



DWPP: Dynamic Windowを用いた 速度・加速度制約を考慮したPure Pursuitの提案と Nav2プラグインの実装公開

2025年9月9日

慶應義塾大学大学院 理工学研究科

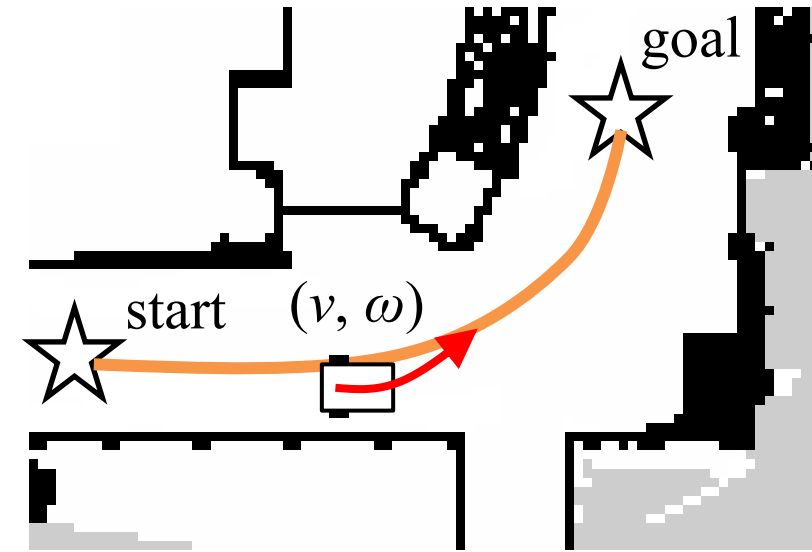
高橋正樹研究室

○大西 史弥、高橋 正樹

- Dynamic Window Pure Pursuit (DWPP) の紹介
 - Pure Pursuitとは
 - 概要
 - 使用例
 - 課題
 - Dynamic Window Pure Pursuit (DWPP)
 - 概要
 - メリット
 - 得意なシナリオ
- 導入方法
 - Nav2への導入
 - テスト環境の導入
- デモンストレーション
- まとめ

- Dynamic Window Pure Pursuit (DWPP) の紹介
 - Pure Pursuitとは
 - 概要
 - 使用例
 - 課題
 - Dynamic Window Pure Pursuit (DWPP)
 - 概要
 - メリット
 - 得意なシナリオ
- 導入方法
 - Nav2への導入
 - テスト環境の導入
- デモンストレーション
- まとめ

- 自律移動ロボットのナビゲーションの仕組み
 - グローバルプランナー
 - スタートとゴールを結ぶグローバルな経路のプランナー
 - ローカルプランナー
 - 経路追従や障害物回避するローカルな経路のプランナー
 - ローカルな経路へ追従するための速度指令値も算出

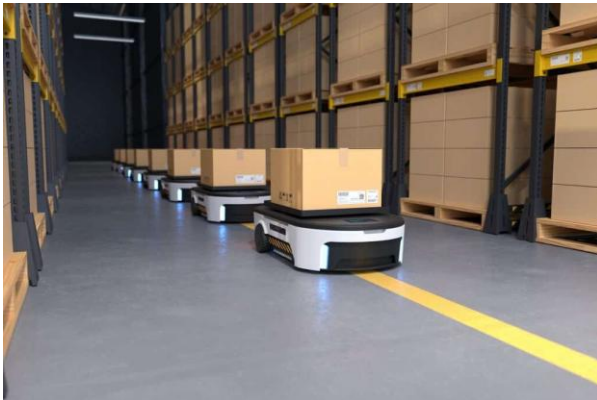


- Pure Pursuitとは
 - 経路追従のみ行うローカルプランナー

名前	ベースのアルゴリズム	機能
DWB Controller	Dynamic Window Approach	経路追従+障害物回避
TEB Controller	Model Predictive Control	経路追従+障害物回避
MPPI Controller	Model Predictive Path Integral	経路追従+障害物回避
Regulated Pure Pursuit	Pure Pursuit	経路追従のみ

