Portfolio: Selected Relevant Projects

Aidan Sussman

Dear Hiring Committee,
I am a passionate Mechanical and Mechatronics Engineer with a deep enthusiasm for robotic systems, their design, control, and real-world application. Through my academic journey at UC Berkeley (B.S. May 2024, M.S. May 2025) and hands-on experiences at Tesla, UC Berkeley Labs, and the competition teams, I have developed expertise in CAD (SolidWorks, CATIA, Fusion 360), FEA/FEM, control algorithms (PID, UKF, EKF, PF), embedded programming (C/C++, Python, ROS), and pneumatic and electro-mechanical system integration. My attached resume provides a concise overview, and the following detailed portfolio highlights key projects that demonstrate my ability to innovate, prototype, and deliver robust robotic solutions. I look forward to discussing how my skills and passion can contribute to your team.

Date Range	**Description**
May 2025	M.S. Mechanical Engineering, UC Berkeley (awarded May 16 2025); **Graduate GPA 3.776**
May 2024	B.S. Mechanical Engineering *with Minor in Electrical Engineering & Computer Sciences*, UC Berkeley (awarded May 10 2024); **Undergrad GPA 3.675**; Regents' & Chancellor's Scholar Finalist; Dean's List; Honors to Date
Feb 2025 – Now	STEM Tutor at Bay Area Tutors both High School and College Level Courses (Subjects: Physics, Calculus, Matlab, Lower Level Math)
Aug 2022 – Dec 2022	Equipment Engineering Intern, Tesla (Fremont, CA)
Fall 2021 – Spring 2022	Combat robotics team member (RoboBears, UC Berkeley): designed and built a 3 lb combat robot
Aug 2021 – Dec 2022	SMV (SuperMileage Vehicle Team) Steering Lead, designing the steering system for competition vehicle.
Aug 2020 – May 2021	Undergraduate researcher in exoplanet habitability (ULab, UC Berkeley)
Jun 2016 – Aug 2023	STEM instructor at ASTEME Learning Center (non-profit STEM School and Camp)

Automated Pneumatic Tooling – *Tesla Equipment Engineering Internship*

Project Overview

Facilitated creation of a fully automated press to install coolant pump mounting brackets, replacing a manual choke point and improving cycle time and takt on the assembly line.

Mechanical Design Highlights

Custom fixture matched bracket and pump profiles for positive seating; SLA printed in durable engineering resin for rapid replacement. Extrusion rail frame with linear guide carriage on a sliding tray table for part loading/unloading. 3D printed pressing foot tailored to allowable touch-off points, with integrated indicator for seating verification.

Process Workflow

Operator loads parts into fixture on tray, then presses two safety buttons. Tray retracts into the semi-enclosed chamber, triggering the pneumatic piston to press-fit the bracket. Upon seating confirmation, releasing buttons reverses sequence: piston retracts, then tray extends.

Pneumatic Logic & Control

Implemented full pneumatic air logic instead of electronic controls apporach. Dual button in series acts as an "AND" gate enabling tray actuation; end of travel switch activates pressing piston via a 5/2 valve. All sequencing done with 5/2 and 3/2 directional valves, pilot lines, and standard factory air at regulated to 60–80 psi.

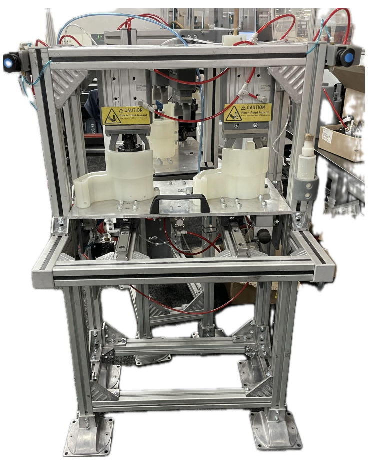
Safety & Ergonomics

Pinch point guards and full acrylic panels enclose moving elements. Buttons ergonomically designed with two hand operation. Warning labels and color coding compliant with Tesla safety standards.

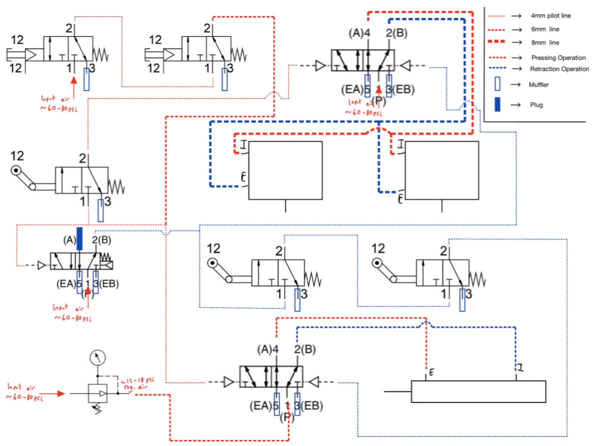
Outcomes & Scalability

Cycle time reduced below takt, eliminating the previous bottleneck. Machine footprint accommodates two fixtures in parallel; maintenance guide includes instructions for scaling multiple units off a single air logic bank.

Figures



(a) Final Press Assembly



(b) Pneumatic Air Logic Control Schematic

Figure 1: Automated pressing tool mechanical design and control

RoboBears Combat Robot

Overview

Design, source, and build a battlebot style 3 lb combat robot with at least one active weapon to destroy another combat robot while minimizing incoming damage.

