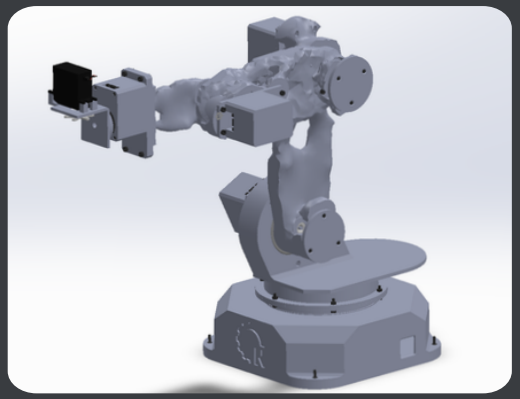
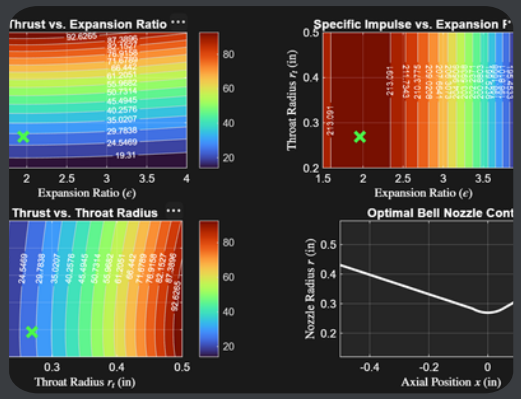
Nathaniel Cavazos

Mechanical Engineering Portfolio

Mechanical Engineer Student at California State University, Long Beach
B.S. Mechanical Engineering | Expected May 2028





Contact Info:
Phone - +1 661-778-8069
Email - nathaniel.cavazos.me@gmail.com
LinkedIn -linkedin.com/in/nathanielcavazos2007
Portfolio - nathaniel-cavazos.com

3 – Technical Skills & Organizations

4 – Six Degrees of Freedom Robotic Arm

4.1 - Requirements & System Architecture
4.2 - Design
4.3 - System Analysis
4.4 - Iterations
4.5 - Testing
4.6 - Software Development
4.7 - Results

5 – Cold-Gas Thruster Study

5.1 - Requirements
5.2 - MATLAB Program Development

CAD/CAE:

- SolidWorks
  - Static Analysis
  - Topology Analysis
- Ansys
  - Static Analysis
  - Topology Analysis
- Siemens NX
- MATLAB
- Simulink

Engineering Software:

- Common Components Workbench
- RSLinx
- BOOTP
- Excel

PCB Design:

- KiCad

Embedded System:

- STM32
- SPI
- UART
- I2C
- Sensor Integration

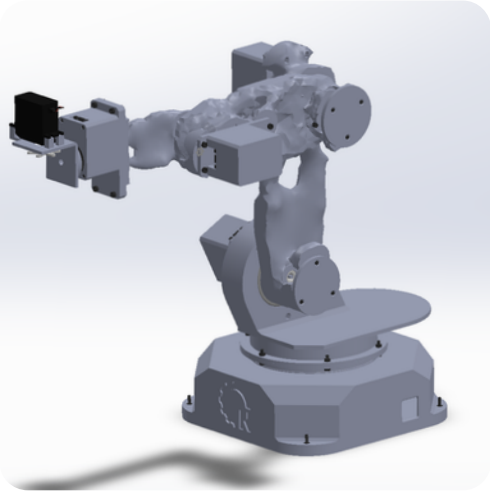
Relevant Coursework:

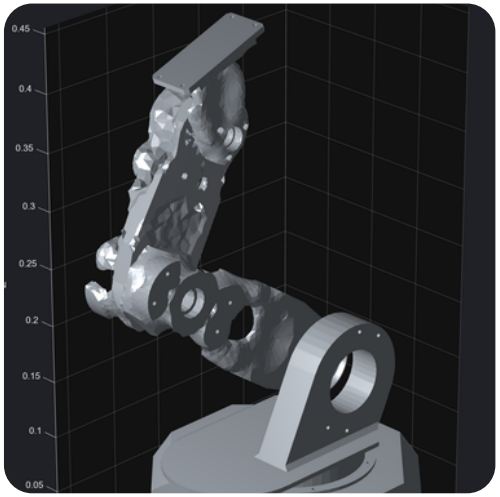
- Applied Math
- Electricity & Magnetism
- Statics
- Dynamics
- Deformable Bodies
- Fluid Dynamics
- Engineering Materials
- Numerical Methods

Organizations:

- Beach Launch Team
- SASE
- Pi Tau Sigma

6-Degree-Of-Freedom (DOF) Robotic Arm





Project Overview: For Version 1, I focused on designing, optimizing, and simulating a 6-DOF robotic arm. I developed this project to explore mechanical design, structural analysis, kinematics, and automation. Using SOLIDWORKS, I designed the mechanical components and joint assemblies, and I used ANSYS to perform structural analysis and topology optimization. I then used MATLAB's Robotics System Toolbox to develop a kinematic model and simulate the first four joints' movement. Future versions will focus on physically building the robotic arm and developing a functional end effector.

