Basavaraj Navalgund

Bengaluru, IN
+919845268868 | ibasavaraj.navalgund@gmail.com
LinkedIn | GitHub | Portfolio

Personal Statement

I'm an experienced professional, who mastered both hardware and software skills throughout a journey of building personal projects, and working in the industry. I am passionate about the autonomous systems field including ROS Robots and Autonomous Driving.

I consider myself to be a strong software engineer who is capable of researching, designing, and developing of software solutions.

Education

KLE Technological University Hubballi, IN
Bachelor of Engineering in Electronics and Communication *Aug. 2016 - June 2020*

Technical Skills

C++, Python, ROS, Gazebo, GitHub, Linux, Arduino, CI/CD

Experience

Mercedes-Benz Research and Development India Bengaluru, IN
Software Engineer *Sep. 2021 - Present*

- Software Development in the Sensor Fusion team for Autonomous Driving
- Good knowledge of simulation tools, software debugging, and analysis of measurement recorded during vehicle test drives
- Perform the Unit Tests and Software test using the G-test, and SBT (Mercedes-Benz internal tool)
- Created ros_cyber_bridge to interface GPS and IMU messages between ROS and Apollo ecosystem
- Experience in Atlassian tool chain - JIRA, Bitbucket, Confluence etc
- Close collaboration with colleagues in Mercedes-Benz AG for software deliverables and release activities
- All software was written in C++

Conigital Group Remote
ROS Developer *Apr. 2021 - Sep. 2021*

- Software Development, Validation, and Simulation for Level 3+ Autonomous Vehicle (AV)
- Modelled sensors on the simulation environment and configured them on the vehicle to create a digital twin
- Performed virtual validation and testing of path planning and control modules, which generates trajectories for the vehicle and calculates control inputs (steering, acceleration, and deceleration!) to execute those trajectories using CARLA with Autoware simulator
- Created ROS package to monitor the Log status messages, sensor failures, and jumps in localization
- Created ROS package to establish the communication between Aslan and ConOP application via vehicle telemetry messages (GPS, IMU, and Odometry)
- Added a custom vehicle in CARLA simulator. Changed characteristics of pre-defined sensors in Unity
- Simulated multiple ego-vehicles running the software stack using distributed computing in the CARLA simulator
- Studied and modified weights and costs for Model Predictive Controller(MPC) for accurate path tracking. Successfully tested in simulation and real world
- Mapped topics published and subscribed on ROS, that serves as the middleware for the AV stack
- All software was based on ROS framework, and was written in C++ or Python

Freelance Remote
ROS Developer *July 2020 - Feb. 2021*

- Software Development for Autonomous Mobile Robot (AMR) with a focus on SLAM and Navigation stack for Holonomic and Differential drive robots
- Made them to drive and map autonomously within a controlled environment
- Hands-on Experience with 3D and 2D SLAM, RPLiDAR S1 and A3, Intel RealSense L515, T265, and D435i
- Sensor Fusion of Wheel Odometry, Visual Odometry, and IMU using robot_localization package
- All software was based on ROS framework, and was written in C++ or Python

Continental Bengaluru, IN

Intern *Feb. 2020 - June 2020*

- Software Development for Autonomous Mobile Robot (AMR)
- Worked primarily as a ROS developer with a focus on SLAM and Navigation stack for Holonomic drive with 4 mecanum-wheeled robot
- Interfaced multiple ROS packages and coded nodes in C++ and Python to autonomously map an environment, localize itself, and navigate to pick-up and drop-off objects in a manufacturing plant
- Expanded my knowledge in both the hardware and software side of engineering especially in Linux, C++, Python, and ROS

Projects

- **lidar_robot_mapping** (ROS, Autoware, Gazebo, C++)
  Hands-on Experience with RS–LiDAR-16, OS2–64, and OS2–128 to build an occupancy grid of controlled environment. Fusion of slam_toolbox with Autoware package based NDT Mapping to get odom_to_baselink transformation and, accurate 2D SLAM
- **t265_robot_navigation** (ROS, Gazebo, C++) [code]
  Using Intel Realsense D435 and T265 to build an occupancy grid and autonomously navigate around using movebase
- **Workplace Service Robot** (ROS, Gazebo, C++, Python) [code]
  Autonomous Mobile Robot (AMR), a holonomic drive with 4 mecanum-wheels. It autonomously maps an environment, localizes itself, and navigate to pick-up and drop-off objects in a simulated environment
- **Sensor fusion-based Tracking** (C++) [code]
  Designed and implemented a Kalman filter, Extended Kalman filter, and Unscented Kalman filter for object tracking
- **Vehicle Localization** (C++) [code]
  Implemented a real-time particle filter to estimate the position and orientation of a moving vehicle
- **Path Planning** (C++) [code]
  Implemented a simple real-time path planner in C++ to navigate a car around a simulated highway scenario, including other traffic, given waypoint, and sensor fusion data
- **Model Predictive Control** (C++) [code]
  Implemented a Model Predictive Control (MPC) to drive a car in a game simulator. The server provides reference waypoints via websocket, and we use MPC to optimize the actuators (steering and throttle), simulate the vehicle trajectory, and minimize the cost like cross-track error