

# Bharath Satheesh

2750 Dwight Way, Berkeley, CA 94704 | (510)-710-2674 | [bharath.satheesh@berkeley.edu](mailto:bharath.satheesh@berkeley.edu) | <http://bharathsatheesh.com/>

## Education

### University of California, Berkeley

Electrical Engineering and Computer Sciences; Applied Mathematics  
Class of 2017; GPA: 3.x

### Carnegie Mellon University

Robotics and Embedded Systems (Team India for Robogames 2013)  
Class of 2013; **Oracle Java Certification**

## Relevant Coursework

Nonlinear Systems (EE 222)  
Introduction to Machine Learning (EE 189)  
Information Theory/ Codes (EE 290S)

Optimization Models and Applications (EE 227A)  
Linear System Theory (EE 221A)  
Probability Theory and Random Processes (EE 126)

## Research Experience

### **Autodesk, Office of the CTO:** *Strategic Innovation Intern* (June 2016 - September 2016)

*The mission of the OCTO team is to explore, distill, and apply what's next for Autodesk and explain why it matters.*

- Lead the development of Autodesk platform for autonomous construction (pick + place + welding) of mesh structures with 6-axis robot arms; *Presented working solution to the entire division, CTO Jeff Kowalski*
- Integrated Autodesk Maya-Fanuc with robot communication to build 1st ever feature film robot-collaboration platform
- Created a robotic *vision and feedback* system ground-up that can build and weld large structures together in tandem without human supervision at a 98.4% accuracy

### **Hybrid Systems lab:** *Research Assistant* (March 2015 - Present)

*Research here covers a wide range of topics, including air traffic control automation, algorithms for decentralized optimization, modeling and analysis of biological cell networks, and unmanned aerial vehicle design and control.*

- Implemented *target and trajectory tracking* for quadrotors with accurate state estimations (Learning model and Reachable Sets)
- Redesigned old non functioning C, Python code base in ROS (Robot Operating System) to create Matlab functionality to test ongoing reachability experiments at the lab
- Created Catkin functionality for ROS code, to collaborate with ETH Zurich on a multi-vehicle test bed (STARMAC)
- Presented poster at NASA UTM (UAS Traffic Management) conference in the summer of 2015 on hybrid systems theory

### **Model Predictive Controls Lab:** *Research Assistant* (August 2014 - December 2014)

*The research lab focuses on the theoretical and real-time implementation aspects of constrained predictive model-based control.*

- Conducted error estimation experiments on autonomous cars to keep track of lane changes with Model Predictive Control
- Designed Kalman filter to accurately correct approximations to match simulations on Carsim and DSpace
- Deployed filtered model to the car and verified that actual highway measurements recorded by a real time camera system, mimic the theoretical estimates of the predicted model

## Publications/ Presentations

- *Secure State Estimation against adversarial Cyber Attacks via. Distributed On-line Secret Sharing* (Dijkstra's Implementation)
- Multivehicle Collision Avoidance systems: A different approach to *Geometric Programming* (Implicit Softmax Affine Defn.)
- Presented Ceramic Artwork at the Wurster Art Gallery in February 2015; 1 of 12 undergraduates in the past 15 years

## Leadership & Activities

- Representing Berkeley Engineering as part of the 6 member *Senior Students Council* for the graduating class of 2017
- Implemented *gesture recognition* with the Spotify API to like or 'upvote' music with OpenCV, Scikit-learn for Python (PennApps)
- Created a *smart calendar* that keeps track of important events with simple single layered neural networks with C optimizations to enhance speed in data recollection (TreeHacks)
- Held *officer positions at the IEEE* in 2014/2015 in the DevOps, Activities committees respectively
- Represented UC Berkeley at the *AIAA Robo-Ops competition* (hosted by NIA and NASA) at the Johnson Space Center in Texas
- Built a *smart fan* that directs wind flow based on user location with low frequency filtering