Initial Design Process

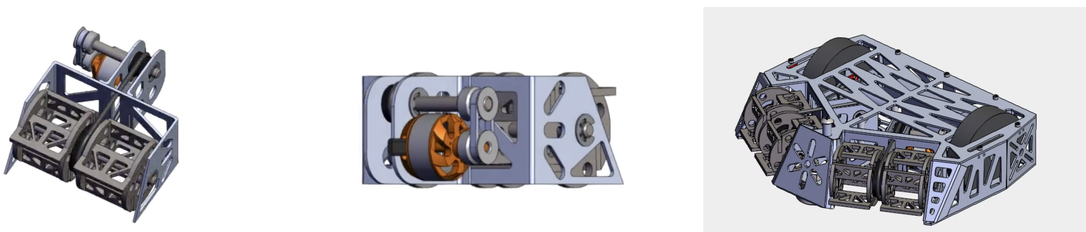
I put together a design matrix ranking the importance elements based on research of previous battlebot competitions, isolating what elements of robots were most effective. Overall I prioritized high maneuverability, invertibility, stability, and damage. The resulting general bot design was a drum bot with a slanted flipper front for better contact and bite. A flat drum bot with good stability would be easily defendable and the concentrated mass of the drum would make for a powerful weapon that wouldn't need a lot of spin up time. I sketched up a generalized design and CAD to get a rough design started.

Modifications

One vulnerability with traditional front-facing drum bots is poor defense and difficulty making contact. To overcome this, I designed the weapon system with two drum segments at a 120° angle, enabling side attacks and 360° defense by rotating to face the opponent. Initial single-motor designs with roundbelts experienced slippage; switching to dual weapon motors improved input reliability. Although dual sections slightly reduced per-hit force, the increased defensibility compensated for the trade-off.

Final Design and Manufacturability

The chassis is laser-cut from 6061 aluminum sheets with pocket patterns for stiffness-to-weight optimization and welded assembly. Weapon drums are waterjet cut from A2 tool steel, welded to the live shaft, and constrained with nylon springs. The drive train uses two direct-drive 750 RPM planetary gear motors, with a front ball caster for invertibility. Low-friction tape on the caster bolt allows self-righting when flipped.



(a) Drum Weapon Assembly (b) Drum Weapon Transmission (c) Final CAD Overview

Figure 2: Key CAD renders of the weapon system.

Contributions and Takeaways

I utilized SolidWorks' workspace and FEA tools for structural analysis, optimizing pocket patterns to reach target weight. Abstracted FEA simulations of drum disk impact allowed precise load placement and magnitude distribution on teeth. I also sourced all components and compiled a Bill of Materials and manufacturing plan.

Mechatronics Design Capstone Project – *ME102 UC Berkeley*

Project Overview

Conceived a flywheel–based module that lets bicycles self-correct lean at low speeds, utilizing a gyroscopic assist. System integrates sensing, computation, and actuation in a compact add-on and can be attached to almost any bicycle.

Core Concept & Architecture

Inverted-pendulum flywheel stores angular momentum; brushless DC motor re-positions the wheel to deliver counter-torque and re-center the bike. Real-time feedback loop: magnetic encoder measures lean angle and ESP32 computes corrective action then sends commands to VESC which drives the motor.

Key Design Decisions

Selected a brushless DC motor (Scorpion SII-4035-450 KV) over brushed for higher power density & efficiency. Designed a steel flywheel using Lagrangian energy model to supply stabilizing torque while keeping weight manageable. Maytech 100 A VESC controller, optical quadrature encoder, steel disk flywheel using precisely balanced weights positioned to create a large moment of inertia.

Embedded Control & Software

Firmware (C++/Arduino) implements state-variable current control on ESP32; PID-like gains tuned via bench testing. State machine governs startup, balancing, fault shutdown; modular encoder and IMU drivers for future features.

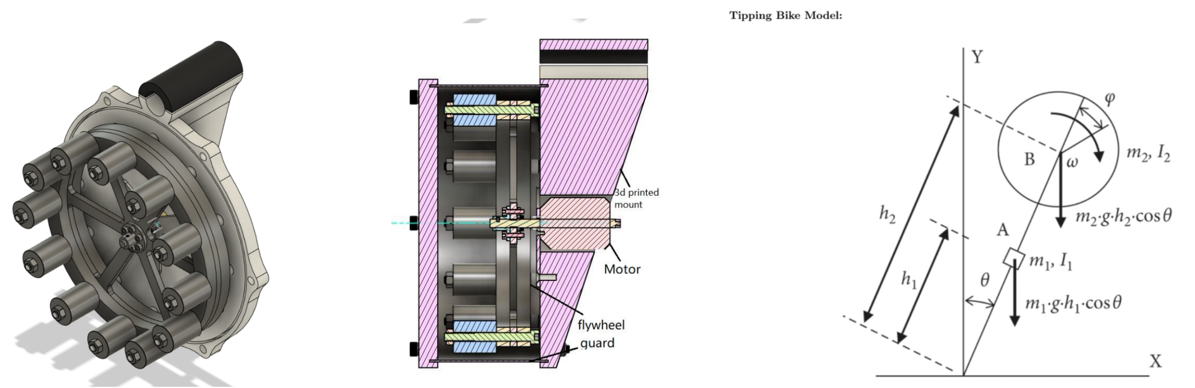
Accomplishments

Demonstrated live low-speed balance correction that counters gusts, uneven terrain, and rider shifts. Fabricated, wired, and programmed a fully functional prototype within 6 weeks, concluding with a successful showcase demo.

Challenges & Lessons

Encountered VESC latency and control saturations; higher-resolution sensors and refined motor-control schemes would allow for better response and recovery.

CAD & System Model



(a) Flywheel Assembly CAD (b) Section View CAD (c) Inverted Pendulum Model

Figure 3: Prototype CAD and dynamic model

ChessBot – ***EECS SawyerBot Lab, UC Berkeley***

Project Goal

Utilize the Sawyer armed robot to play chess against a human: detect the human's move, update the board state, compute the robot's move with Stockfish, and execute it autonomously.

Physical Design

All pieces are uniformly 3D printed for reliable Sawyer gripping. AR markers on board and clock enable camera-based pose estimation.

Perception

ROS's 'ar_track_alvar' node detects AR tags and computes the board's pose in the camera frame. We map board squares into Sawyer's workspace for precise targeting.

