mécaniques		Axes Theta et Z
Poids de l'unité mécanique		120 kg



2-2 LRMate 200 iD



Model		LR Mate 200iD
Controlled axes		6 axes (J1, J2, J3, J4, J5, J6)
Reach		717mm
Installation (Note 1)		Floor, Upside-down, Angle mount
Motion range (Maximum speed)	J1 axis	340°/ 360° (option) (450°/s) 5.93 rad/6.28 rad (option) (7.85 rad/s)
	J2 axis	245° (380°/s) 4.28 rad (6.63rad/s)
	J3 axis	420° (520°/s) 7.33 rad (9.08rad/s)
	J4 axis	380° (550°/s) 6.63 rad (9.60 rad/s)
	J5 axis	250° (545°/s) 4.36 rad (9.51 rad/s)
	J6 axis	720° (1000°/s) 12.57 rad (17.45 rad/s)
Max. load capacity at wrist		7kg
Allowable load moment at wrist	J4 axis	16.6 N·m
	J5 axis	16.6 N·m
	J6 axis	9.4 N·m
Allowable load inertia at wrist	J4 axis	0.47 kg·m ²
	J5 axis	0.47 kg·m ²
	J6 axis	0.15 kg·m ²
Repeatability		± 0.02 mm
Mass (Note 2)		25 kg
Installation environment		Ambient temperature : 0~45°C Ambient humidity : Normally 75%RH or less (No dew nor frost allowed), Short term 95%RH or less (within one month) Vibration : 0.5G or less

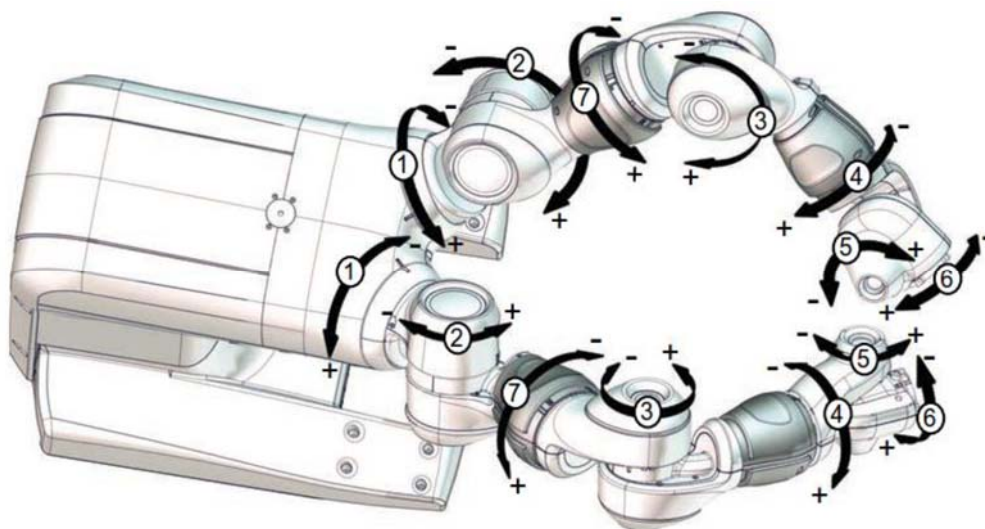
2-3 Unité mécanique du robot ABB IRB 14000 -0,5/0,5 (YUMI)

YUMI est un robot collaboratif à deux bras de 7 axes chacun destiné à l'assemblage de petites pièces.



	Plage de mouvement	Vitesse max.
Axe 1 (rotation)	+168,5° à -168,5°	180°/s
Axe 2 (bras)	+43,5° à -143,5°	180°/s
Axe 7 (rotation)	+168,5° à -168,5°	180°/s
Axe 3 (bras)	+80° à -123,5°	180°/s
Axe 4 (poignet)	+290° à -290°	400°/s
Axe 5 (flexion)	+138° à -88°	400°/s
Axe 6 (pivotement)	+229° à -229°	400°/s

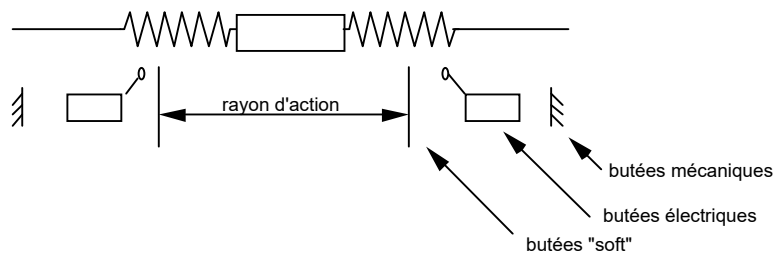
IRB 14000 -0,5/0,5	
Charge utile	0,5kg par bras
Rayon d'action	559mm
Précision	0,02mm
Dimensions	399x497mm
Interface client	Interface socle
Poids	38kg
Position de montage	Sur table
Température de fonctionnement	5°C -40°C
Degré de protection	IP30
Applications en salle blanche / alimentaires	Non



2-4 Capteurs de déplacements

- Contrôle des déplacements : se fait par des codeurs incrémentaux
 - Nécessité de conserver la position courante en permanence : sauvegarde par piles
 - Codeurs absolus

