

# Robot pipeline audit (20.03.2026)

## Introduction

Ce document contient les observations de mon audit du pipeline ur3e-control, réalisé à la demande de Cédric Mariéthoz.

## Audit scope

En m'appuyant sur mon regard neuf de personne n'ayant pas encore travaillé dans la team robot, déterminer, par exemple, s'il manque des fallbacks, si la logique est cohérente ou s'il y a des erreurs

## Review

### DataStore (logger.py)

- `log_tcp_segment` est défini deux fois dans la classe.
- Les méthodes `load_****` retournent `(None, None)` si le fichier est absent, mais une valeur unique si le fichier existe. Pourquoi deux fois `None, None` ?
- `load_trace_segment` affiche "Joint segments data file not found" au lieu probablement de "Trace segments data file not found".

### Calibration (calibration.py)

- `get_tcp_offset` contient une valeur TCP anciennement calibrée codée en dur (`t = np.array([1.30905845e-04, ...])`). Elle n'est probablement plus à jour et du moins il faudrait la mettre dans un fichier de config. afin de ne pas avoir de valeurs hardcodées directement dans le code.
- Lors des opérations de calibration (`spread`, `motion`, `rotation`) (l. 182, 186, 189) il est commenté "(should be small), (should be large)". Ces commentaires laissent penser que l'on ne souhaite qu'une certaine range de valeur mais rien ne bloque la sauvegarde si les seuils ne sont pas satisfaisants (D'ailleurs, qu'est-ce que le "satisfaisant ?").

### Transformation (transformation.py)

- Dans `Transformation.run()` si l'utilisateur répond "n" aux deux questions, le programme rentre en boucle infinie.
- Faute de frappe dans le log: "Transforamtion skipped" (l. 210).

### Filter (filter.py)

- `TraceSegment` est instancié avec les arguments dans un ordre différent que celui attendu : `TraceSegment(waypoints, color, SideType.LEFT)`, alors que le constructeur attend `(color, side, waypoints=None)`.

### General

- Avoir des docstrings d'explication dans toutes les méthodes aiderait à la compréhension de ces dernières.
- Rajouter au maximum les types
- Ajouter une classe (interface) parente (par exemple : `Stage`) pour définir les signatures des méthodes et améliorer les indications de type des paramètres et des variables

Jeremy Duc