Game Logic

A ROS node maintains the chess state. After each human move, we call the Stockfish engine via a custom ROS service to plan the robot's move.

Motion Planning & Control

We convert target piece-and-square commands into Sawyer joint trajectories with ROS moveit, an Inverse Kinematic solver, then execute them with a custom-tuned PID controller on joint velocities for smooth, accurate placement.

Results

Sawyer autonomously plays full games, moving its chosen pieces accurately and demonstrating seamless integration of perception, planning, and control.

System Setup

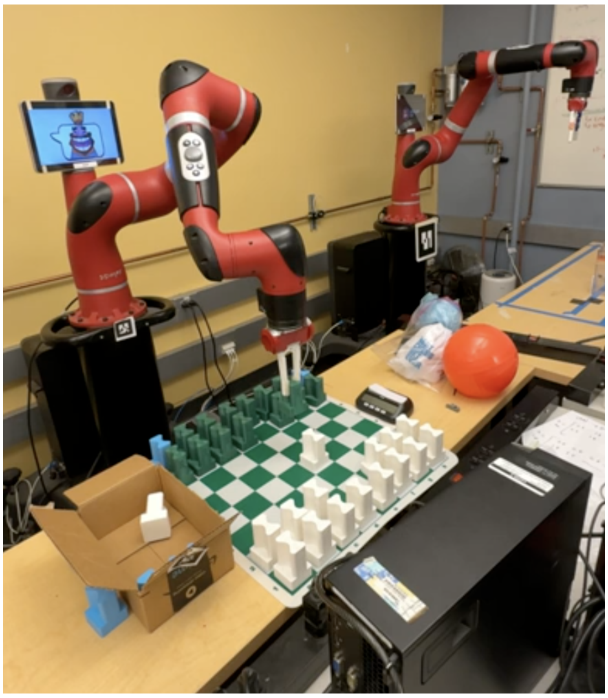


Figure 4: SawyerBot Chess Demo

CNC Pen Plotter – *Design of Microprocessor-Based Systems, UC Berkeley*

Project Overview

Developed a CNC pen plotter that translates digital drawings into precise, disconnected line art. Integrated a custom pen retraction mechanism to enable rapid tool lifts for complex imagery.

Mechanical Design Highlights

Assembled an open-frame XY gantry utilizing extrusion rail with dual stepper drives and DRV8825 microstepping for smooth motion. Added limit-switch homing and drag-chain cable management for reliability. Employed compact 3D-printed brackets and datum features for repeatable assembly.

Pen Lifter Mechanism

Custom servo-driven slider converts rotary motion into vertical pen lift via a linkage puller arm. Idle torque of the servo and pen compression eliminated need for a return spring. Adjustable mounting bracket features chamfered pegs, datum holes, and threaded inserts for rapid prototyping and precise alignment. Integrated limit-switch stopper enforces consistent travel.

Hardware & Integration

ESP32-based control board with Wi-Fi and FluidNC firmware and DRV8825 drivers. Pen lifter mounts on the Z-axis carriage; X/Y axes driven by GT2 belts and T-slot sliders.

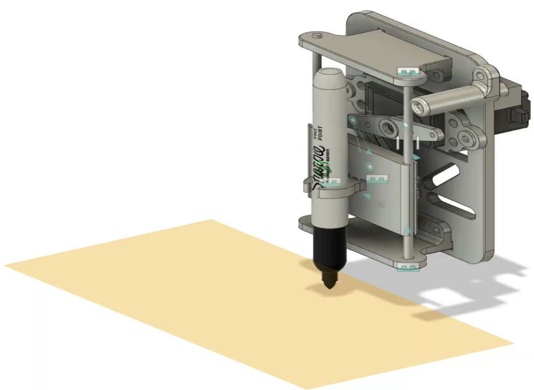
Software & Firmware

FluidNC on ESP32 for motion control; Python/Tkinter GUI for G-code generation, real-time plotting, and low-level commands (E-stop, home, reset).

Outcomes

Achieved clean, disconnected strokes on whiteboard and paper. Manual tool-swap snap-fit posts support multiple pen sizes—foundation for future auto-changer. Real-time feedback via MQTT ensures synchronous drawing.

Figures



(a) Pen lifter mechanism CAD

(b) Final assembled plotter

Figure 5: Mechanical design and integration

Ultrasonic Mapping Car – *Electronics for IoT, UC Berkeley*

Objective

Design and build a small-scale rover that combines object detection, obstacle avoidance, and environmental mapping to demonstrate core IoT robotics principles.

Core Hardware

ESP32 microcontroller running MicroPython Two 9 V DC motors driven by custom NPN-BJT H-bridge with flyback diodes Wheel encoders (20 slots/wheel) for odometry HC-SR04 ultrasonic sensor for distance measurement 9 V battery with on-board 5 V regulator and logic-level shifting

Software & Communications

PWM-based motor speed control with encoder-interrupt feedback. MQTT publish/subscribe for drive/turn/scan commands and sensor-data output. Host-side Python polar-plot visualization; MQTT Explorer for debugging.

Key Functions

- **Drive:** Moves set distance via encoder-pulse counting.
- **Turn:** Single-wheel rotation for heading control.
- **Scan:** 360° spin sampling ultrasonic ranges, streaming points for live mapping.