2-5 Interrupteurs - Fin de course

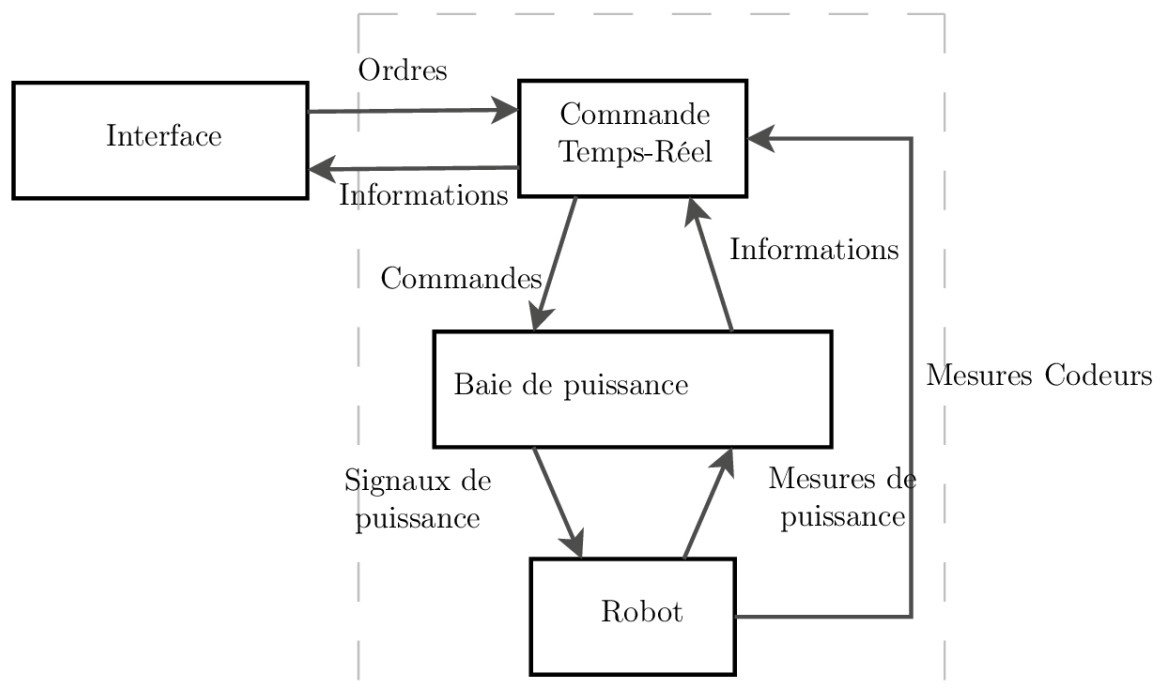


* les butées soft sont des butées logicielles [condition permanente : si pos. cour = pos. réf
→ on coupe tout]. En cas de "contact" avec la butée il suffit de dégager l'axe concerné

* Les butées électriques sont des capteurs [en cas de contact la procédure de dégagement est différente]

2-6 Baie de commande KAREL (LR Mate 200iD)

C'est un système informatisé de commande d'axe simultanée.

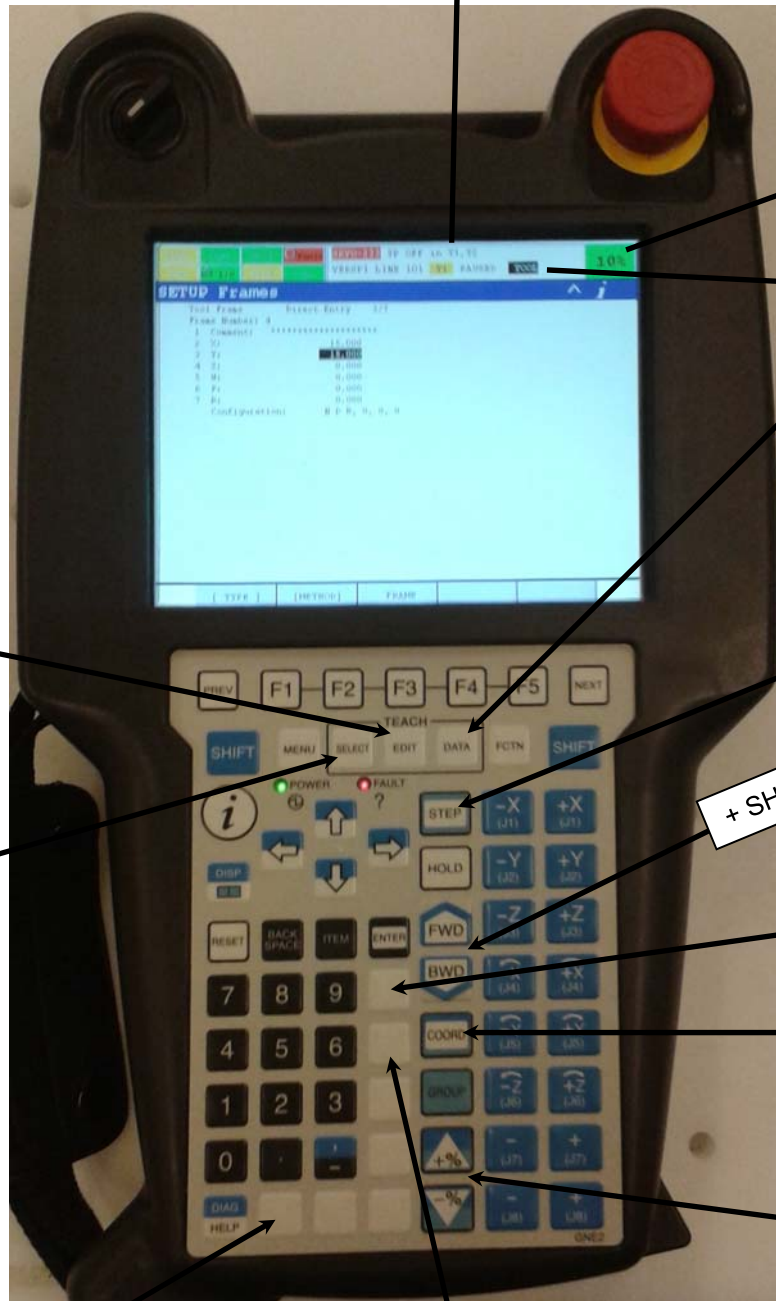


Le contrôleur :

