Manipulation

1) Le robot Poppy Ergo Jr
2) Espaces de trajectoires
3) MoveIt & OMPL
4) Arbres de transformation « tf »
5) Décomposition d'un mouvement de pick-and-place
6) Aspects pratiques

Manipulation

1) **Le robot Poppy Ergo Jr**
2) Espaces de trajectoires
3) MoveIt & OMPL
4) Arbres de transformation « tf »
5) Décomposition d'un mouvement de pick-and-place
6) Aspects pratiques

Poppy Ergo Jr

Robot éducatif opensource et openhardware

Basé sur Raspberry Pi + la caméra Raspberry Pi

Faible précision/répétabilité

Image carte SD originale :

Snap + Jupyter Notebook (Python)

Image ROS : ROS 1 Melodic (Python)

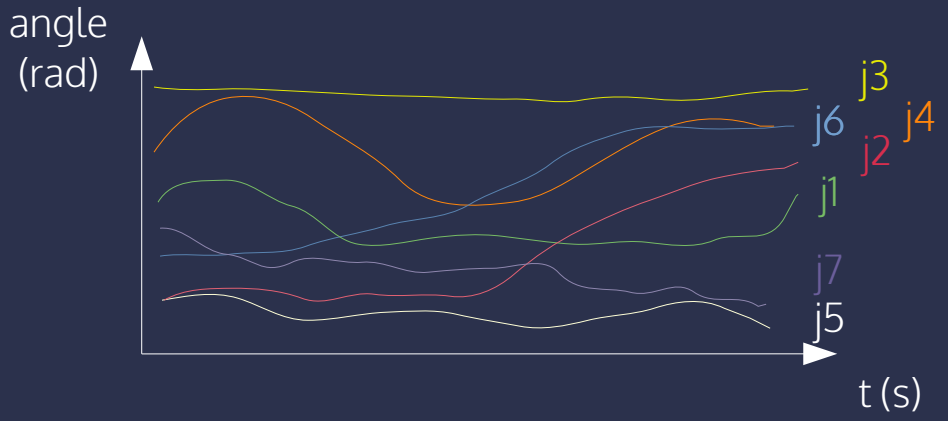
Mode compliant : moteurs mous

https://www.poppy-project.org/fr/robots/poppy-ergo-jr/

Manipulation

1) Le robot Poppy Ergo Jr
2) **Espaces de trajectoires**
3) MoveIt & OMPL
4) Arbres de transformation « tf »
5) Décomposition d'un mouvement de pick-and-place
6) Aspects pratiques