Challenges & Solutions

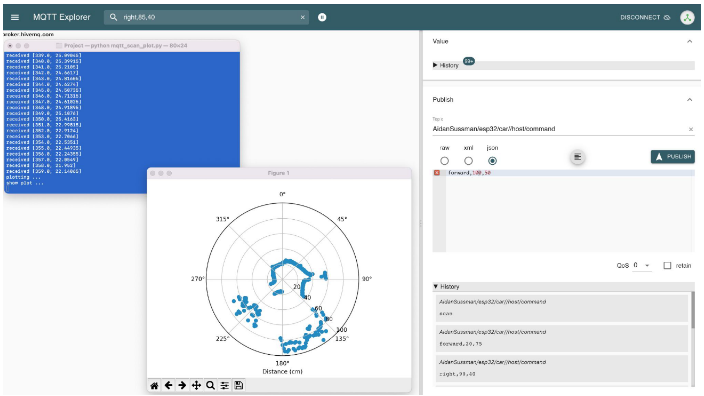
Resolved voltage-level issues with regulator/divider network. Mitigated coarse encoder resolution (5°/pulse) by time-based sampling during scans. Replaced ThingSpeak with direct MQTT workflow to eliminate latency.

Outcomes

Reliable remote control and real-time obstacle visualization. Modular, scalable IoT architecture—applicable to search-and-rescue or inspection robots. Enhanced skills in embedded coding, sensor fusion, motor control, and data pipelines.

Results

(a) Ultrasonic mapping rover



(b) MQTT telemetry dashboard

Figure 6: Rover hardware and real-time control interface

MATLAB Beam-Analysis Project - ***Solid Mechanics, UC Berkeley***

Project Goal

Automate shear, moment, and stress analysis of a simply-supported beam under a point load $F = 100\,\mathrm{N}$ and a uniformly distributed load $w = 10\,\mathrm{N/cm}$, then size a circular cross-section so that maximum principal stress stays below $270\,\mathrm{MPa}$.

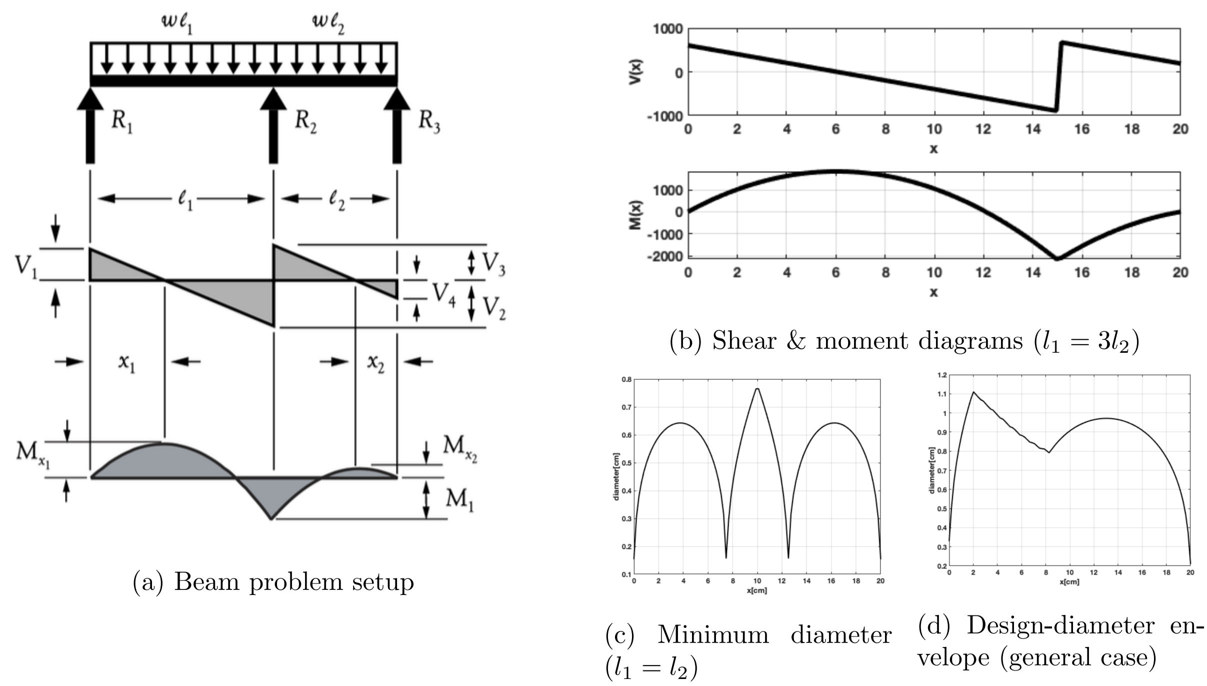
Major Deliverables

- Hand-derived shear-force and bending-moment diagrams for validation.
- Four MATLAB scripts:
  - `vmd.m`: computes $V(x)$ and $M(x)$ along beam for span l_1.
  - `stress.m`: evaluates shear, bending, and principal stresses for a given diameter.
  - `caseEqual.m`: bisection algorithm to find minimum diameter when $l_1 = l_2$.
  - `caseGeneral.m`: sweeps l_1 (2–10 cm) to generate a design diameter envelope.
- Auto-generated plots: shear & moment diagrams, the minimal-diameter diagram when $l_1 = l_2$, and the design-diameter envelope.

Analytical & Computational Workflow

Calculated support reactions via equilibrium; implemented closed-form relations $V(x)/M(x)$ in MATLAB sampled at 100 points. Constructed principal-stress tensors using $\sigma = My/I$ and $\tau = VQ/It$, extracting $\sigma_{\max}$ with `eig`, and used bisection to size diameter so $\sigma_{\max} \leq 270\,\mathrm{MPa}$ ($\pm 0.01\,\mathrm{MPa}$).

Matlab Plotting Results



(a) Beam problem setup

(b) Shear & moment diagrams ($l_1 = 3l_2$)

(c) Minimum diameter ($l_1 = l_2$)

(d) Design-diameter envelope (general case)

Figure 7: Beam analysis results

Manufacturing Design Project - *Manufacturing Communication, UC Berkeley*

Overview

Design and prototype a slide-out side table device that clamps beneath a desk to expand workspace. Applied formal engineering drawing, manufacturing processes, and tolerancing (ASME Y14.5M-1994) to create full working drawings and a prototype.

