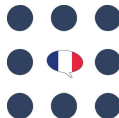


# robot\_arm\_calibration : un outil d'étalonnage cinématique pour les bras robotisés

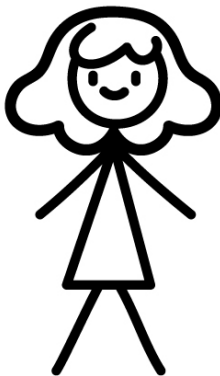
Caroline PASCAL

U2IS & UME – ENSTA Paris

4 Juillet 2023

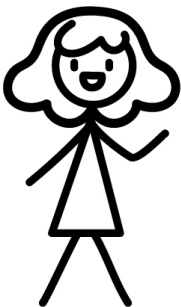


# Introduction

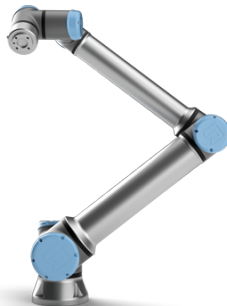


Voici **Billie**.

# Introduction



Billie vient d'acquérir un **bras robotisé**, et est impatiente de tirer profit de ses capacités !



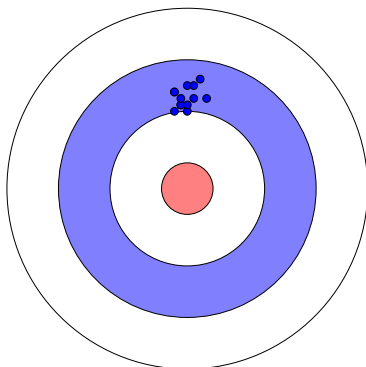
*Universal Robots - UR10e (Datasheet)*

- 6 degrees of freedom;
- 12.5 kg payload;
- 1.3 m reach;
- **0.05 mm precision.**

# Introduction



Ce serait dommage que Billie se rende compte de sa très mauvaise **précision absolue...**



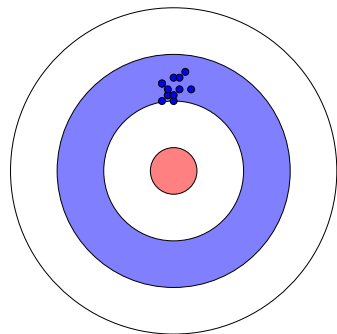
✓ Precise  
✗ Accurate

→ **Centimetric accuracy !**



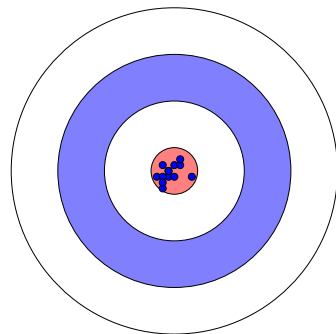
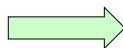
# Introduction

Heureusement pour Billie, il existe une solution à ce problème :  
**l'étalonnage cinématique (*kinematic calibration*) !**



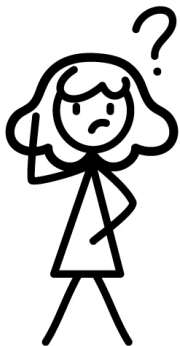
✓ Precise  
✗ Accurate

KINEMATIC  
CALIBRATION



✓ Precise  
✓ Accurate

# Introduction



Cela dit, Billie est encore un peu perdue, et c'est normal :  
l'étalonnage, c'est souvent compliqué.

- Qu'est-ce que l'étalonnage cinématique ? Comment ça fonctionne ?
- Comment l'intégrer sur un bras robotisé ? Quid des interfaces matériel-logiciel-algorithmes ?
- **Comment se fait-il qu'il n'y ait pas déjà d'outil clé en main pour ça ?!**

# Introduction



Cela dit, Billie est encore un peu perdue, et c'est normal :  
*l'étalonnage, c'était souvent compliqué.*