- commande les déplacements
- permet la communication avec des périphériques

- * Interface opérateur : - Panneau opérateur
- Teach-Pendant. (Boîtier d'apprentissage)

Mode de marche : (T2 désactivé)
T1 : Apprentissage et test (250 mm/s) porte ouverte et TP autorisé
Auto : run programme vitesse max (départ bouton vert sur armoire)



EDIT : Pour programme TPE seulement

SELECT : Choix du programme dans MD (Memory device) :
.PC → Prog Karel
.MR → Macro
Sans extension : TPE

Valeur du \$GENOVERRIDE

Mode de déplacement en manuel

DATA : Visualisation des variables
R (Register) numériques
PR : positions

STEP : exécution du Programme pas à pas

Exécution du programme :
FWD : (Forward) Avancer
BWD : reculer

Gestion outil :
Ouverture (release) / fermeture (grab)

COORD :
Modification du mode de déplacement
+Shift : modif Utool

+% ou -% :
Modification de la valeur du \$Genoverride

Macros TPE :

- Prise crayon
- Pose crayon
- GOHOME (Position de départ des programmes en mode auto)
- Chargement outils :
 - o Ventouses
 - o Pince
 - o palpeur

(STATUS)

- Position courante
- UTOOL

2.7 Boîtier d'apprentissage robot ABB IRB14000 (YUMI)



III - LANGAGES DE PROGRAMMATION (LR Mate 200iD)

Le système FANUC propose 3 types de langages pour répondre à tous les niveaux de programmation :

- TPE : langage interprété, simple et intuitif.
- KAREL : langage évolué et compilé, semblable au Pascal, principalement pour les applications complexes (option présente sur les robots de l'AIP).
- PMC : « Programmable Machine Control » langage « Ladder » (optionnel).

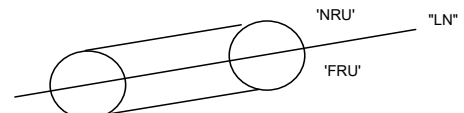
3-1 Déplacements

3-1-1 Positions

Les langages TPE et KAREL utilisent des données de type Position :

- mémorisée sous forme cartésienne :
 - 3 réels : X, Y, Z \Rightarrow localisation en mm
 - 3 réels : W, P, R \Rightarrow orientation en degrés
 - 1 chaîne de caractères : config \Rightarrow position/ligne neutre
- Mémorisée sous forme articulaire
 - 6 réels : J1, J2, J3, J4, J5, J6 \Rightarrow valeur angulaire des axes en degrés
 - 1 chaîne de caractères : 'NRU' ou 'L' ou 'NUT 0, 0, 0' \Rightarrow Configuration : position par rapport à la ligne neutre permettant de différencier les positions pour des robots redondants

Exemple : P [1] = (100, 800, 900, 0, 90, 0, 'FRU')



Angles d'Euler classiques (ISO, VAL +, ...)

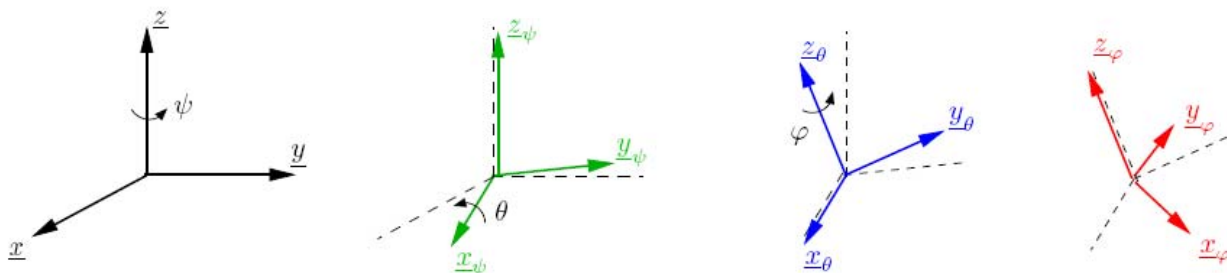
Chaque nouvelle rotation étant par rapport à un repère ayant tourné.

3 rotations successives :

$R(z, \psi)$ précession

$R(x, \theta)$ nutation

$R(z, \zeta)$ rotation propre



Angles d'Euler non classiques (FANUC, KUKA) :

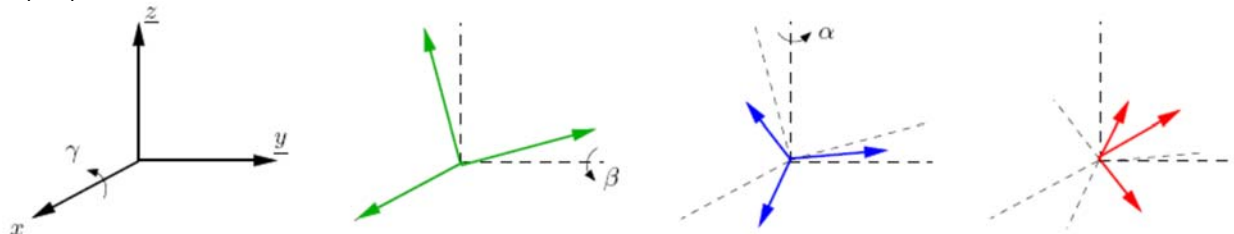
Chaque nouvelle rotation étant effectuée par rapport à un axe du repère fixe R :

Utilisés par les anglo-saxons (yaw, Pitch et Roll) :