GD&T Highlights

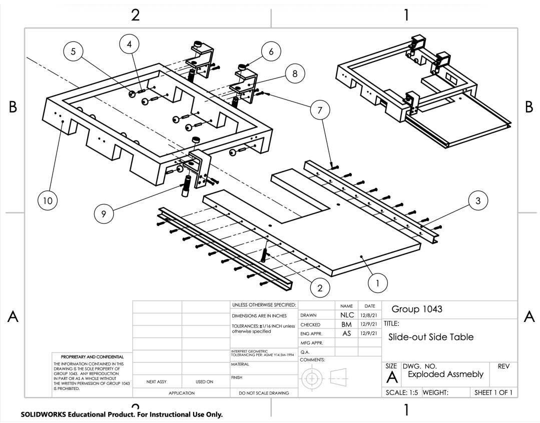
- **Position Tolerance (1/32 in)** on offset wheel holes to ensure consistent stopping feature while allowing minor variation.
- **Flatness Tolerance (1/64 in)** on tray edges to guarantee smooth sliding between base and wheels.
- **Total Runout** on wheel bore relative to pin to maintain roundness and uninterrupted wheel rotation under load.

Fits and Tolerances

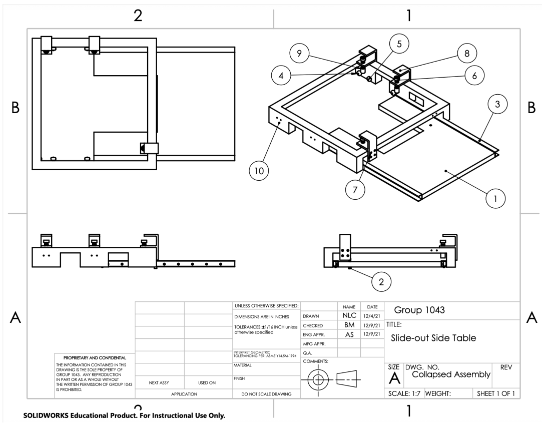
Table 2: Key fit classes (ASME) used

Fit	Components	Class
Clearance	Wheels & Rails	RC7 (free running)
Interference	Pin & Base	FN2 (press fit)
Adjustable	Clamp screw & Frame	RC1 (tight clearance)
Locking	Clamp frame & Desk	LN1 (transition)

Key Drawings



(a) Exploded assembly



(b) Collapsed assembly

Figure 8: Assembly drawings

Process Selection and Materials

Prototyped clamps via FDM/SLA 3D printing (engineering resin for strength, PLA for wheels), MDF tray table with drilling/sawing, and aluminum rails. In production, recommend metal screws and higher-grade wood for durability and finish.

Reflection

Tolerancing profoundly impacts part function; precise GD&T ensures assembly fit and performance. Clear labeling and orientation of drawings proved critical during assembly, and prototyping underscored the mantra "measure twice, cut once."

Acoustic Positioning System Lab – ***EECS 16A, UC Berkeley***

Project Goal

Build a GPS-style indoor acoustic positioning system that estimates a microphone's 2D location from beacon tones played by multiple loudspeakers.

Core Concepts Applied

Cross-correlation to isolate each beacon's signal and measure its sample delay. Time of Arrival distance conversion: delay × speed of sound ($\approx 343m/s$). Trilateration/multilateration via geometric circle intersections or least squares.

Key Implementation Steps

Recorded composite audio from four speakers broadcasting orthogonal tones. Computed cross-correlation peaks to find sample offsets. Converted offsets to distances and solved the least-squares system in Python. Visualized estimated vs. ground-truth locations and iterated to reduce error.

Tools & Technologies

Python - Jupyterlab, NumPy, standard PC microphone and desktop speakers.

Skills Demonstrated

Digital signal processing scripting and debugging. Applied linear algebra modeling and least squares optimization. Hardware/software integration troubleshooting. Clear technical presentation of algorithms, math, and experimental results.

System Setup

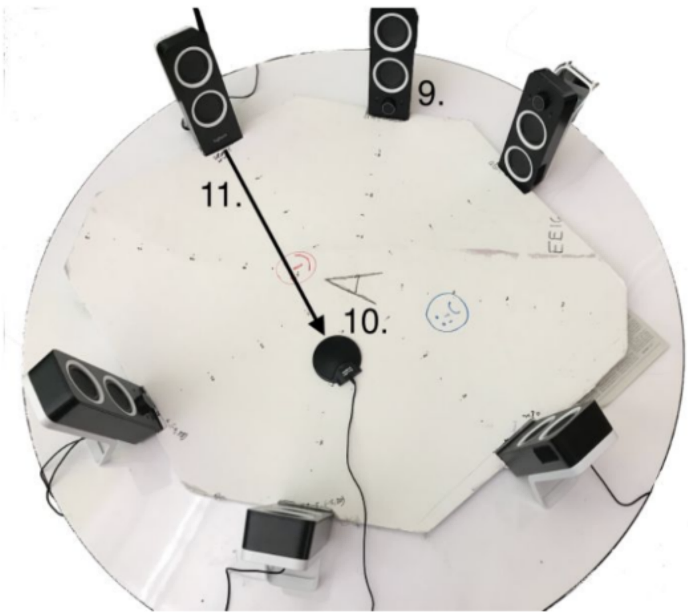


Figure 9: Lab setup: four speakers emit beacon tones for microphone localization

Characterizing Exoplanet Habitability

Abel Castaneda, Anmol Desai, Aneesh Goel, Ashley Maritato, Aidan Sussman
University of California Berkeley, Undergraduate Lab at Berkeley.
Daniel Klyde* Derek Kaplan** Yi Zhu†
*Mentor **Assistant Mentor †Lab Director

Introduction

In modern astronomy, the study of exoplanets is a key way of learning about our own solar system. Currently, the most common method of identifying planets outside of our solar system is through the transit method. Some of the most critical characteristics for identifying the habitability of an exoplanet are its stellar effects, its location within its solar system, and its composition. Throughout our project we have tested the habitability of an exoplanet by analyzing its transit. By using already existing databases, we identified a good candidate for the study of habitability and used the remote telescope, MicroObservatory, to more closely analyze whether the exoplanet fits the criteria for possible habitability. With both the data found through existing databases and the data collected through the MicroObservatory we will compare the characteristics of the selected exoplanet and an ideal habitable exoplanet and determine the extent of the selected exoplanet's habitability.

