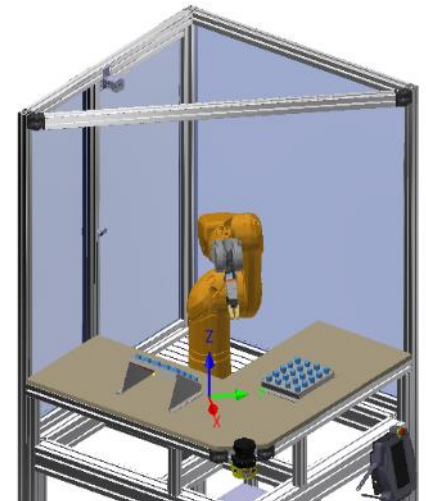


1. Objectif :

Créer la cellule robot du département GEII sur le logiciel de Programmation Hors Ligne Staubli Robotics Suite 2019. Cette cellule se compose d'un bâti fabriqué à l'aide de profilé aluminium, d'un robot Staubli TX2-60 avec contrôleur CS9 et pendant SP2, d'une pince électrique Schunk EGP 40, d'un scrutateur laser de sécurité Sick C300 Mini.

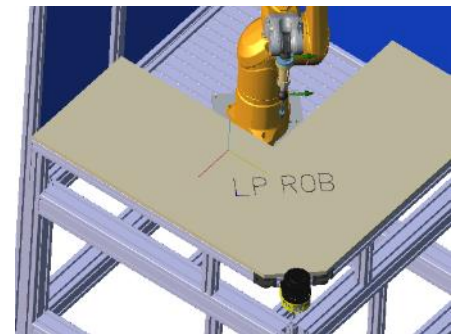
Créer d'une application de traçage et voir l'influence du lissage sur le tracé.

Créer une application de Pick&Place



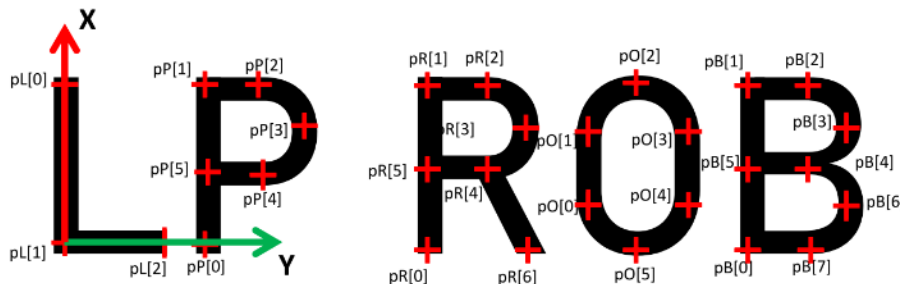
2. Objectifs intermédiaires :

1. Importer le bâti de la cellule issue de CAO
2. Positionner le Robot dans la cellule
3. Manipuler le robot à partir du 3D
4. Manipuler le robot à partir de l'émulateur du pendant



3. Application tracé LP ROB :

1. Créer une application permettant d'écrire « LP ROB »
Pour cela on donne les points de passages pour chaque lettre
Nommer cette application « LP_ROB_Votre NOM »




	x	y		x	y
pL[0]	80	0	pP[0]	0	64
pL[1]	0	0	pP[1]	80	64
pL[2]	0	44	pP[2]	80	84
			pP[3]	60	108
			pP[4]	40	84
			pP[5]	40	64

	x	y
pR[0]	0	164
pR[1]	80	164
pR[2]	80	184
pR[3]	60	208
pR[4]	40	184
pR[5]	40	164
pR[6]	0	208

	x	y
pO[0]	22	228
pO[1]	58	228
pO[2]	80	250
pO[3]	58	272
pO[4]	22	272
pO[5]	0	250

	x	y
pB[0]	0	292
pB[1]	80	292
pB[2]	80	312
pB[3]	60	336
pB[4]	40	312
pB[5]	40	292
pB[6]	20	336
pB[7]	0	312

2. Importer outil Stylo
3. Créer Handle/Handler
4. Importer 3D Logo

5. Créer appli VAL3 (Données/Géométrie (frame/points/JHome/Outil))
6. Ecrire le programme traçage du logo « LP ROB » 
7. Simuler le programme pour vérifier le bon fonctionnement
8. Reprendre le programme pour tracer 5 fois LPROB et en changeant à chaque fois les paramètres de vitesse
 - 8.1. Pour cela on utilisera les instructions : **for – switch et fonction de lissage page 134 du manuel VAL3**
 - 8.2. 1^{er} tracé avec lissage (blend) **Joint** et leave et reach de **50** mm
 - 8.3. 2^{ème} tracé avec lissage (blend) **Joint** et leave et reach de **25** mm
 - 8.4. 3^{ème} tracé avec lissage (blend) **Cartesian** et leave et reach de **25** mm
 - 8.5. 4^{ème} tracé avec lissage (blend) **Cartesian** et leave et reach de **10** mm
 - 8.6. 5^{ème} tracé avec lissage (blend) **off**