- Qu'est-ce que l'étalonnage cinématique ? Comment ça fonctionne ?
- Comment l'intégrer sur un bras robotisé ? Quid des interfaces matériel-logiciel-algorithmes ?
- **Comment se fait-il qu'il n'y ait pas déjà d'outil clé en main pour ça ?!**

robot arm calibration

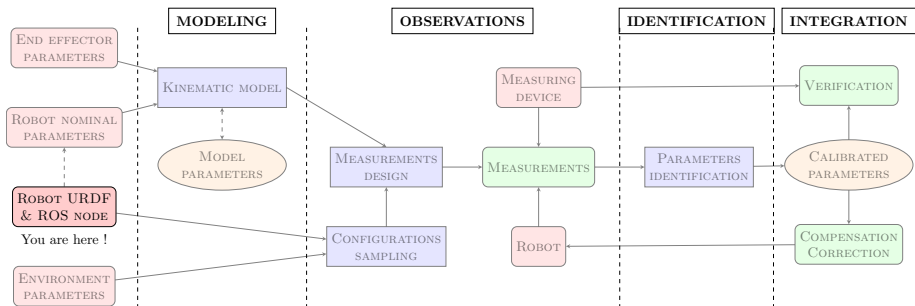
# Sommaire

- 1 Introduction
- 2 Qu'est-ce que l'étalonnage cinématique ?
- 3 Comment gérer les interfaces entre matériel et logiciel ?
- 4 robot\_arm\_calibration, de la théorie à la pratique
- 5 Conclusion & perspectives

# Sommaire

- 1 Introduction
- 2 Qu'est-ce que l'étalonnage cinématique ?**
- 3 Comment gérer les interfaces entre matériel et logiciel ?
- 4 robot\_arm\_calibration, de la théorie à la pratique
- 5 Conclusion & perspectives

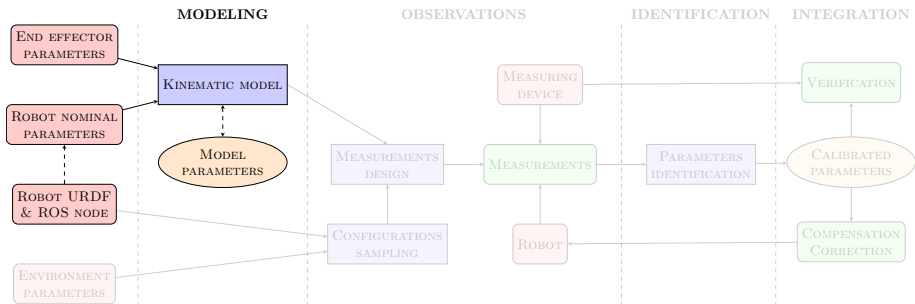
# Les 4 étapes de l'étalonnage cinématique



Description de la procédure d'étalonnage cinématique

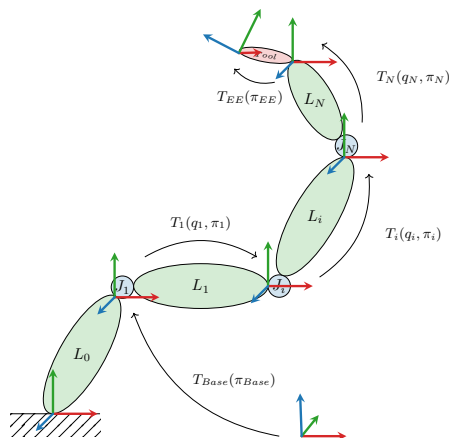
⇒ Ces 4 étapes sont **intégralement implémentées** (python & C++) dans le package `robot_arm_calibration`.

# Etape 1 : Modélisation



Description de la procédure d'étalonnage cinématique - Modélisation

# Etape 1 : Modélisation



## Propriétés du modèle

- **Fidélité;**
  - ↪ Conforme au comportement réel du robot;
- **Completude, sans redondance;**
  - ↪ Modèle défini par un ensemble suffisant, mais irréductible de paramètres;
- **Continuité;**
  - ↪ Fonction continue des paramètres;



# Etape 1 : Modélisation

## Modélisation géométrique complète (*full-pose*)

$$T(q, \pi) = T_{Base}(\pi_{Base}) \cdot [T_{Joint_1}(q_1, \pi_{J_1}) \cdot T_{Link_1}(\pi_{L_1})] \\ \cdot [T_{Joint_N}(q_N, \pi_{q_N}) \cdot T_{Link_N}(\pi_{L_N})] \cdot T_{EE}(\pi_{EE})$$