Tools: SolidWorks • Ansys • MATLAB • Robotics System Toolbox

Requirements:

- 6 Independant controller axes
- Inverse & Forward Kinematics System
- Joint Angle Limiters
- Homing System
- Safety Fault Systems
- Max Load of 4lb for 1 minute & 30 seconds
- Minimum Reach of 16 in
- Mechanical Grippers
- A Factor of Safety >= 2.0
- Optimized Strength-to-Weight Ratio
- End Deflector Position < 5mm
- Yield Strength on All Joints< 8 MPa

System Architecture:

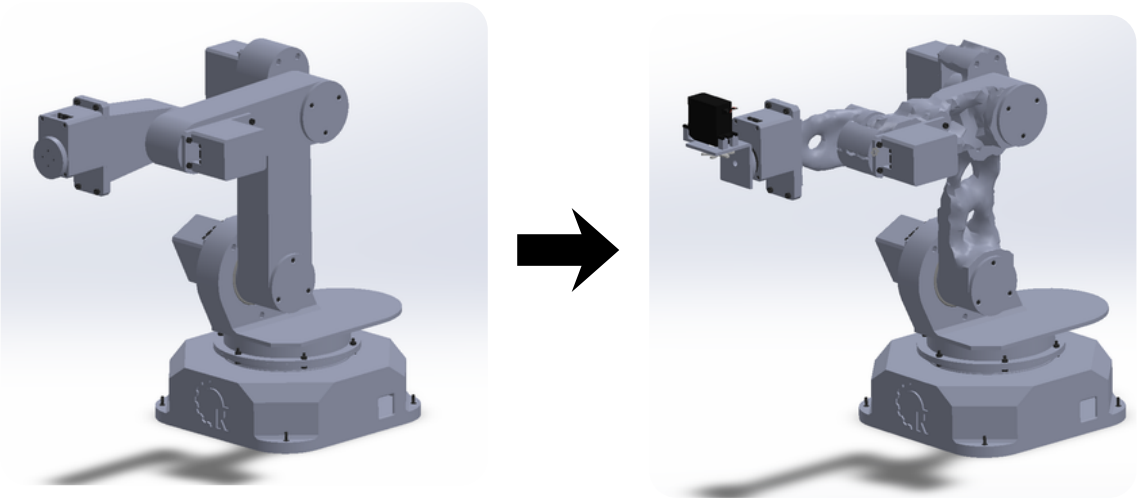
For version 1, I divided the project into three main stages: mechanical design, structural optimization, and kinematic simulation. I first used SOLIDWORKS to develop the robotic arm's components and joint assemblies. I then used ANSYS to analyze the structural performance of the links and optimize their geometry. After completing the design, I imported the optimized links into MATLAB's Robotics System Toolbox to define the joint positions, rotation axes, and transformations. This allowed me to simulate the robotic arm's movement and evaluate different joint configurations.

Future versions will focus on integrating the physical actuators, implementing joint controls, and developing the mechanical gripper.

SolidWorks - Mechanical Design → **ANSYS - Structural Analysis & Optimization** → **MATLAB - Kinematic Modeling & Simulation**

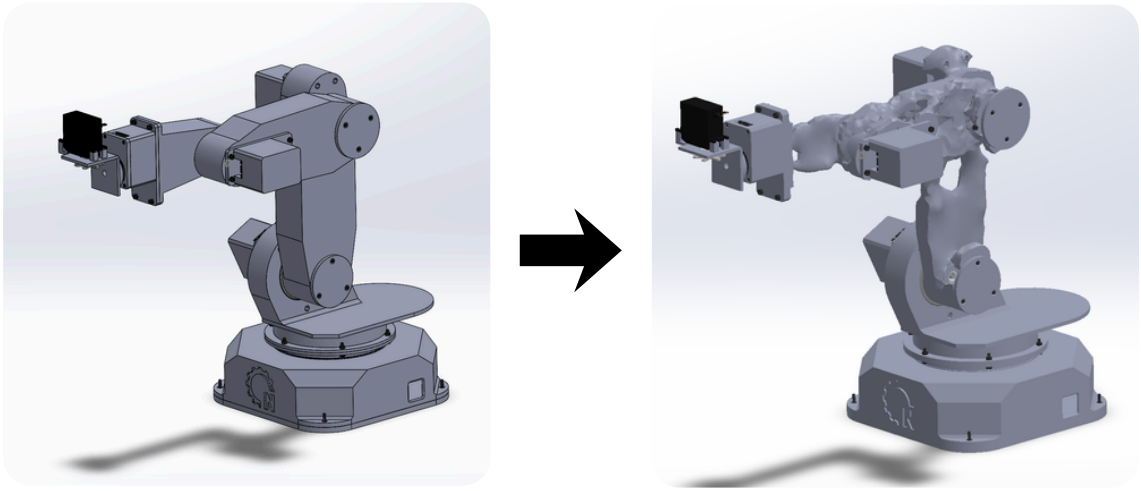
Design Consideration:

Design 1:



(Pre-Optimization) (Optimized)

Final Design:



(Optimized) (V1 Final)

During the design phase, I kept my engineering requirements in mind as I worked through several robotic-arm problems. These included non-concentric mounting holes, joint alignment, and components that would be difficult to manufacture. I originally hollowed out the links to reduce weight, but after starting the topology studies, I realized I was removing material before fully understanding where I needed it. I went back to a solid design and used topology optimization in ANSYS to identify the main load paths under the estimated maximum loading conditions. The resulting geometry helped me better understand how the loads traveled through each link and where material was needed.