Nav2で使用可能な主なローカルプランナー一覧

■ 設定した経路上を走行

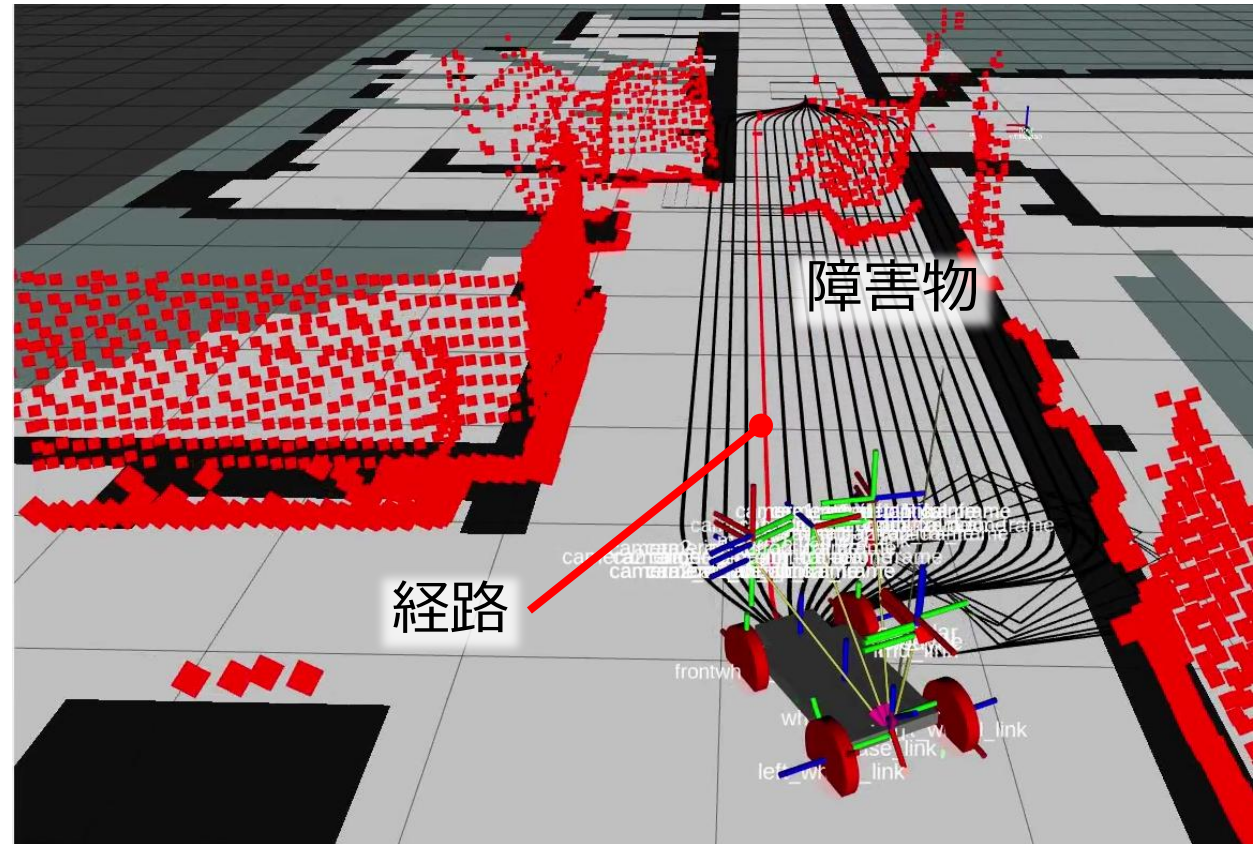


AGV (無人搬送車)



自動運転サービス^[1]

■ 障害物回避するようにリアルタイムに更新される経路上を走行



搬送ロボット (歩行者回避機能付き)

[1] 「WHILL自動運転サービス」 空港での利用方法, <https://youtu.be/5tXsCfeWbso?si=H6xT8fXvoyT5kSaW>

Pure Pursuit (PP)
[Coulter, 1992]

Adaptive
Pure Pursuit (APP)
[Campbell, 2007]

Regulated
Pure Pursuit (RPP)
[Macenski et. al., 2023]



— 速度指令値 — 実現速度

ローカルな経路の計画

前方注視点 p_L の算出

前方注視距離 $L = \text{定数}$

$L_t = g_L v_t$ (並進速度に比例, APP)

前方注視点へ至る (円弧) 経路曲率 ϕ の算出

速度指令値算出

制限並進速度 v_{reg} 算出 (RPP)

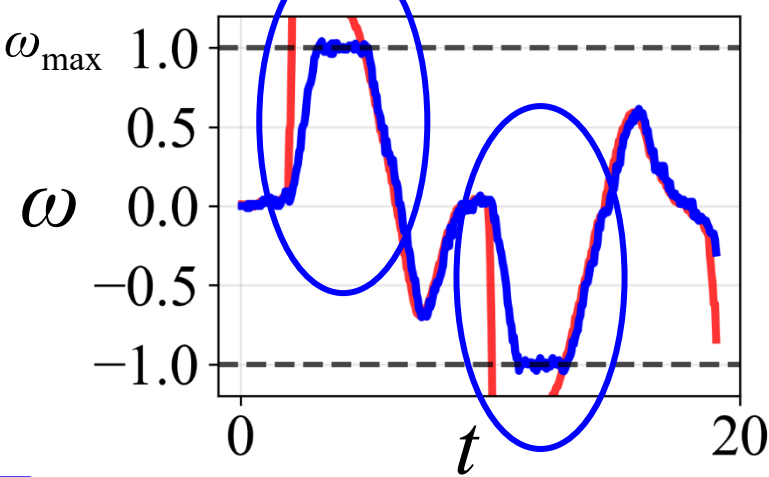
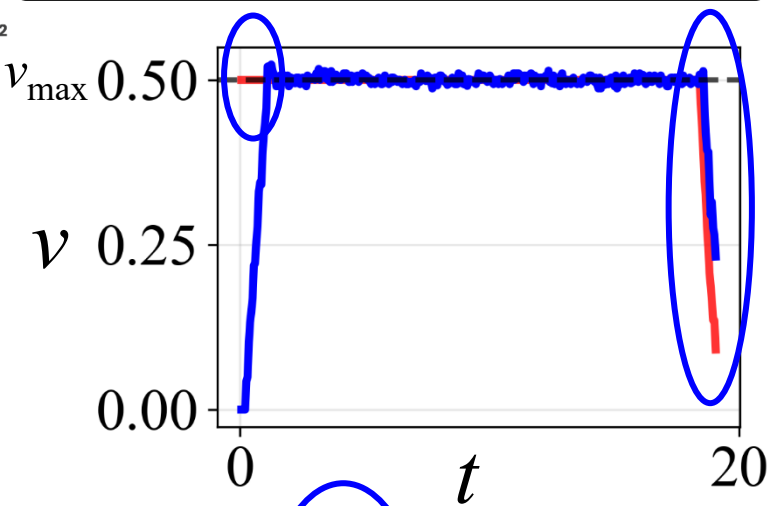
並進速度 v_{ref} の算出 (最高速度で一定, ゴール付近で減速)

v_{ref} を v_{reg} で制限 (RPP)

回転速度 ω_{ref} の算出 $\omega_{ref} = \phi v_{ref}$ (曲率 ϕ の経路を追従する式)

速度指令値
(v_{ref}, ω_{ref})

速度指令値算出時に**速度・加速度制約**を未考慮



PPの速度プロファイル例

Pure Pursuit (PP)
[Coulter, 1992]

