Enhancing Spacecraft Autonomy and Mission Success Via Computational Throttling and Risk Governors

Dr. Christopher “Chrispy” Petersen

Assistant Professor

Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

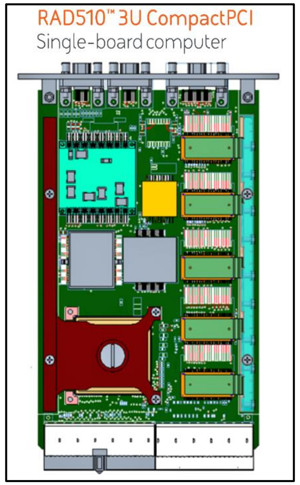
Spacecraft Technology and Research Laboratory

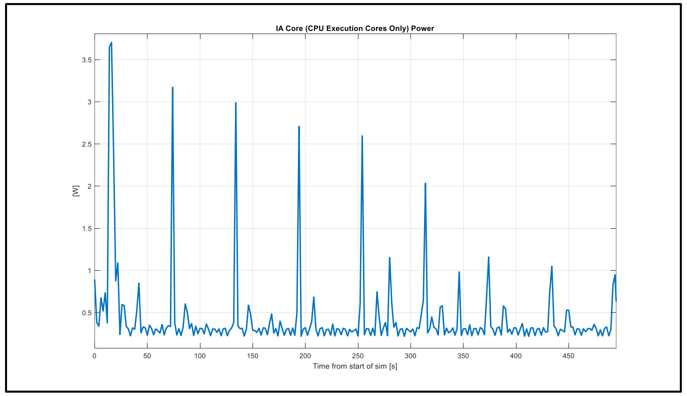
Go Gators!

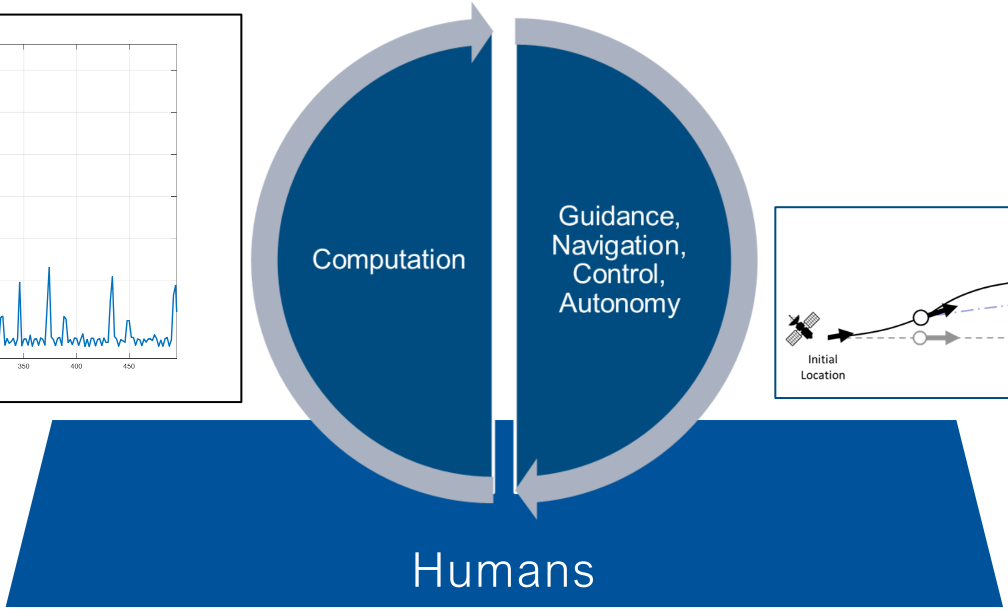

NCAA
2025
DIVISION I
FLORIDA
15
FLORIDA
1
FLORIDA
10
5
FLORIDA
2025 NCAA CHAMPIONS

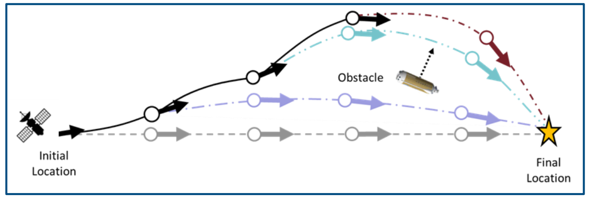
Research Group

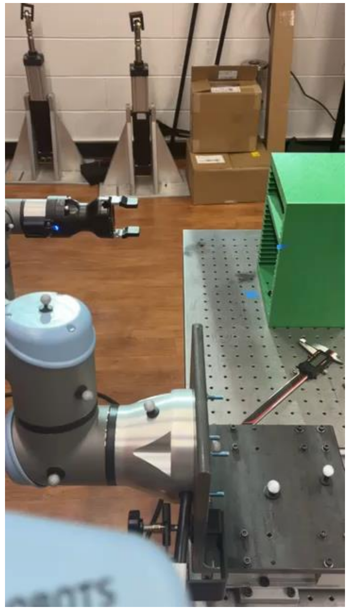
Blending basic research with application to ensure deployment

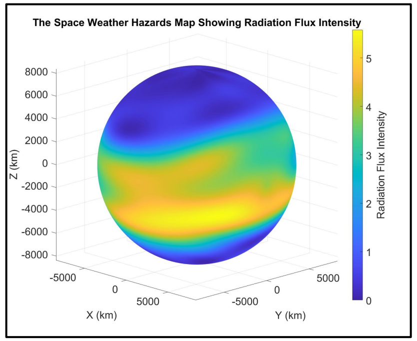


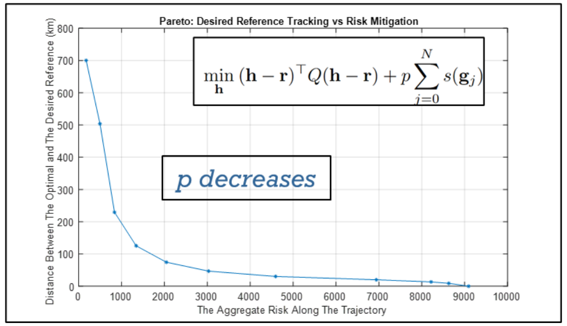












Students and Work

- 3 Graduate Students
  - Faraz Abed Azad (Directly Funded)
  - Channing Ludden (2024 AFRL/RY)
  - Sarah Clees (2025 AFRL/RVS)
- 5 Undergraduates
  - Cannon Whitney (2024,2025 AFRL/RVE)
  - Sara Lin (2025, AFRL/RVS)
  - Eric Stiner (2025 AFRL/RVE)
  - Michael Madden (2025 AFRL/RVE)
  - Jonathan Tindall (2025 AFRL/RVE)
- 2 Summer Faculty Fellowships
  - 2024 AFRL/RY- Dr. Kerrianne Hobbs
  - 2025 AFRL/RV- Dr. Sean Phillip

Successes

- 5 years of invited session at American Control Conference
  - Channing Ludden (2024 AFRL/RY)
  - Sarah Clees (2025 AFRL/RVS)
- 1 Tutorial Paper
- Several Joint Papers (competed or in works)
- 1 DURIP
- UNP Bootcamp for ISAM + C3 Competition
- Several non-traditional space collaborations