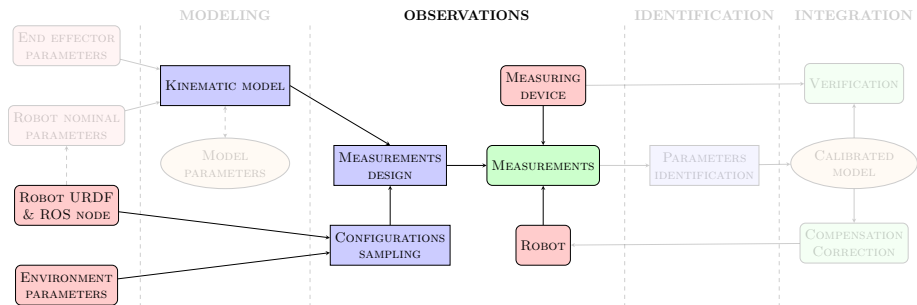
Où  $\pi = (\pi_{Base}, \pi_{J_i}, \pi_{L_i}, \pi_{EE})$  contient les *paramètres géométriques du modèle*.

## Modélisation géométrique partielle (*partial-pose*) [1]

$$\left( P^i(q, \pi) \right)_{i=1 \dots M} = T_{Base}(\pi_{Base}) \cdot [T_{Joint_1}(q_1, \pi_{J_1}) \cdot T_{Link_1}(\pi_{L_1})] \\ \cdot [T_{Joint_N}(q_N, \pi_{q_N}) \cdot T_{Link_N}(\pi_{L_N})] \cdot T_{EE_i}(\pi_{EE_i})$$

Où  $EE_i$  décrit le point  $i \in \{1 \dots M\}$  de l'organe terminal.

# Etape 2 : Observations



Description de la procédure d'étalonnage cinématique - Observations

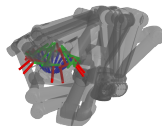
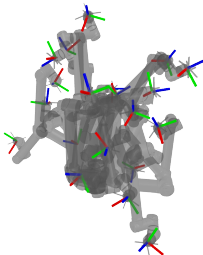
## Etape 2 : Observations

→ **Observations en chaîne ouverte**, avec un suivi interne (encodeurs des moteurs) et des mesures externes (pose de l'organe terminal).

↔ *Comment choisir les configurations du robot mesurées ?*

1) Réaliser un échantillonnage des configurations **atteignables** par le robot, en s'inspirant de la **tâche visée**;

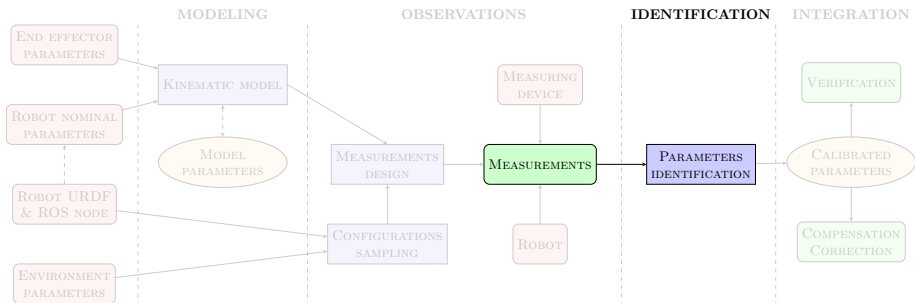
Echantillonnage  
aléatoire



Echantillonnage  
inspiré scan 3D

2) Choisir les configurations maximisant l'**identifiabilité des paramètres**.