Background

The transit method works by taking images of a star and collecting the brightness of the star before, during and after an exoplanet orbits in front of the star. This causes a dip in the brightness of the star which can be analyzed to determine characteristics of the exoplanet.[4]

In addition to this, many of these exoplanets are in a "habitable zone" which is the area around a star in which a rocky, Earth-like planet can possess and sustain liquid water on its surface which is critical to life.[9]

Another method used to check an exoplanet's habitability is through transmission spectroscopy. The key idea behind transmission spectroscopy is that transit depth is wavelength dependent. The absorption of certain wavelengths by atoms and molecules in the atmosphere of the exoplanet affect the amount of stellar flux absorbed[4].

Methodology

The first step in performing the experiment involved picking which exoplanet we wanted to study, from those available on MicroObservatory, a network of telescopes.

Next, we had to collect transit data through MicroObservatory. Specifically, we request data for HAT-P-12b on a night when the transit is visible. Through the request, we received 80-90 images individually and we were able to locate the target star HAT-P-12, two reference stars, and two reference dark spots. For the target star, MicroObservatory measured the relative brightness of the star over a period of about 4 hours.

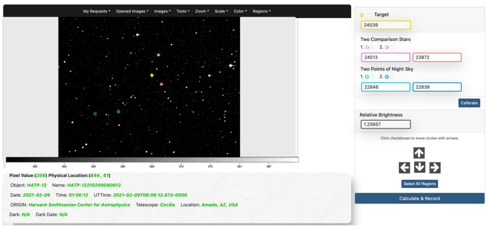


Figure 1: MicroObservatory control panel for measuring brightness of our star. Image shown is HATP-12210209080312 (Image #2/77) taken on 02/08/2021 at 1:00AM in Arizona.

Next, we used Python to clean the data we collected by MicroObservatory and create the transit light curves. Additionally, through the brightness data collected, we were able to determine exoplanet habitability through calculations (orbital radius, transit period, planet radius). The following equations were used:

$$Drop = \frac{r^2}{R^2} \quad \text{axis} = \sqrt[3]{M_s T^2}$$

$$M_{exo} = (2.1 \cdot 10^{-4}) \frac{M_s V_s T}{2\pi a}$$

$$T = \frac{n}{t} \quad \rho = \frac{M}{V}$$

In order to understand the extent to which the exoplanet was habitable, our team had to find an exoplanet that was confirmed as ideally potentially habitable and then compare the characteristics between that planet and the one that we observed and did calculations for. Once this was complete, we were able to make a *limited* prediction on how habitable our exoplanet of study was.

Analysis

Transit Analysis:

Once we collected our data from MicroObservatory, we exported the data into Jupyterhub in order to analyze the data using python.

We began by sorting our data into arrays and plotting it into a relative brightness vs. time graph, using the matplotlib library. With our data plotted we highlighted the images that had substantial noise and were creating outliers during the expected transit (predicted by MicroObservatory). We then used numpy's polyfit with degree 2 to help highlight the transit (Figure 2).

With our outliers identified we removed them and then highlighted the data points inside and outside of the transit (*Figure 3*). We then used python to average the brightness during the transit and outside of the transit and used that information to calculate the dip (*Figure 5*). Lastly we binned our data by a factor of two to illustrate the transit curve clearly (*Figure 4*).

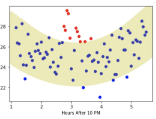


Figure 2: Python plot of data points highlighting outliers

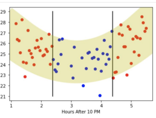


Figure 3: Python plot of transit data points with transit times

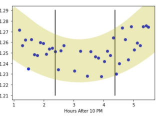


Figure 4: Python plot of transit with binned data

Average Relative Brightness Outside of Transit	1.257102083333333
Average Relative Brightness During Transit	1.2451172413793101
Dip	0.01198484195402294

Figure 5: Average Brightness and Dip calculations

Comparison Between Exoplanets:

	HAT-P-12B[2][5]	Barnard's Star B (Super Earth)[7]	Jupiter[3]	TRAPPIST-1f (Ideal Potentially Habitable Planet)[7]	Earth[1]
Radius (km)	4.5e4(±3.0e3)	2.22e4	6.99e4	6.66e3	6.38e3
Velocity (km/s)	135.39(±10)	3.02	13.07	434.75	29.78
Semi Major Axis (km)	5.7e6(±1.0e6)	6.04e7±(2.69e6)	7.78e8	5.5e6	1.496e8
Eccentricity	0	0.32	0.0489	0.063	0.01671
Period (days)	3.213(± 2.1e-06)	232.8(±0.4)	4,332	9.2	365
Mass (kg)	3.987e26	1.93e25(±1.42e24)	1.898e27	4.061e24	5.9724e24
Type	Gas Giant	Confirmed to be Rocky	Gas Giant	Confirmed to be Rocky	Rocky
Star Type	K4	M4.0V	G2V	M8V	G2V

Figure 6: We compiled a chart comparing characteristics of HAT-P-12B, a super Earth, Jupiter, an ideal potentially habitable planet, and Earth.

Radius: HAT-P-12b has a radius of 4.5e4(±3.0e3), which is much larger than Earth—as well as the Super Earth Barnard's Star B—and aligns more with the radius of Jupiter, which has a radius of 6.99e4.