ローカルな経路の計画

- 前方注視距離 $L = \text{定数}$
- 前方注視点



速度指令値算出

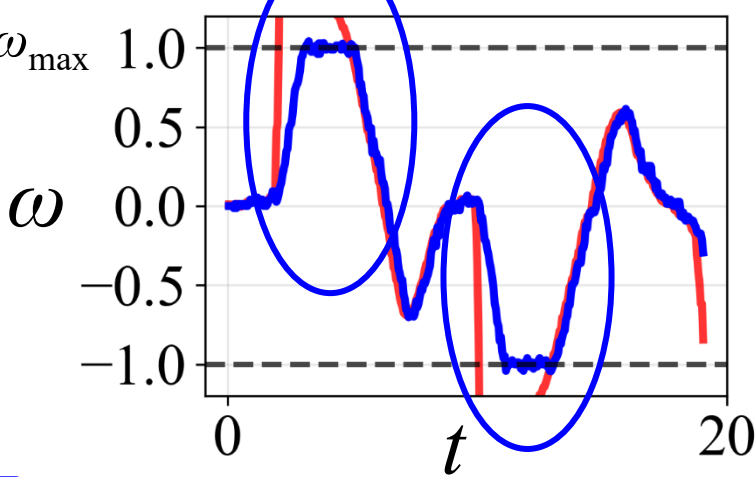
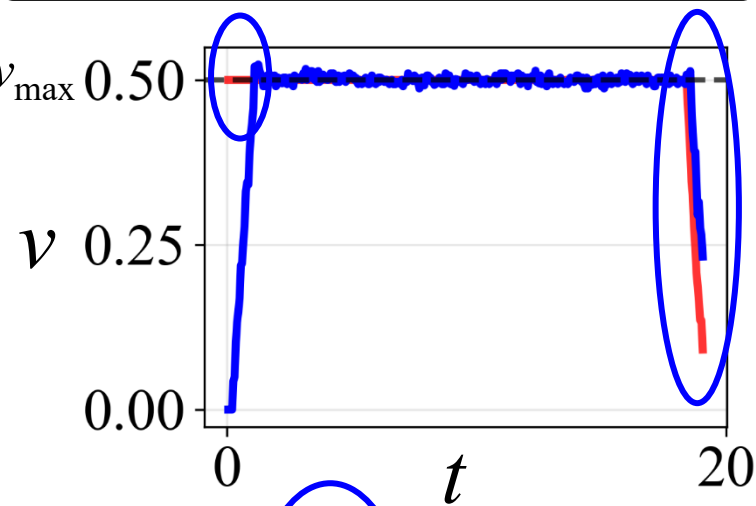
- 並進速度 v_{ref} の算出 (最高速度で一定, ゴール付近で減速)
- v_{ref} を v_{reg} で制限 (RPP)
- 回転速度 ω_{ref} の算出 $\omega_{\text{ref}} = \varphi v_{\text{ref}}$ (曲率 φ の経路を追従する式)

速度指令値
($v_{\text{ref}}, \omega_{\text{ref}}$)

速度指令値算出時に**速度・加速度制約**を未考慮

PP) NAV 2 2023]

— 速度指令値 — 実現速度



PPの速度プロファイル例

Pure Pursuit (PP) [Coulter, 1992]	Adaptive Pure Pursuit (APP) [Campbell, 2007]	Regulated Pure Pursuit (RPP) [Macenski et. al., 2023]	 Dynamic Window Pure Pursuit (DWPP) [Ohnishi et. al., 2025]
--------------------------------------	--	---	---

ローカルな経路の計画

前方注視点 p_L の算出

前方注視距離 $L = \text{定数}$	$L_t = g_L v_t$ (並進速度に比例, APP)
------------------------	--------------------------------

前方注視点へ至る (円弧) 経路曲率 ϕ の算出

速度指令値算出

制限並進速度 v_{reg} 算出 (RPP)

並進速度 v_{ref} の算出 (最高速度で一定, ゴール付近で減速)

Dynamic Windowの
算出 (DWPP)

v_{ref} を v_{reg} で制限 (RPP)

回転速度 ω_{ref} の算出 $\omega_{ref} = \phi v_{ref}$ (曲率 ϕ の経路を追従する式)

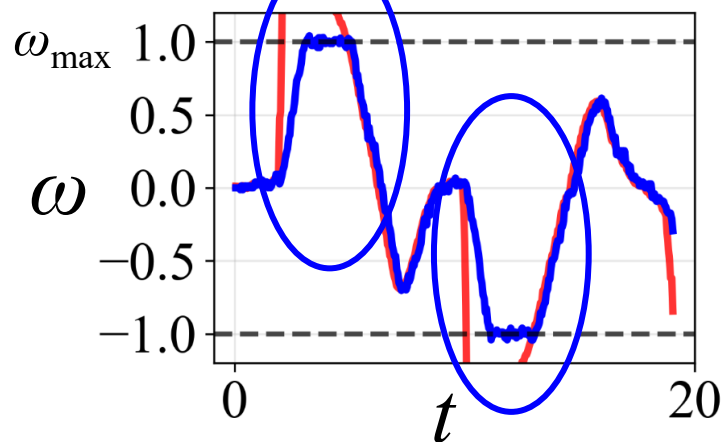
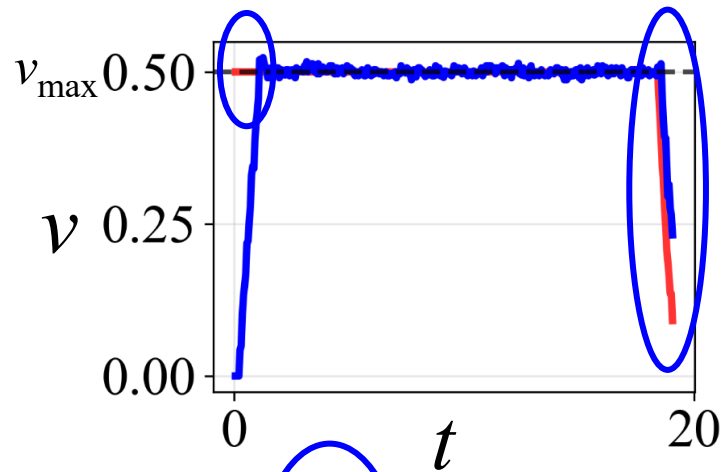
Dynamic Windowを
用いた速度算出 (DWPP)