# Etape 3 : Identification



Description de la procédure d'étalonnage cinématique - Identification

## Etape 3 : Identification

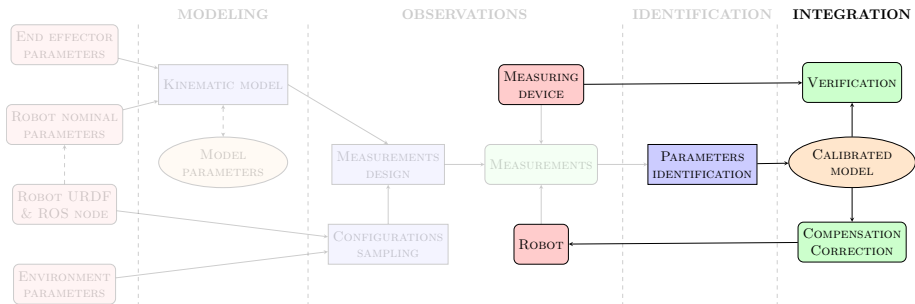
### Identification du modèle géométrique partiel (*partial-pose*) [1]

$$\pi^* = \arg \min_{\pi} \sum_{i=1}^{N_{\text{meas.}}} \sum_{j=1}^{N_{EE}} \underbrace{\left\| P^j(q_i, \pi) - P_{\text{measured}}^j(q_i) \right\|^2}_{\epsilon_i^j}$$

Où  $\epsilon_i^j$  définit l'*erreur de position* du point  $j$  de l'organe terminal lors de la mesure  $i$ .

→ La somme des **erreurs de position** sur toutes les mesures réalisées définit la **précision absolue du robot** (ou *accuracy*).

# Etape 3 : Intégration



Description de la procédure d'étalonnage cinématique - Intégration

# Etape 4 : Intégration

## Vérification

→ **Validation des paramètres étalonnés** grâce aux résultats obtenus sur une *nouvelle série de mesures*, aléatoires ou non.

## Intégration

### → Compensation

↔ Intégration des paramètres étalonnés dans la boucle de contrôle du robot (*plugin logiciel*).

### → Correction

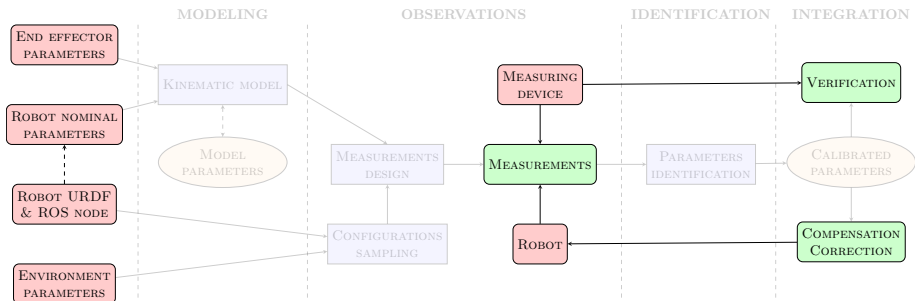
↔ Création d'une nouvelle description du robot à partir des paramètres étalonnés (*fichier de configuration*).

# Sommaire

- 1 Introduction
- 2 Qu'est-ce que l'étalonnage cinématique ?
- 3 Comment gérer les interfaces entre matériel et logiciel ?**
- 4 robot\_arm\_calibration, de la théorie à la pratique
- 5 Conclusion & perspectives

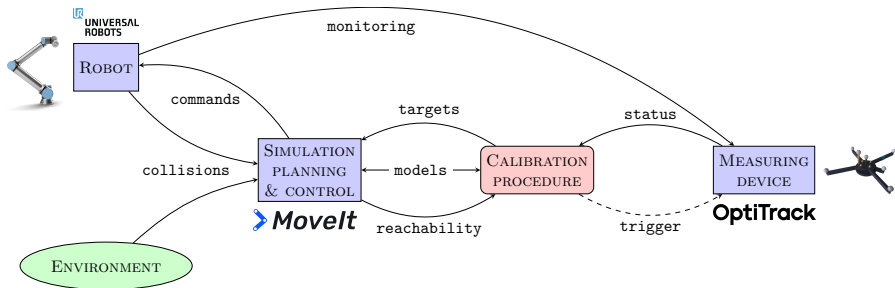


# Interfaces matériel/logiciel - En théorie



Description de la procédure d'étalonnage cinématique - Interfaces matériel/logiciel