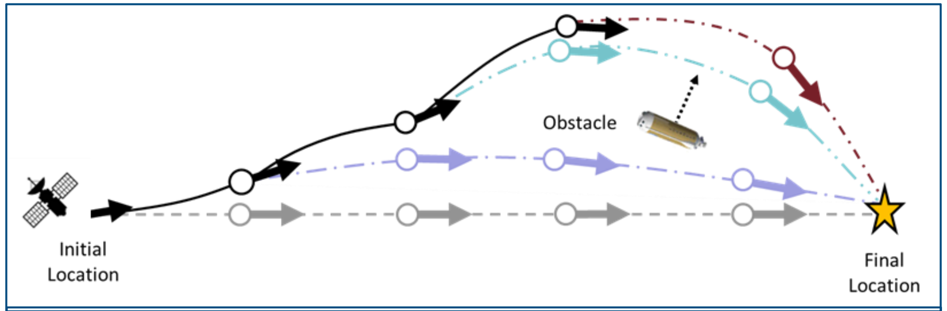
Real-time Recursive Optimization

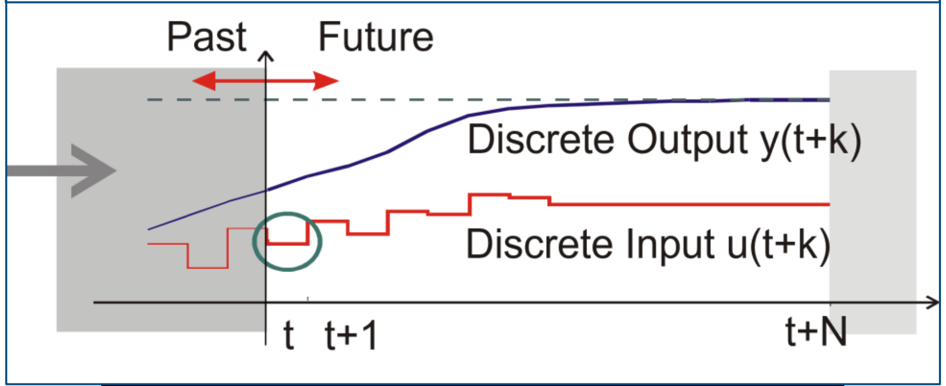
Real-time Recursive Optimization (R2O) & Computation

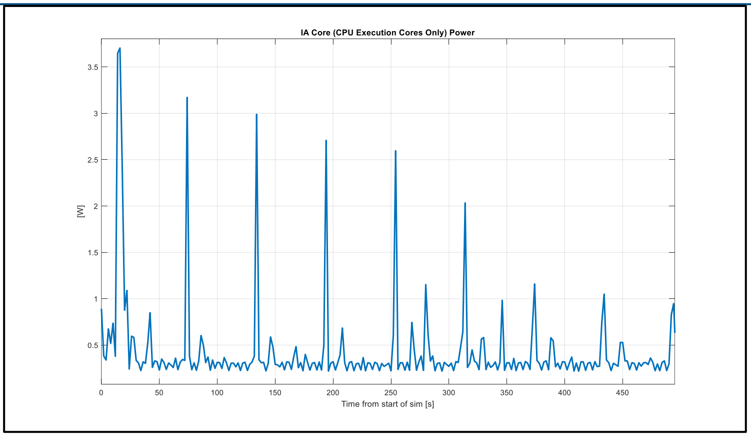
- Any optimization enacted **frequently, on-board**
  - Model Predictive Control
  - Reference Governors
  - AI/ML
  - Non-stabilizing optimizations

- Optimization comes in **different flavors**
  - Local: Interior-point, QP, SQP, MILP
  - Global: Random Search, Particle Swarm

- Computation is:
  - Computation time (sec)
  - CPU %
  - Memory
  - Power





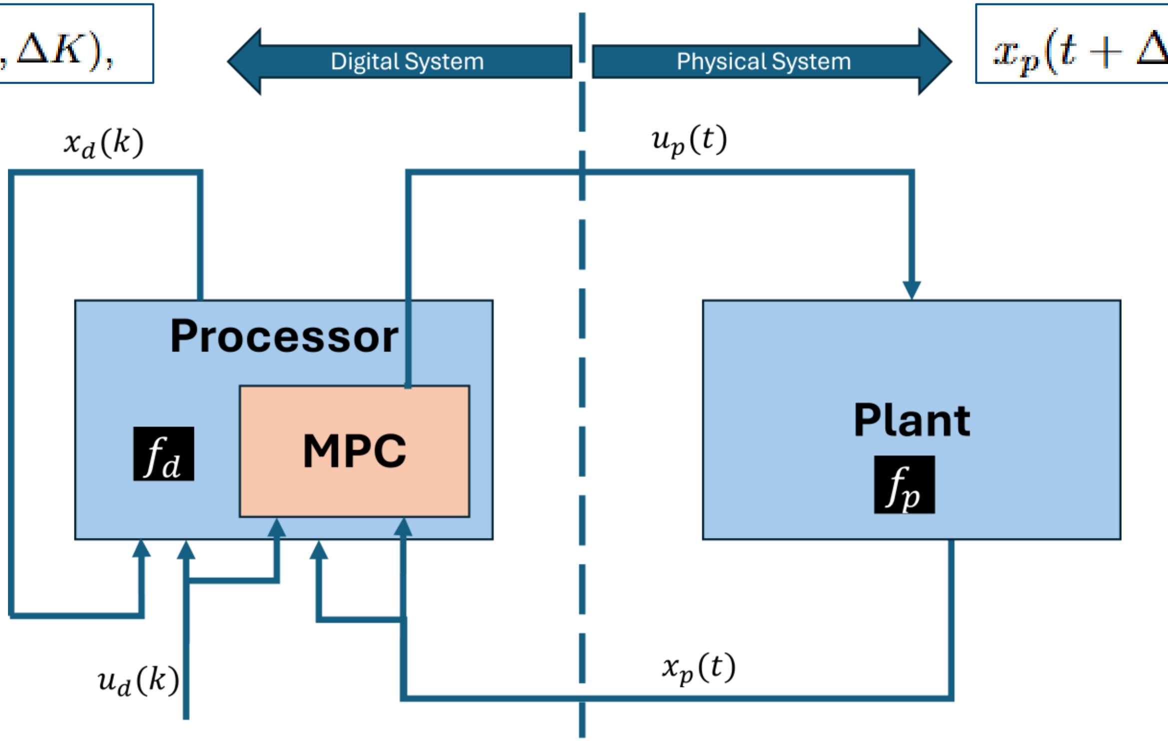


How do R2O drive the computational states?

R2O Modeling

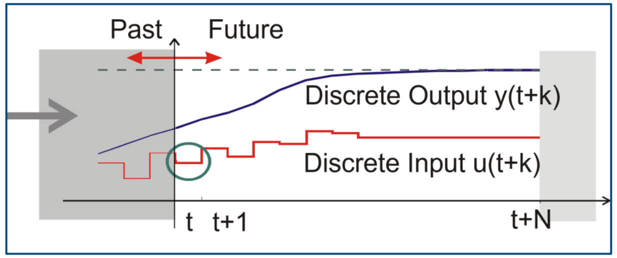
$$x_d(k+\Delta K) = f_d(x_d(k), u_d(k), w_d(k), \Delta K),$$

- States:
  - CPU
  - Power
- Inputs
  - Horizon length
  - Solver Tol

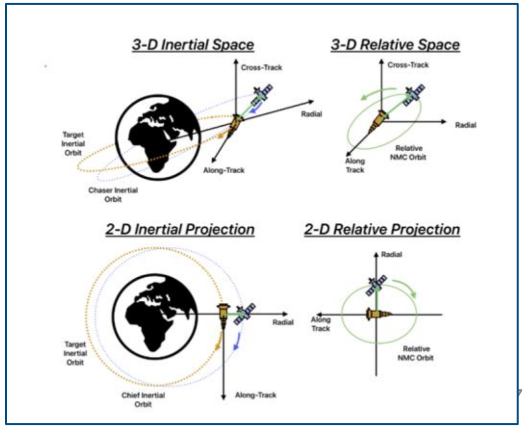


$$x_p(t+\Delta T) = f_p(x_p(t), u_p(t), w_p(t)),$$

- States:
  - Relative Position
  - Relative Velocity
- Inputs
  - Thrusters



How does R2O drive the computational states?



