

# Projet Robotique et Intégration BUT 2 AI

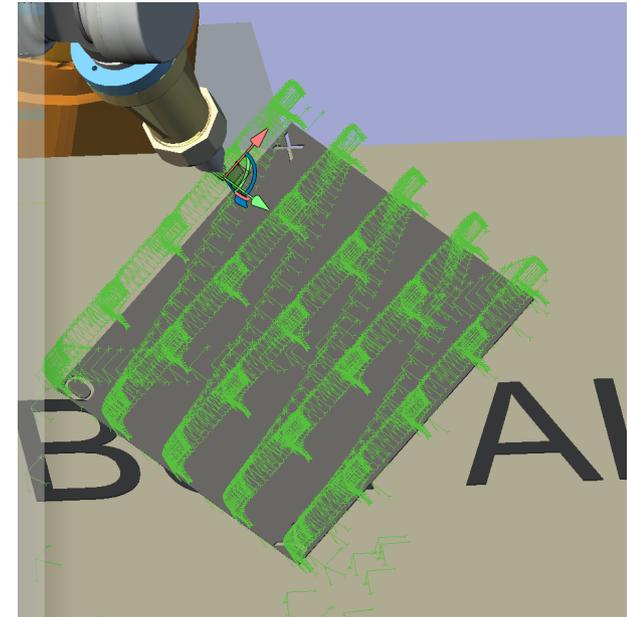
Bertrand  
Vandeportaele

IUT GEII Toulouse



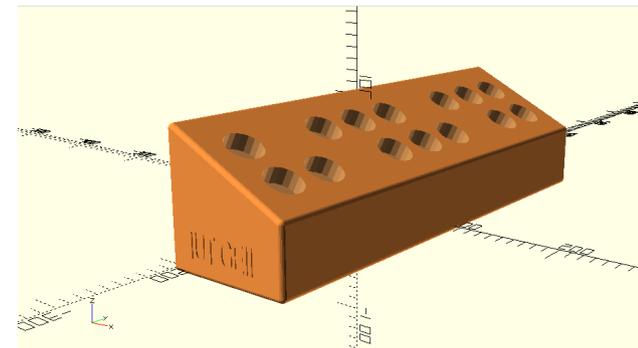
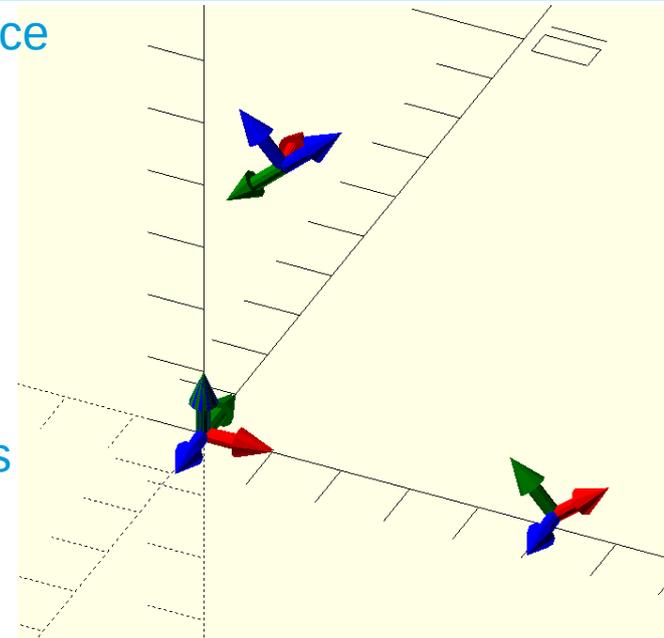
# Enseignements préliminaires liés

- Programmation Hors ligne avec Staubli Robotic Suite
  - Découverte des capacités de déplacement du robot 6R
  - Espace de travail
  - Butées articulaires
  - Changement de configurations du bras et singularités
  - Différences fondamentales entre movel et movej
  - Apprentissage des points
  - Utilisation de repères
- Application Tracé de Logo
- Applications Pick and Place (dé)palettisation et empilement de Kapla
- <https://bvdp.inetdoc.net/wiki/doku.php?id=tprobotique>



