Nominal Beluga AMCL Benchmark Report

Release 0.1.0-alpha

Ekumen Inc.

Nov 03, 2023

CONTENTS:

CHAPTER

ONE

OBJECTIVE

Compare Beluga AMCL and Nav2 AMCL localization systems in terms of localization and computational performance for the nominal case.

CHAPTER TWO

METHODOLOGY

2.1 Dataset

For this report, Magazino datasets, published with the Cartographer Public Data set under Apache License v2.0, were chosen. As these datasets are distributed in rosbag format, equivalent datasets in rosbag2 format were recreated. As both localization systems need a map to work with, and a groundtruth is necessary for performance evaluation, offline mapping was conducted using Cartographer ROS.

2.2 Configuration

For this report, the following baseline configuration:

```
/**:
  ros__parameters:
    # Odometry motion model type.
    robot_model_type: nav2_amcl::DifferentialMotionModel
    # Expected process noise in odometry’s rotation estimate from rotation.
    alpha1: 0.1
    # Expected process noise in odometry’s rotation estimate from translation.
    alpha2: 0.05
    # Expected process noise in odometry’s translation estimate from translation.
    alpha3: 0.1
    # Expected process noise in odometry’s translation estimate from rotation.
    alpha4: 0.05
    # Expected process noise in odometry's strafe estimate from translation.
    alpha5: 0.1
    # The name of the coordinate frame published by the localization system.
    global_frame_id: map
    # The name of the coordinate frame published by the odometry system.
    odom_frame_id: odom
    # The name of the coordinate frame of the robot base.
    base_frame_id: base_link
    # The name of the topic where the map is published by the map server.
    map_topic: map
    # The name of the topic where scans are being published.
    scan_topic: scan_front
    # The name of the topic where an initial pose can be published.
    # The particle filter will be reset using the provided pose with covariance.
    initial_pose_topic: initialpose
    # Maximum number of particles that will be used.
    max_particles: 2000
    # Minimum number of particles that will be used.
    min_particles: 500
```

(continues on next page)

(continued from previous page)

```
    # Error allowed by KLD criteria.
    pf_err: 0.05
    # KLD criteria parameter.
    # Upper standard normal quantile for the probability that the error in the
    # estimated distribution is less than pf_err.
    pf_z: 3.0
    # Fast exponential filter constant, used to filter the average particles weights.
    # Random particles are added if the fast filter result is larger than the slow␣
↪filter result
    # allowing the particle filter to recover from a bad approximation.
    recovery_alpha_fast: 0.1
    # Slow exponential filter constant, used to filter the average particles weights.
    # Random particles are added if the fast filter result is larger than the slow␣
↪filter result
    # allowing the particle filter to recover from a bad approximation.
    recovery_alpha_slow: 0.001
    # Resample will happen after the amount of updates specified here happen.
    resample_interval: 1
    # Minimum angle difference from last resample for resampling to happen again.
    update_min_a: 0.2
    # Maximum angle difference from last resample for resampling to happen again.
    update_min_d: 0.25
    # Laser sensor model type.
    laser_model_type: likelihood_field
    # Maximum distance of an obstacle (if the distance is higher, this one will be␣
↪used in the likelihood map).
    laser_likelihood_max_dist: 2.0
    # Maximum range of the laser.
    laser_max_range: 100.0
    # Maximum number of beams to use in the likelihood field sensor model.
    max_beams: 100
    # Weight used to combine the probability of hitting an obstacle.
    z_hit: 0.5
    # Weight used to combine the probability of random noise in perception.
    z_rand: 0.5
    # Weight used to combine the probability of getting short readings.
    z_short: 0.05
    # Weight used to combine the probability of getting max range readings.
    z_max: 0.05
    # Standard deviation of a gaussian centered arounds obstacles.
    sigma_hit: 0.2
    # Whether to broadcast map to odom transform or not.
    tf_broadcast: false
    # Transform tolerance allowed.
    transform_tolerance: 1.0
    # Execution policy used to apply the motion update and importance weight steps.
    # Valid options: "seq", "par".
    execution_policy: seq
    # Whether to set initial pose based on parameters.
    # When enabled, particles will be initialized with the specified pose coordinates␣
↪and covariance.
    set_initial_pose: true
    # If false, AMCL will use the last known pose to initialize when a new map is␣
↪received.
    always_reset_initial_pose: false
    # Set this to true when you want to load only the first published map from map_
```