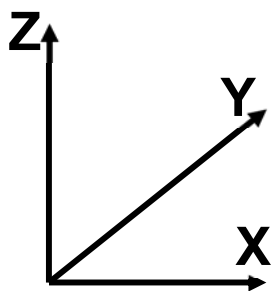
$R(x, \gamma)$ roulis

$R(y, \beta)$ tangage

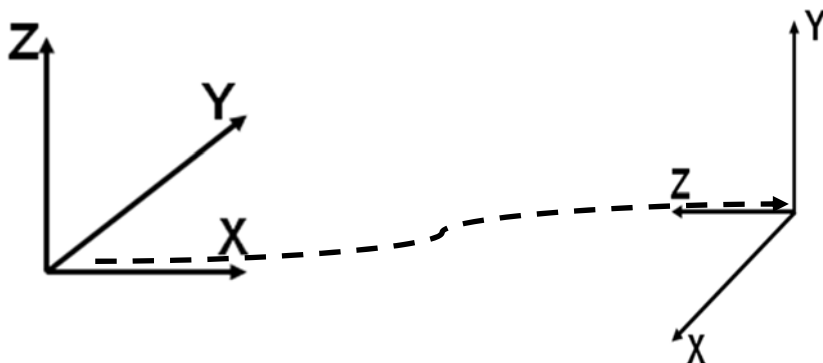
$R(z, \alpha)$ lacet.



Exemple : $P[1] = (\dots, \dots, -90, 0, -90, \text{'NRU'})$
W P R



Exemple : $P[1] = (\dots, \dots, \dots, \dots, \dots, \text{'NRU'})$
W P R



Exemple sur robot GMF A600 :
P1=pos(100,800,900, 0 , 90, 0 , 'L')

↑ elbow **L**eft ou **R**ight (axe du coude à gauche ou à droite)



Exemple sur robot GMF S10 :
P1=pos(100,800,900, 0 , 90, 0 , 'NRU')

↑↑ elbow **U**p ou **D**own (axe du coude en haut ou en bas)
Shoulder **L**eft ou **R**ight (Epaule droite ou gauche)
wrist **N**o-flip or **F**lip (poignet non plié ou plié)



Exemple sur robot FANUC LRMate 200 :
CNV_STR_CONF('NUT', NUT , status) – la chaîne de caractères 'NUT' est stockée dans la variable de type CONFIG nommée NUT
P [1] = (100, 800, 900, 0, 90, 0, 'NUT')

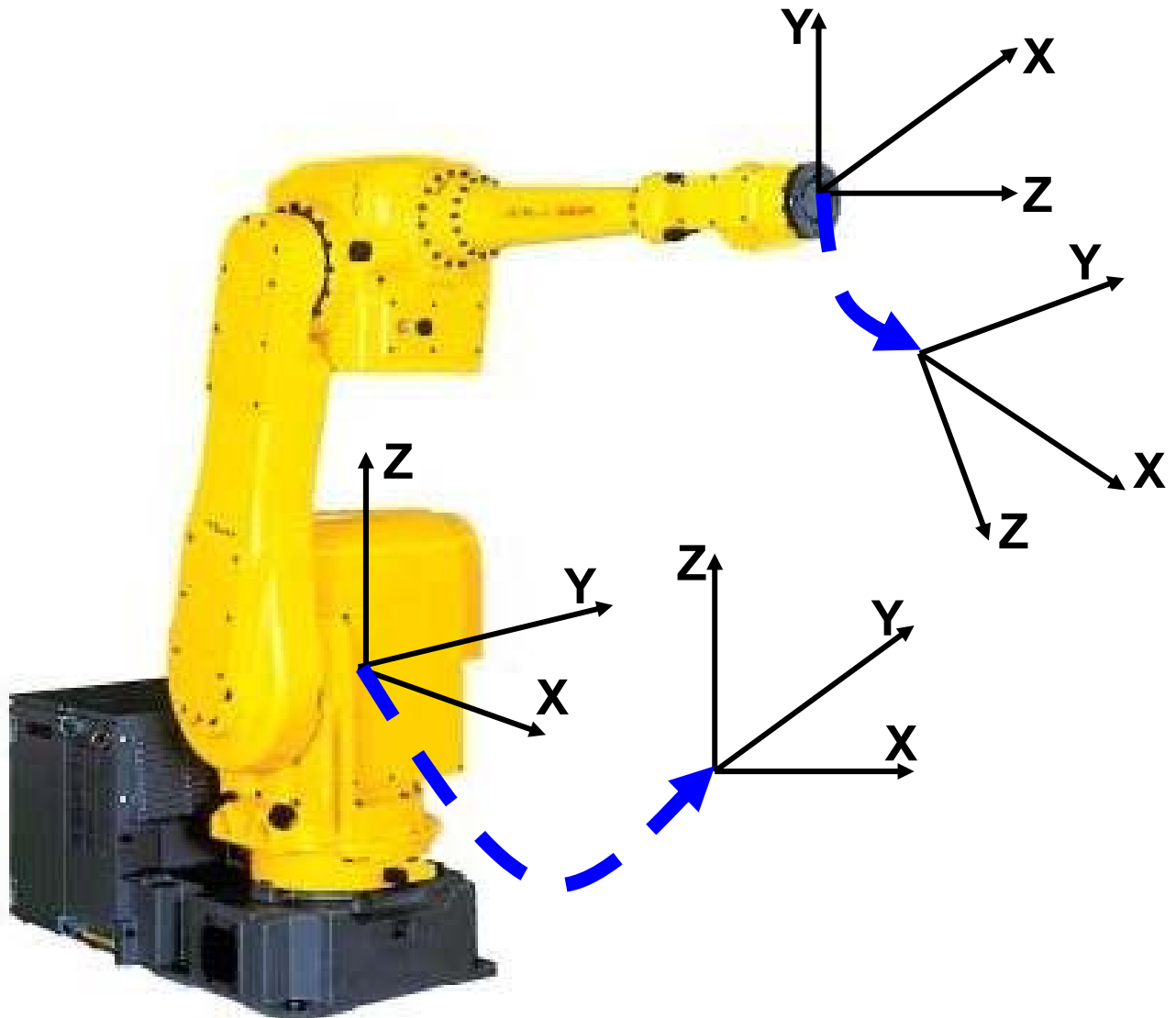
↑↑↑ fron**T** ou **B**ack (avant ou arrière)
elbow **U**p ou **D**own (axe du coude en haut ou en bas)
wrist **N**o-flip or **F**lip (poignet non plié ou plié)



Le Karel du LRMate permet également d'utiliser des variables de type XYZWPR et JOINTPOS

3-1-2 Repères de référence

Le système TPE définit la localisation et l'orientation des données de position par rapport à un repère de référence.