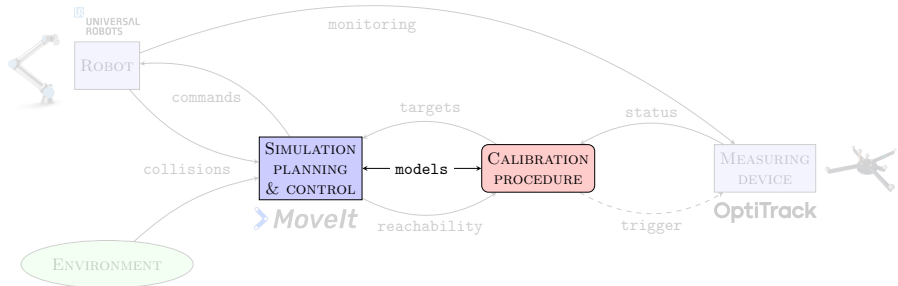
# Interfaces matériel/logiciel - En pratique



Description des interfaces matériel/logiciel

⇒ Chaque interface est **complètement intégrée** et traitée de manière **générique** dans le package `robot_arm_calibration`.

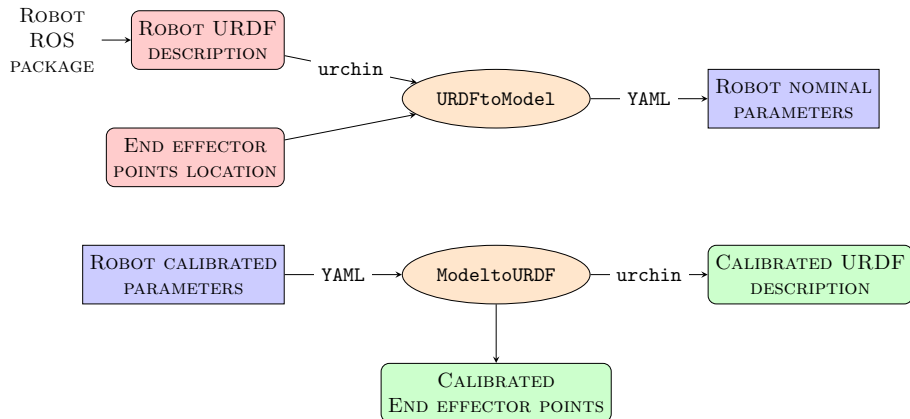
# Intégration du modèle cinématique



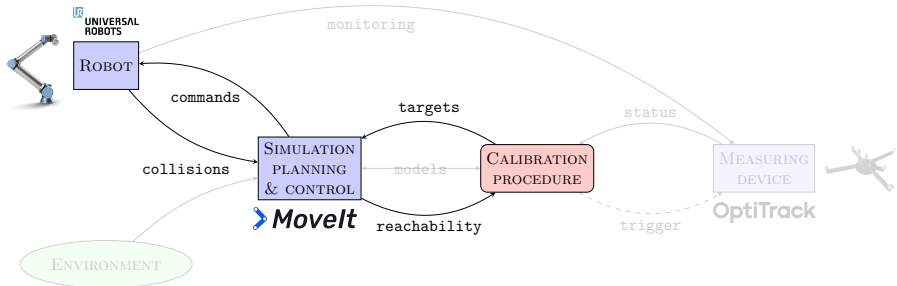
Description des interfaces matériel/logiciel - Modèle cinématique

# Intégration du modèle cinématique

→ Traduction et génération automatisées des descriptions URDF via des fichiers YAML standardisés.



# Intégration du robot



Description des interfaces matériel/logiciel - Robot

# Intégration du robot

- API du framework de planification ➤ **Movel**
- + surcouche personnalisée : `robot_arm_tools`

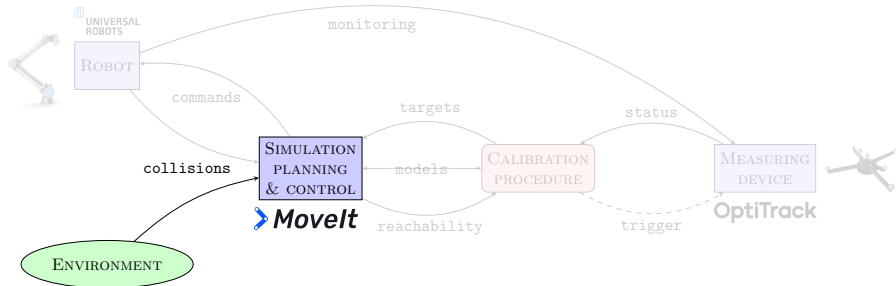
## Simulation

- Plannification de mouvement avec évitement des *collisions* et *singularités*;
- Pipelines de planification et de cinématique *modulaires* et *génériques*;
- Définition et intégration simplifiées de cas d'usages communs.