Espaces de trajectoires

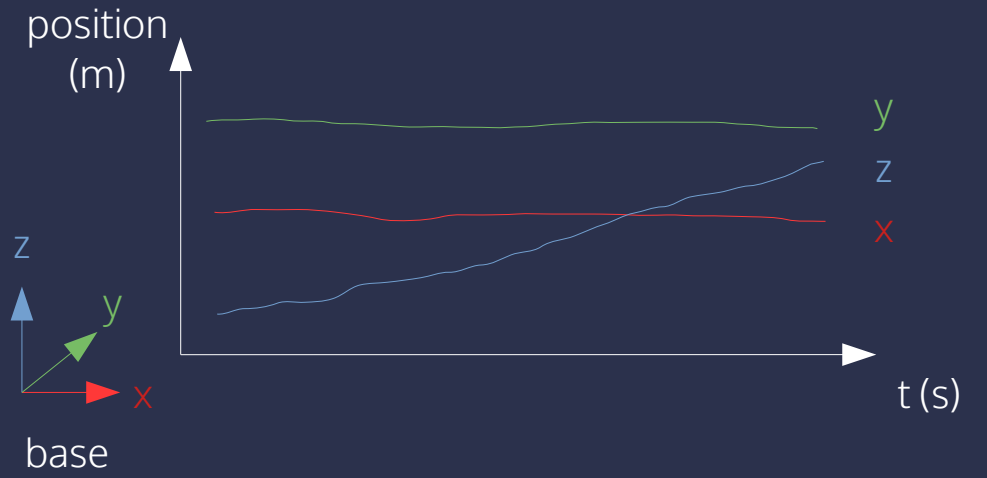


Espace des joints

Espace des configurations

1 trajectoire par joint

Angle en fonction du temps



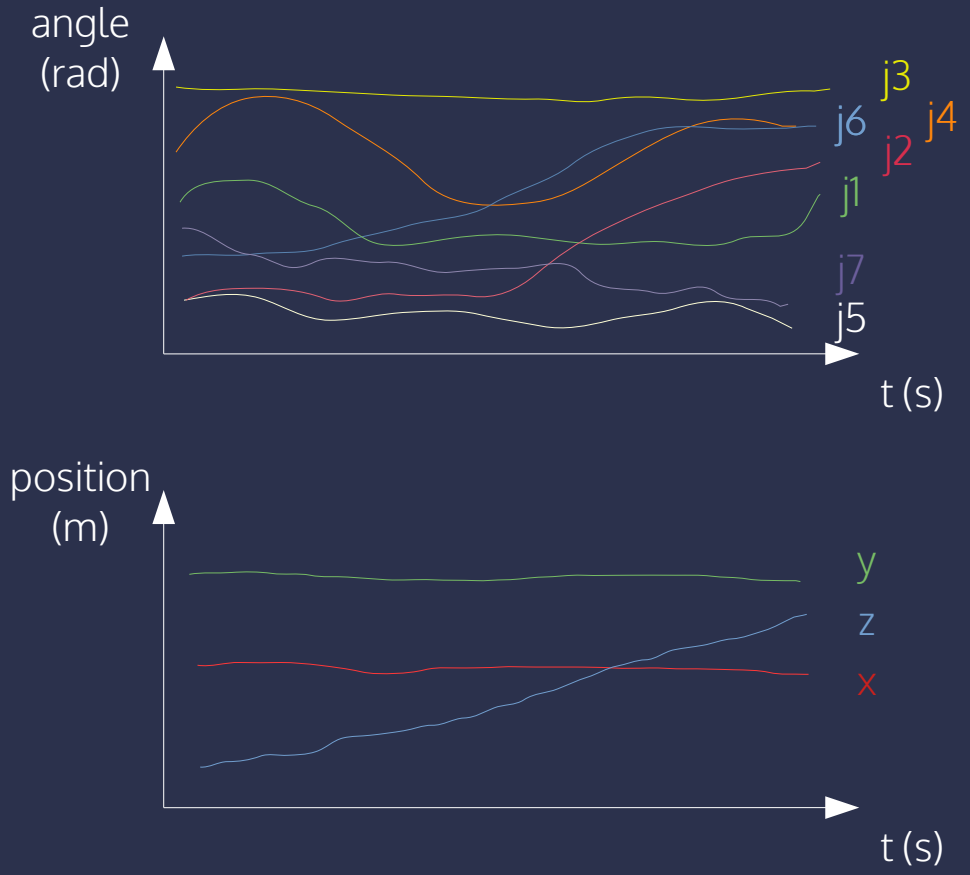
Espace cartésien

Espace de l'effecteur

1 trajectoire par axe

Position en fonction du temps

Espaces de trajectoires



Espace des joints

Géométrie directe
(Forward Kinematics)
FK
/compute_fk

Géométrie inverse
(Inverse Kinematics)
IK
/compute_ik

Espace cartésien

Manipulation

1) Le robot Poppy Ergo Jr
2) Espaces de trajectoires
3) **MoveIt & OMPL**
4) Arbres de transformation « tf »
5) Décomposition d'un mouvement de pick-and-place
6) Aspects pratiques

MoveIt

Une boîte à outils pour la manipulation :

- Calcul de cinématique directe et inverse
- Génération de trajectoires sans collision :

  La biliothèque OMPL

  Sémantique de l'env. de collision : SRDF, CO, ACO
- Prise en compte de contraintes
- Manager de controlleurs moteurs : vitesse, accélération ...