Risk-Mitigation Governor (RRG)

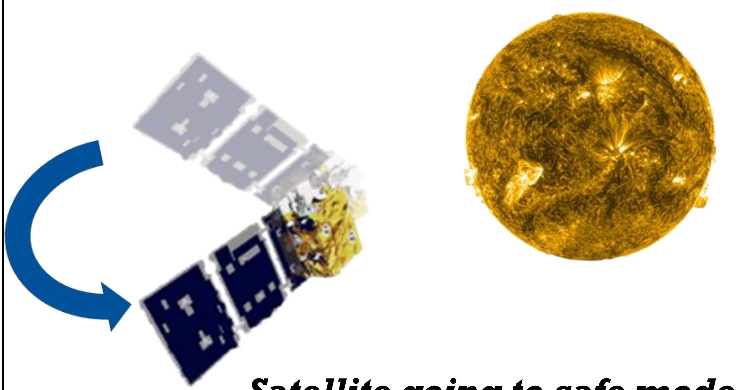
Space Weather Risk

Satellite Safety and Operations

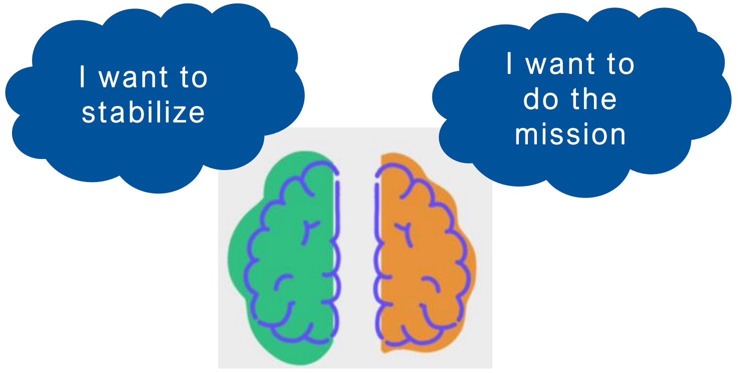
- Satellites need to perform a mission
- On-orbit safety often results in going to safe mode

Safe Complex Operations

- Takes time to diagnose and recover
- Takes time off mission
- Decouples safety and risk



Satellite going to safe mode



Left-Right Brain of System Engineers

Problem: Spacecraft autonomy needs to reflect goals and operator “risk”

Solution: Create autonomy that reflects human goals (not just controls) while **being safe.**

Risk-Mitigation Reference (RRG) Governor

Standard RG

Standard RG

$$\mathbf{h} = \mathbf{v}_k$$

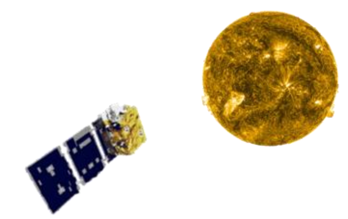
$$\min_{\mathbf{h}} \boxed{(\mathbf{h}-\mathbf{r})^\top Q(\mathbf{h}-\mathbf{r})} + \boxed{p\sum_{j=0}^{N} s(\mathbf{g}_j),}$$

$$\text{subject to : } \boxed{(\mathbf{x}_k, \mathbf{h}) \in \mathcal{O}_\infty,}$$

$$\boxed{\mathbf{g}_j = \bar{A}^j \mathbf{x}_k + \sum_{i=0}^{j-1} \bar{A}^i \bar{B}\mathbf{h},}$$