Another design consideration was making the robotic arm easier to assemble and maintain. My original idea was to use 25 lb magnets to hold the joints in place and simplify mounting. However, I realized this could become a problem since I was only allowing two magnets per joint. To improve the mounting design, I added four 3.2 mm holes to each joint, so M3 screws could prevent unwanted rotation around the joint's center axis.

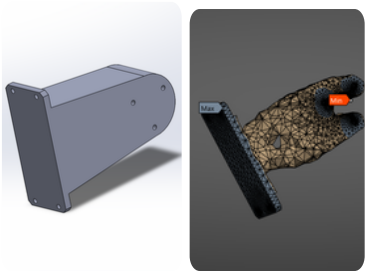
For the final V1 design, I used ANSYS topology optimization to reduce each link's mass while preserving key mounting features, including screw holes, nut pockets, and bearing inserts. I also adjusted the optimized geometry to make the components easier to manufacture and assemble in future versions.

System Analysis:

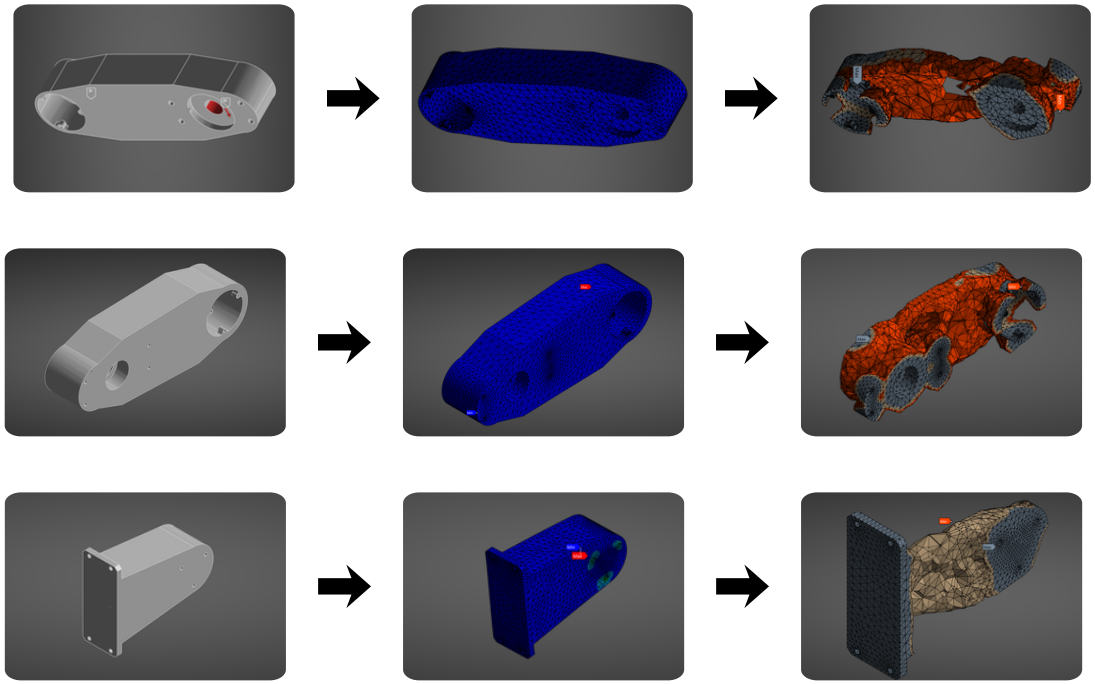
Design 1 Optimization:


(J1 to J2 Link)


(J2 to J3 Link)


(J4 to J5 Link)

Final V1 Design Optimization:


(Raw) (FEA) (Optimized)

As I continued developing each subassembly, I used ANSYS to perform structural analysis and topology optimization on the robotic arm's links. I created a custom PETG material profile using Elegoo's material properties and considered how FDM printing would affect the final parts' strength. Since 3D-printed components are anisotropic, I also considered the printing orientation so the main load paths would rely less on interlayer strength.
For each link, I used a target factor of safety of 2.0 when evaluating the estimated loading conditions. After the initial structural analysis, I used topology optimization to identify areas where I could remove material without significantly affecting the main load paths. I aimed to retain approximately 40% of the original mass while preserving important mounting features.

After each optimization study, I used the resulting geometry as a reference to refine the CAD models and improve their surfaces. I then checked the updated designs against the stress and deflection requirements to evaluate whether the optimized geometry could be used for the final V1 assembly.

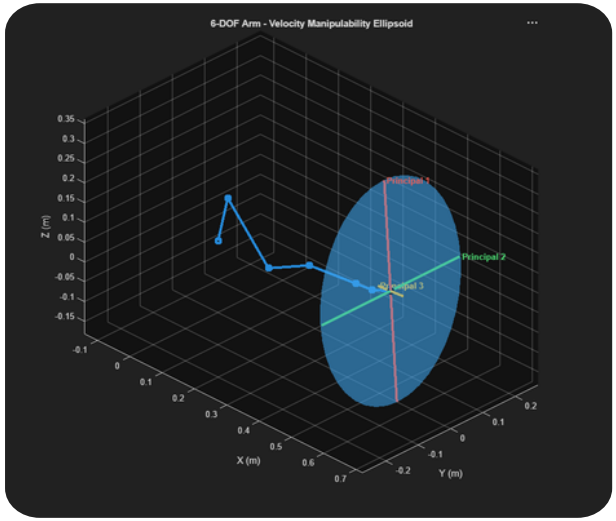
PETG Profile:

- $\rho = 1260\ kg/m^3$
- $E = 1.072\ GPa$
- $\nu = 0.39$ (PETG assumption)
- Tensile >= 51 MPa
- Flexural = 64 MPa
- Impact = $4.8\ kJ/m^2$

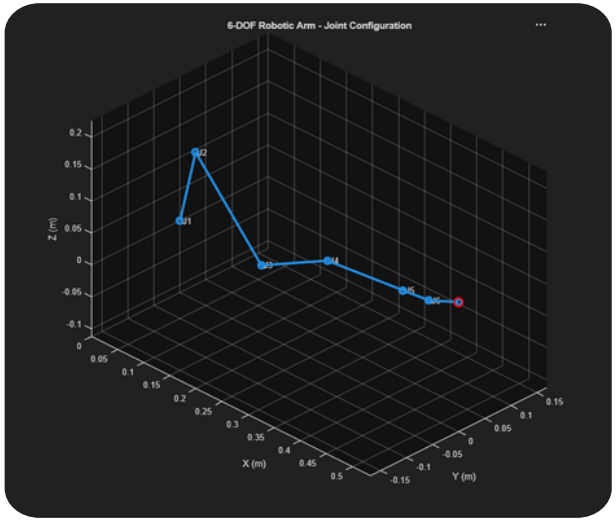
Iterations:

	Problem	Design Change	Rationale
Rev. A	J5 Servo did not fit properly inside the housing.	Reduced the main housing height by 4mm.	Allows the servo to sit flush with the mounting flange, improving fit and alignment.
Rev. B	Joint flanges exhibit excessive predicted deflection > 5mm.	Reduced the M3 insert hole diameter to 3.2mm	Increases the surrounding material thickness to improve flange stiffness and reduce defelction.
Rev. C	J4 servo does not contain a magnetic connector.	Added mounting hole for magnet and re-optimized J4 to J5 link.	Adding the magnetic connector allows for easy maintenance and assembly in the future.
Rev. D	Improper flange engagement at J4 connecting link J4 to J5.	Added longer mounting legs.	Adding longer legs stabalizes the joint while screwing in the magnet.
Rev. E	J2 flange has improper tolerance with bearing.	Switched 0.6mm tolerane to 0.2mm tolerance.	Shortening the toleance ensures proper engangement along the entire inner diameter of the bearing.

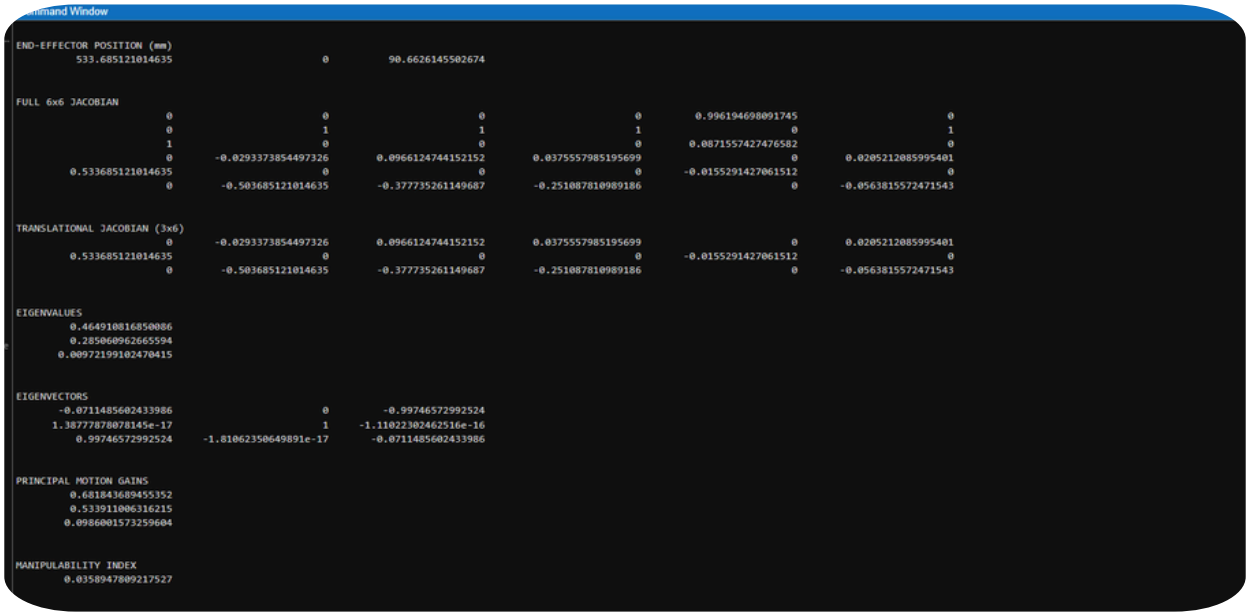
Calculations For All Six Joints: Joint Angles [0 45 -70 20 0 -15]



(Figure 1)



(Figure 2)