Options de lissage :

Il est possible de réduire de manière importante la durée du mouvement en " lissant " la trajectoire au voisinage des points **pDepart** et **pAppro**. Pour cela, on utilise le champ **blend** du descripteur de mouvement. Lorsque la valeur de ce champ est **off**, la trajectoire du robot s'arrête à chaque point. En revanche, quand le paramètre est réglé sur **joint** ou **Cartesian**, la trajectoire est lissée au voisinage de chaque point et le robot ne s'arrête plus aux points de passage.

Quand le champ **blend** a la valeur **joint** ou **Cartesian**, deux autres paramètres doivent être définis : **leave** et **reach**. Ces paramètres déterminent à quelle distance du point d'arrivée on quitte la trajectoire nominale (début du lissage), et à quelle distance du point d'arrivée on la rejoint (fin du lissage).

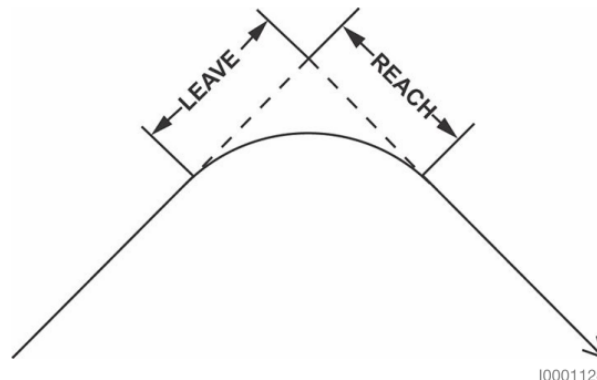




Figure 10.5 : Définition des distances : 'leave' / 'reach'

- 8.7. Rajouter au programme le calcul de la durée d'un tracé et l'afficher dans une fenêtre « popup »
 - 8.7.1. Rajouter à la ligne avant le tracé : `nInstant=clock()` (créer le paramètre local **nInstant**)
 - 8.7.2. Rajouter à la fin du tracé le calcul de la durée : `nDuree=clock()-nInstant` (créer la donnée **nDuree**)
 - 8.7.3. Rajouter la ligne pour afficher la durée dans un fenêtre « popup » : `popUpMsg(toString(".3",nDuree))`
- 8.8. Simuler le programme
- 8.9. Avec l'accord de l'enseignant transférer l'application sur le robot. 
- 8.10. Visualiser les vidéos de manipulation robot avant d'aller sur la cellule réelle 

4. Application : Echange de pions

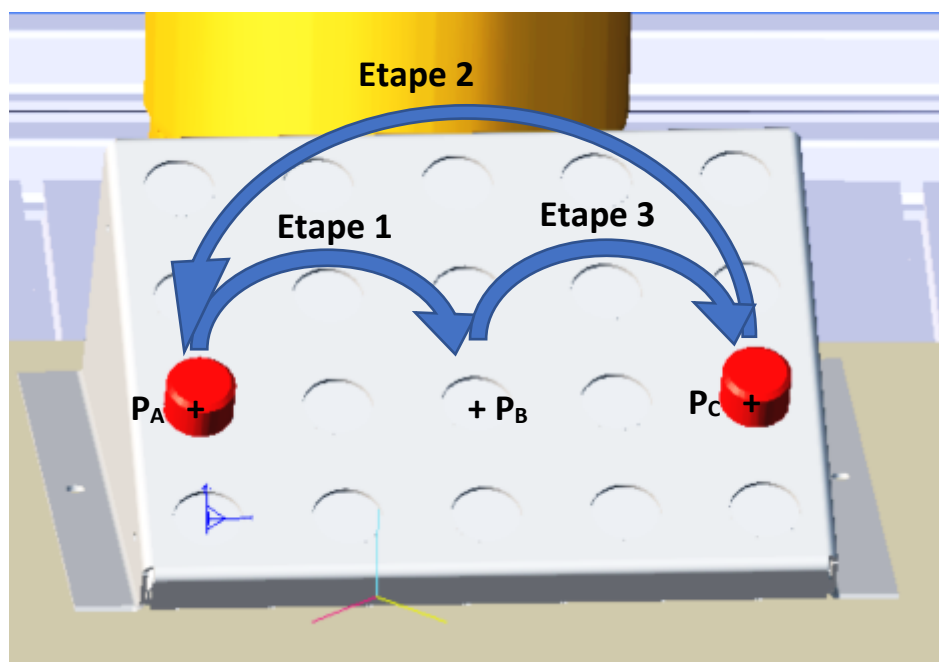
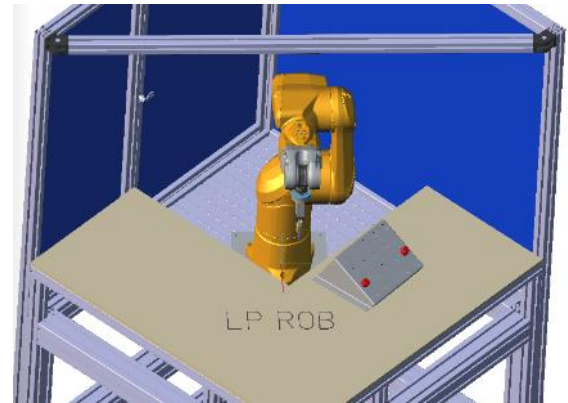
Objectifs :

