

AJ Chandrasekaran

U.S. Citizen • (248) 719-2472 • ajaay@umich.edu • Novi, MI 48375 • ajaayc.github.io

Summary and Mission

I am a software engineer with 7 years of experience in robotics, automated driving, and healthcare applications, seeking to leverage my industry experience, critical thinking, and problem-solving skills to develop and implement solutions that improve system and process efficiency. I have broad subject-matter experience in various areas of AI and robotics, including machine learning, motion planning, and state estimation. My primary software development experience is in C++ and Python on Linux-based platforms, but I am prepared to learn and leverage any necessary technology.

Experience

Neya Systems

Staff II Robotics Engineer

Aug 2024 - May 2025
Sterling Heights, MI

- Collaborated in teams of 5-10 contractor engineers on autonomous ground vehicle projects owned by the U.S. Department of Defense, implementing various features requested by the U.S. Army Rapid Capabilities and Critical Technologies Office
- Contributed to the RTKLite project, an R&D simulation interface software written with a C++/ROS2/Nav2/TypeScript-based software stack for enabling a soldier to command a Boston Dynamics Spot robot to follow waypoints, explore unknown environments, and navigate via manual teleoperation
- Contributed to the ARCS project, an autonomous robotic C++/ROS2/Python-based software system for enabling a soldier to command U.S. Army ground vehicles to perform reconnaissance tasks

Stellantis

Senior Automated Driving Software Engineer – AI Motion Planning Team, Core Development Team

Dec 2019 - Mar 2024
Auburn Hills, MI

- Collaborated in global organization of ~700 software, systems, calibration, and product engineers to build STLA AutoDrive, an L2 autonomous driving platform, in C++ and Simulink with Bazel build system, Docker development environment, and Git version control
- Dissected and documented 10,000+ lines of esoteric BMW C++ codebase obtained from BMW-Stellantis partnership to help team adapt primary BMW codebase for Stellantis vehicles
- Conducted dozens of in-vehicle tests to diagnose and resolve problems across AUTOSAR/ROS publisher-subscriber stack
- Validated system requirements through simulation of AUTOSAR software components via Google Test (GTest) framework
- Built emergency failover motion planning module to decelerate the vehicle to a stop in case of primary planning module failure, which—on average—facilitated complete braking of the vehicle within ~15 seconds in failure-mode situations on highways
- Facilitated migration of data visualization plugins from RViz to TypeScript-based Foxglove Studio extensions to enhance software diagnostics
- Prototyped and validated trajectory planning algorithms in Python and visualized algorithm execution via Jupyter Notebook
- Trained ~10 new team members and pioneered new training curriculum used by dozens of new software developers, reducing onboarding workloads for team leads and facilitating more rapid involvement of new hires in product engineering efforts

Epic

Software Developer – Epic ASAP Team

Jul 2018 - Aug 2019
Madison, WI

- Migrated several features of electronic medical record software to TypeScript and C# given existing Visual Basic codebase
- Fixed security vulnerabilities and various bugs in SVN version-controlled codebase touched by hundreds of developers

University of Michigan Transportation Research Institute

Graduate Student Research Assistant – Biosciences Group

May 2017 - Jun 2018
Ann Arbor, MI

- Constructed vehicle perception system to monitor the safety and quality of construction materials along the US-23 highway over a 2-year period, merging data from a Novatel GPS, Quanergy M-8 Lidar, and a Mobileye unit
- Developed multithreaded C++ code utilizing VTK and PCL libraries to extract, interpret, and visualize Lidar readings of lane marking visibility between various months of data collection
- Interfaced with a Mobileye unit via vehicle CAN message network to evaluate lane marking quality across several months

Tools

Languages: C++/C, Python, Matlab/Simulink, Java, BASH, SQL, JavaScript, TypeScript, C#, CSS

Software & Libraries: ROS2, ROS1, Bazel, CMake/Make, Git, Emacs, VSCode, Docker, Linux terminal, GTest, GitHub Copilot

Subject Knowledge: LLM design, Planning algorithms, Machine learning, Kalman filtering, Multithreading, Sockets, AUTOSAR

Education

University of Michigan

MSE in Electrical and Computer Engineering, Robotics — GPA: 3.730/4.000

Sept 2013 - Apr 2018

Ann Arbor, MI

BSE in Computer Science — GPA: 3.759/4.000

Relevant Coursework: Autonomous Robotics Laboratory, Self-Driving Cars: Perception & Control, Machine Learning, Motion Planning, Mobile Robotics Algorithms, Robot Kinematics & Dynamics, Software Engineering, Control Systems Design, Data Structures & Algorithms, Operating Systems, Computer Architecture

Academic Projects

Collision Estimation for Safe Planning

Graduate Motion Planning Course Project

Mar 2018 - Apr 2018

- Devised and implemented a Gaussian Mixture Model-based algorithm in C++ to expand upon existing Monte Carlo simulation methods to estimate the probability that a robot following a motion plan would collide with an obstacle
- Simulated a linear feedback controller and a Kalman Filter estimate for robot position via OpenRAVE and Armadillo

Deep Neural Network Vehicle Detection System

Graduate Self-Driving Cars Course Project

Nov 2017 - Dec 2017

- Collaborated with a team to develop a classifier to detect cars in various images captured from GTA simulation
- Utilized TensorFlow Python Object Detector API and NVIDIA CUDA library on AWS EC2 GPU instances to train convolutional neural network (faster RCNN) on ~6,000 training images with known bounding box locations of cars
- Ranked 5th place among 35 teams in class competition by achieving 63% mean absolute error on instructor test set

TrashBot

Senior Undergraduate Capstone – Autonomous Robotics Laboratory Course Project

Mar 2017 - Apr 2017

- Collaborated with a team to build a robot that would navigate unknown urban environments, detect trash, and remove it
- Implemented forward/inverse kinematics and trajectory planning algorithms in Python for controlling a robotic arm to efficiently pick up and dispose of objects detected via 3D point cloud data from a Microsoft Kinect
- Wired up embedded systems parts such as Arduino to fuel and interface with four-wheeled differential drive robot base

KinEval Robotics Framework

Undergraduate Robot Kinematics and Dynamics Course Project

Sept 2016 - Dec 2016

- Built a JavaScript-based robot simulation framework to better understand modeling and control of autonomous agents
- Implemented various robotics algorithms, such as path planning via A*, forward kinematics via homogeneous transforms and quaternions, inverse kinematics via the Jacobian matrix, and motion planning via rapidly-exploring random trees

Cloud-Based Ocular Disease Diagnosis with Machine Learning

Undergraduate Software Engineering Course Project

Jan 2016 - Dec 2016

- Collaborated with a multidisciplinary team and the Kellogg Eye Center to develop a web-based tool, which utilizes a machine learning model to assist clinicians worldwide in diagnosing hundreds of patients' genetic retinal dystrophy cases
- Gathered app requirements from Kellogg Eye Center clinicians and UofM HIPAA Compliance Office and implemented back-end infrastructure with Python-Flask and a PostgreSQL database, ensuring usability and security of patient data
- Enhanced web interface and ML model iteratively via an agile development workflow to maximize product quality
- Communicated project requirements and design processes with Kellogg Eye Center clinicians and UofM HIPAA Compliance group to ensure that clinicians' needs are satisfied