Il existe trois repères :

- WORLDFRAME : prédéfini, axe Z vertical vers le haut passant par axe 1, situé à la base du 2ème axe
- USER FRAME : déterminé par l'utilisateur défini dans le repère WORLD en attribuant une valeur type position à l'un des USER [0 à 9]
- TOOLFRAME : c'est le centre outil, défini par la variable système TOOL [1 à 9]

3-1-2-1 USER FRAME

- USER FRAME [0] Correspond à WORLDFRAME, non éditable
- On peut modifier les USER FRAME [1] à [9] :

- Par entrée directe :

USER FRAME [5] = POS (100, 200, 800, 0, 0, 45)

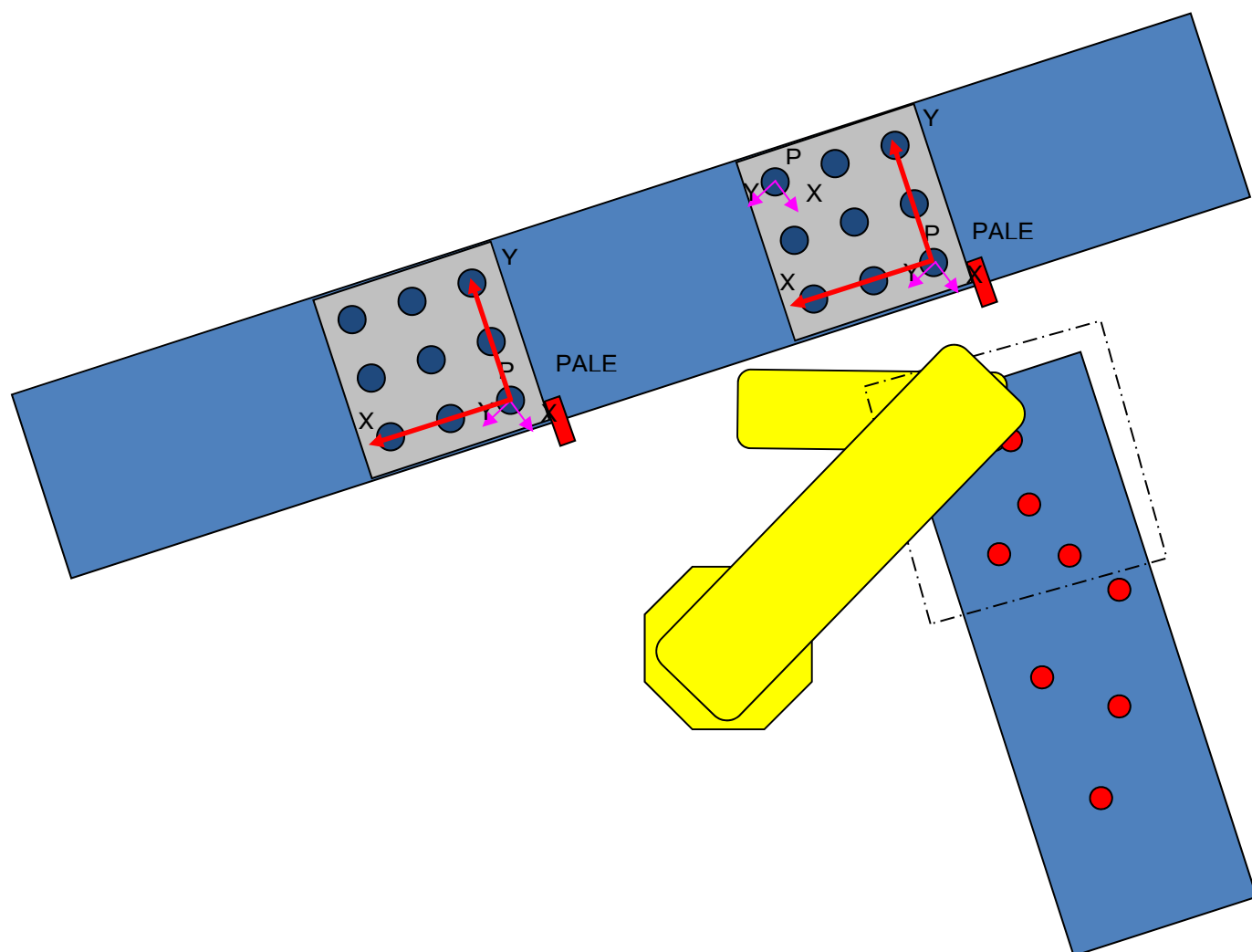
- Par la méthode des 3 points correspondant à :

UFRAME = FRAME (P0, PX, PY) en Karel

- Par la méthode des 4 points correspondant à :

UFRAME = FRAME (P0, PX, PY, P1) en Karel

3-1-2-2 Exemple d'application d'un USER FRAME



Nota : En plus de l'entrée directe, deux méthodes d'apprentissage du USER FRAME (utilisant la fonction FRAME) sont disponibles sur le robot LRMATE : MENU / SETUP / FRAME puis OTHER / FRAME et DETAIL / METHOD, choisir 3 points (P0, PX, PY) ou 4 points (P0, PX, PY, P)

