

ML Shukai

Weekly P-AMI<Q>

2025/04/02 GesonAnko

今週の進捗 Summary

- **P-AMI<Q> Core プロジェクト**

自律機械知能 P-AMI<Q>のコアシステムのライブラリ化

- システム (スレッド) 制御系モジュール完成 with アイシアさん!
- PyTorch のモデル統合機能を実装 by zassouさん

- **P-AMI<Q> ROS2**

P-AMI<Q> Core を ROS2 (Robot Operating System 2) と統合するライブラリ

- 入出力処理 (Pub / Sub)の部分は統合完了

システム（スレッド）制御系モジュール完成

- P-AMI<Q>の並列処理の制御モジュール
 - `thread_control.py`: 332行！

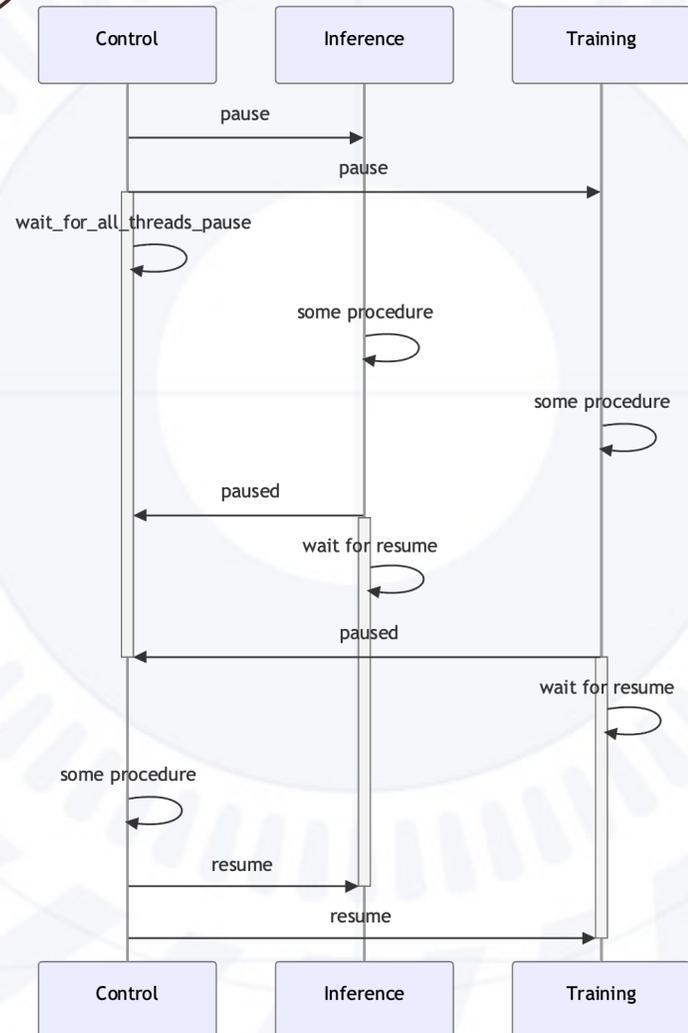
機能

並行して動く処理に対して…

- 一時停止 → 停止確認
- 再開処理
- 例外通知 → システム終了

感想

- 🙄 安全に制御処理の実装は本当に大変だったよ！
- 🥰 アイシアさんとめっちゃ議論！ありがたい！



PyTorch のモデル統合機能を実装

• TorchTrainingModel & TorchInferenceModel

- P-AMI<Q> Core のモデルプロトコルのアダプター
- 学習用モデルと推論用モデルクラスを定義
- ラッパー形式になっているので、既存のPyTorchモデルを使用可能
- 自動的な演算デバイス転送機能
- 学習モデルと推論モデルの同期処理実装
- などなど…

• 感想とか

- 🥲 色々必要な機能が多くて大変だったね。
- 🤨 zassouさんと意見が衝突した部分は Claudeに調停してもらう。

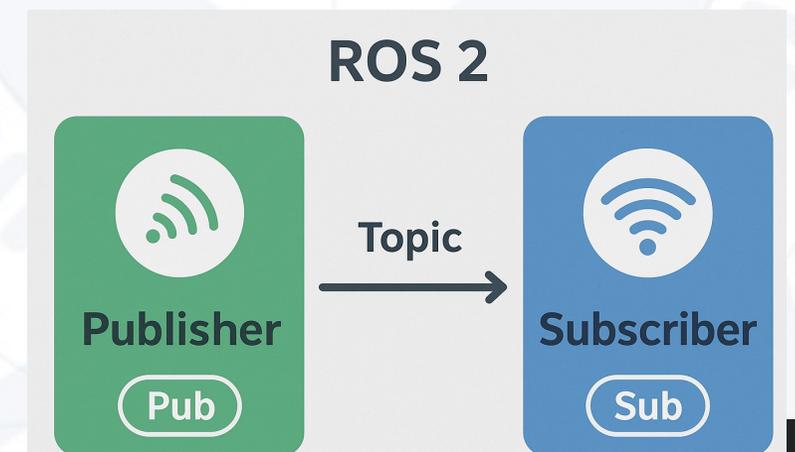
このメソッドの名前は `get_device` と `seek_device` , どちらが良いか？

Analyzed method naming conventions for device retrieval function.

`get_device` の方が適切だと思います。理由は以下の通りです：

P-AMI<Q> の ROS2統合

- ROS2
 - 分散型のロボットソフトウェアシステムを作成するためのフレームワーク
- 統合をするために
 - 分散システムのためのメッセージングシステムを取り込む
 - メッセージングシステムの基本概念: “Pub/Sub通信”
 - Publisher: データを送信する
 - Subscription: データを待ち受けて処理する。
 - Topic: データの名前



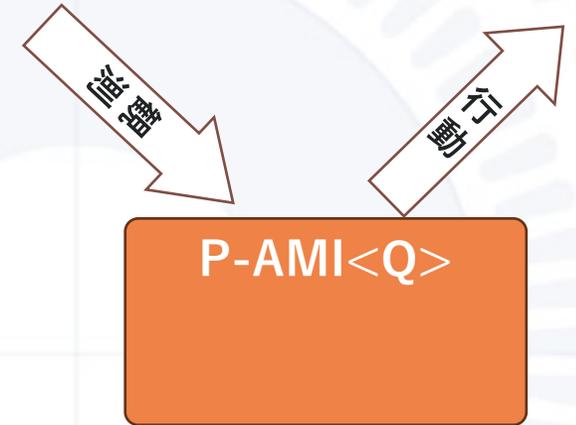
P-AMI<Q> の ROS2統合

- P-AMI<Q> Coreへの統合

- Subscription: 観測を受け取る部分
- Publisher: 行動を送信する部分。 適切なTopic名で公開

- 🥵 難しいところ

- 観測が来るまで待ち受けなければならない。
→ パッシブな処理が前提
- しかし、 行動は一定間隔で行いたい…
→ 処理間隔が保証されたアクティブな処理



P-AMI<Q> の ROS2統合

- パッシブ受信・アクティブ送信の折衷案

- 非同期に観測を受け取り、保持。
- 新規の観測がある
 - すぐに処理して行動を返す。
- 新規の観測がない
 - 観測が来るまで“T”秒まで待つ。

🧐 *threading.Condition*

- 同期プリミティブ“条件変数”を使用

- Producer-Consumerパターンの同期処理で使う。
 - Producer: 観測の受信
 - Consumer: 観測を処理し行動を送信