La bibliothèque OMPL

Planners géométriques :

- Kinematic Planning by Interior-Exterior Cell Exploration (KPIECE)
- Probabilistic Roadmap Method (PRM)
- Expansive Space Trees (EST)
- Lower Bound Tree RRT (LBTRRT)
- Rapidly-exploring Random Trees (RRT)

  RRTConnect

RRTConnect

Single-Query randomized planning : pas de précalcul nécessaire

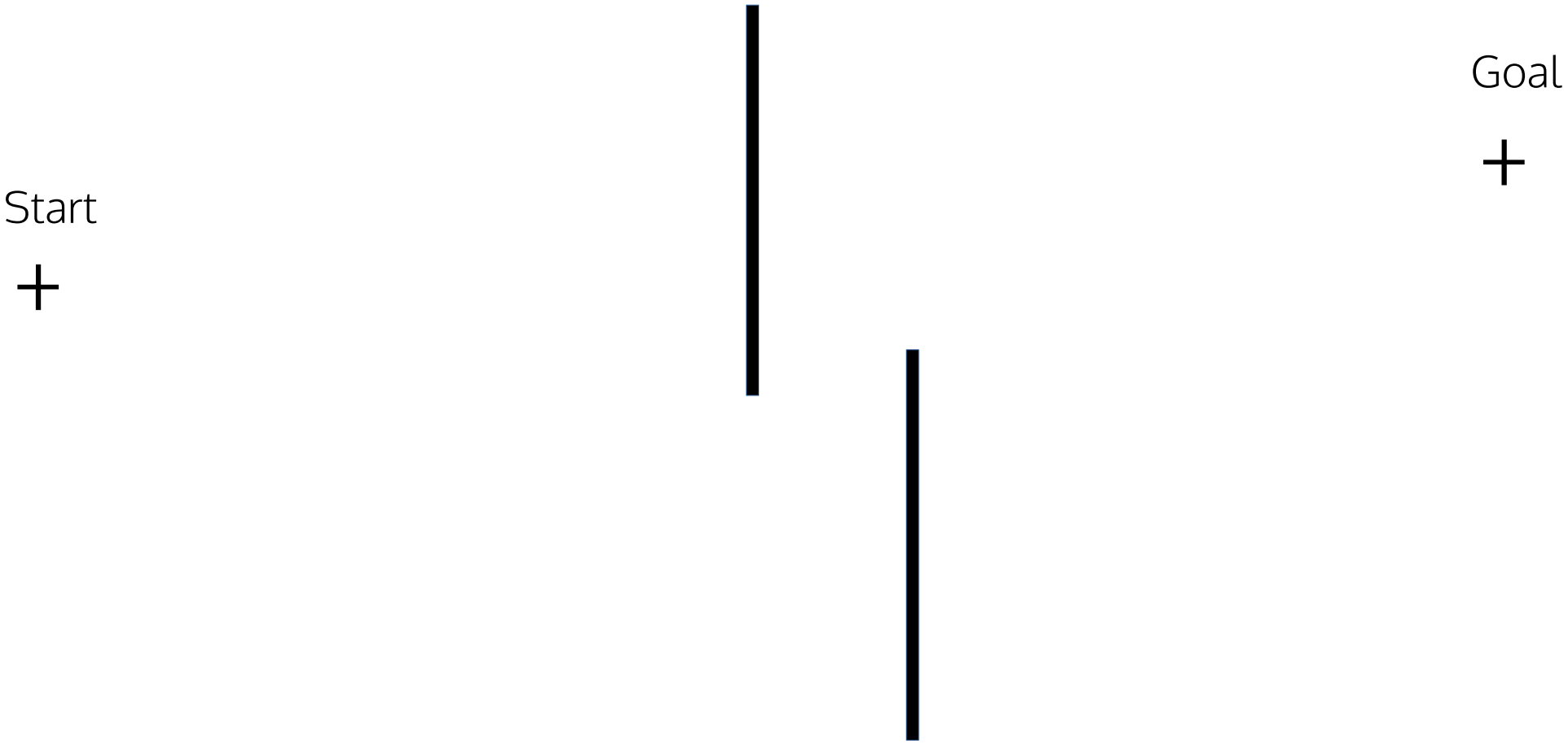
Entrée :

- Start state
- Goal state
- Contraintes

Sortie :