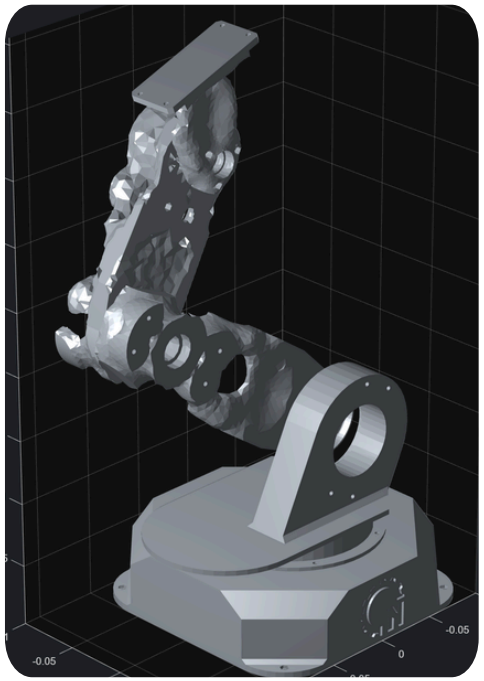
(Figure 3)

Performing kinematic calculations was an important part of understanding how my robotic arm moves. Using MATLAB, I calculated the Jacobian matrix to determine how changes in joint angles affect the end-effector's velocity and position. I then used eigenvalue and eigenvector analysis to identify the principal directions of movement and how effectively the arm could move in each direction.

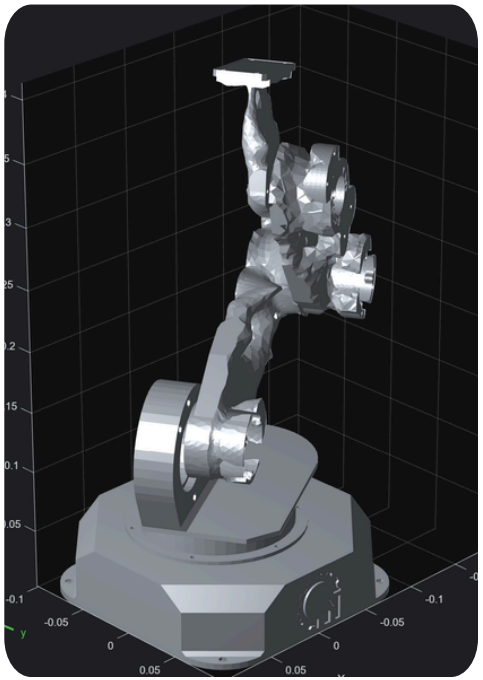
Using the joint angles [0 45 -70 20 0 -15], I calculated the Jacobian matrix and its eigenvalues and eigenvectors. Figure 1 shows the velocity manipulability ellipsoid, where the longer axes represent directions with better motion capability and the shorter axes represent more limited movement. Figure 2 shows the resulting joint configuration, while Figure 3 displays the calculated Jacobian, eigenvalues, eigenvectors, and principal motion gains.

Simulation:

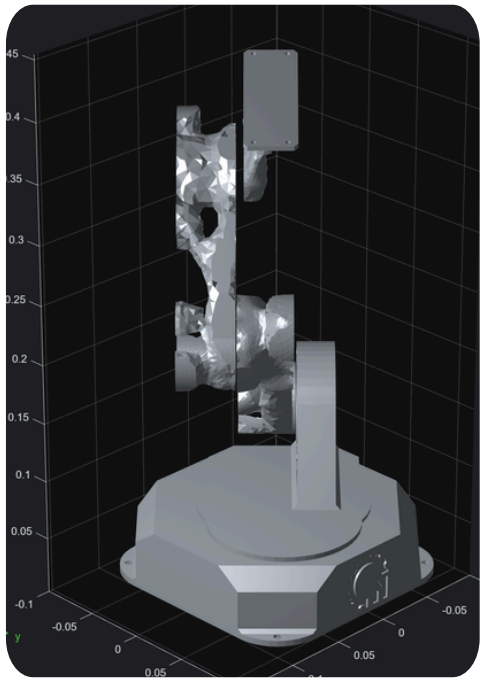
To simulate the robotic arm's movement, I used MATLAB's Robotics System Toolbox to develop a kinematic model using my optimized SOLIDWORKS components. I imported the STL files and manually configured the joint positions, rotation axes, and transformations. Once the links were aligned, I simulated the motion of joints J1 through J4 using different joint-angle configurations. I also implemented smooth motion interpolation to gradually move the arm between positions and plotted the joint angles throughout the simulation. This let me visualize the arm's range of motion, see how each joint affects overall movement, and evaluate the arm's kinematic behavior.



