Navigation Toolbox™ Release Notes

MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Navigation Toolbox™ Release Notes

Contents

R2019b

R2019b

Version: 1.0

New Features

Simultaneous Localization and Mapping (SLAM): Create 2-D and 3-D occupancy maps using SLAM algorithm and lidar scan data

Use the SLAM algorithm to tune parameters for scan matching and loop-closure detection. The `lidarSLAM` object takes lidar scan data and builds a map as your vehicle moves through it. The algorithm generates a `poseGraph` and continuously optimizes edge-constraints based on detected loop closures. As more loop closures are detected, you can continuously build a map of your environment and adjust for odometry drift.

For an example using 2-D lidar scans, see “Implement Online Simultaneous Localization And Mapping (SLAM) with Lidar Scans”.

For an example using 3-D lidar point clouds, see “Perform SLAM Using 3-D Lidar Point Clouds”.

For more information, see “SLAM”.

SLAM Map Builder App: Interactively modify loop closures and adjust overall map using SLAM algorithm

Use the **SLAM Map Builder** app to load and filter lidar scans and estimated poses from a log file or data in the workspace. Tune and run the SLAM algorithm to automatically build the map. Pause at any time to modify relative poses between scans. Modify or delete loop closures from the pose graph to improve the overall map. After you are done with the entire data set, output the map as an occupancy grid to use with path planning or other navigation algorithms.

Pose Estimation: Accurately estimate vehicle poses using IMU and GPS sensors and Monte Carlo Localization

Use localization and pose estimation algorithms to orient your vehicle in your environment. Sensor pose estimation uses filters to improve and combine sensor readings for IMU, GPS, and other sensors. Localization algorithms, like Monte Carlo localization and scan matching, estimate your pose in a known map using range sensor or lidar readings. Pose graphs track your estimated poses and can be optimized based on edge constraints and loop closures.

For more information, see “Localization and Pose Estimation”

Customizable Sampling-Based Path Planners: Plan a path from start to goal locations using RRT and RRT* algorithms

Plan paths through a 2-D environment using provided path planning algorithms:

- `plannerRRT`
- `plannerRRTStar`
- `plannerHybridAStar`

Specify parameters for provided 2-D state-space representations:

- `stateSpaceSE2`
- `stateSpaceDubins`
- `stateSpaceReedsShepp`

Validate your planned paths using occupancy maps or vehicle cost maps:

- `validatorOccupancyMap`
- `validatorVehicleCostmap`

Write your own custom state space or state validator using class interfaces:

- `nav.StateSpace`
- `nav.StateValidator`

Path-Planning Metrics: Use metrics to check and compare the output of path planners

Calculate path metrics to evaluate planned paths using the `pathmetrics` object. Check the `clearance` and `smoothness` based on your path constraints.

Sensor Models: Use simulated models for IMU, GPS, and range sensors

Perform sensor modeling and simulation for accelerometers, magnetometers, gyroscopes, altimeters, GPS, IMU, and range sensors. Analyze sensor readings, sensor noise, environmental conditions, and other configuration parameters. Generate trajectories to emulate these sensors traveling through a world, and calibrate the performance of your sensors.

Sensor models include:

- `gpsSensor`
- `imuSensor`
- `rangeSensor`

For other sensors and more information, see “Sensor Models”.

Trajectory and Waypoint Following Algorithms: Use built-in algorithms to generate trajectories and control commands for robots

Use the `waypointTrajectory` and `kinematicTrajectory` objects to generate trajectories for sensors or vehicles and control commands to send to your vehicle