速度指令値

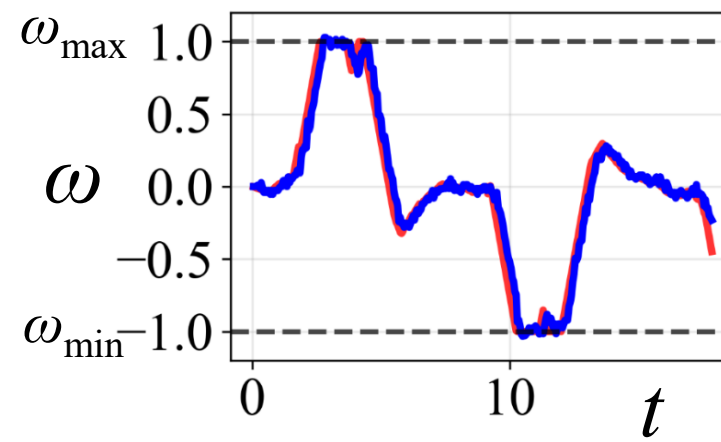
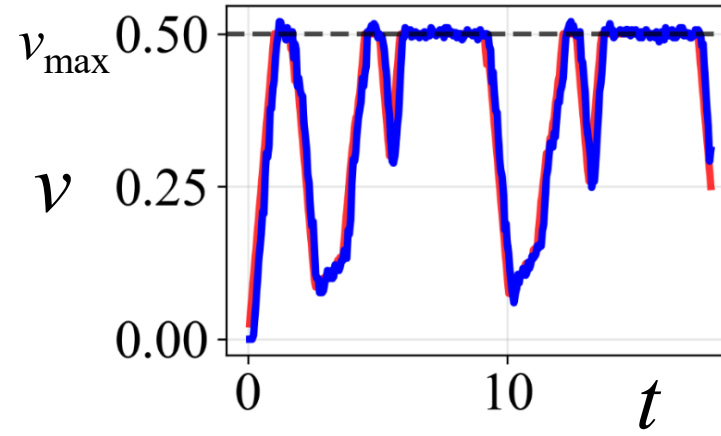
(v_{ref}, ω_{ref})