[90 55 -85 50] Degrees

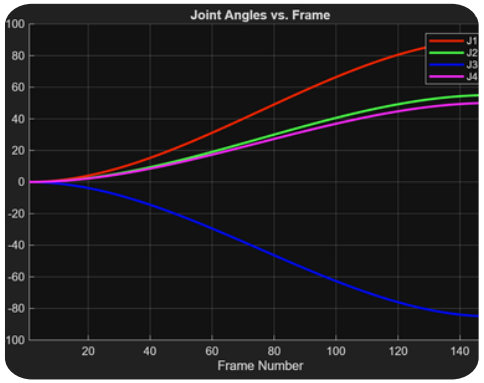


[-60 65 -105 55] Degrees

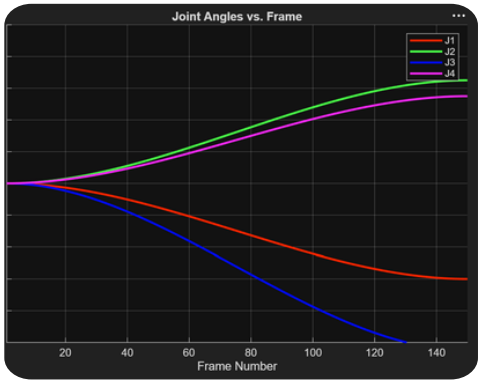


[135 50 -90 70] Degrees

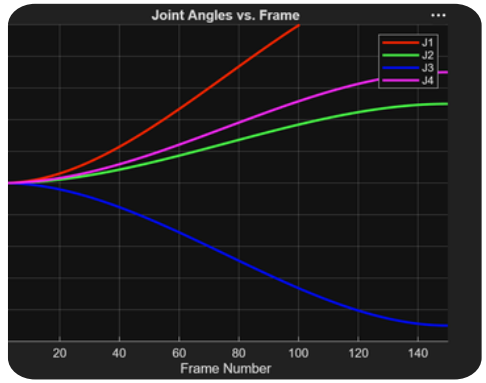
(MATLAB Simulated Joint Angles)



[90 55 -85 50]



[-60 65 -105 55]



[135 50 -90 70]

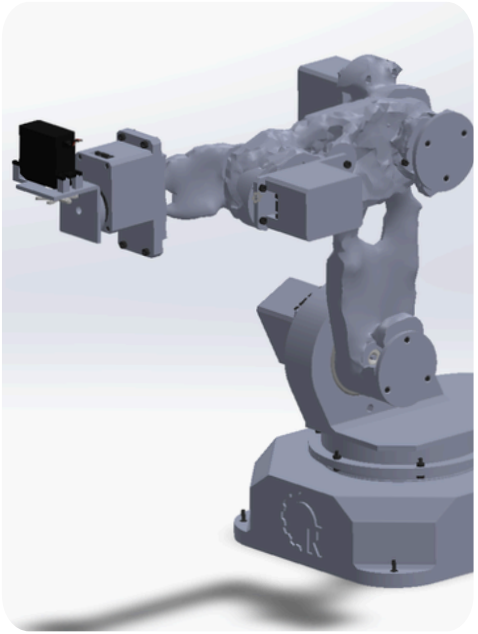
Results & Future Development:

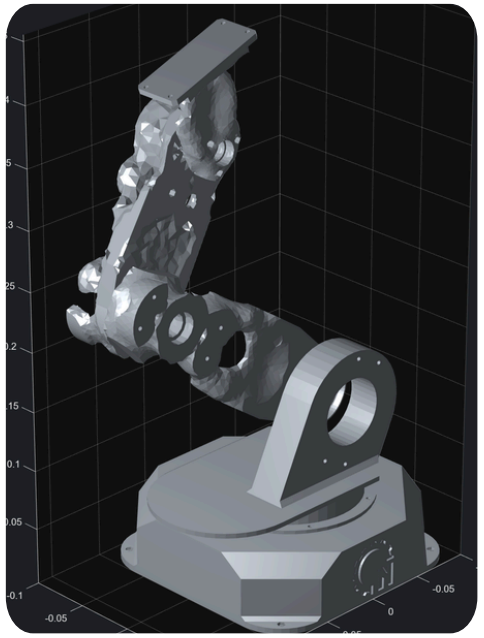
For Version 1, I focused on the mechanical design, structural optimization, and simulation of my 6-DOF robotic arm. Using SOLIDWORKS and ANSYS, I designed and optimized the mechanical components to improve their strength-to-weight ratio while maintaining important mounting features. I also tested several components' tolerances and revised the design to improve fit and overall assembly.

For the kinematic analysis, I developed a mathematical model of the entire 6-DOF robotic arm and calculated the Jacobian matrix to better understand the relationship between joint velocities and end-effector motion. I then used MATLAB's Robotics System Toolbox to simulate joint movements for J1 through J4. By implementing smooth motion interpolation, I simulated different joint configurations and visualized how the robotic arm moves.

For future development, Version 2 will focus on physically assembling the robotic arm, integrating the actuators, and testing the joint movements. Version 3 will focus on developing and integrating a mechanical gripper so the robotic arm can perform basic object manipulation.

Overall, this project helped me better understand mechanical design, structural analysis, and kinematic modeling. I also gained experience troubleshooting joint alignment, evaluating design tolerances, and improving the design throughout the process.