(continues on next page)

(continued from previous page)

```
↪server and ignore subsequent ones.
    first_map_only: false
    # Initial pose x coordinate.
    initial_pose.x: 0.0
    # Initial pose y coordinate.
    initial_pose.y: 0.0
    # Initial pose yaw coordinate.
    initial_pose.yaw: 0.0
    # Initial pose xx covariance.
    initial_pose.covariance_x: 0.25
    # Initial pose yy covariance.
    initial_pose.covariance_y: 0.25
    # Initial pose yawyaw covariance.
    initial_pose.covariance_yaw: 0.0685
    # Initial pose xy covariance.
    initial_pose.covariance_xy: 0.0
    # Initial pose xyaw covariance.
    initial_pose.covariance_xyaw: 0.0
    # Initial pose yyaw covariance.
    initial_pose.covariance_yyaw: 0.0
```

was modified for each benchmark case in terms of:

- the laser sensor model, to assess both beam and likelihood models (see sections 6.3 and 6.4 of Probabilistic Robotics, by Thrun et al);
- the execution policy, to compare single-threaded and multi-threaded performance. Note this feature is only provided by Beluga AMCL.

so as to have a reasonably complete picture of how both localization systems perform.

2.3 Platform

- Hardware
  - CPU: Intel(R) Core(TM) i9-9900 CPU @ 3.10GHz @ 5000.0 megahertz MHz x 8 cores (16 threads)
    - L1 instruction cache: 8 x 32 KB, 8-way set associative (64 sets), 64 byte lines, shared by 2 processors
    - L1 data cache: 8 x 32 KB, 8-way set associative (64 sets), 64 byte lines, shared by 2 processors
    - L2 unified cache: 8 x 256 KB (exclusive), 4-way set associative (1024 sets), 64 byte lines, shared by 2 processors
    - L3 unified cache: 16 MB (inclusive), 16-way set associative (16384 sets, complex indexing), 64 byte lines, shared by 16 processors
  - Memory: 16689.225727999998 MB
- Software
  - OS: Ubuntu 22.04.3 LTS
  - ROS:
    - Distribution: Humble Hawksbill
    - Packages:
      - `beluga_amcl` 1.0.0
      - `nav2_amcl` 1.2.0

2.4 Metrics

To characterize the localization performance of both systems, this report uses:

- **APE**. The Absolute Pose Error is the difference between estimated and reference trajectories after alignment when taken as a whole. It is a measure of global accuracy and consistency.

Metrics are aggregated across multiple runs of each parameter variation to ensure statistical significance.

CHAPTER THREE

RESULTS

Table 1: APE metrics for `hallway_localization` trajectories

Implementation	Likelihood field sensor model				Beam sensor model			
	rms	mean	stddev	max	rms	mean	stddev	max
Beluga AMCL	0.117 m	0.09 m	0.074 m	0.535 m	0.098 m	0.076 m	0.062 m	0.406 m
Nav2 AMCL	0.072 m	0.058 m	0.042 m	0.194 m	11.243 m	6.931 m	8.855 m	52.297 m

Table 2: APE metrics for `hallway_return` trajectories

Implementation	Likelihood field sensor model				Beam sensor model			
	rms	mean	stddev	max	rms	mean	stddev	max
Beluga AMCL	0.111 m	0.085 m	0.071 m	0.623 m	0.101 m	0.079 m	0.062 m	0.526 m
Nav2 AMCL	0.07 m	0.058 m	0.039 m	0.189 m	20.694 m	15.525 m	13.684 m	77.223 m