Nolan Holden

Contact

mobile: +1.502.994.7601
email: holden.nolan@gmail.com

Professional Experience

Tesla *Palo Alto, CA*
Senior Autopilot Software Engineer *Nov 2021* – **Jun 2022**
Autopilot Software Engineer *Oct 2020* – *...*
Autopilot Intern **Apr 2020** – *Sep 2020*

Multi-functional engineer reporting to CEO and director staff on weekly/monthly basis.

Projects created or significant contribution (order not important):

- **Feature toggle service**: support canary rollouts and generic runtime configurability across >1M vehicles without need for OTA update (zero customer downtime)
- **Realtime-friendly performance tooling**: always-on collection (DRAM-buffered) & on-demand logging (UFS) of any qualitative metric of autopilot + low-level metrics e.g. task latencies, bus/link utilization, memory usage, performance counters of various IP blocks, across multiple SoCs.
- **Autopilot latency reduction**: space and time reduction of {planning/control algorithms, application & kernel layer scheduling, shared memory IPC}. 100s of ms saved in pure scheduling overhead, 100s of ms saved in incidental compute & re-allocated to truly important tasks. Aggressive use of parallelism (userspace process splits, multithreading, SIMD) to minimize the longest path in the compute graph.
- **"Photon to rubber" whole-vehicle latency reduction**: hardware-in-the-loop testing of autopilot on a chassis-dynamometer vehicle, performing safety-critical maneuvers in a 1:1 real-time, pixel-injection simulation. Validate software improvements in closed-course track tests (Model S,X,3,Y, various generations). Indentify & characterize hardware bottlenecks to latency & control bandwidth (incl. vehicle dynamics, powertrain, brakes, comms (CAN/ethernet), compute, etc) and specify parts in-budget for future vehicle programs.
- **SLAM-based localization**: localization for vehicles of any configuration online and offline. *Online*: progressive SLAM, real-time; feature detection by small-footprint neural net, parallel tracking & mapping. *Offline*: larger PyTorch model for feature detection & tracking. Single, global bundle adjustment.
- **Fleet logging & runtime triggers**: Clang-based C++ parser & corresponding code generation framework for zero-copy logging + offline reconstruction, & runtime "reflection" (using static or Turing-complete triggering conditions)
- **End-to-end fleet performance metrics**: from kernel & userspace data collection to dashboards, alerts & reports. Including {measurement, realtime logging & packaging, cloud infra (primarily AWS), batch processing, heavily query-optimized indexing, automated emails}. Includes true in-car & CI-test derived metrics, both at massive scale.
- **"Last-built" CI pipeline**: guaranteed build incrementality & passing mainline tests for developers (minutes to hours saved per-developer-per-day on local or remote builds/tests)
- **Dramatic reduction of uncached build times**, from hours to minutes: primarily C++, reduction of compiler (Clang) front-end and back-end load, w/ measured, minimal impacts to generated code latency in-car (w/ LTO).
- **C++ $\rightleftarrows$ Python interoperability** for single-implementation shared libraries configurable for use online (in-car, C++, small {CPU, memory, latency} budget) and offline (training cluster, Python + PyTorch, large compute budget).
- **Simulation-based eval & regression testing**: use simple Python scripts to evaluate any observable characteristic of autopilot over 1000s of closed-loop simulations or near-100% deterministic open-loop replays of recorded scenarios, on the order of 1-2hrs.

Google *Mountain View, CA*
Software Engineering Intern **Sep 2018 – Dec 2018**

Built auditability into production ad indexers handling petabytes of data/day. ∼100x faster resolution time for system issues & customer complaints (days → minutes). Frontend made diagnosis by non-technical (e.g. sales) staff possible.

US Navy, Naval Surface Warfare Center *Crane, Indiana, USA*
Software Engineering Co-op (Secret Security Clearance) **Spring 2017 – Fall 2017**

Authored new surveillance system deployed worldwide to the US Marine Corps. Refactored underlying storage & reduced latency of a US govt distributed database service. Helped develop/maintain a US govt automation framework.

University of Louisville, Computer Engineering & Computer Science Dept.
Teaching Assistant **Fall 2016, Sum & Fall 2017, Spr & Fall 2018, Spr 2019**

Teaching assistant for undergraduate courses in software design.

Tutor, Computer Engineering & Computer Science **Dec 2017 – Mar 2019**

Tutored students of various undergraduate courses in individual and lecture settings.

Redbird Robotics, University of Louisville **Summer 2018**

Co-captain of team. Built in-house autonomous drone for aerial herding in a GPS-denied environment (computer vision + localization + autopilot). Tested using software-in-the-loop physics simulation.

River City Rocketry, University of Louisville **Summer 2017 – Summer 2018**

Built integrated hardware & software stack for flight visualization. Prototyped a realtime, long range telemetry system. Heavy use of C++ on bare-metal.

Computational Thermal & Fluid Dynamics Laboratory, University of Louisville **Fall 2017**

Designed C software & hardware spec for NASA Blue Origin additive manufacturing (FDM) research payload.

Education

University of Louisville, Louisville, Kentucky, USA

Bachelor of Science: Computer Engineering & Computer Science **May 2015 – Jan 2020**

Programming

C++, Python, Bash, Go, JavaScript (Node.js, React), C#. Heavy focus on testing. At home in a Unix/Linux environment.

Other Skills, Interests

FAA certified private pilot, multi-instrument musician, songwriter. Interested in robust CNS nerve regeneration & genetic cellular reprogramming.