|  |
| --- |
| Sanjar Normuradov |

Seattle, WA | 206-677-1874 | [snn98@uw.edu](mailto:snn98@uw.edu) | [LinkedIn](https://www.linkedin.com/in/sanjar-normuradov/) | [Portfolio](https://sanjarnormuradov.github.io) | [GitHub](https://github.com/SanjarNormuradov)

Dear Hiring Team at ,

I want to express my interest in the .   
As you will find on my resume, I have had a variety of experiences from **Autonomous Robotics** to the **Amazon Manipulation Project**. I am looking to take the next step in my career by leveraging my experience in **Robotics Engineering** to

As a prospective engineer, I immediately can bring the following qualities to the team:

**Autonomous Robotics:** Throughout my graduate course projects, I have successfully implemented various state-of-the-art methods and algorithms on diverse robotic platforms. This includes a 1/10th scale MuSHR rally car, and both 2-DoF/3-DoF robotic arms in a simulation environment.

My work encompassed several key aspects of autonomous robotics:   
● Control: PID / MPC.  
● State Estimation: Particle Filter (PF) / Extended Kalman Filter (EKF) with Odometry/Velocity Motion and LIDAR/Landmark-based Sensor Models.  
● Path Planning: RRT/RRT\*, A\* / LPA.  
● Reinforcement Learning: Behavior Cloning, DAgger, and Policy Gradient.

ROS / PyBullet / MuJoCo were used for simulations, while for an actual robotic platform operation and interface control, I used Linux, Python, and ROS.

Currently, I am helping other enthusiasts to gain hands-on experience as a Teaching Assistant for this course.

**Robotic Manipulation:** During my research assistantship in the Amazon & University of Washington Manipulation Project (UR16e), I have accomplished the following results:

Motion Planning:  
● Optimized pod positioning relative to the robot workstation by automating test-run procedures and refining control and MoveIt parameters, such as collision checking frequency and discretization.

Simulation:

● Enhanced simulation environment precision and robustness by revising URDF/Xacro/XML files and creating a single source for all pod models, thereby expanding the system’s versatility.

Reachability Test:

● Reduced UR16e reachability failures from 20/1600 to 1/1600 via modifications above.

Synthetic Data Generation:

● Expanded synthetic imagery training dataset, from 30K to 140K, generated by utilizing Scanned Objects from Google Research and NVIDIA Scene Imaging Interface to simulate varying real-world scenarios:   
different background configurations (bin ratio aspect ratios and textures, additional neighboring bins);   
various object arrangements (horizontal/vertical stacking, random placement/reorientation).

● Improved the Average Precision, from 0.424 to 0.646 overall and from 0.336 to 0.573 for stacked bins, of our novel framework “Discrete-Frame Segmentation and Tracking of Unseen Objects for Warehouse Picking Robots (STOW)”, built on top of ResNet-50 and utilizing ViT with an additional multi-frame attention layer for discrete-frame tracking.

Baseline Comparison:

● Run training of STOW on the new dataset using HYAK supercomputing clusters with SLURM job scheduler, and tested it along MinVIS, Mask2Former-Video, and VITA in the real UR16e robot to [show](https://openreview.net/pdf?id=48qUHKUEdBf) its superior performance.

Architecture Development:

● Investigated upcoming STOW2’s performance boosts in Instance Segmentation by integrating the Segment Anything Model by Meta AI.

**Embedded Engineering:** In my Bachelor’s Thesis project, I designed a 5-DoF portable robotic arm with diverse object grasping, and an adaptable human hand-like end-effector for in-depth mechatronics study.  
● Mechanical: Utilized CAD and FDM/FFF 3D printing technologies on Tevo Flash/Tornado, employing  
diverse filaments (PLA, PETG, TPU, Nylon), to refine and optimize the robot's frame for functionality.

● Electrical: Designed dual-sided PCBs for the robot's base and controller utilizing Altium Designer,  
fabricated through photolithography, leveraging SLA 3D printer (Anycubic Photon) and Ordl Alpha 350 dry film

● Software: Developed software for the robot base (6 servo motors, nRF24L01 module) and controller  
(nRF24L01, OLED 128x64, 2 joysticks) using STM32F103 MCU and state machines.

**Software Engineering:** In my project of “TetrArm” in C++ during the course “Software Engineering for Embedded Applications”, I achieved the following objectives:

● Gained proficiency in testing code across various scenarios using Docker container images.  
● Implemented a Finite State Machine using user-defined classes and events in ELMA and ENVIRO images.  
● Leveraged STL containers, including vectors, maps, and deques to create custom ADTs.

● Currently, revising the course materials as a Teaching Assistant.

**Machine Learning:** During my graduate coursework in “Machine Learning” at the University of Washington, I have successfully implemented:

● Linear Regression with Polynomial to study Bias-Variance Tradeoff.

● Ridge Regression and Neural Networks for multi-class classification of handwritten digits from MNIST dataset.  
● Lasso Regression to analyze local crime statistics for 1,994 US communities.

**Industrial Robotic Engineering:** During the Airbus Robotics & University of Washington ENGINE Capstone Project: “Robotic Fuselage Inspection for Dents and Scratches”, I accomplished the following objectives:  
● Set up workstations in Gazebo, RoboDK, and RoboGuide with UR5e, UR16e, and Fanuc CRX 20-iA/L  
● Generated motion plans using MoveIt and other built-in packages to automate the inspection.  
● Completed on-site Core Training and online Core, Advanced, Application Training for UR e-series

Sincerely,

**Sanjar Normuradov**