Joey Carpinelli · Technical Résumé

Professional Experience

NASA Johnson Space Center (JSC)

August 2017 — Present

GN&C Engineer, Pathways Intern, USRA Intern

- Full time as of July 2021; Orion's Launch Abort System GN&C development, analysis, verification
- Reduced, analyzed, and implemented flexible body (structure) model (MATLAB, C++)
- Uses monte-carlo simulations for parameter tuning; improved vehicle performance noticeably
- · Uses linear analysis to analyze vehicle performance, verify stability margins; validates linear models
- Serves as backup regression data approver for simulated Orion Launch Abort System performance
- Led development for polarity tests; created novel 6DOF kinematics simulation (Julia, Python); represented Orion GN&C at multiple lab tests in three states: Texas, Colorado, Florida

Space Systems Laboratory (SSL)

August 2016 — May 2021

Graduate Assistant

- Graduate Assistant under Dr. Dave Akin as of August 2019; led robot software development (C++)
- Developed novel Julia package to generate symbolic manipulator kinematics models; implemented
 and merged required changes to ModelingToolkit.jl; intermediate Jacobian performance
 substantially improved over <u>Orocos</u> iterative solvers; implemented fast inverse-kinematics algorithm
- Created C++ interfaces (templates) and implementations for control, including force/torque control
- · Maintained operator GUI (Python); diver for Neutral Buoyancy Research Facility Maintenance

Harris Corporation

May 2016 — August 2016

Electrical Engineering Intern

- Automated Excel task with VBA; 20 worker hours \rightarrow 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

SRI International

May 2015 — December 2015

Data Annotation Intern

- · Collected and annotated data to train deep-learning algorithms; improved process with scripting
- Designed LED Array and circuit layouts for gaze tracking project using Eagle CAD

Education

M.S. Aerospace Engineering

August 2019 — May 2021

University of Maryland, College Park

- · 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. Electrical Engineering

May 2019

University of Maryland, College Park

- Four control theory courses, four computer science courses; major emphasis in control theory
- Undergraduate Research Assistant under Dr. Akin at SSL; ROS/Orocos software lead in third year
- $\bullet \ \ Implemented \ inertial \ and \ viscous \ friction \ compensation \ for \ SSL's \ MGA \ Exoskeleton \ (Galil, \ UART)$

Technical Skills

Computer Programming

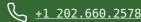
- Experienced Julia & Python developer; aerospace dynamics, simulation, and analysis applications
- Productive C/C++ developer; robot core software, kinematics, & control applications

Modeling & Simulation

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

Circuit Design

• Digital & analog lab experience, including Verilog, SPICE, PSpice, Xilinx, oscilloscopes, Eagle CAD



joey@carpinelli.email

Open Source

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. Presented at JuliaCon!

SPICEApplications.jl

All SPICE toolkit applications, wrapped in Julia functions with high-level syntax.

<u>SPICEKernels.jl</u>

All general kernels provided by NASA, exposed and cached through Julia functions.

HorizonsAPI.jl

A word-for-word wrapper for the JPL Horizons REST API. Fetch solar system ephemeris for free! See also: HorizonsEphemeris.jl.

PolynomialGTM.jl

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

<u>CommonLicenses.jl</u>

Inline any <u>standard license</u> into your executable document! For example: CommonLicenses.MIT().

<u>module-hygiene</u>

Provides an __export__ key, and an associated cleanup function to reduce namespace clutter. See also, <u>block-scopes</u>.

rich-admonitions

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

Personal Media



@cadojo



in/joeycarp



https://loopy.software