Result

- Add-on to stable system
- Enforces safety
- Balances Risk



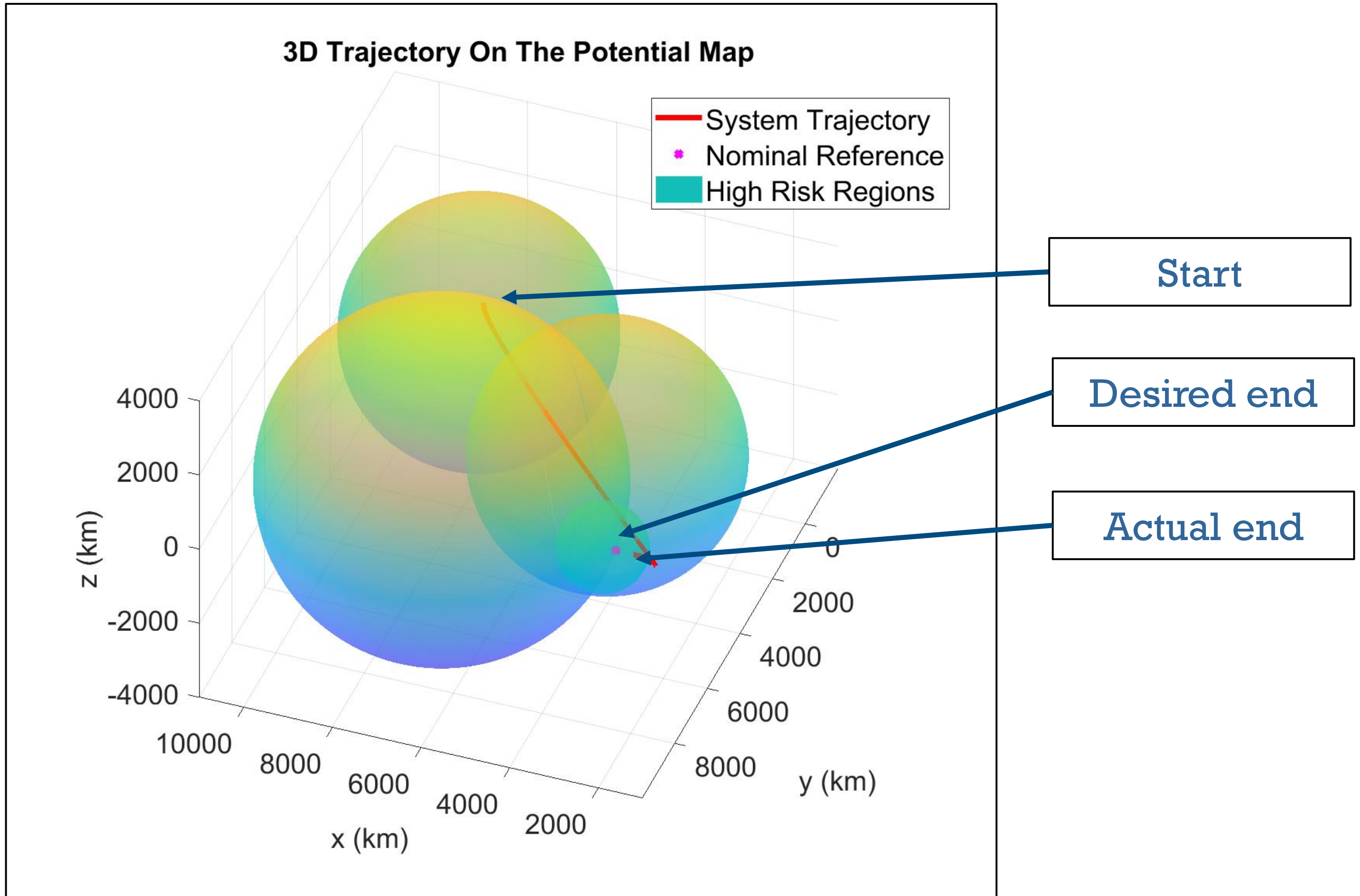
$$\mathbf{x}_\mathbf{k}$$

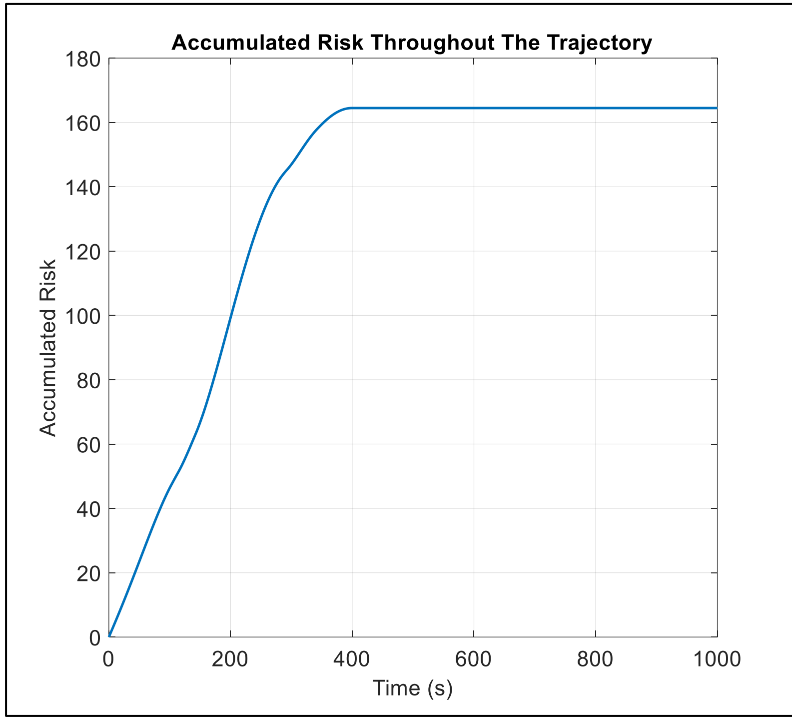
Dynamics

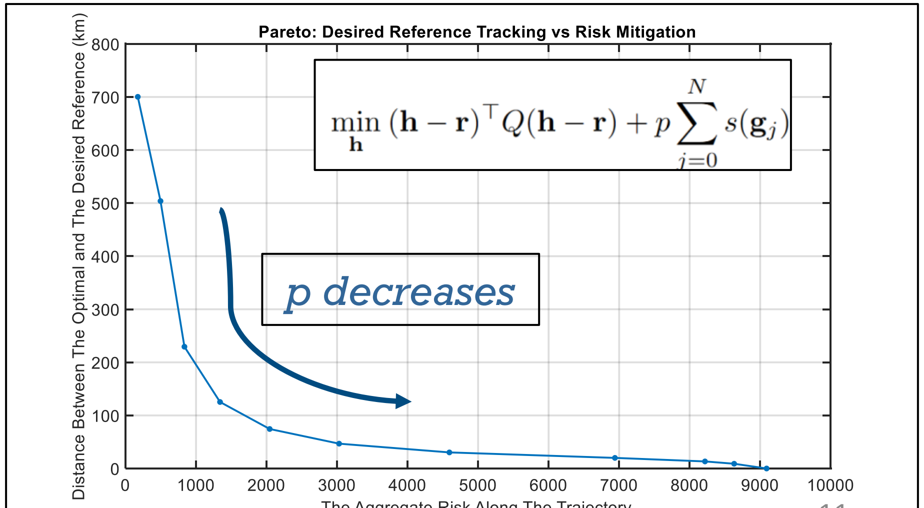
Safety

Spacecraft Rendezvous Problem

- **Problem:** Spacecraft in GEO to get as close as possible to goal while balancing risk
- **Assume:** Existing controller
  Risk in terms of “bubbles” (GPS sparse measurements)



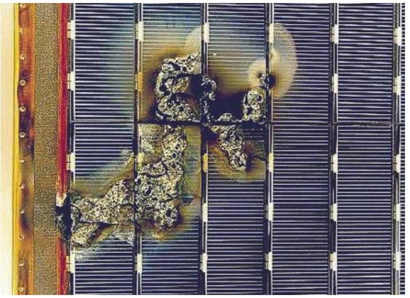




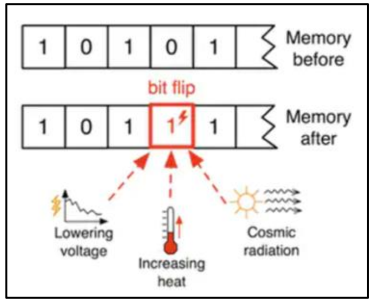
Risk In Terms Of Space Weather

- Space weather generates risk of radiation particles hitting a spacecraft
  - Generates failures

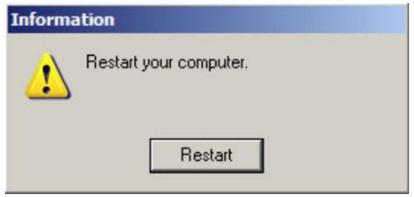
Highly nonlinear



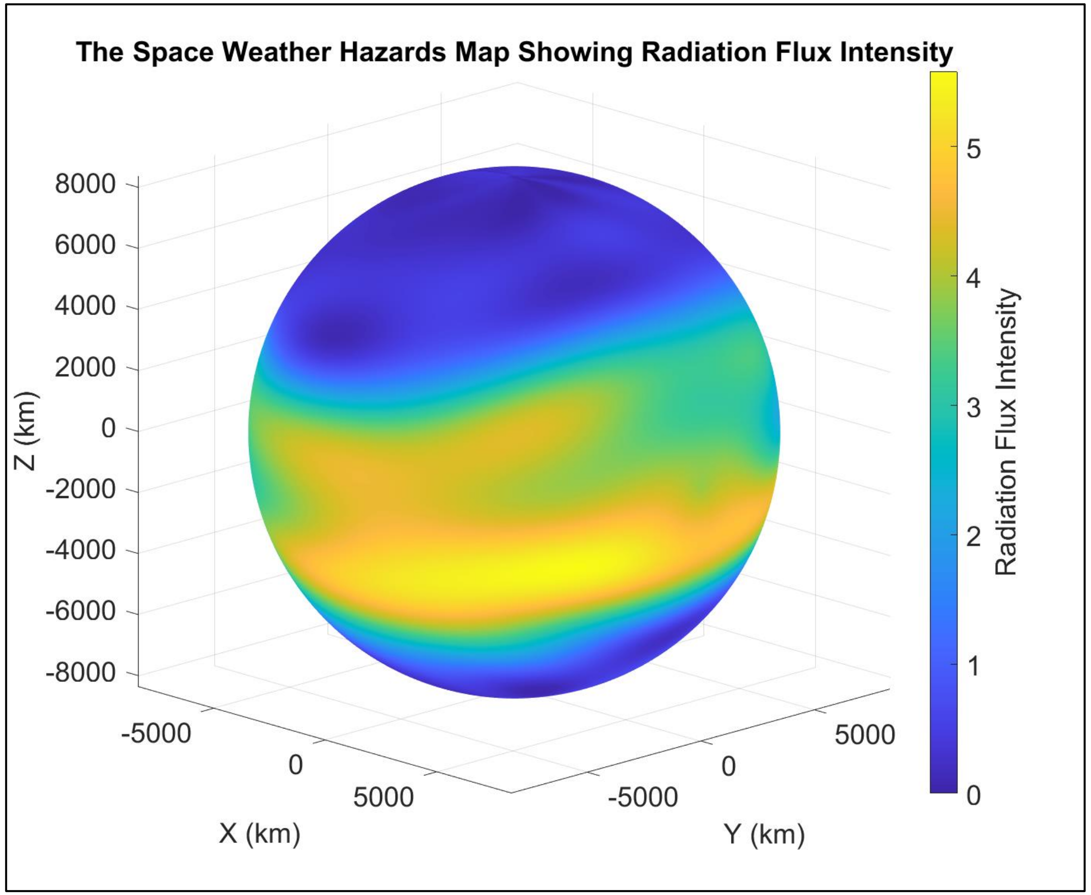
Electrostatic Discharge



Bit Flip

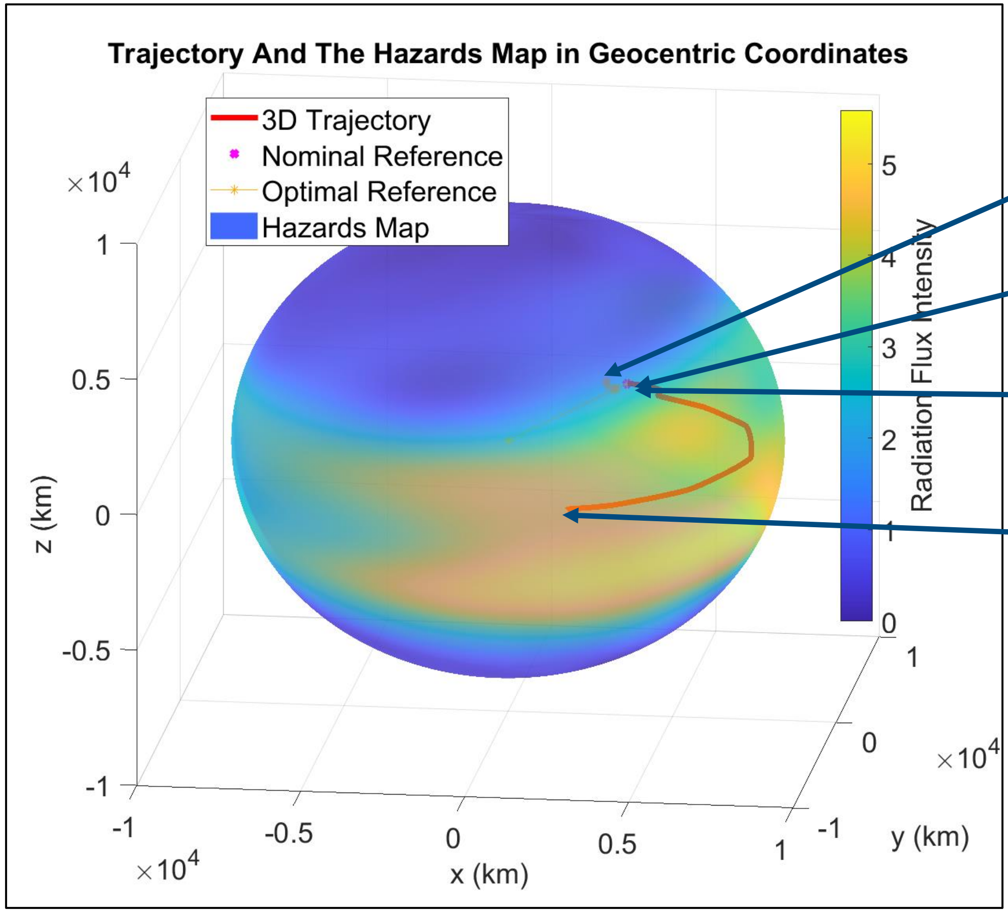


Reboot



Spacecraft Rendezvous Problem

- **Problem:** Spacecraft in GEO to get as close as possible to goal while balancing risk
- **Assume:** Existing controller
  Risk in terms of space weather