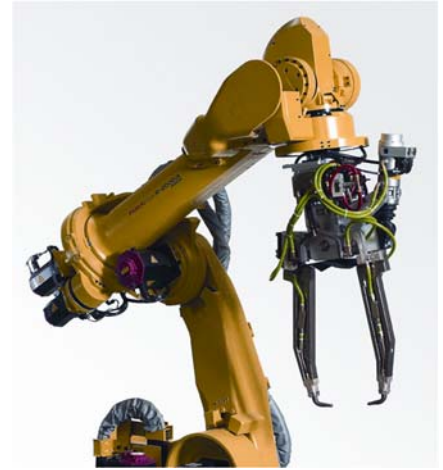
3-1-2-3 TOOL FRAME

En programmation TPE :

- Les TOOL FRAME de [1 à 10] sont modifiables
- La fonction *CLEAR TOOL* [1] lui renvoie la position TOOL [1] = (0, 0, 0, 0, 0, 0, 'NRU') qui correspond au FACEPLATE du robot
- On peut modifier les USER TOOL par 6 méthodes différentes (MENU / SETUP / F1 [TYPE] / FRAMES / TOOL FRAME

SETUP : Sélectionner le centre-outil puis faire F2 [DETAIL]) :

- 3 points : apprentissage de la localisation (bien éloigner les orientations des 3 points)
- 6 points XZ : apprentissage de la localisation et de l'orientation (3 premiers points idem) puis décaler la bille de 200 mm, palper le point 4 axe Z utool vertical, point 5 suivant X utool, revenir au point 4 par MOVE_TO, descendre suivant Z Utool pour le point 6)
- 6 points XY
- 2 points + Z
- 4 points
- DIRECT ENTRY : saisie des valeurs du TOOL FRAME [n]



3-1-3 Systèmes de coordonnées de déplacement

Le système TPE offre 4 systèmes de coordonnées de déplacement manuel :

- JOINT :
 - système de coordonnées articulaire, on déplace axe par axe
 - déplacement par interpolation des articulations
- WORLD :
 - système de coordonnées cartésienne, le centre outil se déplace parallèlement ou autour des axes X, Y, Z, du WORLD FRAME
 - déplacement par interpolation linéaire
- TOOLFRAME :
 - système de coordonnées cartésienne, le centre outil se déplace par rapport par rapport au TOOLFRAME sélectionné, de [1 à 10]
 - déplacement par interpolation linéaire
- USER :
 - système de coordonnées cartésiennes, le centre outil se déplace parallèlement ou autour des axes du USER FRAME sélectionné, de [0 à 9]
 - déplacement par interpolation linéaire

3-1-4 Contrôle du déplacement

Les caractéristiques du déplacement sont :

- le type de trajectoire
- la vitesse
- la terminaison (ou précision)
- l'accélération/décélération

3-1-4-1 Trajectoire

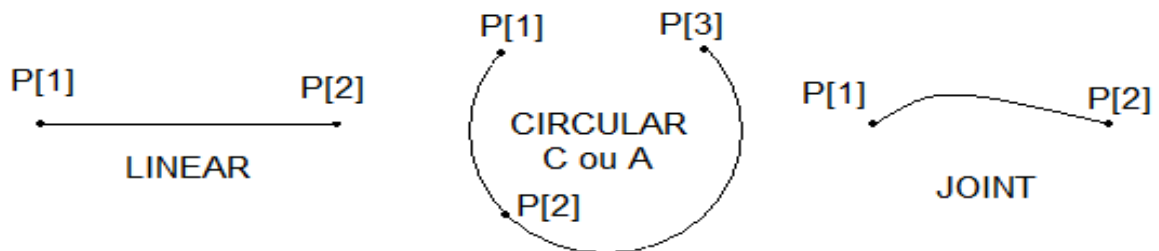
Les deux méthodes d'interpolation de base sont :
- articulaire
- cartésienne

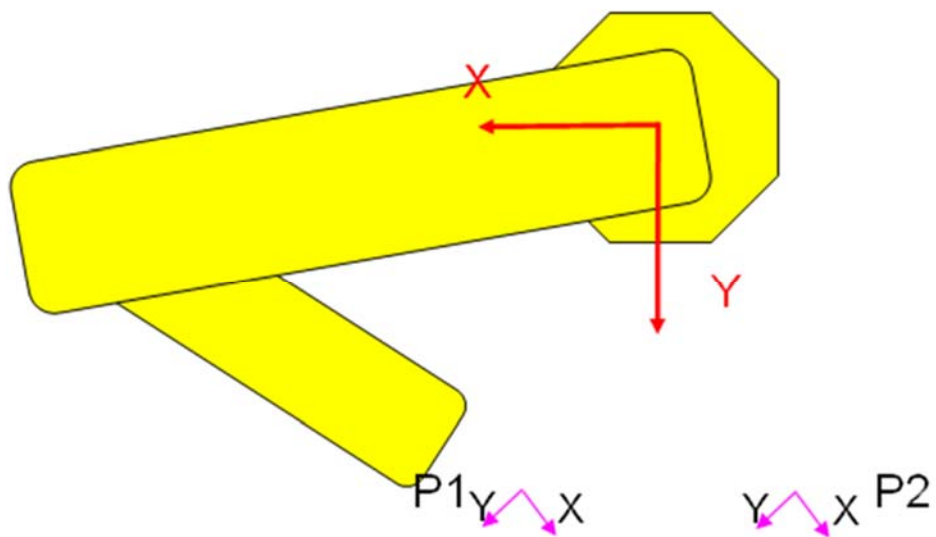
La localisation et l'orientation sont interpolées pendant tous les déplacements du robot.