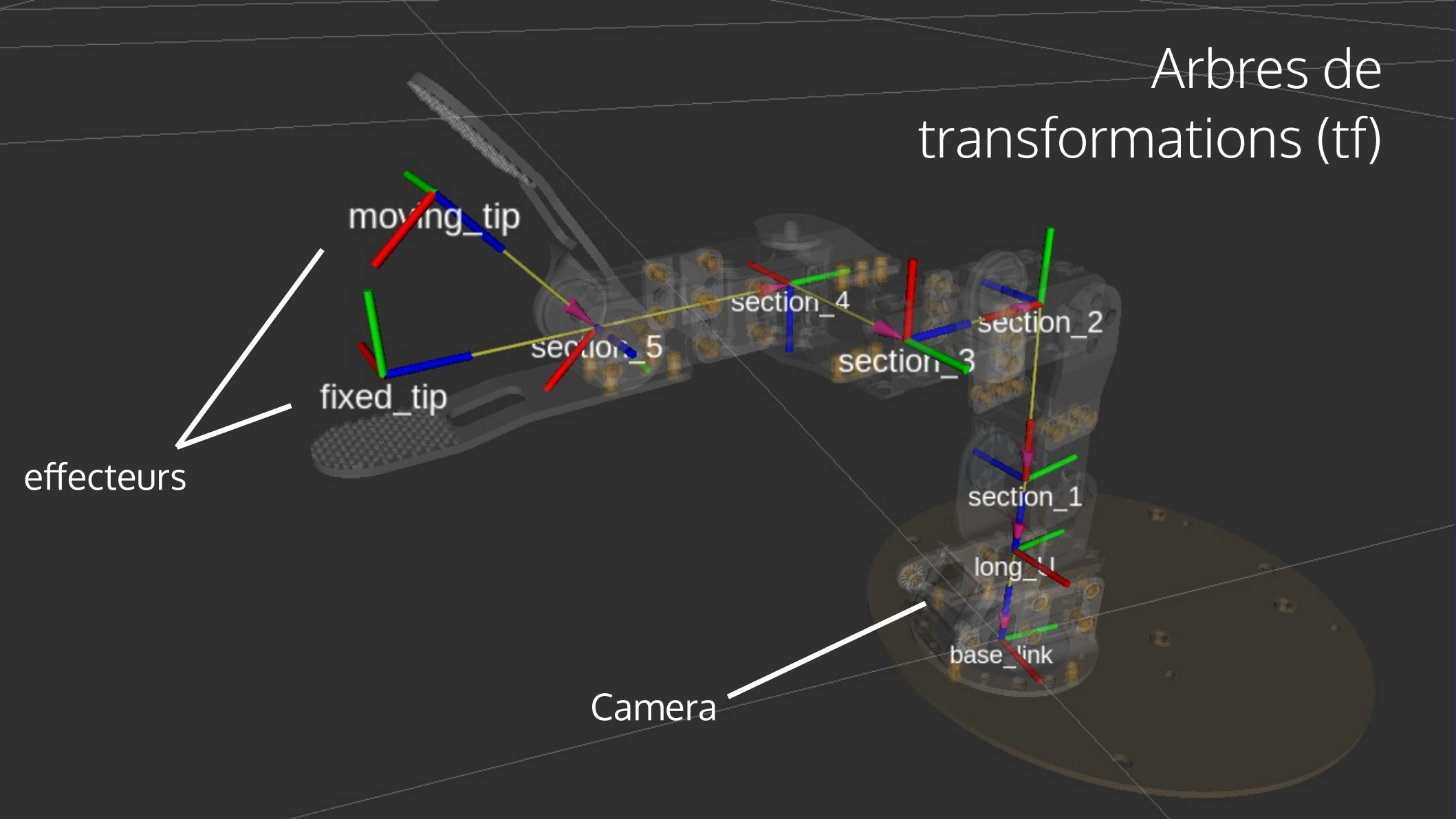
- Trajectoire géométrique dans l'espace des joints

