Tomas' Mechatronics Projects Portfolio
Cal Poly SLO | Robotics / Mechatronics
tofranco@calpoly.edu | 650-722-9645 | github.com/TomasEzFranco

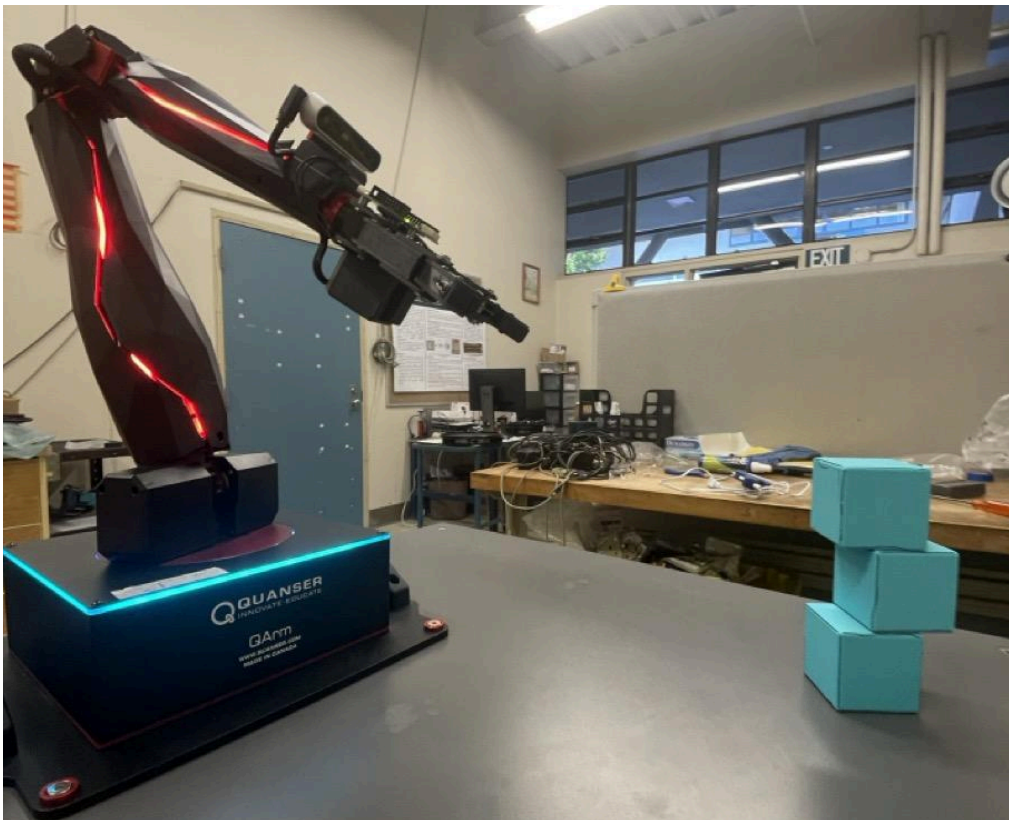
Thesis: LLM-led Autonomous Pick-and-Place Pipeline (In Progress)

Video Demo link: v6_6cubes_placed_and_validated_full_run.MOV

Summary: For my thesis, I'm developing a robust autonomous manipulation pipeline using RGB/depth sensing, current sensing, encoders, and YOLO instance segmentation. The platform I'm using is a 3.5 DOF Quanser Qarm, the same robot used in Healbot, because I felt it didn't get enough reps in last time.

The primary tasks are cube color sorting and stacking tasks based on text prompts. Initial grasping failures led to current-based grasp checks, integral and proportional gain tuning for smoother centering on detected objects, pre-grasp scene scans to avoid blindly placing cubes into occupied positions, and motion safety limits to keep the robot from getting too creative.

The broader goal is to show how far an LLM can generalize a robotic manipulation pipeline while still keeping the robot grounded through perception, validation, and safety checks. Since LLMs can hallucinate, the robot gets multiple verification layers before, during and after actions instead of trusting it completely.



Autonomous Driving Pipeline with MXCarkit

Video link: https://youtube.com/shorts/HvuyGzoIKhY?feature=share

Summary: For our final project in an autonomous vehicles course at Cal Poly, our team built a full autonomous driving pipeline on the MXCarkit platform, a small vehicle with an onboard computer and multi-sensor stack. The goal was smooth lane following and accounting for traffic lights.

We built two parallel pipelines in Docker: a primary behavioral cloning model based on PilotNet that mapped camera images directly to steering and speed commands, and a YOLOv11 backup for traffic light detection as a safety net. We collected driving data with Foxglove, ran 10 training trials across different architectures and hyperparameters tracked in Weights and Biases, and compared RGB vs. grayscale inputs. The color variant won as grayscale introduced lag and overshoot during turns. Our best model hit strong steering accuracy and the YOLO backup reached 0.98 F1 across red, yellow, and green classes. The YOLO model was exported to ONNX and integrated into a ROS2 node that published speed commands in response to detected light states, all written in Python.

It was a great end-to-end experience, from raw data collection to training neural networks and validating models in simulation before live hardware deployment on the car.