Lors d'un déplacement vers une position la première lettre commande le type de déplacement :

- J : JOINT (articulation)
- L : LINEAR (linéaire) Dx et Dq cste
- C : CIRCULAR (mouvement en arc de cercle)
- A : CIRCLE ARC (mouvement en arc de cercle avec terminaisons différentes)

Exemples de déplacement vers un point à partir du point P [1] :





P1=pos(50,250,100,180,0,135,'R')

P1=pos(50,250,100,180,0,135,'R')

P1=pos(50,250,100,180,0,135,'L')

P1=pos(50,250,100,180,0,135,'L')

JOINT	LINEAR

P2=pos(-50,250,100,180,0,135,'R')

P2=pos(-50,250,100,180,0,135,'L')

P2=pos(-50,250,100,180,0,135,'L')

P2=pos(-50,250,100,180,0,135,'R')

3-1-4-2 Vitesse de déplacement

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

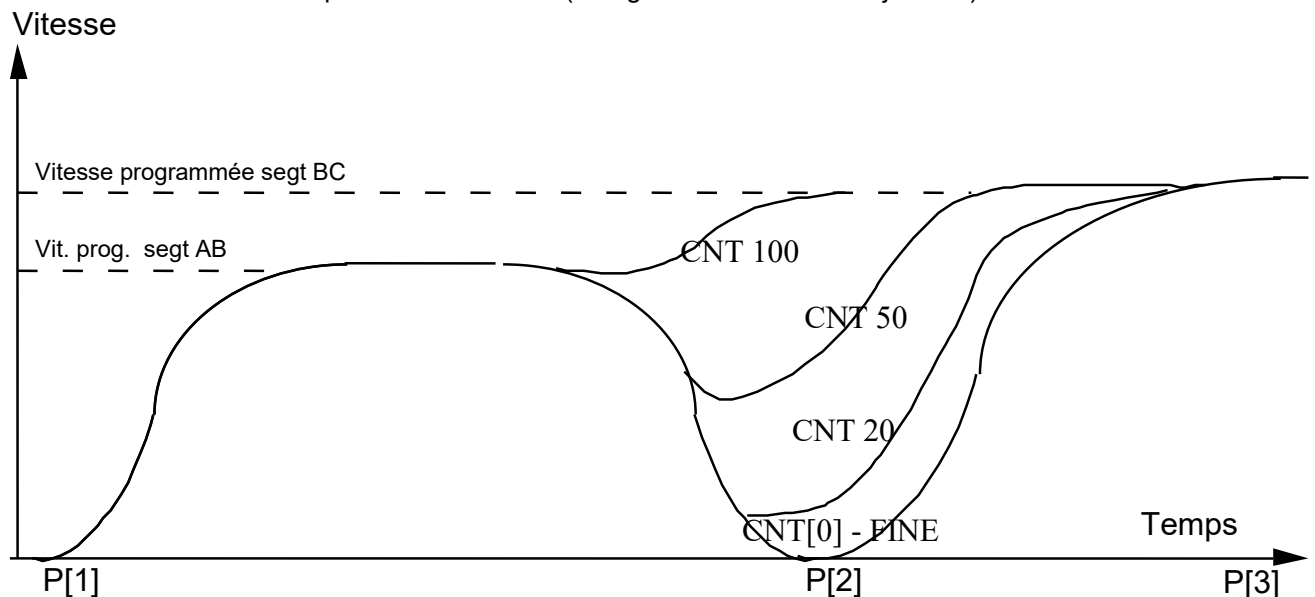
- en JOINT :
Valeur en % de la vitesse maximale, J P [1] 50%
Valeur en secondes : temps de déplacement imposé, J P [1] 2,0 sec
- en LINEAR, CIRCULAR ou CIRCLE ARC :
Valeur en mm/s de la vitesse maximale, L P [1] 4000 mm/sec
Valeur en secondes : temps de déplacement imposé, L P [1] 2,0 sec
Valeur en degrés par secondes : vitesse angulaire imposée, L P[1] 1200,0 deg/sec

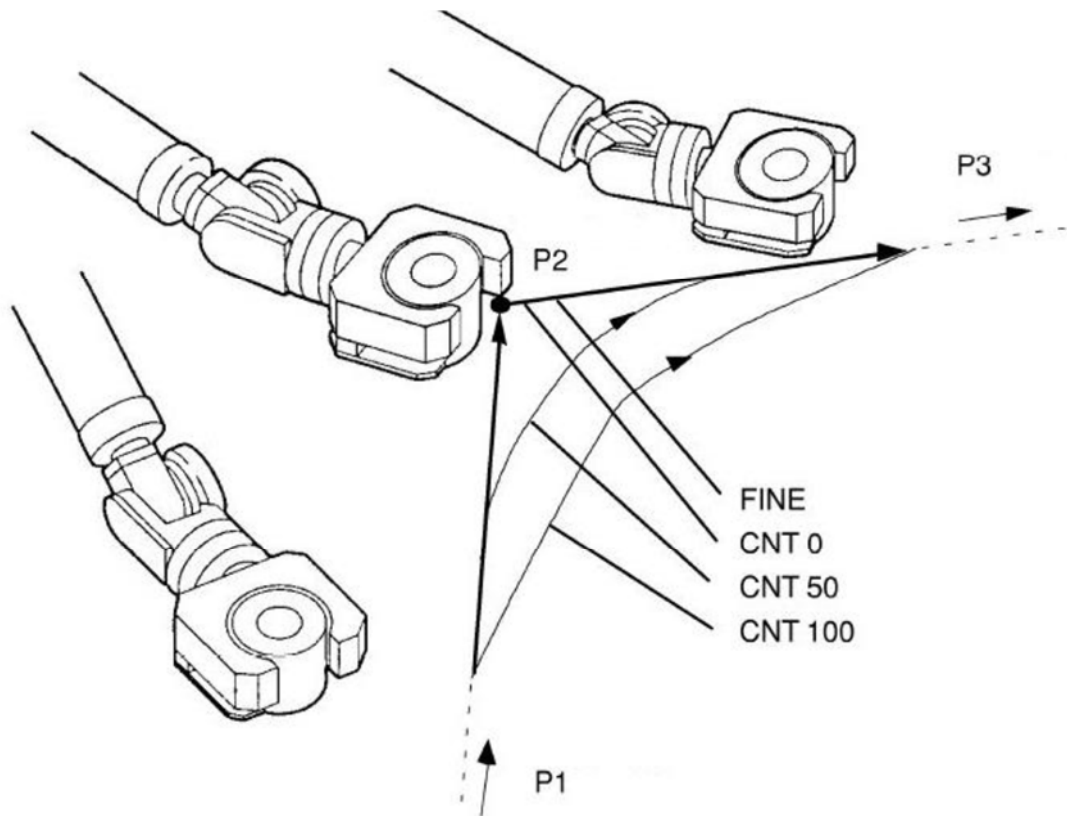
Nota : Les valeurs indiquées sont toujours pondérées en (%) de 0 à 100 par la valeur affichée sur Teach-Pendant