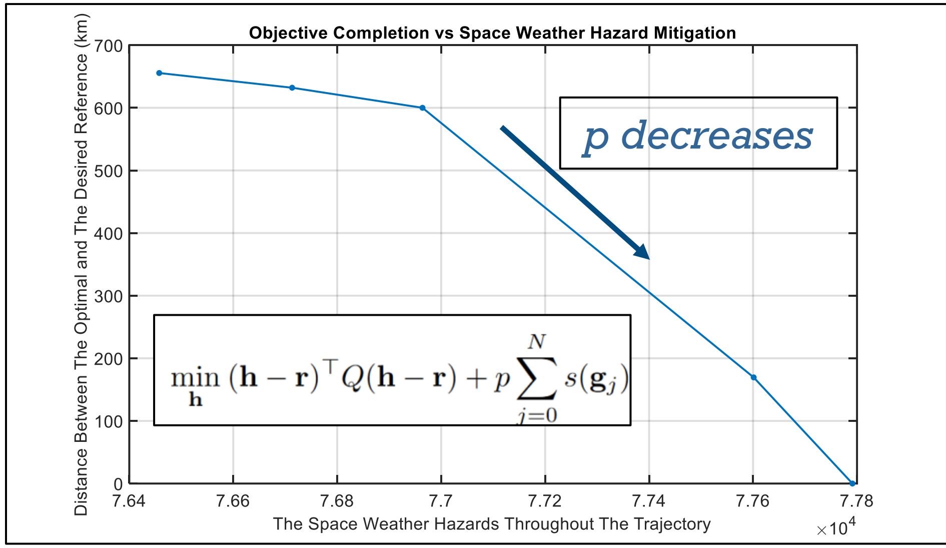


Mod ref.

Actual end

Desired end

Start

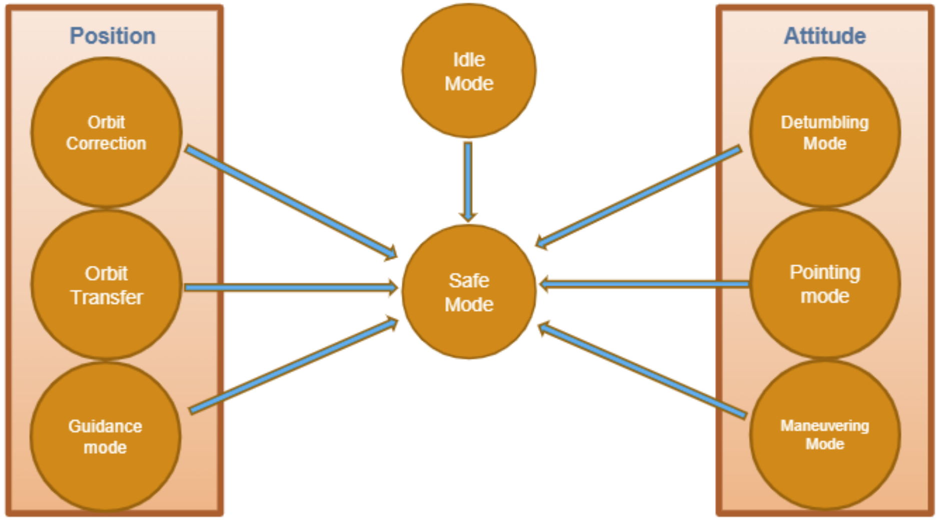


WORKS! But properties being investigated (CDC -2025)

What's Next

- Spacecraft discrete mode switching under risk
- Incorporation of "Human-in-the-loop" that balances "stabilization" vs "risk"
  - "Bi-directional, Adaptive User Interfaces for Successful Adoption of Intelligent Decision Aids by Spacecraft Operators" – Jain, Purdue

Look beyond GNC and build Guardian Trust In Autonomy

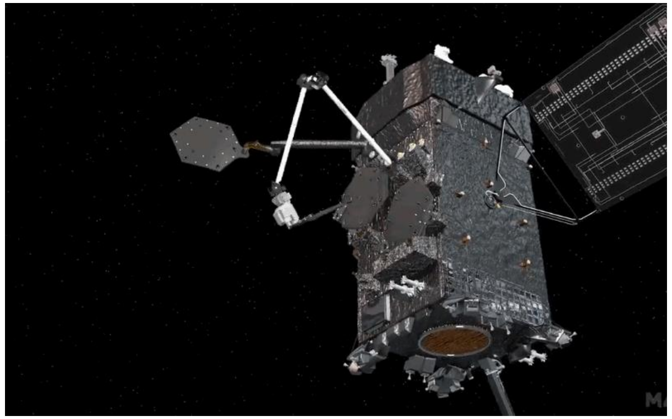


Mode Number	SCS Mode	Specifications	Objective	Admissible Transitions to
1	Idle Mode	Actuators are off	Standby for the next objective	all modes
2	Safe Mode	Point the solar panels towards the Sun	$\alpha < \frac{\hat{n}\vec{d}^T}{\|\|\vec{d}\|\|_2} \leq 1$	1
3	Detumbling Mode	$\dot{k} > \dot{\gamma}_k$, $k = \theta, \varphi, \psi$	$\dot{k} < \dot{\gamma}_k$, $k = \theta, \varphi, \psi$	2,4,5
4	Attitude Pointing Mode	$\|e_k\| < \beta_k$, $\|\dot{e}_k\| < \dot{\beta}_k$, $k = \theta, \varphi, \psi$	minimize attitude error	2,3,6
5	Attitude Maneuvering Mode	$\|e_k\| > \beta_k$, $\|\dot{e}_k\| > \dot{\beta}_k$, $k = \theta, \varphi, \psi$	minimize attitude error	2,3,5
6	Orbit Correction Mode	$X - X_{des} < \chi$	minimize tracking error	2,8,7
7	Orbit Transfer Mode	$X - X_{des} > \chi$	minimize tracking error	2,8,6
8	Guidance Mode	Avoid certain regions	Navigate to objective	2,6,7

Modes and Transition for Spacecraft

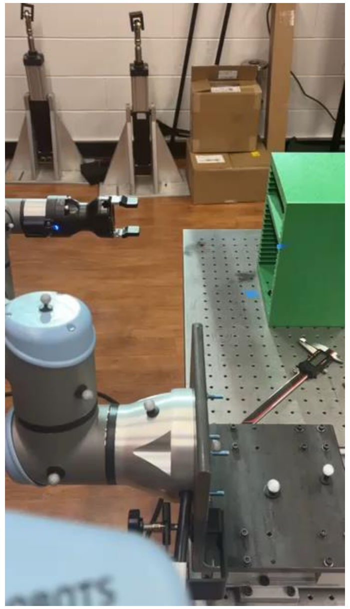
Computational Throttle

Computation and RPO



Servicing and Manufacturing

- Servicing with satellites requires a number of complex subfunctions



Complex Operations

- Dock with safety constraints and complex environment (e.g. lighting, contact mechanics)
- 20+ DoF systems (robotic arms, satellite bus…)
- Subsystem interconnections (power, GNC, thermal)

Problem: Spacecraft autonomy is needed for a variety of future missions, but hardware is 10+ years behind terrestrial spacecraft

Solution: Instead of waiting for hardware to catch up, provide a software solutions with algorithms that are **computationally aware.**