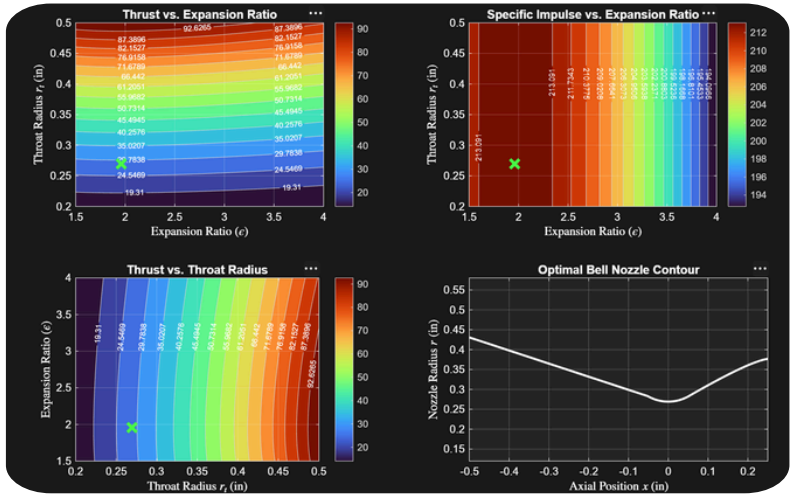


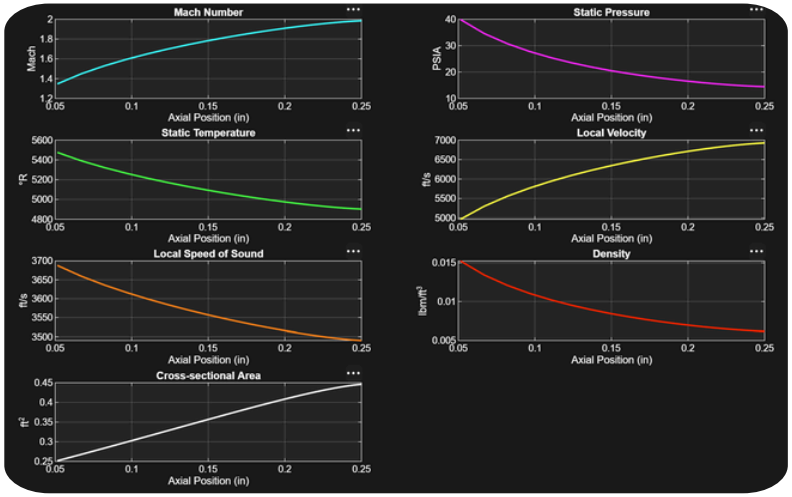


Engineering Requirements Satisfaction:

1. 6 DOF Design: ✓
2. Structural Analysis & Optimization: ✓
3. Component Tolerance Testing: ✓
4. Jacobian & Kinematic Calculations: ✓
5. J1-J4 Motion Simulation: ✓
6. Physical Assembly & Actuator Testing: **Planned V2**
7. Functional Gripper: **Planned V3**

Cold-Gas Thruster Study





Project Overview: This project was coded in MATLAB and integrated with NASA's Chemical Equilibrium with Applications in MATLAB (CEAM) software to compute combustion thermodynamic data. This algorithm takes user-defined geometry ranges (throat/exit radii, expansion ratios), ideal thrust, chamber scale, and gas reservoir parameters. It evaluates 2,000+ geometries via 1-dimensional isentropic flow analysis, filters down to a single optimal nozzle, prints flow results to the MATLAB terminal, generates three figures (nozzle optimization, flow profiles, and transient performance), and exports geometry data for ANSYS Fluent. Next, I will validate the performance metrics in a 2D ANSYS Fluent flow simulation using matching parameters.

Tools: MATLAB • CEAM• MATLAB

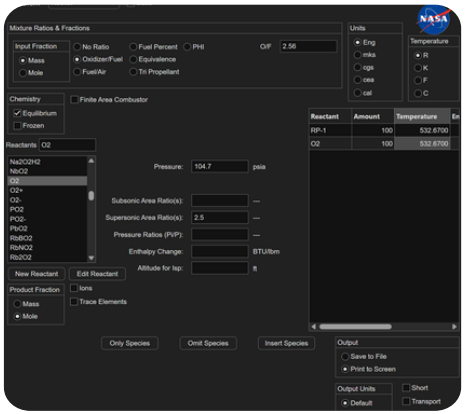
References: Rocket Propulsion Elements, Modern Compressible Flow

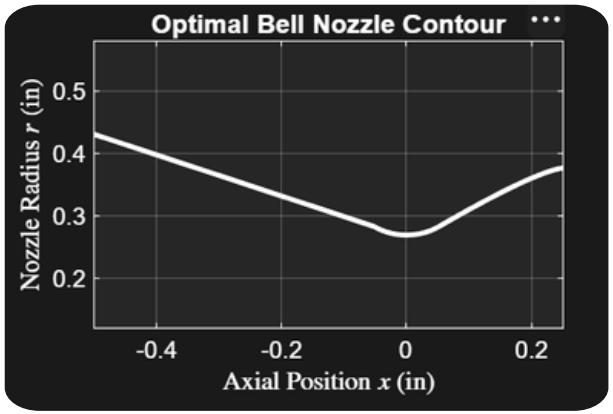
Requirements:

V1 - MATLAB
1. Thermodynamics
2. Compressible Flow
3. Nozzle Performance
4. Thrust & Specific Impulse
5. Transient Behavior
6. Automated Plots

V2 - CAD & CFD
1. Nozzle Geometries
2. Boundary Conditions
3. Mach Distributions
4. Static Pressure
5. Velocity Fields
6. Mesh Indepenence
7. Analytical Comparison

V3 - Testing
1. Modular Test Stand
2. Air Compressor
3. Load-cell Measurement
4. Data Acquisition
5. Measurement Uncertainty
6. Repeatability Testing
7. Experimental Comparison





Note:

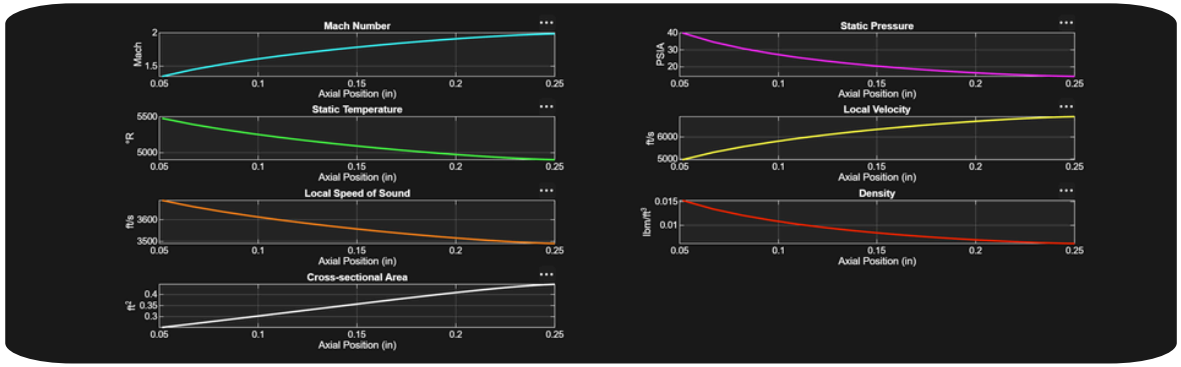
Although my algorithm includes thermodynamic data for bipropellant nozzles, in phase 3 of the project, I will perform the thrust analysis using an air compressor under ambient conditions at a set chamber pressure, which I will discuss later in my portfolio.

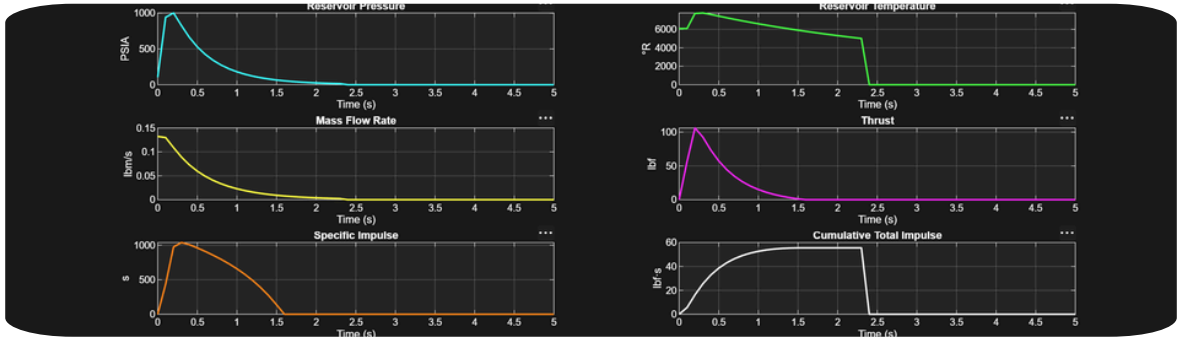
Nathaniel Cavazos
Mechanical Engineer at California State University Long Beach

MATLAB Program Development:

I built this program in MATLAB and integrated NASA's Chemical Equilibrium with Applications in MATLAB (CEAM) to pull combustion thermodynamic data into the model. From user-defined inputs, it runs a 1D isentropic flow analysis, selects a single nozzle geometry, and exports coordinate data for an easy workflow into ANSYS Fluent.

- **Design Inputs & Sweep:** Accepts throat and exit radius ranges, expansion ratios, ideal thrust, chamber scale, and gas reservoir parameters, then sweeps 2000+ geometries (adjustable via linspace) and filters them based on flow characteristics.
- **Terminal Flow Analysis:** Prints the full flow breakdown and performance metrics for the selected nozzle geometry directly to MATLAB's terminal window.
- **Automated Figures:** Generates 3 main figures covering nozzle optimizations, internal flow profiles, and transient performance.






```
----- Parameters -----

-- Geometry --
Specific Heat Ratio: k = 1.122
Throat Radius Range: [0.200 to 0.500] in
Exit Nozzle Radius Range: [0.300 to 2.500] in
Expansion Ratio Range: [1.500 to 4.000]
Nozzle Length Max: [0.200 to 2.000] in
Bell Length: 0.80 (percent)
Inflection Angle: 30.000° --- Exit Angle: 8.000° --- Reference Angle: 19.000°
Ideal Thrust: 30.000 lbf

----- 1D Flow -----

Chamber Pressure: 90.000 PSI
Stagnation Temperature: 5621.666 F
Gas Constant R: 68.817 (ft.lbf)/(lbm.°R)
External Pressure: 14.700 PSIA
```

```
----- 1D Isentropic Flow Analysis -----

Pressure Ratio: 0.137970
Critical Pressure Ratio: 0.237838
Density Ratio: 0.171127
Critical Density Ratio: 0.278036
Temperature Ratio: 0.806241
Critical Temperature Ratio: 0.855422
Critical Area Ratio: 1.959184

----- 1D Nozzle Performance Analysis -----

Exit Velocity: 6927.260 ft/s or 4723.132 mph
Exit Mach Number: 1.984876
Exit Mach Angle: 30.252381°
P-M Angle: 33.456525°
Ideal Mass Flow Rate: 0.132 lbm/s
```