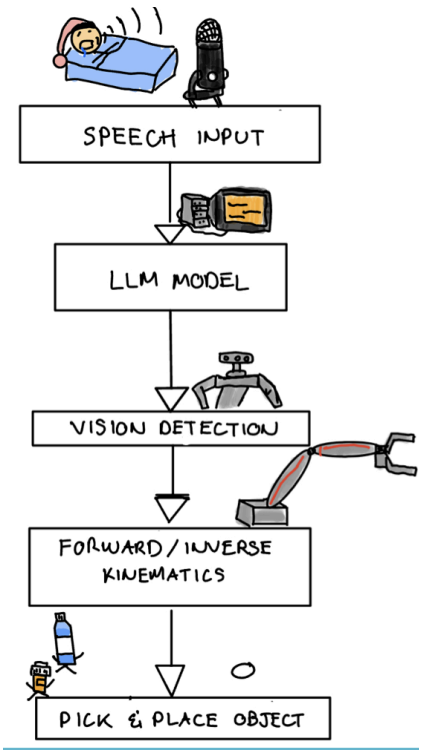
Healbot

Project Link: https://github.com/TomasEzFranco/Healbot

Video Link: https://www.youtube.com/watch?v=xKzFN9P7FqU&feature=youtu.be

Summary: Healbot is a robotic assistant to help people with mobility limitations retrieve important items like water, medicine, or an energy drink based on their symptoms or request. If a patient says they are thirsty, our robot will quickly retrieve water for them, as long as we remembered to stock the very sophisticated robot pantry.

Built on the Quanser Qarm educational robot platform, this robot manipulation pipeline takes speech input, processes it through an LLM, uses YOLO for object detection, and performs pick and place operations at different waypoints using forward and inverse kinematics. Please enjoy the fun pipeline diagram below.

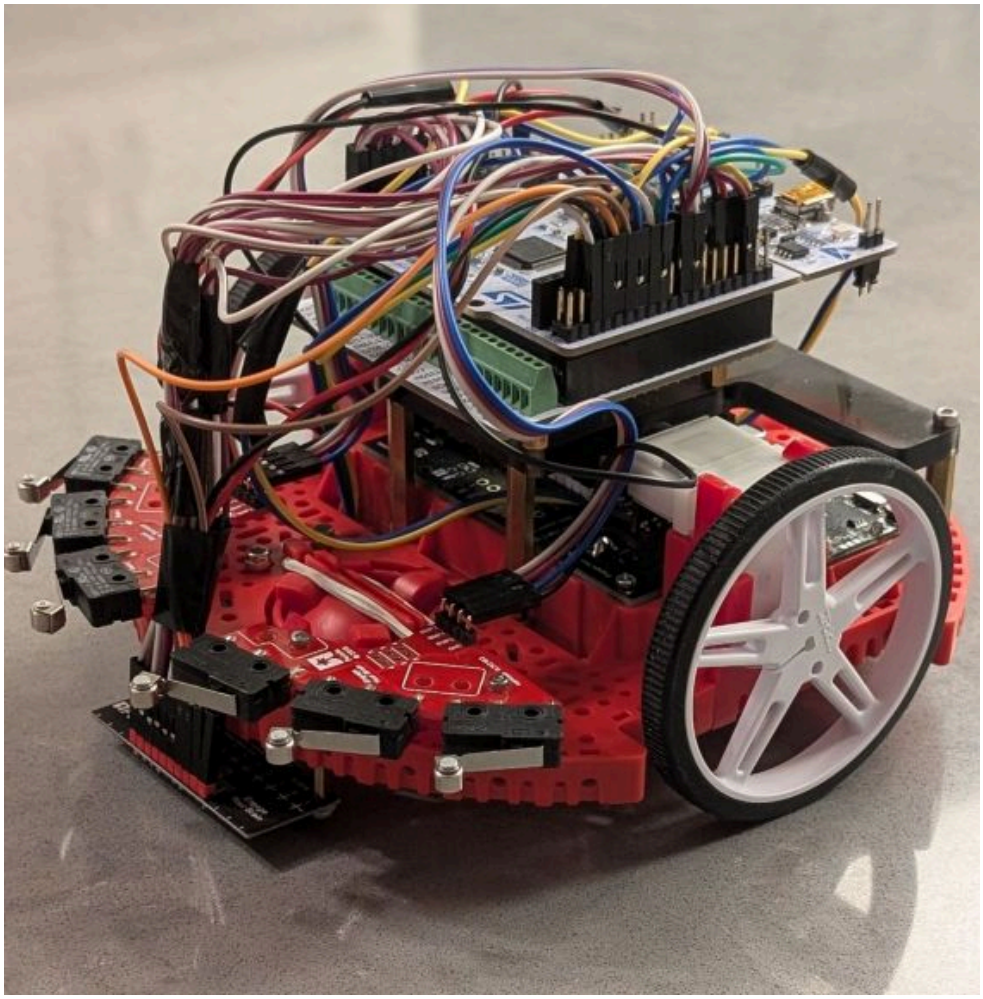


ROMI

Project Link: https://github.com/TomasEzFranco/Romi_Line_Follower
Video Link: https://www.youtube.com/watch?v=yved8C4mFfU

Summary: ROMI is a line-following differential drive robot that had to navigate a game track, hit checkpoints and detect a wall without getting lost. We handled part selection, assembly, wiring diagrams, schematics, sensor integration, and firmware.

ROMULUS, as we called it, used an IR sensor array for line following, encoders for tracking wheel motion, and a BNO055 IMU for the parts of the track where the line disappeared and the robot had to believe in itself. We also characterized the motors and used closed-loop control with IR, encoder and IMU feedback rather than open-loop moves. The final system had line following, heading control and dead reckoning with bump sensing to complete the course consistently.



DAL-I

Project Link: https://github.com/TomasEzFranco/Robot-Aided-Drafter

Summary: DAL-I is our attempt at a robotic drafting robot. Unlike most printers which follow a 2P or 3P design, DAL-I uses an RRP robot architecture because apparently we wanted a challenge.

We designed the mechanical assembly, PCB, wiring diagrams and software needed to bring DAL-I to life. I worked mainly on electrical design, board bring-up and software development. While DAL-I's art may not make it to the Louvre anytime soon, it was a great experience taking a project from idea to a working physical system.