Faire une application de pick&place qui échange les pions de place.

L'application sera lancée par un bouton connecté sur l'entrée rapide 1 et sera stoppée par un bouton sur l'entrée rapide 2.

L'application comprendra **3** programmes en plus du *start* et *stop*.

Le programme « **Cycle** » gèrera les opérations à effectuer et appellera au besoin le programme de « **prise** » et de « **pose** »



Travail à faire

1. Importer le préhenseur « SchunkEGP40 » et configurer
2. Importer la palette et les pions mettre en place
3. Créer appli VAL3 (Données/Géométrie)
 - 3.1. Voir donnée « Prog change_pion »
4. Ecrire les programmes
 - 4.1. Voir programmes « Prog change_pion »
5. Simuler le programme en mode débogage pour pouvoir piloter les entrées et simuler les boutons départ cycle et fin de cycle

5. Tour de Kapla

Objectif et données :

A partir de kapla disposés à intervalle régulier sur la tranche, sur le plan de travail le robot devra constituer une tour de kapla à un autre endroit sur le plan de travail.

Les dimensions d'un Kapla sont : $L \times l \times h = 110 \times 22 \times 8$ (mm)

Conseils :

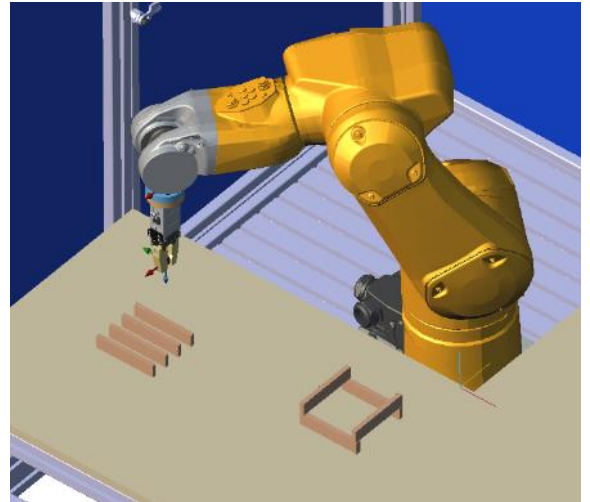
Données à créer :

- Un repère pour la prise
- Un point de prise du premier Kapla
- Une transformation de décalage pour la prise
- Un repère de pose centré à la base de tour
- Un point de pose à l'origine du repère de pose
- Une transformation de décalage pour la pose
- Une transformation de décalage d'approche
- Une Variable numérique pour le nombre de Kapla

Créer en variable locale :

- Un point de prise courant
- Un point de pose courant
- Une variable numérique pour compter le Kapla courant
- Au besoin des variables numériques

On donne ci-dessous un exemple de disposition avec les transformations pour constituer la tour.



nKapla Courant	nPaire Kapla	Tx	Ty	Tz	Rx	Ry	Rz
1	1	45	0	0	0	0	0
2	1	-45	0	0	0	0	0
3	2	0	45	-22	0	0	90
4	2	0	-45	-22	0	0	90
5	3	45	0	-44	0	0	0
6	3	-45	0	-44	0	0	0
7	4	0	45	-66	0	0	0
8	4	0	-45	-66	0	0	0
9	5	45	0	-88	0	0	90
10	5	-45	0	-88	0	0	90