## Robot réel

- Intégration *ergonomique* et *haut-niveau* des contrôleurs ROS;
- Interruption et récupération dynamique de l'exécution des mouvements;
- Création de *logs* pour suivre et reprendre l'exécution de trajectoires multi-points.

# Intégration de l'environnement extérieur

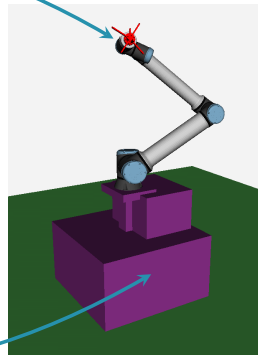
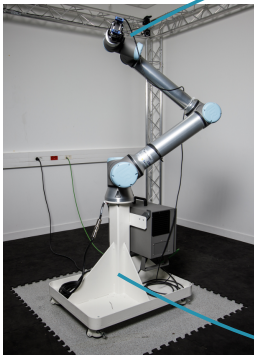


Description des interfaces matériel/logiciel - Environnement extérieur

# Intégration de l'environnement extérieur

Génération automatisée  
d'un fichier URDF/XACRO  
depuis un maillage STL.

↪ *Modification de  
/robot\_description*

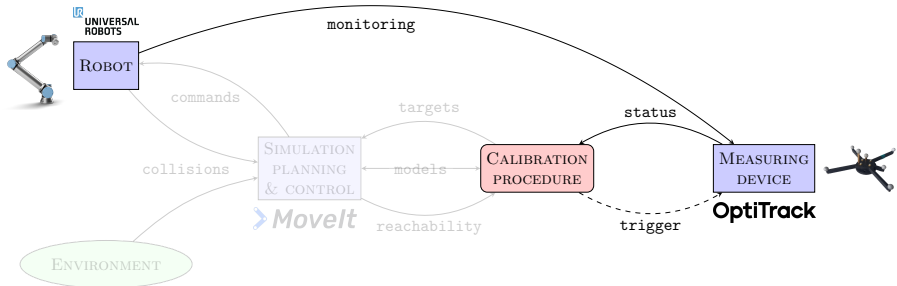


Fichier YAML  
contenant des  
primitives géométriques.

↪ *Encapsulation de  
moveit\_visual\_tools*



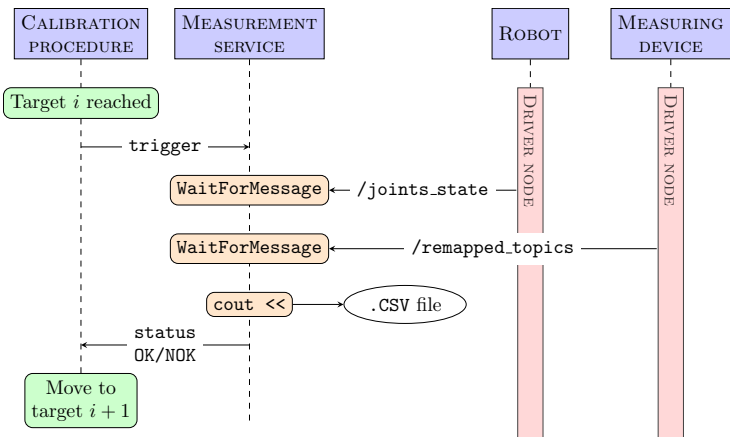
# Intégration des appareils de mesure externes



Description des interfaces matériel/logiciel - Appareils de mesure externes

# Intégration des appareils de mesure externes

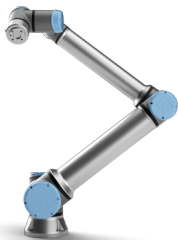
→ Création d'un **système client-serveur générique (ROS service)** pour la synchronisation des mouvements et des mesures.