RRTConnect

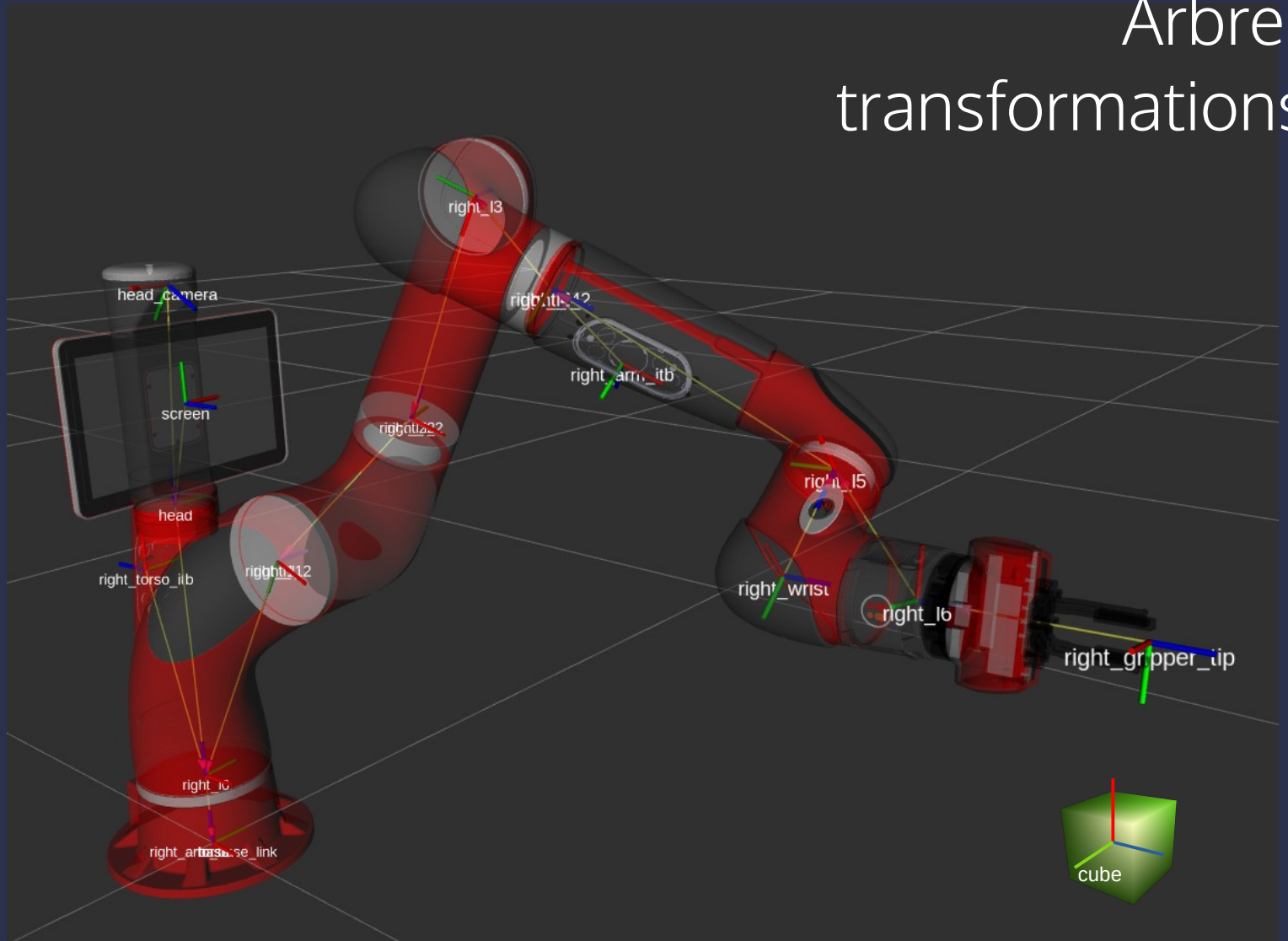


1) Le robot Poppy Ergo Jr
2) Espaces de trajectoires
3) MoveIt & OMPL
4) **Arbres de transformation « tf »**
5) Décomposition d'un mouvement de pick-and-place
6) Aspects pratiques

Arbres de transformations (tf)



Arbres de transformations (tf)



Arbres de transformations (tf)

