AA 274A
Principles of Robot Autonomy I

Open-source Automated Driving Stack „Autoware Hands-on“
Agenda

• Localization
• Path planning
Localization pipeline

- Map loader `[points_map_loader]`
  - PCD loader from map

- Voxel Grid Filter `[voxel_grid_filter]`
  - Downsampling lidar data
  - Leaf size: 2m (60MB/s → ~1MB/s)

- Lidar based localization `[ndt_matching]`
  - NDT matching
  - Input: `/filtered_points`, `/vehicle/twist`
  - Output: `/ndt_pose`

- EKF Localization Fusion `[ekf_localizer]`
  - Input: `/ndt_pose`, `/vehicle/twist`
  - Output: `/ekf_pose_with_covariance`

Localization / Roborace / Croix-en-Ternois

Autonomous Racing Graz
EKF Localizer

11/15/2019

DD 274A | Lecture 22
EKF Localizer / Interface

Input:
/devbot/twist ... twist from Devbot (velocity, yaw_rate)
/ndt_pose ... position from localization (lidar or noisy GPS data)

Output:
/ekf_pose_with_covariance ... output from the EKF for localization

Ground truth: /gps_local/pose
Localization modes

1) GPS based localization with noisy gps data:
   localization_pose: /ndt_pose (gps_pose + noise)
lidar_localization_active: false

2) Lidar based localization (localization running online)
   Localization_pose: /ndt_pose (ndt_localization)
lidar_localization_active: true

Localization mode: vifware_launch/launch/localization_devbot/Devbot_localization.launch
EKF localizer setting: vifware_launch/launch/localization_devbot/ekf_localizer.launch

After every change in a launch file you need to rebuild the source!
Tasks

1) Localization only with Odometry
2) Localization with GPS without noise + Odometry
   \[ \text{stddev}_x, \text{stddev}_y: 0, \ \mu_x, \ \mu_y: 0 \]
3) Localization with GPS with noise + Odometry
   \[ \text{stddev}_x, \text{stddev}_y: 1, \ \mu_x, \ \mu_y: 0 \]
4) Localization with GPS with noise incl. bias + Odometry
   \[ \text{stddev}_x, \text{stddev}_y: 1, \ \mu_x, \ \mu_y: 1 \]
5) Localization with lidar + Odometry
   \[ \rightarrow \text{parameter tuning (lidar pose has an unknown time delay and unknown noise)} \]
   \[ \textbf{Goal: the ekf\_pose should match the gps\_local/pose} \]
Path Planning / Trajectory Tracking

Global Planner

Local Planner

Local Planner

Trajectory Tracking
Path planning

• Global planner \([\text{op\_global\_planner}]\)

• Local planner \([\text{op\_trajectory\_generator}, \text{op\_motion\_predictor}, \text{op\_trajectory\_evaluator}, \text{op\_behavior\_selector}]\)
  • Input: /tracked\_objects, /global\_path
  • Output: /final\_waypoints

• Trajectory Tracking \([\text{pure\_pursuit or mpc\_follower, twist\_filter}]\)
  • Input: /final\_waypoints
  • Output: /twist\_cmd

• Autoware Simulator \([\text{wf\_simulator}]\)
  • Input: /twist\_cmd
  • Output: /simulated\_objects
Thanks for your attention!

Questions?

Daniel Watzenig, Markus Schratter
daniel.watzenig@v2c2.at
markus.schratter@v2c2.at