# Sommaire

- 1 Introduction
- 2 Qu'est-ce que l'étalonnage cinématique ?
- 3 Comment gérer les interfaces entre matériel et logiciel ?
- 4 **robot\_arm\_calibration**, de la théorie à la pratique
- 5 Conclusion & perspectives

# Validation expérimentale - Robots



*Universal Robots - UR10e*

- 6 axes;
- 18 paramètres géométriques;
- Packages ROS:
  - `universal_robots`
  - `ur_robot_driver`



*Franka Emika - Panda (FR3)*

- 7 axes;
- 22 paramètres géométriques;
- Packages ROS:
  - `panda_moveit_config`
  - `franka_ros`

# Validation expérimentale - Appareil de mesure externe

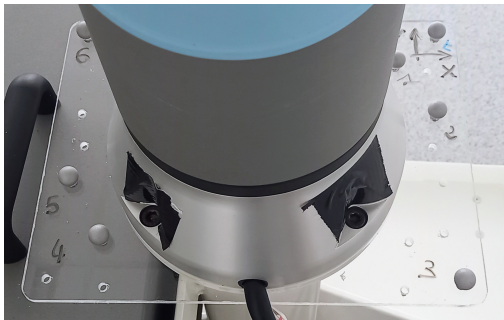
→ Outil de *tracking* et de mesure de position **OptiTrack**



↪ 6 caméras *Prime 13*  
⇒ Précision absolue de  
 $\pm 0.2 \text{ mm}$ .

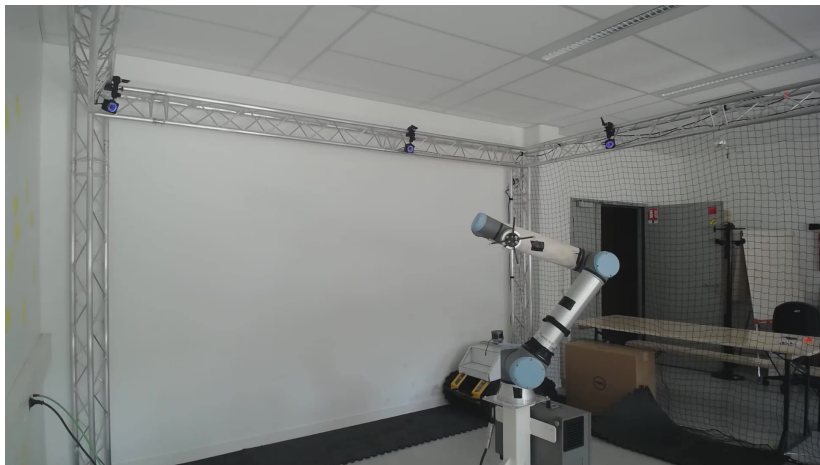
↪ Package ROS :  
`mocap_optitrack`

# Validation expérimentale - Corps d'épreuve



→ Corps d'épreuve associés à la **base** et à l'**organe terminal** du robot - 7 sphères réfléchissantes fixées sur une structure 3D asymétrique.

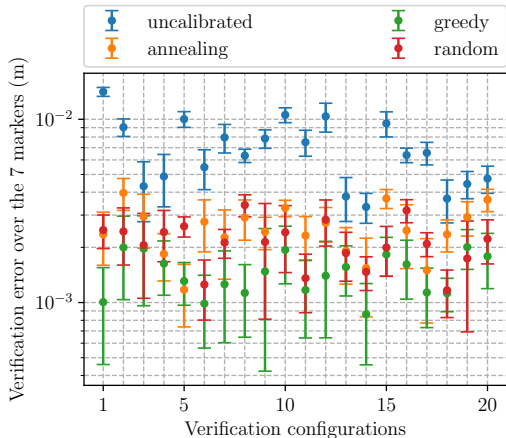
# Validation expérimentale



Exemple de mesures d'étalonnage sur un UR10e (vitesse  $\times 2$ )