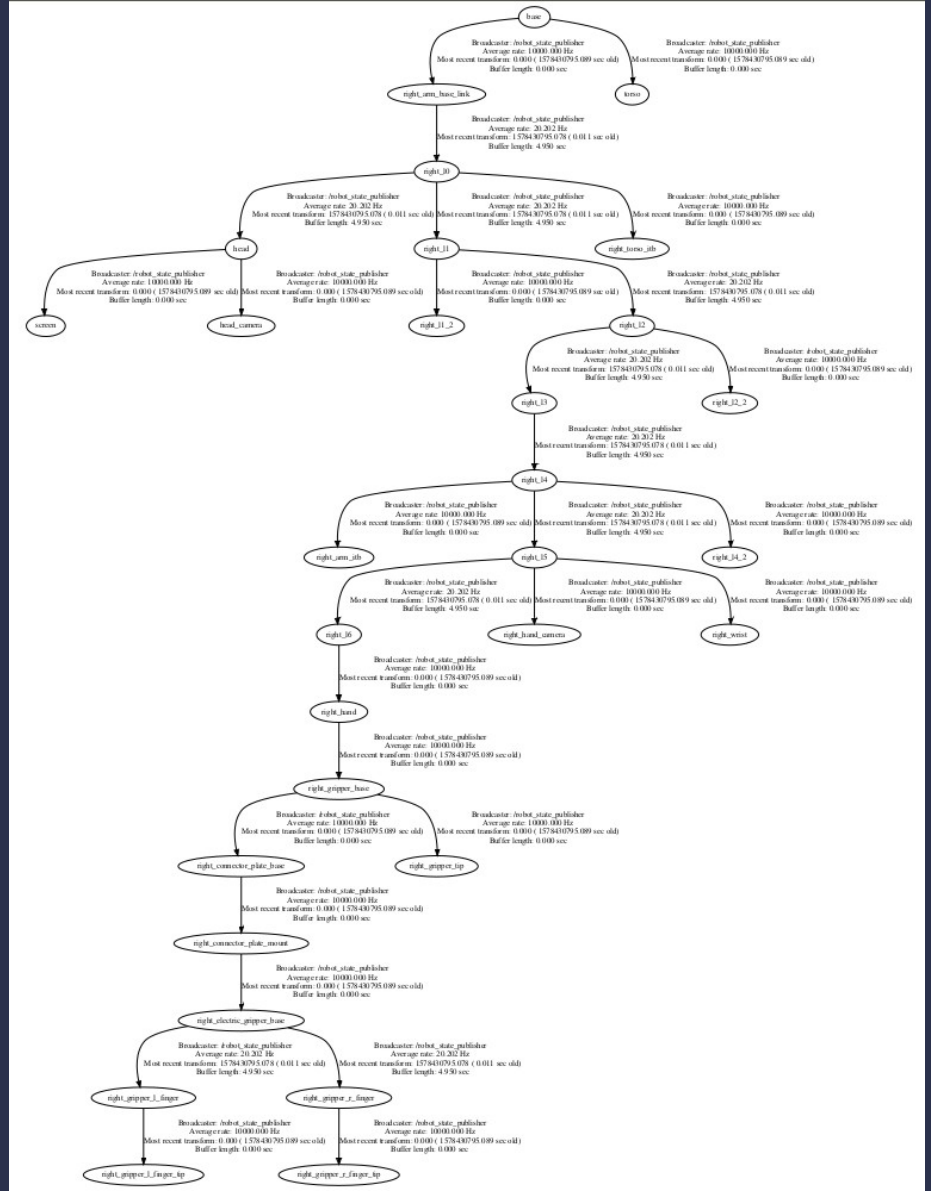
(ROS) rosrun tf :

echo : obtenir la tf actuelle base right_gripper_tip

static_transform_publisher : publier des tf d'objets fixes

view_frames : dessiner les arbres de transformations dans un pdf

view_frames


base
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_arm_base_link
torso
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
right_s0
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
head
right_s1
right_torso_stb
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
screen
head_camera
right_s1_2
right_s2
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_s3
right_s2_2
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
right_s4
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_arm_stb
right_s5
right_s4_2
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_s6
right_hand_camera
right_wrist
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_hand
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_gripper_base
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_connector_plate_base
right_gripper_tip
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_connector_plate_mount
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_electric_gripper_base
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
Broadcaster: /robot_state_publisher
Average rate: 20.202 Hz
Most recent transform: 1578430795.078 (0.011 sec old)
Buffer length: 4.950 sec
right_gripper_l_finger
right_gripper_r_finger
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1578430795.089 sec old)
Buffer length: 0.000 sec
right_gripper_l_finger_tip
right_gripper_r_finger_tip

Arbres de transformations (tf)

(Python) `TransformListener` :

- Quelle est la pose actuelle entre « base » et « right_gripper_tip » ?
- Quelle était cette pose à la date t ?

