

# David Kooi

Website: <http://davidkooi.pythonanywhere.com>

Linkedin: <https://www.linkedin.com/in/david-kooi/>

Email : [dkooi@ucsc.edu](mailto:dkooi@ucsc.edu)

Mobile : +1-626-421-5418

## EDUCATION

---

### University of California, Santa Cruz

Santa Cruz, CA

— *Master of Science in Electrical and Computer Engineering (GPA 3.86)*

*Expected Graduation: June 2020*

- **Affiliation: Hybrid Systems Lab:** Graduate Researcher
- **Thesis: Auto-Generation of Run-time Monitors for Embedded Systems:** Formal specifications are used to generate run-time monitors as compilable C++ code. Theoretical perspectives on sample time and safety bounds.

### University of California, Santa Cruz

Santa Cruz, CA

— *Bachelor of Science in Computer Engineering (GPA 3.80)*

*Sept. 2016 – June. 2018*

- **Emphasis:** Robotics and Control; **Organizations:** Tau Beta Pi, IEEE

## EXPERIENCE

---

### SGT at NASA Ames

Moffet Field, CA

— *Requirements Engineering Intern (Formal Methods, Model Checking)*

*June 2019 - September 2019*

- **Realizability Testing:** Developed and proved a compositional testing approach. Used model checking techniques to test, analyze, and document realizability of formal flight requirements.
- **Development:** Extended the FRET software to support generation of realizability specifications in Lustre.

### UC Santa Cruz Summer Internship Program

Santa Cruz, CA

— *Project Mentor (Computer Vision, Reactive Motion Planning)*

*June 2019 - Sept. 2019*

- **Autonomous Control of PacMan:** Led a team of high-schoolers to reconstruct the PacMan game-state using computer vision and demonstrated real-time control of PacMan using potential functions.
- **Accomplishments:** Poster accepted into Sigma Xi conference.

### NASA Jet Propulsion Laboratory (Python, C++, Unix)

Pasadena, CA

— *Flight Software/Ground System Development Intern*

*June - September of 2015 - 2018*

- **2018:** • Used static analysis tools to extract memory allocation information from flight code. Used information to create binary images for in-flight parameter updates. • Created an auto-coder to take memory segment specifications and output flight verified C code.
- **2017:** • Used Python ZeroMQ to create a dynamic, many-many, publish-subscribe ground system. • Developed a C++ interface to server. • Created "pluggable" packet-protocol translation.
- **2016:** Ground system UI and API development and maintenance.
- **2015:** Ground system and auto-coder development and maintenance.

### UC Santa Cruz Senior Design Project / Topcon Corporation

Santa Cruz, CA

— *Software Systems Lead (C++, Python, ROS, Perception)*

*September 2017 - June 2018*

- **Result:** Used computer vision to demonstrate autonomous operation of full sized hydraulic front-loader.
- **Perception:** Developed C++ functions to analyze stereo-camera data to find optimal approach towards stockpile as measured by maximizing of material ahead of front-loader.
- **Software Systems:** Developed modular Python framework to allow independent development of scheduling, perception, and control tasks. Developed asynchronous Python module for actuation of front-loader.
- **Testing:** Development on 1/20 scale front-loader, field testing of full-size hydraulic front-loader.

## SKILLS

---

- **Programming Languages:** C, C++, Python, Unix shell scripting, Verilog, Java, MATLAB
- **Software Frameworks:** ROS, ZeroMQ, PointCloudLibrary, Tkinter, Git, Nose, Android SDK
- **Software Methodologies:** Automated Code Generation, AST Traversal, OOP Design Patterns, UML Modeling
- **Robotics/Control:** Hybrid System methodology; classical, state-space, optimization based control; path planning techniques like RRT, A\*; MPC based planning, minimum-jerk curve fitting
- **Electrical:** Signal Conditioning; Circuit design, analysis; Troubleshooting with DMMs, O-scopes, logic analyzers

## LINKS TO PROJECTS

---

- [Path Planning for Robotic Manipulator](#) (MATLAB, Dynamics, Computed Torque)
- [Perceptron Branch Predictor Implementation](#) (Micro-Architecture, C++)
- [Minimum Jerk Path Planning](#) (MATLAB, A\*)