

# NALIN BENDAPUDI

☎ (734) 882-1436 📧 nalin-bendapudi ✉ bnaln@umich.edu 🌐 nalinbendapudi.ml 📧 nalinbendapudi

## EDUCATION

**University of Michigan, Ann Arbor, MI**  
*Master of Sciences in Robotics, GPA:3.98/4*

September 2019 - Dec 2020

**Indian Institute of Technology, Delhi, India**

July 2013 - May 2017

*Bachelor of Technology in Mechanical Engineering (Minor in Computer Science), GPA: 8.91/10*

## RELEVANT COURSES AND SKILLS

<b>Coursework</b>	Self-driving Vehicles, Robotics Systems Lab, Robot Modelling and Control, Mobile Robotics (SLAM), Deep Learning for Computer Vision, Data Structures and Algorithms, Machine Learning
<b>Skills</b>	C++, Python, MATLAB, R, Java, ROS, Linux

## EXPERIENCE

**BOSCH Engineering and Business Solutions, Bengaluru, India**  
*Senior Engineer, Automation Team*

Aug 2017 - May 2019

- Analyzed trend and seasonality of time series data, and developed applications for anomaly detection and forecasting of automobile metrics. These modules, used across nine departments in Mercedes-Benz, have minimum accuracy of 92.7%
- Utilized deep-NLP on unstructured error logs to develop a recommendation engine to identify potential high-impact issues
- Applied unsupervised learning to associations among data that helped reduce resolution & root-cause analysis time by 40%

## PROJECTS

**Trajectory Optimization using Digital Model Predictive Control**

Feb 2020 - Apr 2020

- Built a PID controller to drive a non-linear vehicle model on Texas Austin's F1 circuit to achieve a track-time of 92.1s
- Designed a model predictive controller for a linearized trajectory to achieve safe tracking even with sensor noise of 5%
- Solved a non-linear optimization problem to calculate control inputs to track an obstacle course with 0% collision rate

**Visual Inertial SLAM with Invariant Extended Kalman Filtering**

Feb 2020 - Apr 2020

- Utilized invariant kalman filter to propagate IMU pose uncertainty and parallelly maintained the visual correction algorithm
- Improved OpenVINS code by substituting the ORB feature extractor with deep-learning based SuperPoint visual descriptor

**Occupancy-Grid SLAM for Autonomous Ground Robot**

Sep 2019 - Dec 2019

- Implemented particle filter localization using 2D-LiDAR sensor and an efficient mapping algorithm on a 40,000 cell grid
- Incorporated an exploration strategy to search for new frontiers in the SLAM map and reach them using A\* path planning

**Linear Quadratic Regulator for Wheeled Inverted Pendulum**

Sep 2019 - Dec 2019

- Formulated state space model of a 2-DOF mobile inverted pendulum by linearizing the Lagrangian dynamics of the system
- Combined balancing LQ regulator, in-plane angle stabilizer and a tracking controller to limit position error to 0.05m

**Computer Vision assisted Pick & Place Robotic Arm and Gripper**

Sep 2019 - Dec 2019

- Employed Kinect depth sensor and RGB camera to implement block detection algorithm with a mean accuracy of 98%
- Developed a trajectory module to generate smooth and safe paths by integrating obstacle avoidance with path-smoothing

**Computed Torque Controller for Wheeled Mobile Robot (WMR)**

Jan 2017 - Aug 2017

*This work was published in 15<sup>th</sup> IEEE International Workshop on Advanced Motion Control (AMC-2018)*

- Devised a novel three-stage controller which included a Lyapunov stability based controller on position error, a PID controller applied on the robot velocity, and a non-linear compensator to calculate motor torques using inverse dynamics
- Simulated the controller for a system with modelling uncertainties & disturbances. Demonstrated max tracking error is 4cm
- Designed and manufactured a mobile robot to implement these control algorithms using RaspberryPi 3, encoders & IMU

**Off- line Programming of Industrial Robot**

Jul 2016 - Dec 2016

- Developed a kinematic simulator for KR5-arc robot on MATLAB SimMechanics using STL files & DH parameters
- Built an automatic KUKA Robot Language code generator to translate the simulated motion to robot motion
- Implemented a technique for traversing any joint-space trajectory using series of differential Point-to-Point motions