DWPP: 速度・加速度制約の下で最適に経路追従する速度を算出可能

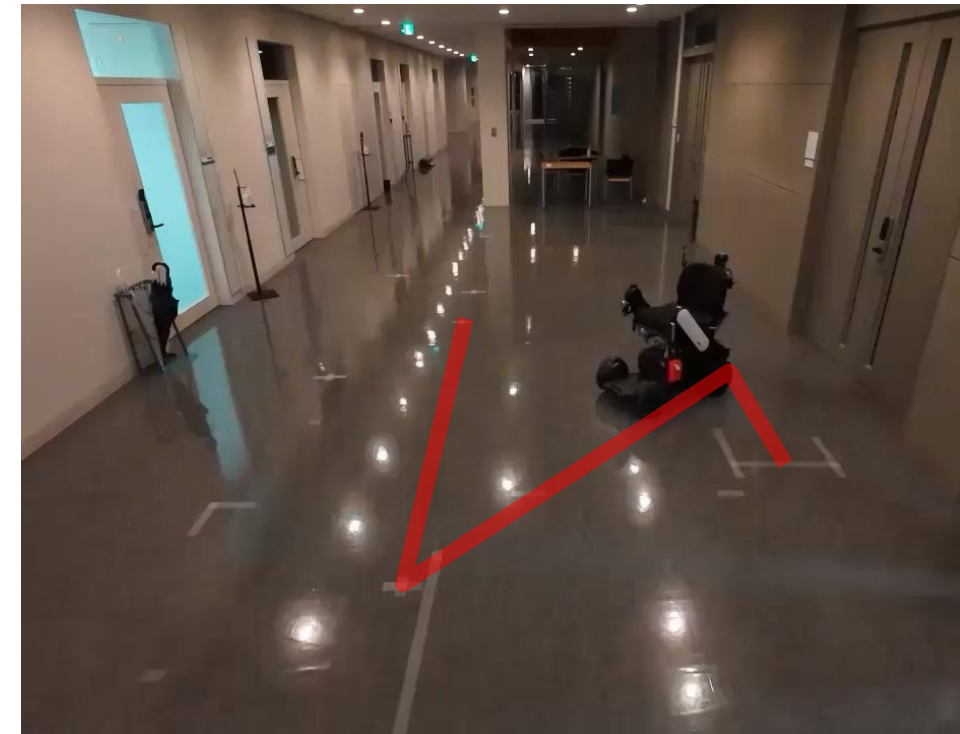
— 速度指令値 — 実現速度



PPの速度プロファイル例



DWPPの速度プロファイル例



DWPPによる経路追従

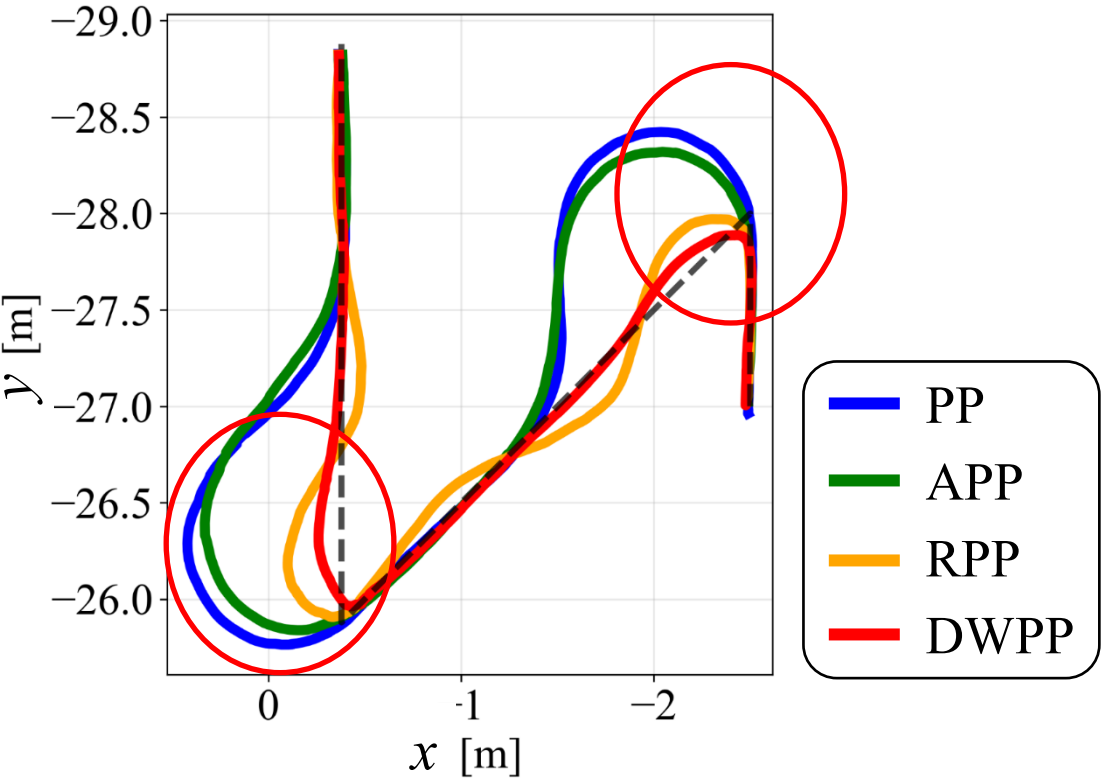
DWPP: 速度・加速度制約の下で最適に経路追従する速度を算出可能

■ 正確な経路追従

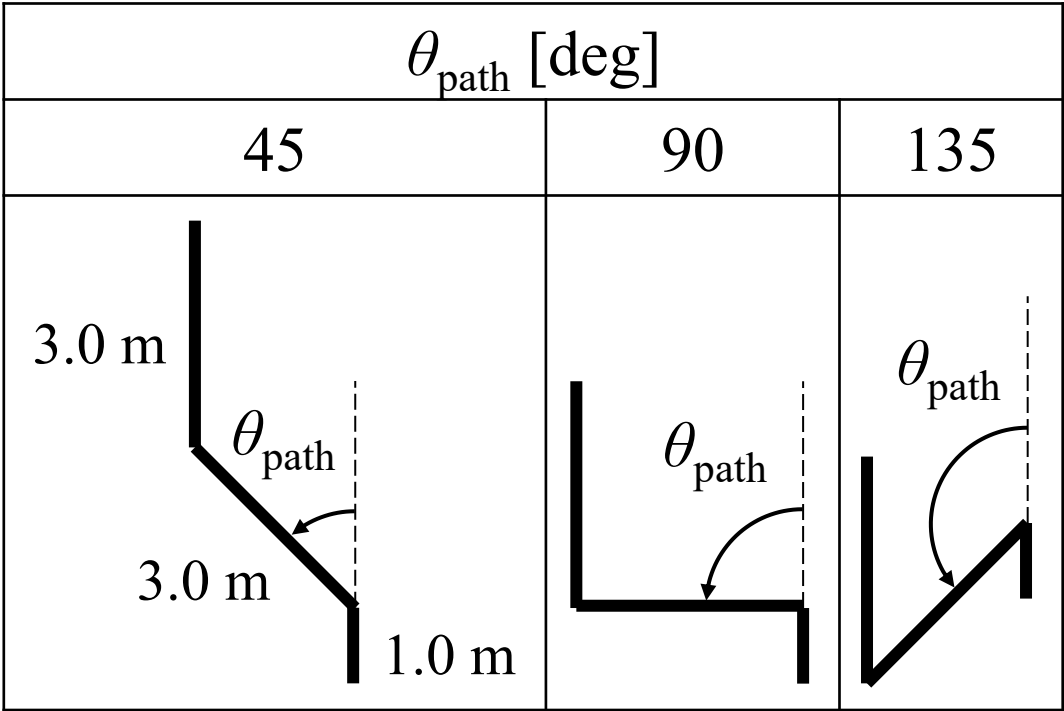
- 追従誤差の低減

■ 乗り心地の向上

- 設定した速度・加速度制約下での運転
- オーバーシュートの低減



評価に使用した経路

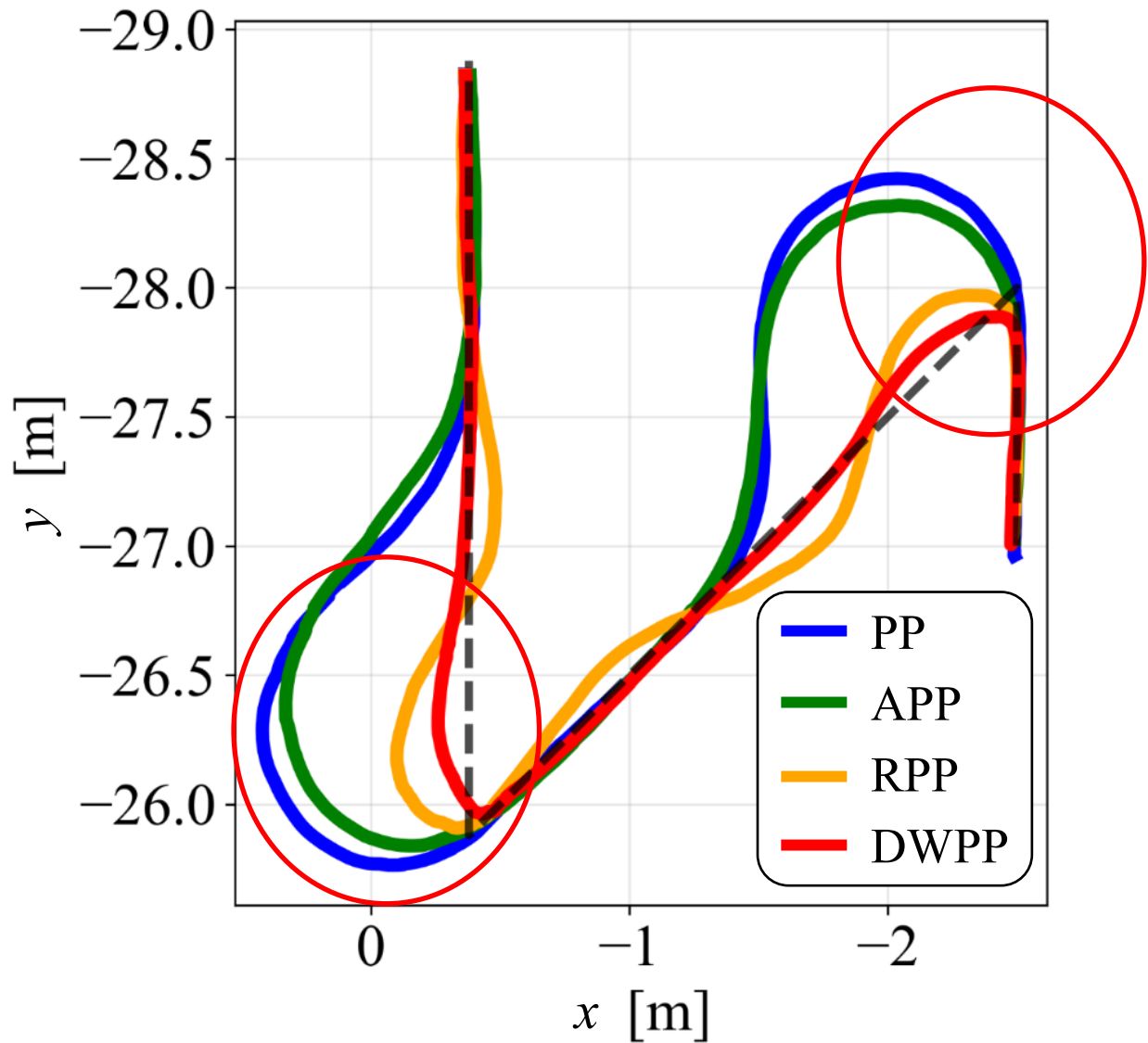


経路追従誤差 (RMSE of cross-track error [m])

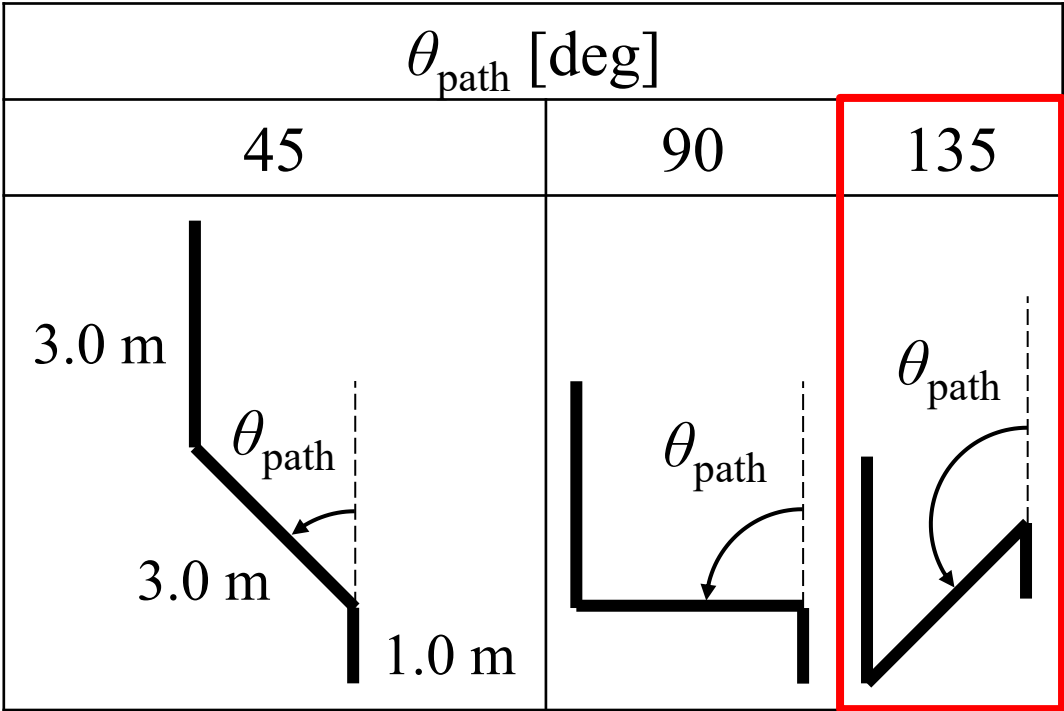
θ_{path}	PP	APP	RPP	DWPP
45	0.035	0.032	0.024	0.023
90	Failed	0.15	0.040	0.029
135	0.38	0.35	0.10	0.039

誤差
低減

■ 曲率が大きくな経路



評価に使用した経路



経路追従誤差 (RMSE of cross-track error [m])

θ_{path}	PP	APP	RPP	DWPP
45	0.035	0.032	0.024	0.023
90	Failed	0.15	0.040	0.029
135	0.38	0.35	0.10	0.039

誤差
低減

- Dynamic Window Pure Pursuit (DWPP) の紹介
 - Pure Pursuitとは
 - 概要
 - 使用例
 - 課題
 - Dynamic Window Pure Pursuit (DWPP)
 - 概要
 - メリット
 - 得意なシナリオ
- 導入方法
 - Nav2への導入
 - テスト環境の導入
- デモンストレーション
- まとめ

■ DWPP Nav2プラグイン GitHubリポジトリ (ROS2 Humble)

– https://github.com/Decwest/nav2_dynamic_window_pure_pursuit_controller



■ 導入手順

1. ROS2環境のsrcディレクトリにコピー

```
git clone https://github.com/Decwest/nav2_dynamic_window_pure_pursuit_controller.git
```

2. パッケージをビルド

```
colcon build --symlink-install
```

3. Nav2のパラメータファイルのプラグインを変更

```
controller_server:  
  ros_parameters:  
    controller_plugins: ["FollowPath"]  
  FollowPath:  
    plugin: "nav2_dynamic_window_pure_pursuit_controller::DynamicWindowPurePursuitController"
```