R2O Optimization Breakdown

$$u_p(t) = v_0^*$$

General Horizon Optimization

$$\min_{\mathcal{Y}} \sum_{i=0}^{N-1} l(y_i, v_i) + F(y_N) \qquad \textit{Cost}$$

$$\text{Subject to}:$$

$$y_0 = x_p(t) \qquad \textit{Real-life}$$

$$y_{i+1} = f_p(y_i, v_i, 0), \ i = 0, ..., N-1 \qquad \textit{“Dynamics”}$$

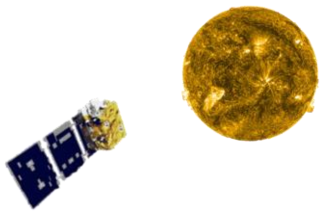
$$g_j(y_i, v_i) \leq 0, \ i = 0, ..., N-1, \quad j = 1, ...h \qquad \textit{Constraints}$$

$$y_N \in \mathcal{Y}_f$$

Math Parameters
- **Horizon**
- **Constraints**

Solver Parameters
- **Exit Criteria**
- **Step-size**

<u>Result</u>
- Two Categories for understanding & “throttling” computation

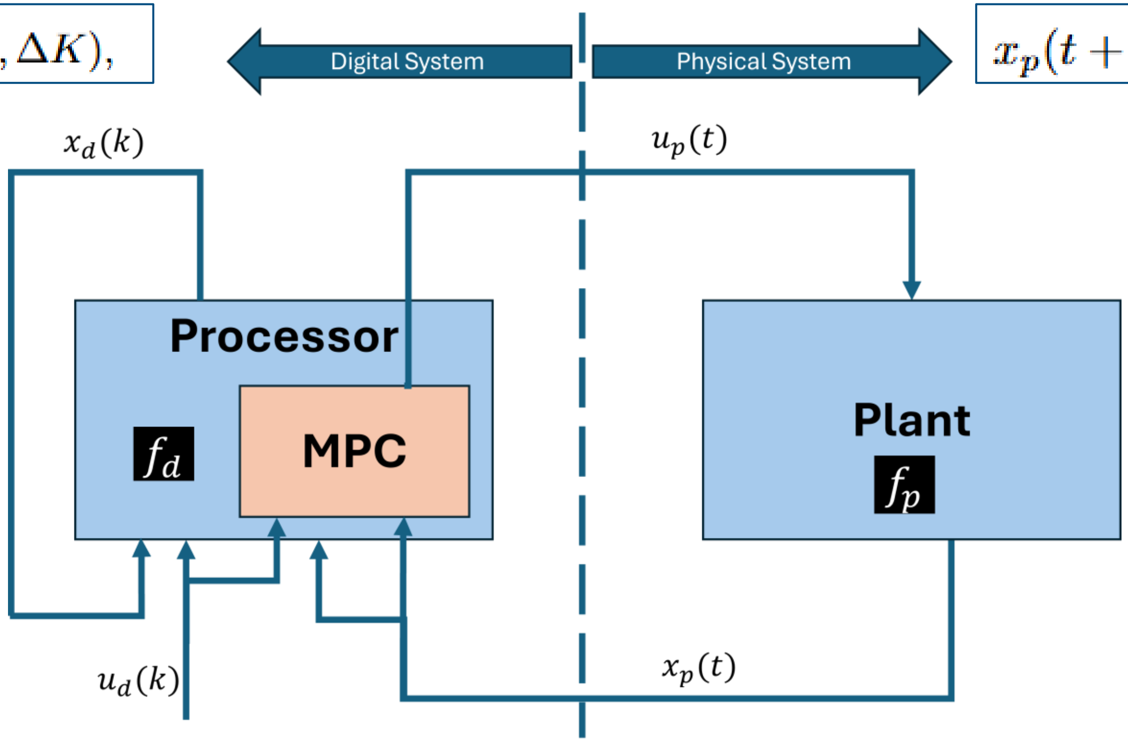


$$x_p(t)$$

Computation States and Inputs

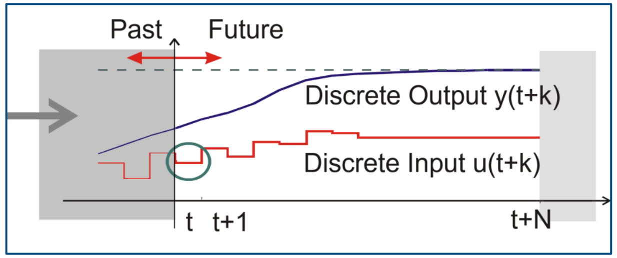
$$x_d(k+\Delta K) = f_d(x_d(k), u_d(k), w_d(k), \Delta K),$$

- States:
  - CPU
  - Power
- Inputs
  - Horizon length
  - Solver Tol

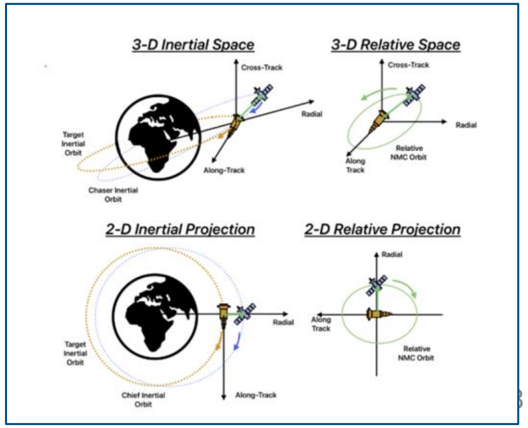


$$x_p(t+\Delta T) = f_p(x_p(t), u_p(t), w_p(t)),$$

- States:
  - Relative Position
  - Relative Velocity
- Inputs
  - Thrusters

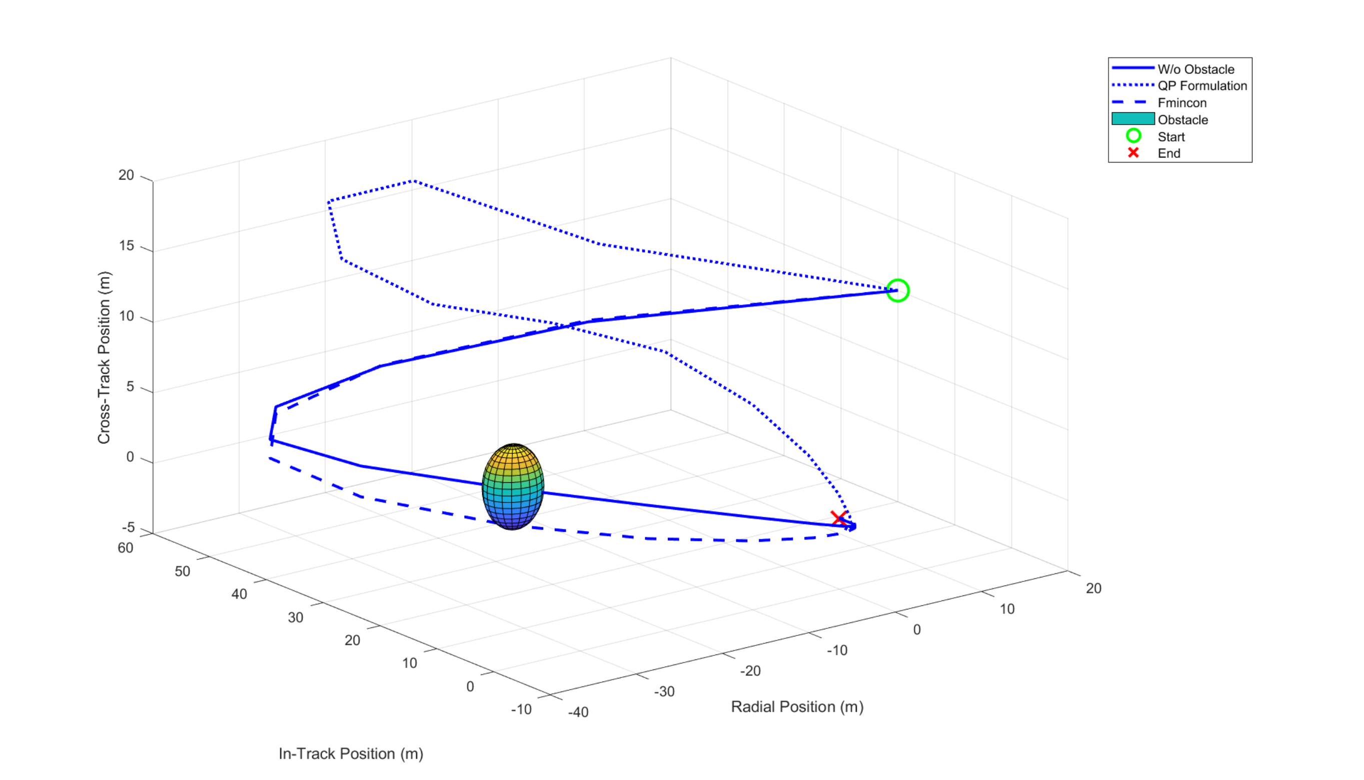


How does MPC drive the computational states?