(Python) `TransformBroadcaster` :

- Publier un nouvel objet

Opérations sur les transformations

Le TransformListener pour obtenir une pose relative entre deux objets dans un code Python :

```
tfl = TransformListener()

pose = tfl.lookupTransform('base', 'right_gripper_tip',
                           rospy.Time(0))

pose : [[x, y, z], [x, y, z, w]]   # [[position], [rotation]]
```

Opérations sur les transformations

Le TransformBroadcaster pour publier (régulièrement) une pose relative entre deux objets dans un code Python :

```
tfb = TransformBroadcaster()

tfb.sendtransform([x, y, z], [x, y, z, w],
                  rospy.Time.now(),‘cube’, ‘base’)
```

Opérations sur les transformations

tf ne peut pas les faire pour nous si l'objet n'est pas publié

e.g. l'objet est une cible à atteindre dans l'espace

${}^{base}P_{gripper}$ = pose du « gripper » dans le repère « base »

Changement de repère : ${}^{base}P_{gripper} = {}^{base}T_{cube} \times {}^{cube}P_{gripper}$

Manipulation

1) Le robot Poppy Ergo Jr
2) Espaces de trajectoires
3) MoveIt & OMPL
4) Arbres de transformation « tf »
5) **Décomposition d'un mouvement de pick-and-place**
6) Aspects pratiques

Mouvement de « Pick »

1) Approche :

   Mouvement sans collision 18cm au dessus du cube

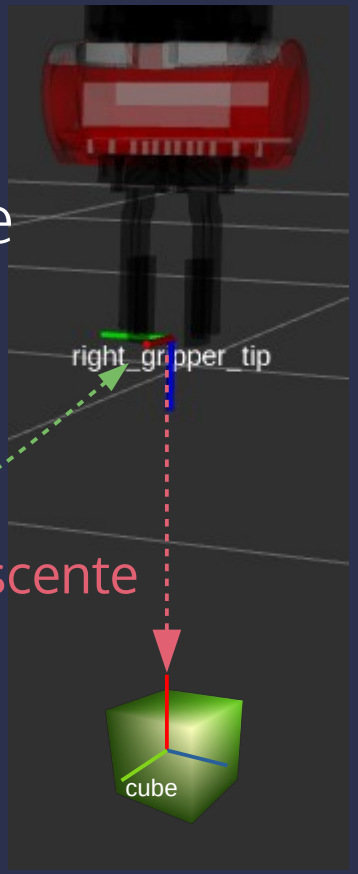
   Cible = une pose cartésienne

2) Descente :

   Mouvement linéaire

   Cible : une trajectoire de 50 poses cartésiennes

   Collision partielle autorisée (« touch links »)



Mouvement de « Pick »

3) Retraite :

Mouvement linéaire

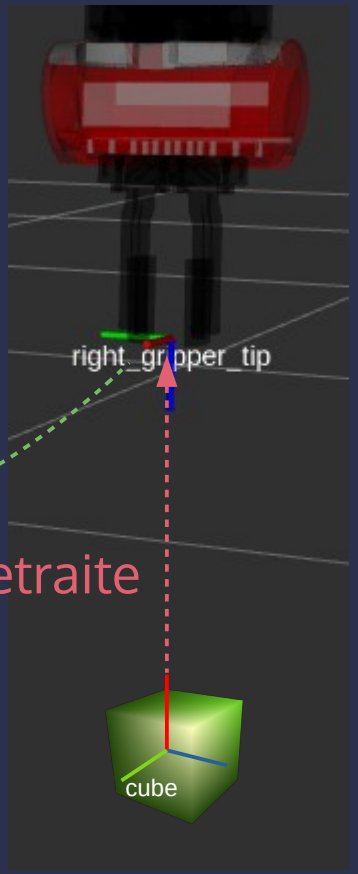
Cible = une trajectoire de 50 poses cartésiennes

4) Retour :

Mouvement sans collision ailleurs

Cible : une pose cartésienne ou autre chose …

Si le pick est un succès, procéder au « place »



Manipulation

1) Le robot Poppy Ergo Je
2) Espaces de trajectoires
3) MoveIt & OMPL
4) Arbres de transformation « tf »
5) Décomposition d'un mouvement de pick-and-place
6) **Aspects pratiques du TP**

Objectifs du TP

- Comprendre la représentation d'un robot en ROS (URDF)
- Découvrir et tester la planification de mouvement (MoveIt)
- Comprendre les groupes cinématiques
- Comprendre le rôle du contrôleur moteur
- Savoir différencier un mouvement dans l'espace cartesien ou moteur
- Découvrir et tester l'évitement d'obstacles
- Tester l'API Python pour MoveIt