コピーしてビルド→プラグイン変更！

■ DWPP テスト環境 GitHubリポジトリ

- https://github.com/Decwest/dwpp_test_environment

■ テスト環境 **デモを実施します！**

- 手法間比較シミュレーション
- Nav2シミュレーション with DWPP



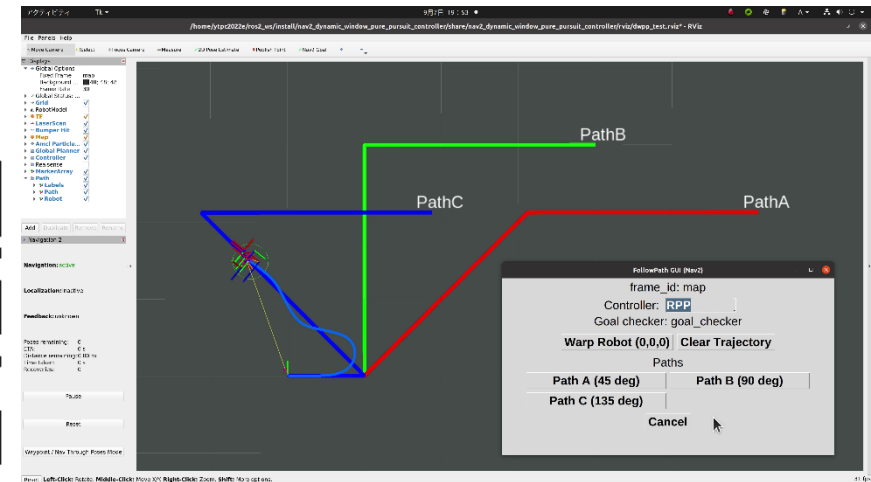
■ 導入に必要なもの

- Docker, Docker compose
 - 仮想環境の作成
- Task
 - コマンドの管理
- GPUを使う場合：nvidia-container-toolkit