# Validation expérimentale - Résultats (Panda)

	F.E. Panda
<b>Paramètres (total)</b>	49
Configurations mesurées	98
<b>Durée totale (h)</b>	4.38
Modélisation (min)	55
Mesures (h)	3.1
Identification (min)	22
Accuracy initiale (mm)	7.66
Accuracy finale (mm)	1.63
<b>Taux d'amélioration</b>	<b>78.7%</b>

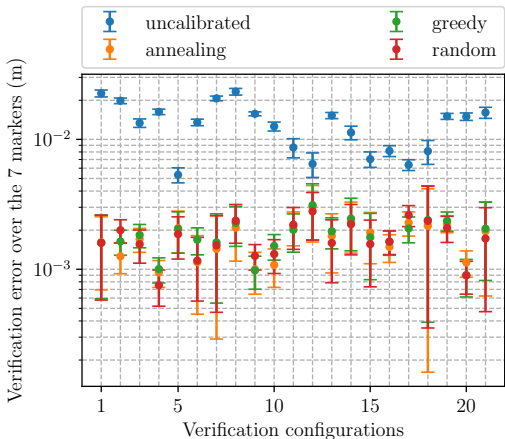


Erreur en position moyenne sur les 7 sphères, avec et sans étalonnage



# Validation expérimentale - Résultats (UR10e)

UR10e	
<b>Paramètres (total)</b>	45
Configurations mesurées	90
<b>Durée totale (h)</b>	<b>7.03</b>
Modélisation (min)	45
Mesures (h)	6.0
Identification (min)	17
<i>Accuracy</i> initiale (mm)	14.4
<i>Accuracy</i> finale (mm)	2.03
<b>Taux d'amélioration</b>	<b>85.9%</b>



Erreur en position moyenne sur les 7 sphères, avec et sans étalonnage

# Sommaire

- 1 Introduction
- 2 Qu'est-ce que l'étalonnage cinématique ?
- 3 Comment gérer les interfaces entre matériel et logiciel ?
- 4 robot\_arm\_calibration, de la théorie à la pratique
- 5 Conclusion & perspectives

# Conclusion

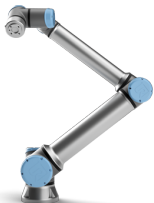
## Comment résumer robot\_arm\_calibration en 3 points ?

- Une **procédure d'étalonnage cinématique complète** pour les bras robotisés (modélisation, observations, identification et intégration);
- Un **package ROS**, offrant des interfaces matériel-logiciel génériques et ergonomiques, s'adaptant à différents robots et outils de mesure;
- Un **projet libre d'accès**, dont l'ambition est de permettre de futures améliorations algorithmiques et personnalisations.

# Et pour la suite ?

- **Migrer le projet vers ROS 2 !**
  - ↔ Passer de ROS 1 *Noetic* à la dernière version en date de ROS 2.
- **Introduire la souplesse des moteurs dans le modèle cinématique**
  - ↔ Prendre en compte les effets de la gravité sur les moteurs du robot [2], dans un *plugin* dédié (compensation).
- **Intégrer les mesures de pose complètes et indirectes**
- **Augmenter la robustesse des mesures**
  - ↔ Eviter les obstructions causées par le robot lors du choix des configurations mesurées grâce à des contraintes de planification.

# Merci pour votre attention !



[gitlab.ensta-paris.fr/caroline.pascal.2020/robot\\_arm\\_tools](https://gitlab.ensta-paris.fr/caroline.pascal.2020/robot_arm_tools)

[gitlab.ensta-paris.fr/caroline.pascal.2020/robot\\_arm\\_calibration](https://gitlab.ensta-paris.fr/caroline.pascal.2020/robot_arm_calibration)

C. Pascal, A. Chapoutot, and O. Doaré, "A ROS-based kinematic calibration tool for serial robots (accepted paper)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023



Y. Wu, A. Klimchik, S. Caro, B. Furet, and A. Pashkevich, “Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments,” *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 151–168, 2015.



A. Klimchik, Y. Wu, S. Caro, B. Furet, and A. Pashkevich, “Geometric and elastostatic calibration of robotic manipulator using partial pose measurements,” *Advanced Robotics*, vol. 28, 2014.



C. Pascal, A. Chapoutot, and O. Doaré, “A ROS-based kinematic calibration tool for serial robots (accepted paper),” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.