3-1-4-3 Terminaison d'un déplacement

La terminaison d'un déplacement définit comment le robot termine le mouvement :

- Terminaison **FINE** : arrêt du robot avec une précision maximale.
- Terminaison **CNT 0 à 100** (Continu Terminaison) :
 - CNT 0 : décélération complète mais pas d'arrêt précis au point ; (ce mode de terminaison est recommandé par rapport au Fine)
 - CNT 50 : considère que le point est atteint lorsqu'il reste 50% de décélération pour l'atteindre.
 - CNT 100 pas de décélération (lissage maximum de la trajectoire).

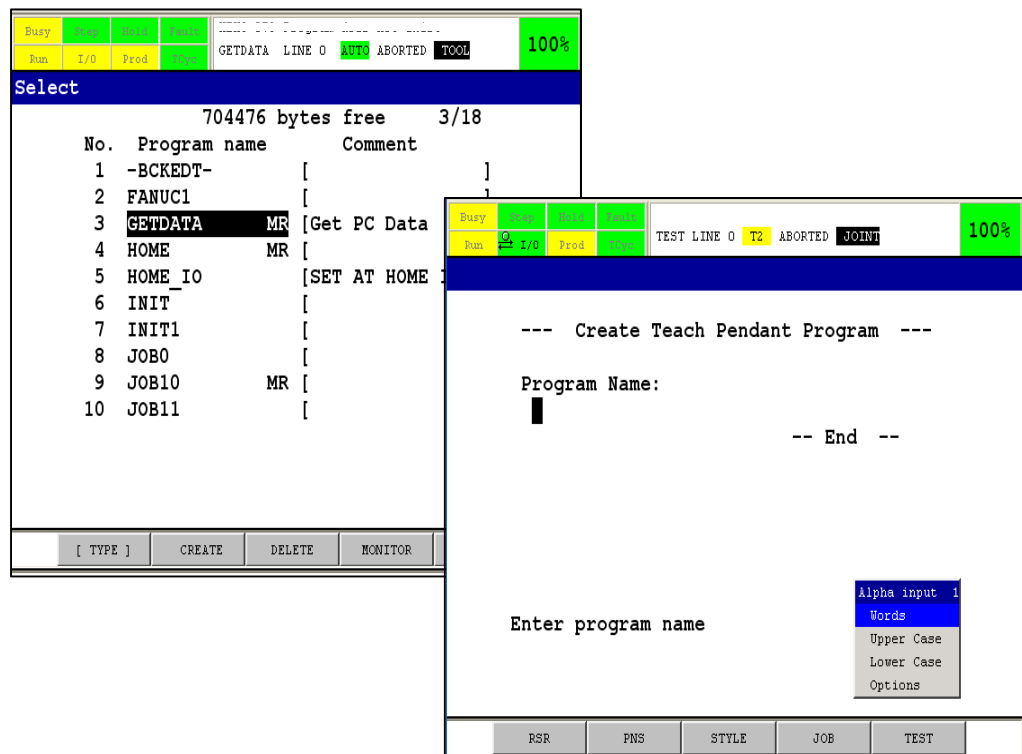




3-2 Système de gestion des fichiers

3-2-1 Création d'un programme

- Appuyer sur la touche **SELECT**
- Appuyer sur la touche **CREATE**



3-2-2 Édition d'un programme

Lorsqu'un programme est créé, il est possible de le sélectionner par la touche **SELECT** du « Teach Pendant ».

Une fois le programme sélectionné, le programme peut être édité à partir de n'importe quel menu du « Teach Pendant » via la touche **EDIT**.

3-2-3 Exécution d'un programme

Avec les touches SHIFT +  pour exécuter vers le bas le programme à partir de la position actuelle ou SHIFT +  pour reculer, en mode continu ou pas à pas STEP.

IV - LANGAGE TPE

Voir Manuel **FANUC** « Aide à la programmation TPE »

4.1. Instructions de saut conditionnel « IF/SELECT »

Une instruction de saut conditionnel permet d'effectuer un saut à une étiquette située dans un même programme si certaines conditions sont vraies.

4-1-1 Instruction IF

Effectue un saut en fonction d'une condition vraie.

IF [valeur 1][opérateur][valeur 2][opérande]

Exemples :

IF R[1]=2 JMP LBL[1] Si le registre R[1] est égal à 2 sauté à l'étiquette 1

IF DI[3]=ON [1] CALL SUB1 Si l'entrée digitale [3] est ON, appeler le sous-programme 1

4-1-2 Instruction SELECT

Effectue un saut ou plusieurs saut en fonction de la valeur d'un registre.

SELECT R[n] = [valeur1] , saut
 [valeur2] , saut
 [valeur3] , saut
 [valeurn] , saut
 ELSE , saut

Exemple :

SELECT R[1] = 1, JMP LBL [1]
 = 2, JMP LBL [2]
 = 3, JMP LBL [3]
 = 4, JMP LBL [4]
 ELSE , CALL SUB2

4.3. Sous-programmes

4.4. Divers

4.5. Transformations :