Test Scenario

- Satellite is docking with another satellite
  - There exists an obstacle in the way
- Path is solved using two methods
  - Convex quadratic program (Simple)
  - Nonlinear interior point (Complex)
- Useful parameters
  - Satellite ~ 30 m away, stagged for docking
  - Control rate/discretization 60 seconds
  - Horizon length is 15 steps (~1/5 orbit)
- Computation metrics measured on Microsoft Surface 3, executed as if in "real time"



Computational State

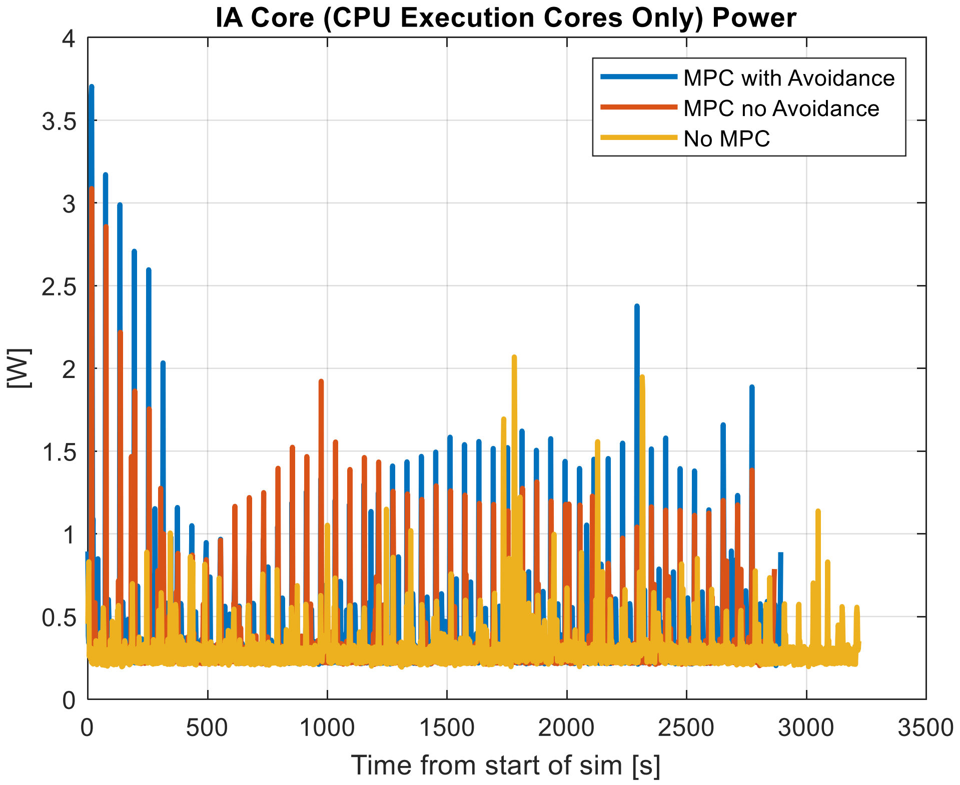
a) *CPU*
b) *Power*

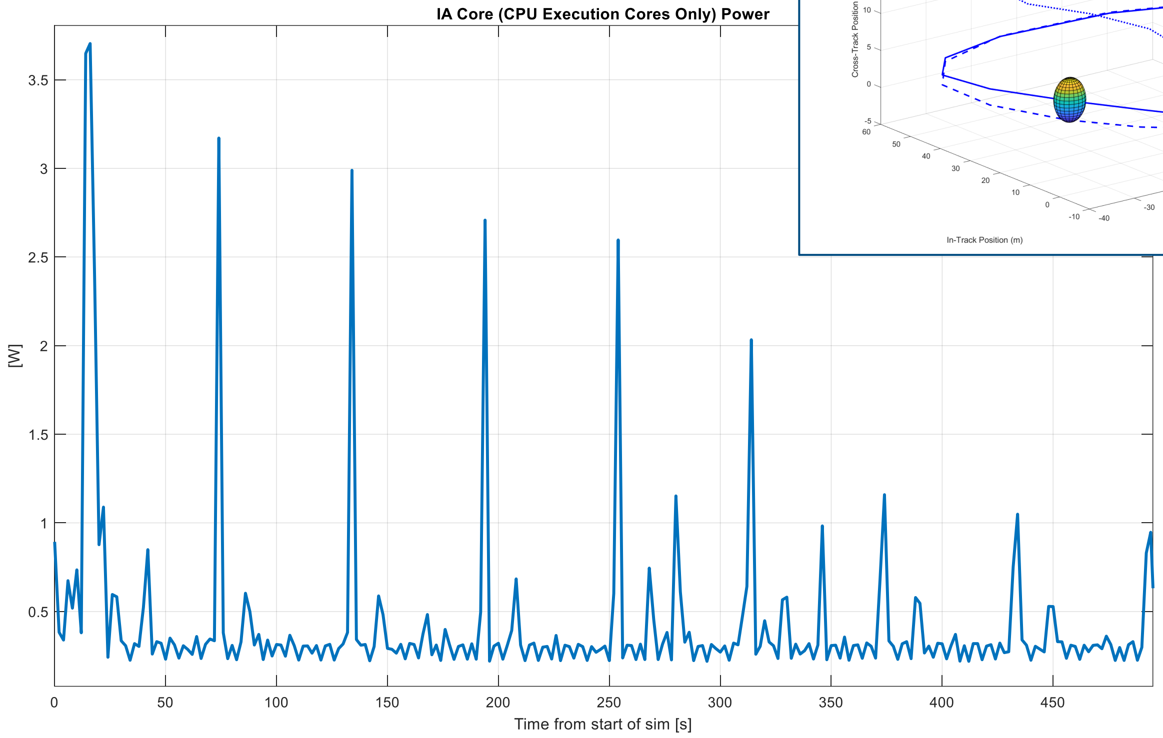
Computational Input

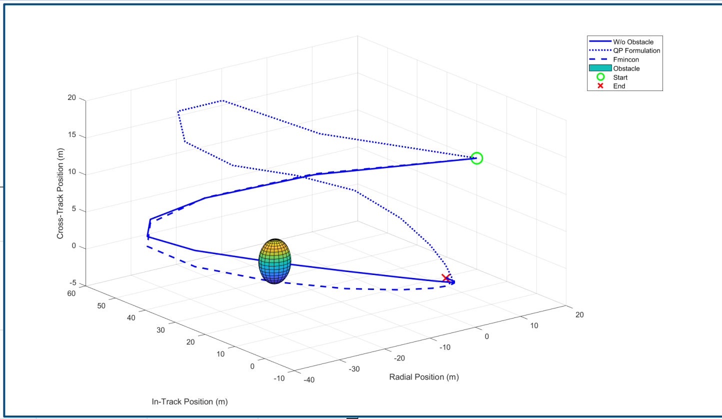
a) *Algorithm*
b) *Horizon*
c) *Solver Tol*

Objective: How do the computational metrics vary and evolve temporally?

