Joey Carpinelli | Engineering Portfolio

+1 (202) 660-2578 | joey@carpinelli.email

Professional Experience

GN&C Engineer @ NASA Johnson Space Center (JSC)
July 2021 — Present

- Supports Orion's Launch Abort System development, analysis, and verification
- Reduced, analyzed, and implemented flexible structure model (MATLAB, C++)
- Uses monte-carlo simulations for parameter tuning; improved performance
- Uses linear analysis to analyze vehicle performance, verify stability margins
- Serves as backup regression data approver for Orion Launch Abort System
- Led development for polarity tests through novel kinematics simulation (Julia, Python); represented Orion GN&C at multiple lab tests

Research Assistant @ Space Systems Laboratory (SSL)
August 2019 — May 2021

- Graduate Assistant under Dr. Dave Akin; led manipulator software development
- Developed novel Julia package to generate symbolic manipulator kinematics models; implemented and merged required changes to `ModelingToolkit.jl`
- Created C++ templates, and implementations, for multiple manipulator kinematic controllers, including force/torque control and Cartesian control
- Maintained operator GUI for all manipulators; primary operator for neutral-buoyancy testing; maintained facility as diver; received open water certification

Inertial & Viscous Friction Compensation Project
Spring 2017 — Summer 2017

- Independent study to implement Dr. Carignan's inertial and viscous friction compensation for Maryland-Georgetown-Army (MGA) exoskeleton within SSL; used Galil, Python, ROS, UART

Intern @ Harris Corporation
May 2016 — August 2016

- Automated Excel task with VBA; 20 worker hours → 2 minute runtime
- Worked with one other intern to implement rain attenuation ITU Propagation Model; MATLAB functions written to implement model calculations, C# used to gather terrain data

Intern @ SRI International
May 2015 — December 2015

- Collected and annotated data to train deep-learning algorithms
- Designed LED Array and circuit layouts for gaze tracking project using Eagle CAD

Education

M.S. in Aerospace Engineering @ University of Maryland

- Research assistant under Dr. Akin; space robotics (manipulator) software lead, primary operator
- Halo orbit & invariant-manifold research project with Professor Barbee; released as open source tools
- Emphasis in space systems, prioritized dynamics & controls in coursework

B.S. in Electrical Engineering @ University of Maryland

- Emphasis in control theory, prioritized software in coursework through four computer science classes
- Undergraduate Research Assistant under Dr. Akin at SSL; manipulator software lead as junior

Technical Skills

Programming

- Productive in C++, used for robotic manipulator control software
- Experienced with Julia, used for astrodynamics research; hobbyist FOSS
- Experienced with Python, used for post-simulation analysis & scripting at NASA JSC; hobbyist FOSS

Circuit Design

- Undergraduate digital & analog lab experience, including Verilog, SPICE, PSpice, Xilinx, oscilloscopes
- Internship experience using Eagle CAD to design PCB for gaze-tracking project

Modeling & Simulation

- Experienced with linear model reduction, linear analysis, and nonlinear analysis methods

FOSS Highlights

`SolarSystemSurrogates.jl`

A work in progress that I'm excited about! This unreleased package attempts to provide surrogate models for solar system ephemeris.

`AstrodynamicalModels.jl`

Model generation — with optional state transition matrix dynamics — through `ModelingToolkit.jl`.

`GeneralAstrodynamics.jl`

General calculations, visualizations, iterative & analytical periodic orbit solvers, and orbit-manifold solvers.

`HorizonsAPI.jl`

A word-for-word wrapper for the JPL Horizons REST API. Fetch solar system ephemeris for free!

`HorizonsEphemeris.jl`

A user-friendly wrapper around the word-for-word wrapper for the JPL Horizons REST API.

`PolynomialGTM.jl`

Implements publicly available polynomial models for NASA's Generic Transport Model using `ModelingToolkit.jl`.

`module-hygiene`

Provides an `__export__` key, and an associated `cleanup` function to reduce namespace clutter.

`block-scopes`

Provides a single context manager, `only`, which creates block-style copes within Python. This package isn't unique — it's just for fun!

`rich-admonitions`

Extends the excellent terminal formatting package `rich` with Julia-style Markdown admonition blocks!

Socials

`loopy.software` `@cadojo`

`@cadojo#6211` `@code_typed`