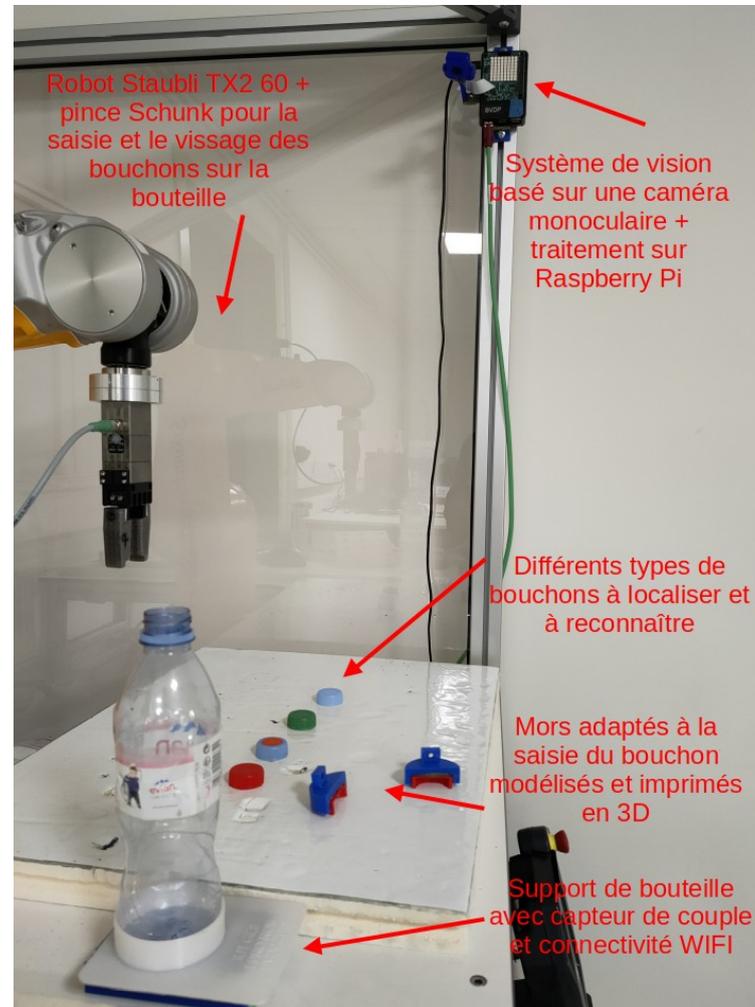
# Enseignements préliminaires liés

- Base de modélisation 3D avec Openscad: Libre et Opensource
  - Description textuelle
  - Hiérarchique
  - Paramétrique
  - Algorithmique
  - Primitives géométriques 2D et 3D et multiples opérateurs
  - Énormément de bibliothèques disponibles
  - Antisèche sur une page A4
  - Animation
  - Export 3D et 2D (simulation de caméra)
- Modélisation d'un repère orthonormé
- Modélisation d'un robot 6R avec Denavit-Hartenberg avancé
- Modélisation d'une palette
- <https://bvdp.inetdoc.net/wiki/doku.php?id=openscad>



# Objectifs du projet

- Aborder une tâche robotique "complexe": Saisie et vissage d'un bouchon sur un bouteille à l'aide d'un robot 6 Axes Staubli



# Objectifs du projet

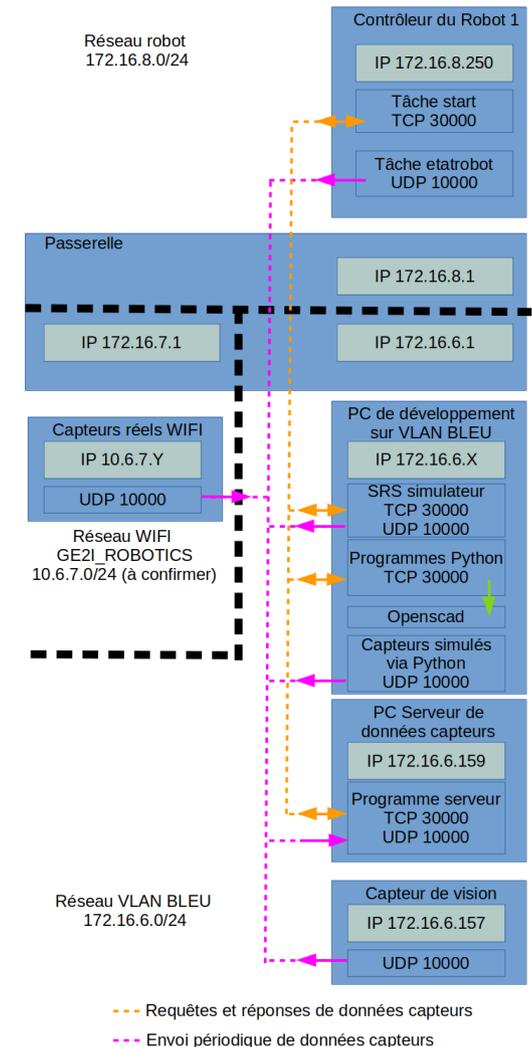
En termes de mouvements:

- Positions de saisie variable (test d'atteignabilité dynamique, vérification de non changement de configuration du bras)
- Mouvement hélicoïdal (combinaison de translation et de rotation), pré-dévisage
- Mouvements paramétriques (décrire le mouvement en fonction de pas de vis et de débattement angulaire). Les "bonnes" valeurs sont apprises à posteriori
- Utilisation d'articulation multi-tours: multiples solutions pour le MGI
  - Nécessité de jongler entre commande cartésienne et articulaire (par exemple : attraper le bouchon à une pose cartésienne mais avec  $j_6=0$ )
- Orientation relative entre le bouchon et le goulot non connue:
  - Rattrapage de jeu: lâcher le bouchon avant de le serrer pour mettre en contact les filetages
  - Arrêt du vissage quand un couple de serrage est atteint : capteur de couple et parallélisme de tâche pour avoir un mouvement lisse
  - Remonter la pince après vissage sans tourner le poignet
- Utilisation des mouvements du robot pour mesurer la conformité de la réalisation de la tâche
- Mesure et optimisation des temps de cycle

# Objectifs du projet

En termes de système:

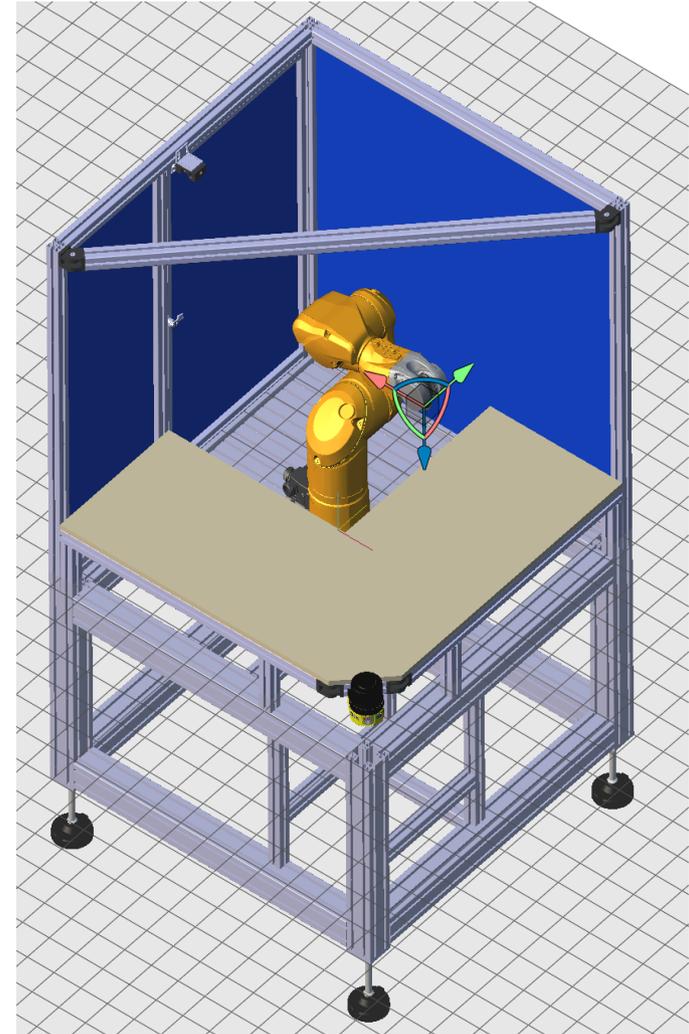
- Intégration de capteurs « boites noires »
  - Capteur de Vision basé sur une Raspberry Pi (Projet BUT3 All)
    - Étalonnage géométrique (transformation rigide 3ddl entre repère vision et repère robot)
  - Capteur de couple (Jauge de contrainte TP BUT3 et uC WIFI)
    - Apprentissage du couple de consigne
- Middleware maison et communication réseau via sockets TCP et UDP
  - Communication avec protocole simple basé sur chaînes de caractères
  - Debug à l'aide de Wireshark
  - Simulateurs des capteurs fournis en python
    - Gestion des modes « à problèmes »
    - Chaque groupe dispose de « cases » pour simuler ce qu'il souhaite avec des programmes python
- **Approche Logicielle et matérielle modulaire :**
  - Les différents composants produisent et consomment des données capteurs (ex : le robot est un capteur qui informe le système de sa pose, de son action en cours...)
    - Ex :Affichage synchronisé de jumeau numérique dans Openscad



# Objectifs du projet

En termes de modélisation 3D:

- Reproduire le robot et tous les éléments hors de SRS
- Importer les fichiers stl des parties du robot et les articuler
- Prise de cotes sur la scène réelle :
  - Cellule
  - Champ de vision camera/éclairage
  - Outil + préhenseur
  - Bouchon
  - Support bouteille
  - Référencement des modèles par rapport au robot
- Impression 3D de certains éléments