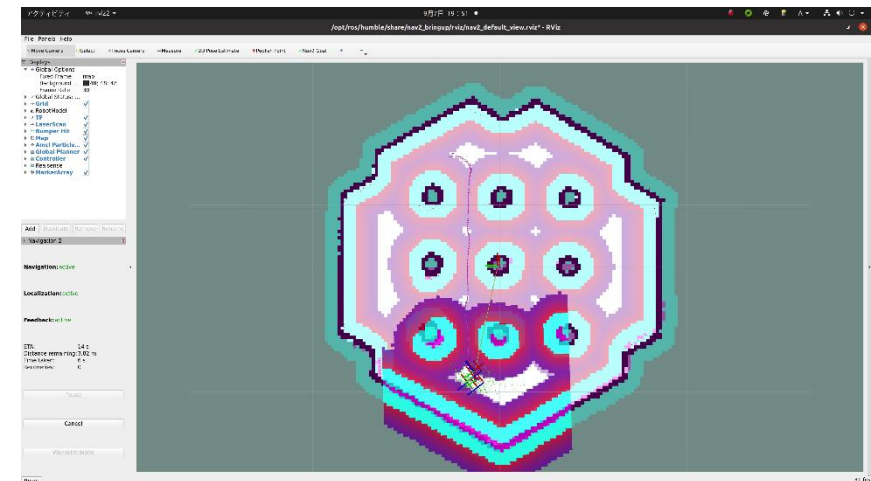


■ 導入方法

- README参照

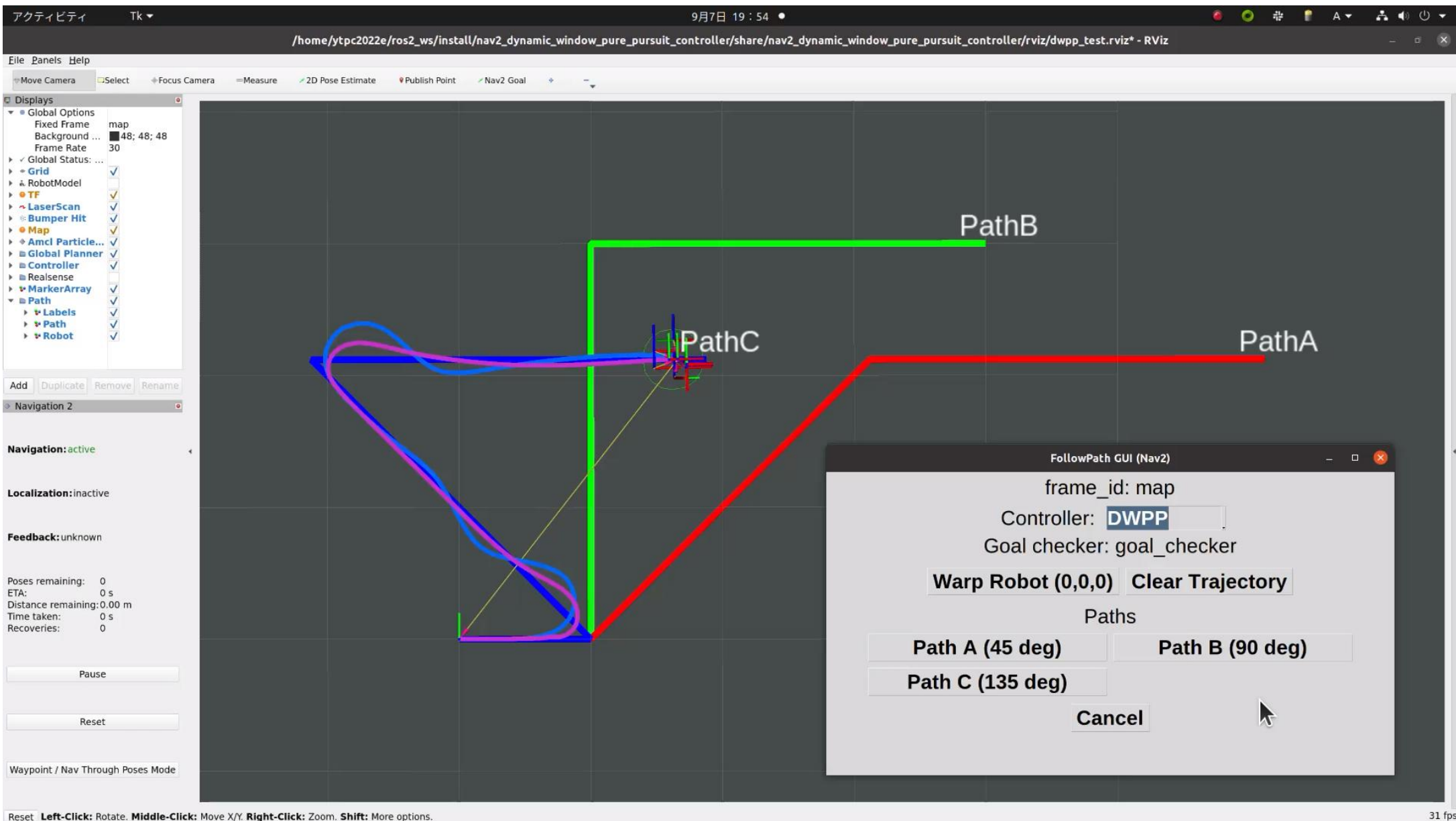


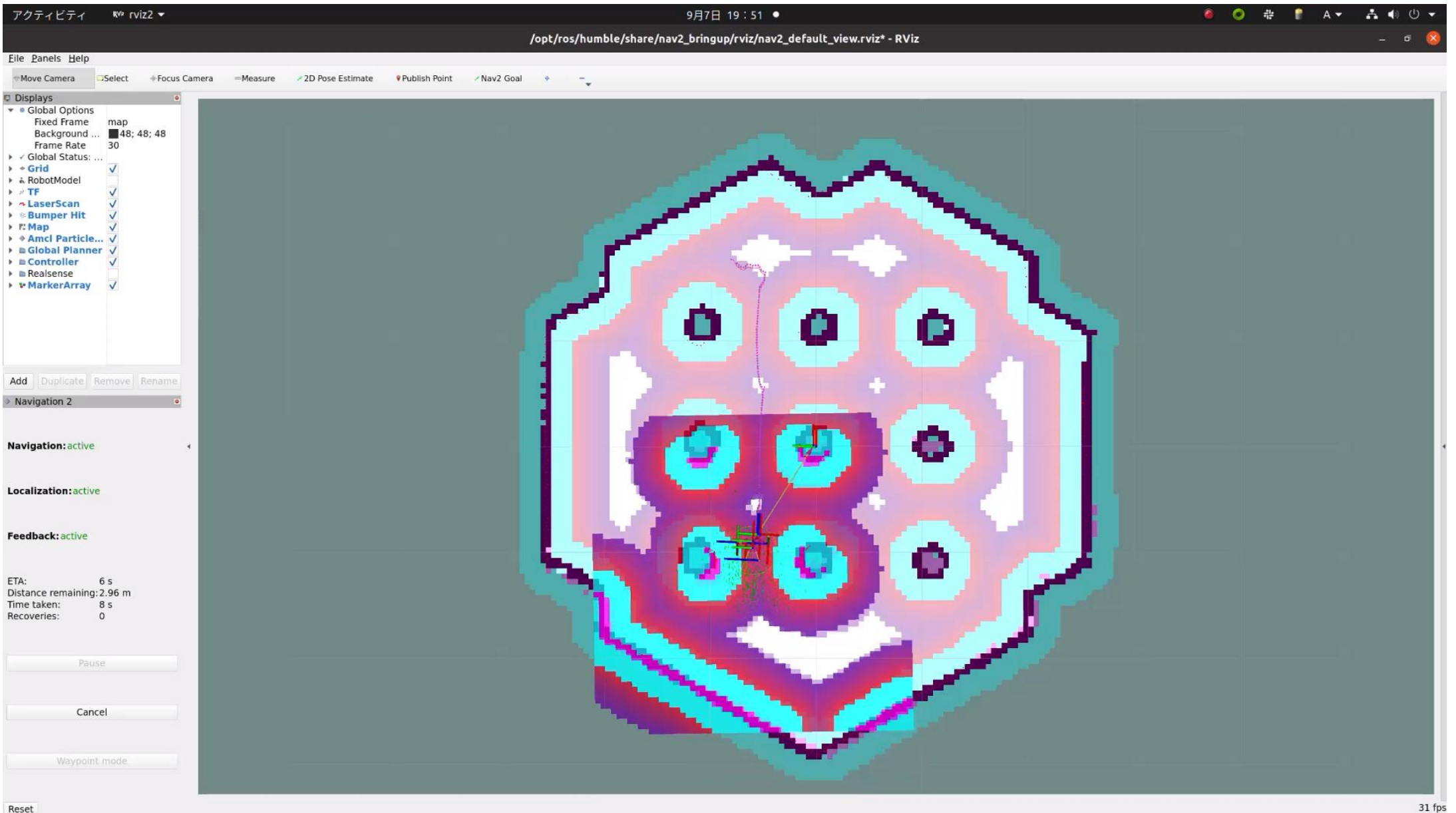
手法間比較シミュレーション



Nav2シミュレーション with DWPP

- Dynamic Window Pure Pursuit (DWPP) の紹介
 - Pure Pursuitとは
 - 概要
 - 使用例
 - 課題
 - Dynamic Window Pure Pursuit (DWPP)
 - 概要
 - メリット
 - 得意なシナリオ
- 導入方法
 - Nav2への導入
 - テスト環境の導入
- デモンストレーション
- まとめ





- Dynamic Window Pure Pursuit (DWPP)とは
 - **速度・加速度制約**の下で最適に経路追従する速度を算出可能なローカルプランナー
- 使用場面：**経路追従のみ**行う場合
 - 設定した経路上を走行
 - 障害物回避するようにリアルタイムに更新される経路上を走行
- 導入のメリット
 - 正確な経路追従
 - 乗り心地の向上
- 得意なシナリオ
 - 曲率が大きな経路
- 導入方法（簡単！）
 - **コピってビルド→プラグイン変更！**
 - テスト環境も用意



DWPP Nav2プラグイン



DWPP テスト環境

ぜひお使いください！