Velocity: HAT-P-12b's velocity is significantly faster than all of our comparison planets, with the exception of Trappist-1f, an exoplanet widely considered potentially habitable. When compared to other planets, we noticed HAT-P-12B is nearly 4.5 times faster than Earth, over 10 times faster than Jupiter, and about 45 times faster than Barnard's Star B.

Semi-Major Axis: HAT-P-12B has a semi-major axis of 5.7e6(±1.0e6). The habitable zone for HAT-P-12 is between 4.9052603e7 km and 1.03317554e8 km[2], which puts HAT-P-12B outside the habitable zone, as it is too close to its star. Earth is the only planet in our solar system within the habitable zone, as Mercury and Venus are too close to the sun and Mars and the outer planets are too far from the sun. TRAPPIST 1-f, though it has a similar semi major axis to HAT-P-12B is within the habitable zone for the TRAPPIST system, as its star is a M8V type, which is much cooler than HAT-P-12, a K4. Barnard's Star B, on the other hand, borders the snowline of the habitable zone, in which water will not be liquid, making it unlikely to be habitable.

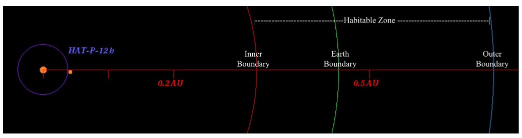


Figure 7: A visualization of HAT-P-12B's orbit in comparison to its star's habitable zone.

Eccentricity: An ideal habitable planet will have a low eccentricity, as a nearly circular orbit lessens the temperature fluctuations on the surface of the planet. HAT-P-12B has an eccentricity of 0 which essentially means it has a circular orbit. Earth, while not having a completely circular orbit, has a low eccentricity of 0.01671. Jupiter and TRAPPIST-1f also have relatively low eccentricities, at 0.0489 and 0.063 respectively. Barnard's Star B has a much higher eccentricity, with 0.32.

Period: HAT-P-12B has an extremely short period of approximately 3 (Earth) day—substantially shorter than all of our comparison planets, with the exception of Trappist-1f. The closest related period would be Trappist-1f with an orbital period of about 9 days.

Mass: In comparison to Earth and Trappist-1f, we noticed that HAT-P-12B is two orders of magnitude more massive. When compared to our super earth model (Barnard's Star b), HAT-P-12B was found to be one order of magnitude higher. HAT-P-12B was slightly smaller than a magnitude less massive than Jupiter, making it close in mass to a small gas giant like Jupiter.

Results

Through our research we can conclude that our exoplanet, HAT-P-12B is not likely habitable. In fact, our exoplanet more closely resembles a Jupiter-like gas giant than Earth. We've found that our exoplanet's relatively fast velocity is likely a direct result of its relatively high mass and large size along with its proximity to its host star. These factors lead us to best model our exoplanet after a small gas giant, as it most directly relates to features shown by Jupiter. Additionally since our host star is a K type, our exoplanet is well below the habitable zone making it far too hot and too close to its star, exposing it to dangerous amounts of ultraviolet rays. We did find that HAT-P-12B had a near perfect circular orbit, which would be a beneficial factor for habitability, as a less eccentric orbit keeps surface temperatures more stable. Unfortunately, that would be the only habitable factor in HATP-12B's favor. Based on the current factors that we have looked at, HAT-P-12B doesn't uphold enough characteristics of our habitable models in order to be classified as potentially habitable.

Figure 8: Artist rendering of HAT-P-12b

Limitations of our Experiment

It is important to consider the constraints of our analysis on exoplanet habitability for HAT-P-12B. Specifically, the exoplanet's atmospheric conditions and compositions were not considered. Atmospheric composition can dictate biosignatures that can allude to possible habitability. Another limitation of our experiment includes that we only compared one prospective habitable planet to a confirmed one. This sample size is too small to make any large conclusions or to see a greater (more holistic) trend in how planet composition and characteristics affect their habitability. Lastly, there was also the component of human error when we created our dataset from the images and our software was not the most accurate either.

Future Work

In order to reduce the limitations of our experiment, we would use spectroscopy to determine the atmospheric composition and see if the planet could support different forms of life. We can also look at a lot more exoplanets and use the 'Earth Similarity Index' or some sort of habitability index to quantify the probability of habitability given certain exoplanet characteristics and composition[8]. For example, if we take the presence of oxygen as a measure of habitability, we could look for oxygen or indicators of oxygen in our spectroscopy data. Lastly, considering the atmospheric evolution of exoplanets over time and tides will also be critical in improving the accuracy of characterizing exoplanet habitability[9].

References

1. Earth By the numbers - NASA. (2020, June 26). Retrieved from https://solarsystem.nasa.gov/planets/earth/by-the-numbers/
2. "HAT-P-12." Exoplanet Kyoto, www.exoplanetkyoto.org/exohtml/HAT-P-12.html.
3. Jupiter By the numbers - NASA. (2018, January 12). Retrieved from https://solarsystem.nasa.gov/planets/jupiter/by-the-numbers/
4. Kreidberg, Laura. "Exoplanet Atmosphere Measurements from Transmission Spectroscopy and other Planet-Star Combined Light Observations." arXiv preprint arXiv:1709.05941 (2017).
5. Planet hat-p-12 b. (n.d.). Retrieved from http://exoplanet.eu/catalog/HAT-P-12_b/
6. Planet trappist-1 f. (n.d.). Retrieved from http://exoplanet.eu/catalog/trappist-1_f/
7. Ribas, I.; Tuomi, M.; Reiners, A.; Butler, R. P.; et al. (2018-11-14). "A candidate super-Earth planet orbiting near the snow line of Barnard's star" (PDF). Nature. Holtzbrinck Publishing Group.
8. Earth similarity index (esi). (n.d.). Retrieved from http://phl.upr.edu/projects/earth-simi--larity-index-esi
9. Exoplanets: Detection, HABITABILITY, BIOSIGNATURES. (n.d.). Retrieved from https://depts.washington.edu/astrobio/wordpress/research-areas/exoplanets-detection-ha bitability-biosignatures/

Acknowledgements

We would like to thank Ronan Alam, Katie Lamar, Aditya Sengupta, Carrie Zuckerman, and Prof. Dan Kasen.