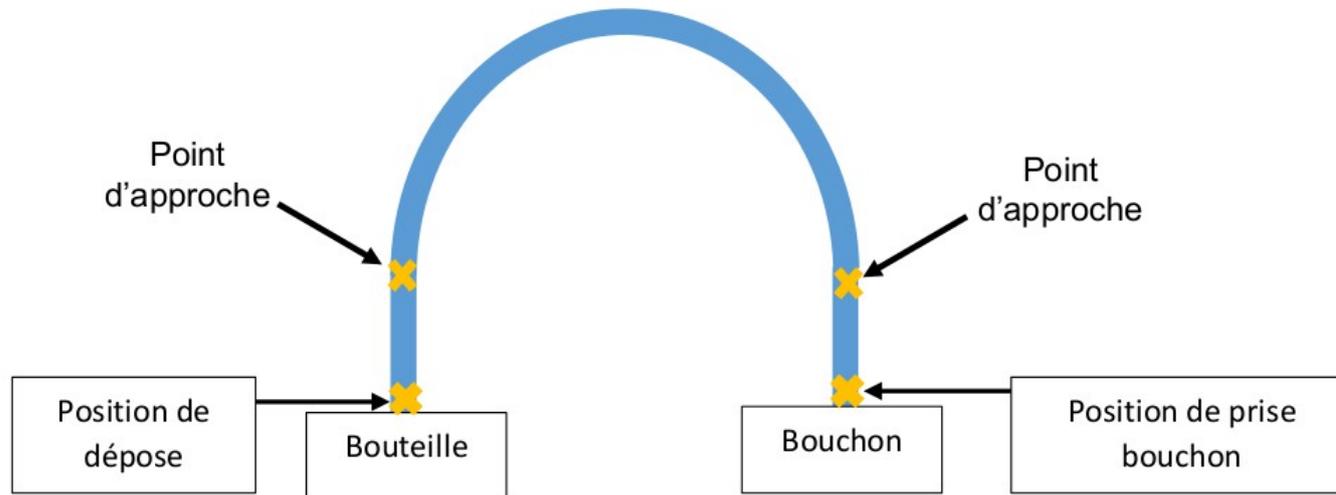
# Déroulement du projet

- Projet sur 2 semaines (coupé par 2 semaines de vacances)
  - 1° semaine:
    - Mouvement en Boucle Ouverte
    - Modélisation 3D des éléments de la cellule
    - Tests en simulation
  - 2° semaine:
    - Intégration des capteurs pour mouvement en Boucle Fermée
    - Animation en temps réel de la cellule dans openscad
    - Enregistrement de trajectoires pour le rejeu
    - Étalonnage Vision/Robot
    - Tests sur le robot
- Gestion de versions avec Git et Gitlab
- Rédaction d'un rapport à chaque fin de semaine avec un canevas imposé

# Modélisation des trajectoires

- Dessin simple et nommage des points

Dessin du cycle :



- Décomposition de la trajectoire en tronçons délimités par des points (dans l'espace articulaire ou cartésien), utilisant une interpolation (dans l'espace articulaire ou cartésien) et avec des paramètres de lissage

# Modélisation des trajectoires

- Dessin en perspective et représentation des contraintes

Pick Bouchon:

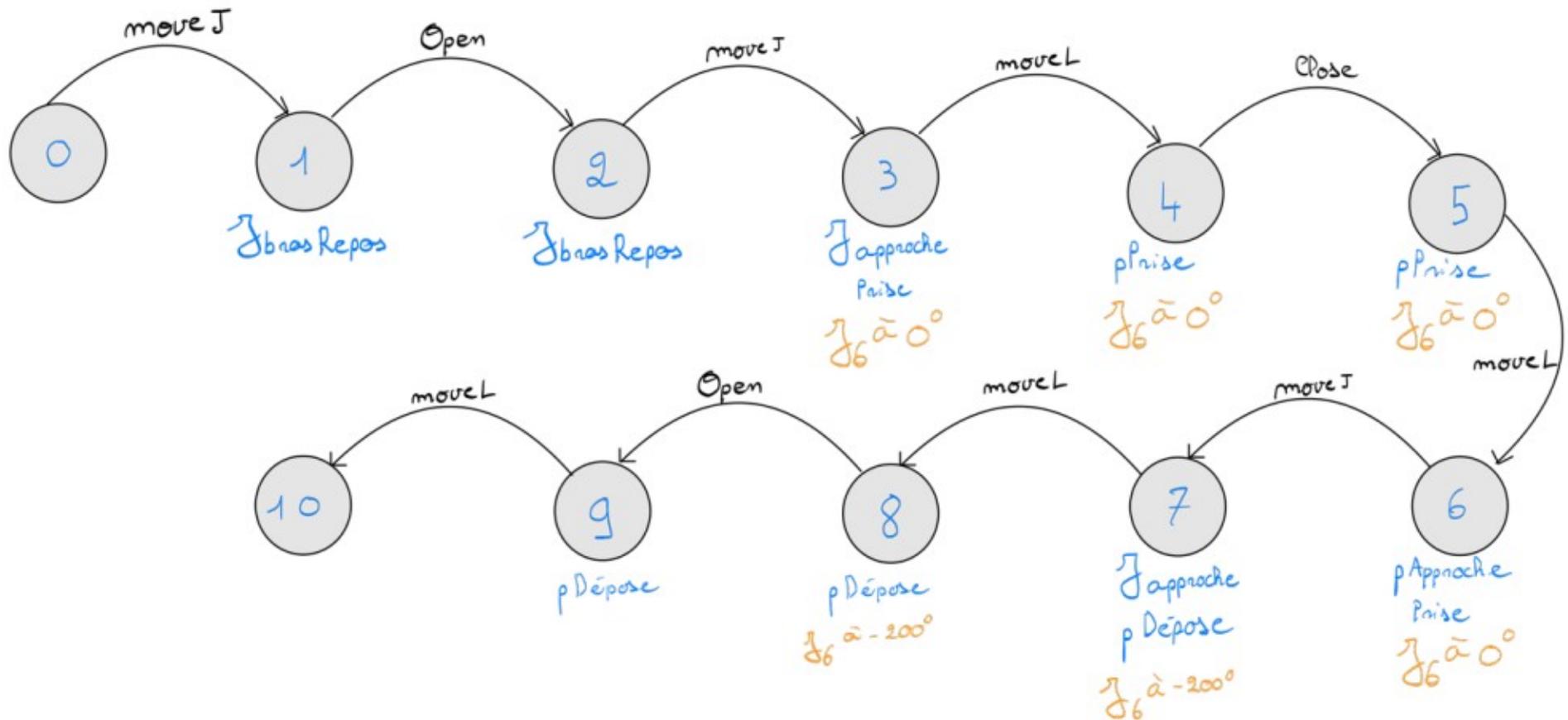
- J'approfondis**  
 ↳ calculé à partir de planche, qui a été corrigé pour avoir  $J_6 = 0^\circ$
- planche corrigé**  
 ↳ corrigé pour avoir  $J_6 = 0^\circ$  pendant le PICK au bouchon
- p'approfondissement corrigé**  
 ↳ calculé à partir de j'approfondis pour avoir  $J_6 = 0^\circ$

PLACE Goulot:  
(prévisage + visage)

- J'approfondit**  
 ↳ calculé à partir de planche en appes, avec  $J_6 = -200^\circ$  pour obtenir le prévisage
- planche corrigé**  
 ↳ corrigé pour avoir  $R_2$  tel que  $J_6 = -200^\circ$  pour maintenir le prévisage  
 → puis open (t'ad)  
 → puis close (t'ad)  
 ↳ pour bien positionner le bouchon pour le visage
- trif Visage.  $R_2 = 0$  vers  $z = 0$**   
 ↳  $R_2$  vers  $z = 0$  paramètre visage  
 attention → Bouchon (visage)  
 → Mvt rotation  $R_2$ :  
 ou inversement  $R_2$  visage?  
 de  $n$  rotations régulières / 10  
 → Mvt vertical  $z$ :  
 ou inversement  $R_2$  visage?  
 de  $(n \text{ Rotat } / 360) * n \text{ Pas Delvis}$   
 \* (n Iteration / 10)  
 → mvt avec  $R_2$  visage a paramètre

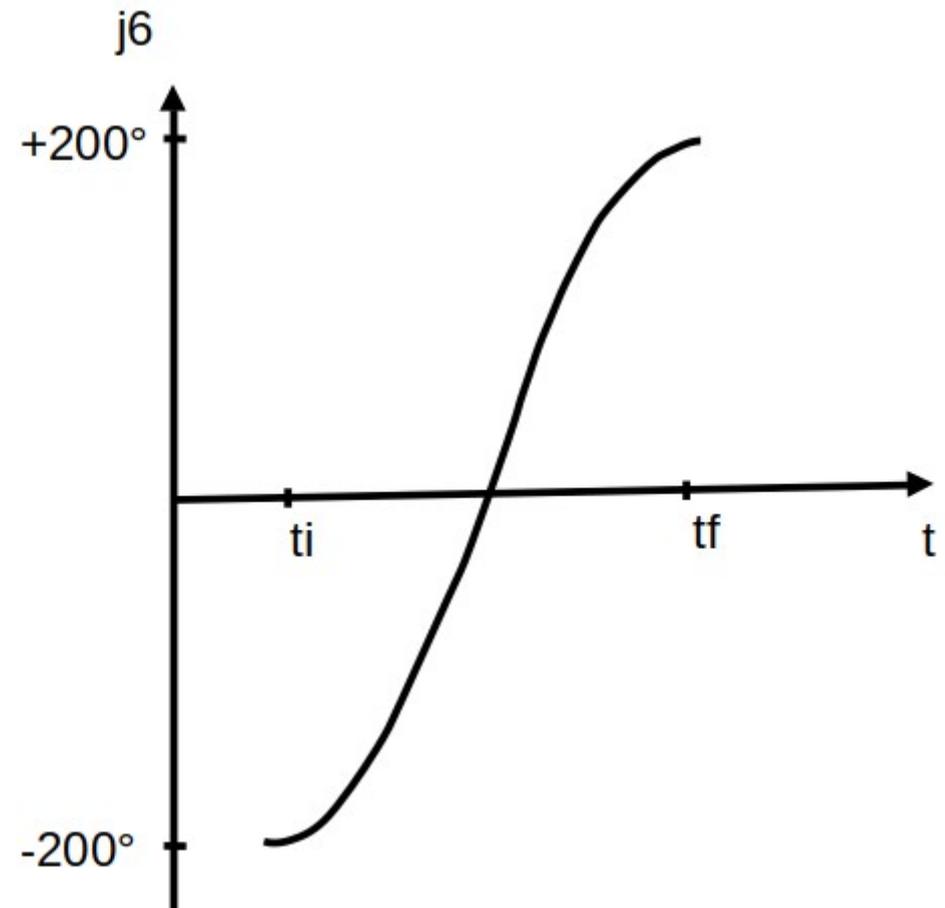
# Modélisation des trajectoires

- Représentation par un graphe d'états



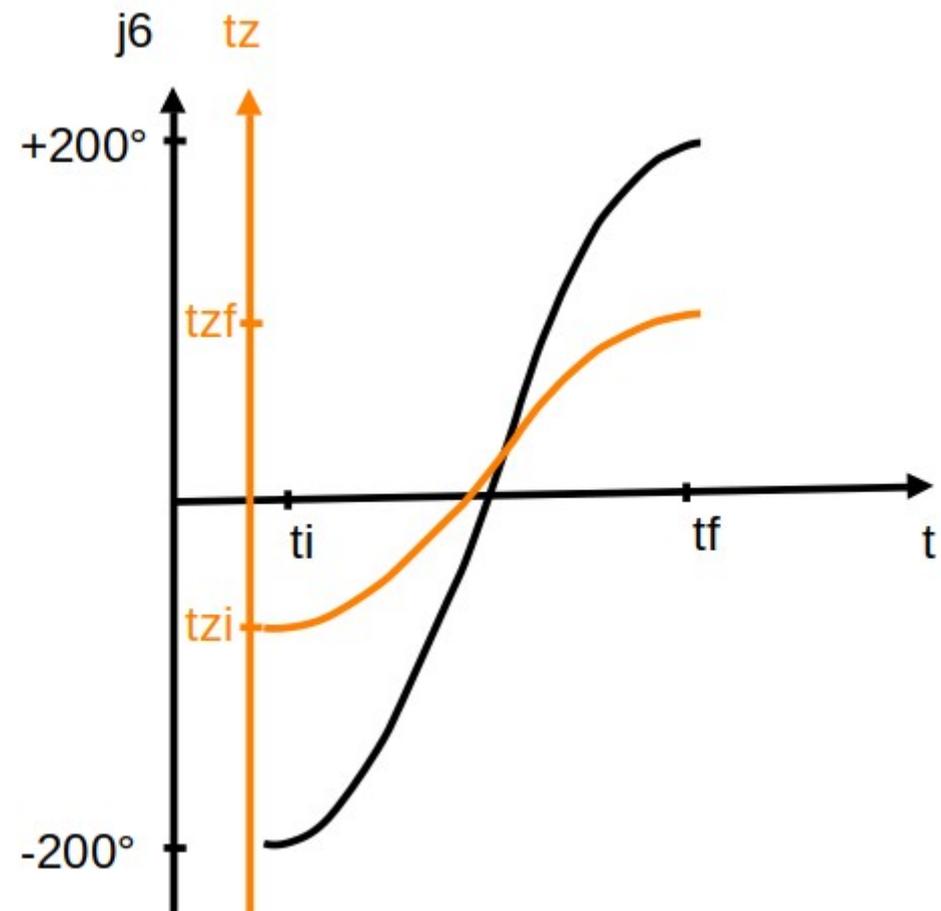
# Modélisation des trajectoires

- Tracé et analyse temporelle : trajectoire hélicoïdale
- J6 avance de  $400^\circ$



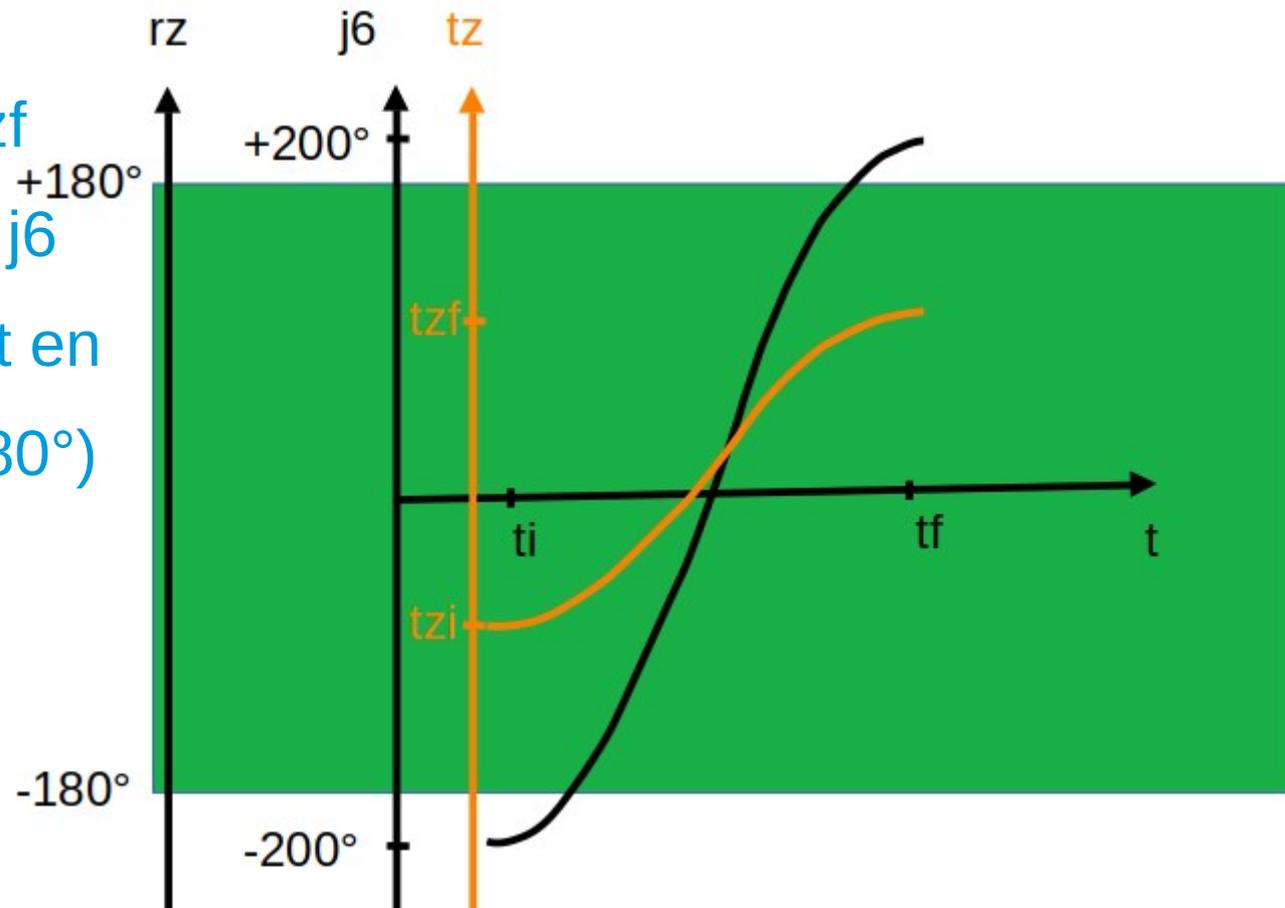
# Modélisation des trajectoires

- Tracé et analyse temporelle : trajectoire hélicoïdale
- J6 avance de  $400^\circ$
- tz doit parcourir de tzi à tzf comme fonction affine de j6