Berkeley, CA
(310)-746-6330
ajsuss29@berkeley.edu

Aidan Sussman

Mechanical Engineering Graduate

linkedin.com/in/
Aidan-Sussman

Education

University of California, Berkeley: M.S. Mechanical Engineering, GPA: 3.776 **May 2025**

- **Graduate Coursework:** Dynamics & Control of Autonomous Flight, Graduate Introduction to the Finite Element Method, Graduate Introduction to Continuum Mechanics, Advanced Special Topics: Flight Mechanics, Advanced Control Design II, Engineering Aerodynamics, Statistics & Data Science for Engineers

University of California, Berkeley: B.S. Mechanical Engineering, Minor in Electrical Engineering and Computer Science, GPA: 3.675 **May 2024**

- Regents' and Chancellor's Scholar Finalist; Dean's List; Honor's to date
- **Relevant Coursework:** 3D Modeling for Design, MATLAB for Engineers, Solid Mechanics, Dynamics, Mechanical Behavior of Materials, Manufacturing and Design, Experimentation and Measurements, Thermodynamics, Heat Transfer, Fluid Mechanics, Physics I/II, Multivariable Calculus, Linear Algebra and Differential Equations, Designing Information Devices and Systems I and II, Data Structures, Intro to AI, Microprocessor-Based Systems, Intro to Robotics, Mechatronics Design, Dynamic Systems and Feedback, Product Development

Technical Skills

Programming & Software	MATLAB, Python, C/C++, Java, Robot Operating System (ROS), CAD (SolidWorks, CATIA, Creo, Fusion 360, AutoCAD), FEA (SolidWorks Sim, ANSYS, MATLAB FEM, structural/thermal), Git, LaTeX
Mechanical & Fabrication	CNC machining, 3D printing, waterjet, waterjet, pneumatic systems, rapid prototyping, additive + subtractive manufacturing, Design for Manufacturing (DFM), Design for Assembly (DFA), GD&T

Selected Technical Projects

Real-Time EKF Bicycle State Estimator, Graduate Control Design, University of California, Berkeley **Jan 2025 – May 2025**

- Built a MATLAB Extended Kalman Filter that fuses steering, speed, and intermittent GPS data to estimate bicycle pose in real time; ran in ≈15ms per 30s ride data, placing in the top 10% of all course submissions.
- Derived nonlinear bicycle kinematics, implemented analytical Jacobians, and grid searched noise covariances to balance drift versus correction, achieving sub meter position accuracy over 100 test rides.
- Compared EKF, UKF, and particle filter approaches, selecting EKF for significantly lower compute cost with comparable accuracy; documented findings in a technical report and live demo.

Quadcopter Autonomous Flight Control, Flight Dynamics Lab - Prof. Mueller, University of California, Berkeley **Sept 2024 – Dec 2024**

- Built C++ firmware in a Ubuntu VM (VirtualBox) and Eclipse IDE; calibrated the powertrain by fitting PWM to RPM and thrust (ω^2) models, then implemented force command converters used in all downstream controllers.
- Designed a complementary filter attitude estimator and cascaded PID loops for rate, angle, and hover control; step response tuning delivered stable hover with < 5 cm altitude drift and fast 0 to 30° attitude commands.
- Fused IMU, optical flow, and range sensor data for full 3D state estimation, then executed a fully scripted mission take off, navigate to (1m, 0m, 1.5m), and soft land all within 25s during the course competition.

ChessBot - Sawyer-Arm Chess Player, EECS Robotics Lab, University of California, Berkeley **Aug 2023 – Dec 2023**

- Integrated Sawyer robot with Stockfish via ROS service node to autonomously play full chess games against human opponents.
- Implemented enhanced AR tag recognition using a head camera for accurate chessboard and piece positioning, and developed custom 3D-printed chess pieces for optimal robot interaction.

Voice Controlled Car (S1XT33N), EECS 16B Project, University of California, Berkeley **Jan 2022 – May 2022**

- Built a battery powered two-wheel robot that recognizes four spoken commands via an onboard PCA/SVD audio classifier and executes precise drive or turn command with encoder feedback.
- Modeled motor dynamics from encoder data and tuned cascaded PWM controllers, reducing straight line drift to <2cm over 2m demo runs.
- Integrated custom circuitry and firmware for signal processing and robotic control: analog microphone, filter bank, custom motor drivers, encoders, and classifier/controls code; achieved 100% command execution success in end of term showcase.

Industry Experience

Equipment Engineering Intern, Tesla – Fremont, CA **Aug 2022 – Dec 2022**

- Designed and validated new equipment assemblies using SolidWorks, ensuring seamless integration in the General Assembly of Model S and X vehicles, and developed control logic for large scale pneumatic systems, verifying function and safety constraints.
- Created datum tooling and features in fully automated areas, and validated the geometry and dimensions of existing tooling using CATIA to ensure proper fit and eliminate potential out of spec displacements, leading to improved reliability of installed parts and reduced downtime.
- Improved the overall manufacturing process by identifying and resolving equipment issues, implementing mutilation prevention measures, and developing operator assistance tooling, resulting in increased efficiency and improved quality.

Lead STEM Instructor, ASTEME Learning Center – Los Angeles, CA **Jun 2016 – Aug 2023**

- Created 30+ K-8 lessons on robotics, 3D printing, aeronautics, and AI; maintained lab of 3D printers, laser cutter, and drones.
- Managed $2k lesson budget and kept supply costs under target through vendor negotiation and custom kit design