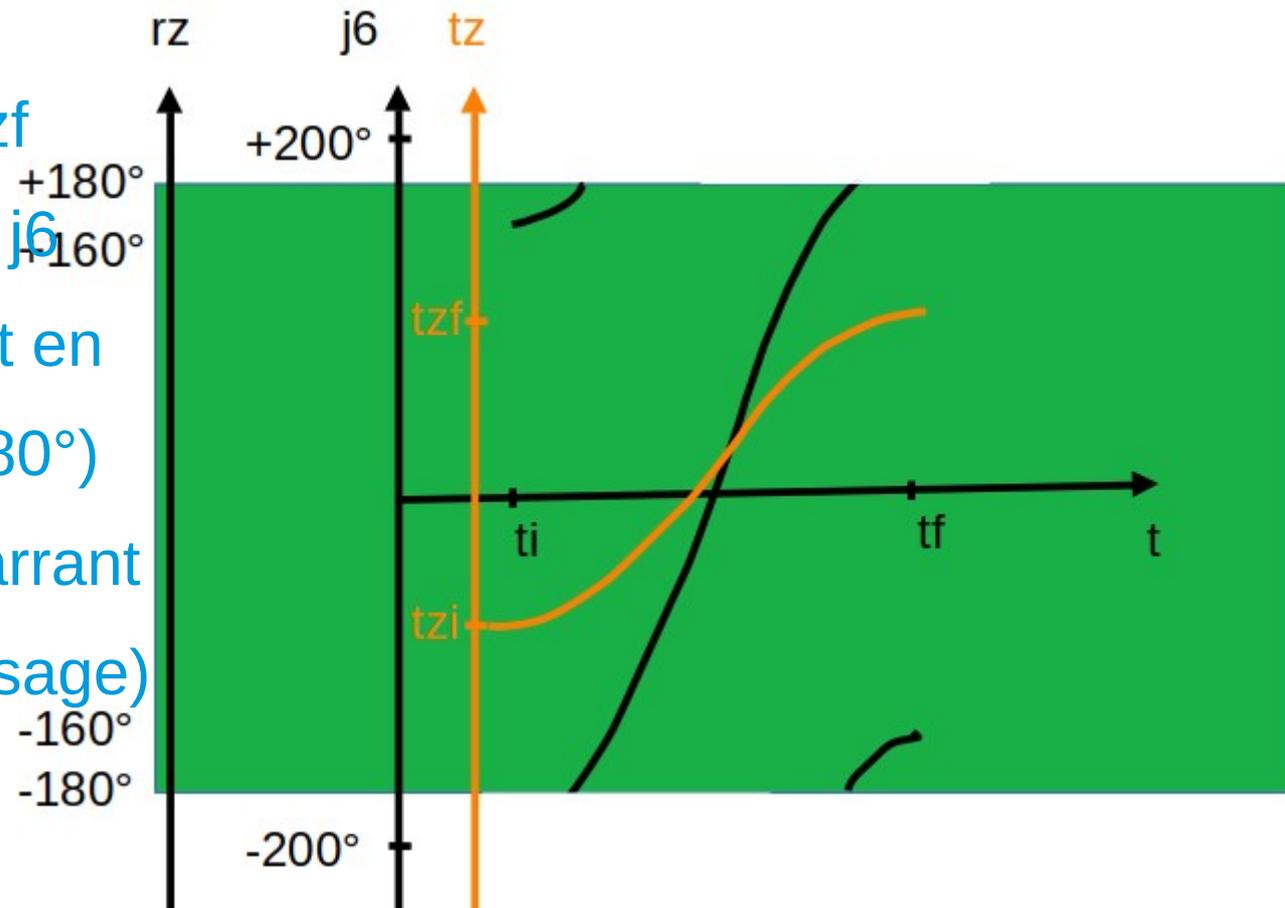
# Modélisation des trajectoires

- Tracé et analyse temporelle : trajectoire hélicoïdale
- J6 avance de  $400^\circ$
- tz doit parcourir de tzi à tzf comme fonction affine de j6 comme fonction affine de j6
- Il faut donc piloter le robot en cartésien avec tz/rz ( $\pm 180^\circ$ )



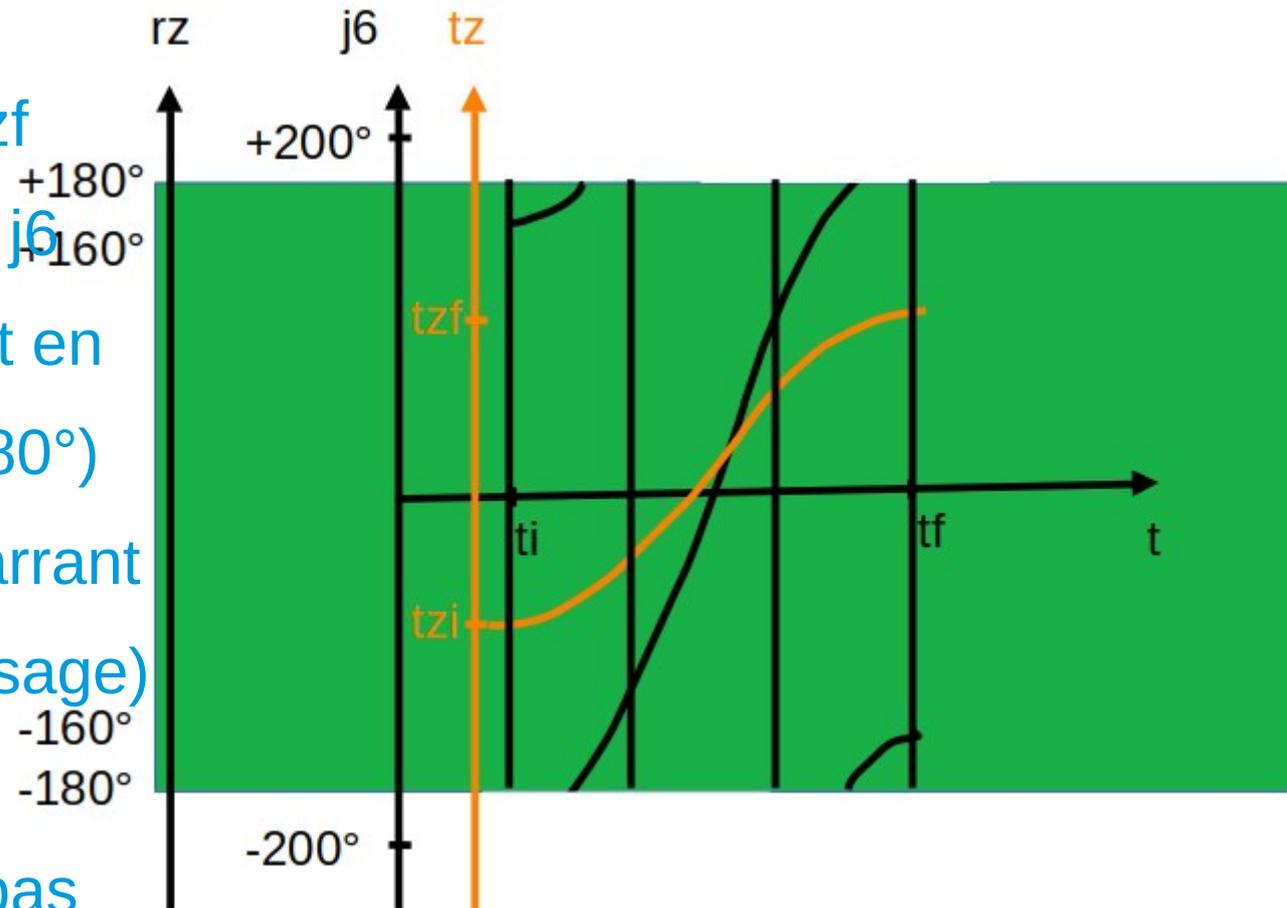
# Modélisation des trajectoires

- Tracé et analyse temporelle : trajectoire hélicoïdale
- J6 avance de  $400^\circ$
- tz doit parcourir de tzi à tzf comme fonction affine de j6
- Il faut donc piloter le robot en cartésien avec tz/rz ( $\pm 180^\circ$ )
- Traduire j6 en rz en démarrant « au bon tour » (prédéviage)



# Modélisation des trajectoires

- Tracé et analyse temporelle : trajectoire hélicoïdale
- J6 avance de  $400^\circ$
- tz doit parcourir de tzi à tzf comme fonction affine de j6
- Il faut donc piloter le robot en cartésien avec tz/rz ( $\pm 180^\circ$ )
- Traduire j6 en rz en démarrant « au bon tour » (prédéviage)
- Il faut échantillonner plusieurs poses pour ne pas aller « au plus court »



# Gestion de l'arrêt du vissage

- Analogie pilote/copilote
  - Le programme utilisateur informe le pilote à l'avance du futur de la trajectoire du robot (via une file)
  - Le pilote réalise la trajectoire lissée
- Pré-programmation de la trajectoire hélicoïdale "trop grande" dans la file d'ordre du robot
  - Puis consultation en boucle du capteur de couple
    - Si couple de consigne atteint avant l'intégralité du mouvement, interruption du vissage et vidage de la file: la bouteille est conforme
    - Sinon détection de non conformité

# Étalonnage Vision/Robot

- Dépose du bouchon à 3 positions connues du robot dans le champ de vision de la caméra
- Collecte des 3 positions correspondantes fournies dans le repère du système de vision
- Estimation simple ou moindre carrés des 3 paramètres de poses  $t_x, t_y, \theta$  et mise à jour du système de vision pour qu'il fournisse la pose du bouchon dans le repère robot

# Développement de programmes Python

- Récupération de la pose du robot et écriture dans fichiers openscad pour animer le robot, en temps réel ou pour le rejeu de trajectoire
- Comptage des bouteilles (non) conformes
- ...





# • Démonstration

- Démonstration demain en C04 de 9 à 10h
- Plus d'infos sur : <https://bvdp.inetdoc.net/>