Power Load (Simple Solver)





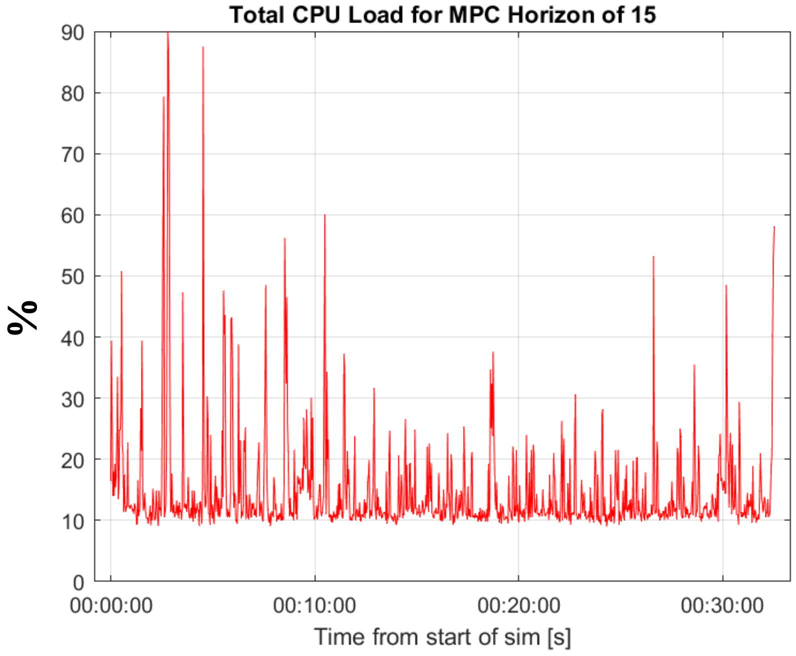


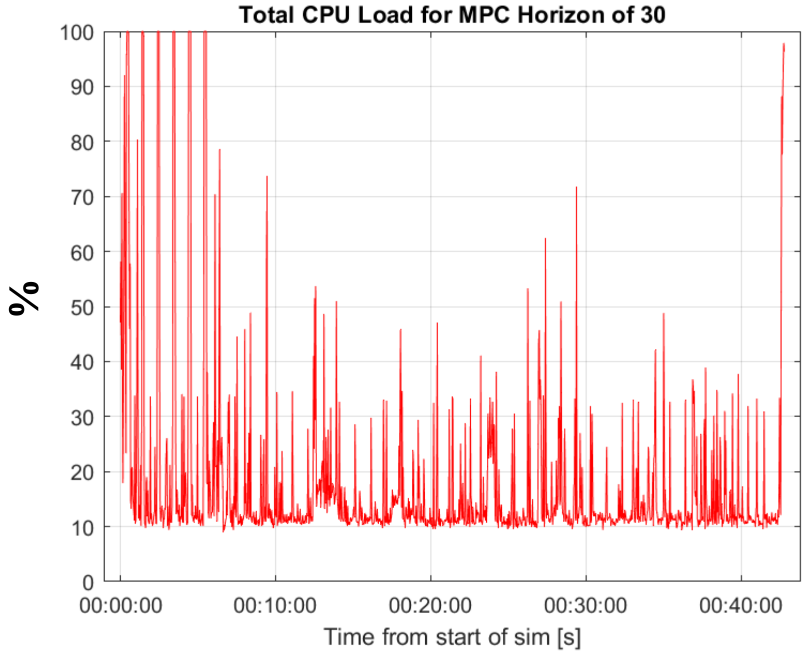
Power >4x most the time
Still more complex with obstacle

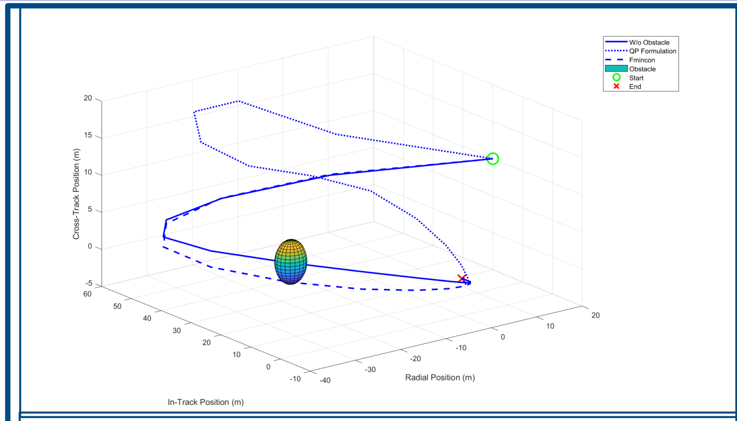
Asym. stable with small disturbance
Results as an impulse to computation

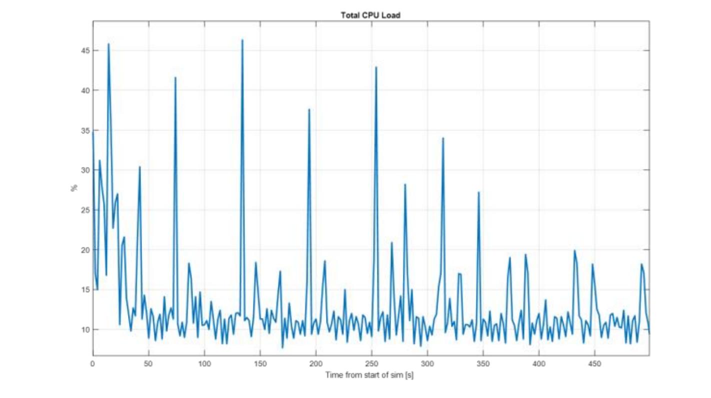
Computational dynamics do exist!

CPU (Complex Solver)









More computation when it sees the obstacle

Maxes computation like a step instead of an impulse

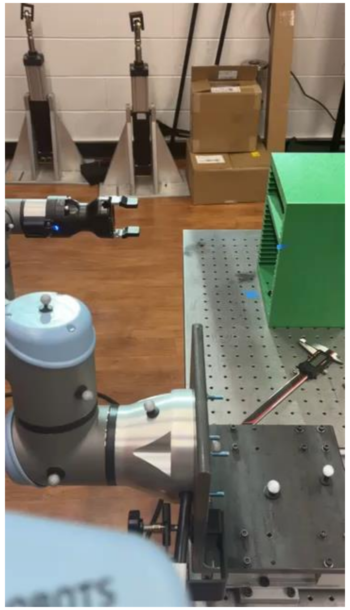
Simple solver,
Max %CPU = 45%

Though a bit noisier, complex solver still has a temporal trend of dynamics

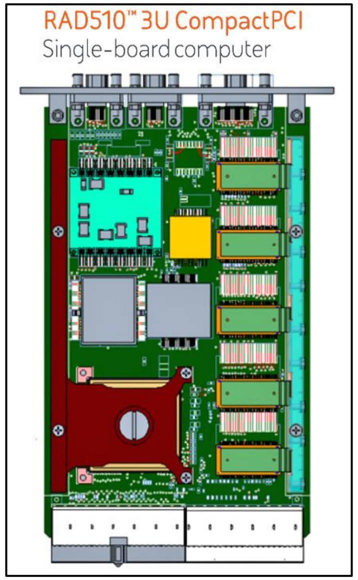
What’s Next

- **Space robotics** for in-space servicing
  - COSMIC C3 Competition
- Benchmarking on **space hardware**
  - AFRL/RY AI/ML algorithms
  - AFRL/RV inspection algorithms
  - Full in-house hardware-software digital twin
- Suboptimal, Safe Solvers & R20
  - Safe, local solvers
  - Suboptimal MPC and R20
- Hybrid, Computational-R2O Models
  - Mathematically formalize
  - Quantify solver and math parameters
  - Quantify existence and suboptimal parameters

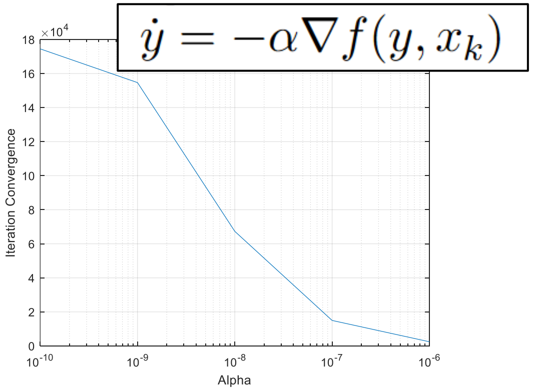
Bridge Computation and Autonomy



In-house space robotics testbed



Benchmarking on RAD510



Suboptimal, Safe Solvers & R20

$$\psi_\epsilon(y,\eta) := \{y \in \mathbb{R}^n |\ y \in H(\eta), \phi_\epsilon(y) \le V(y)+\epsilon\}$$

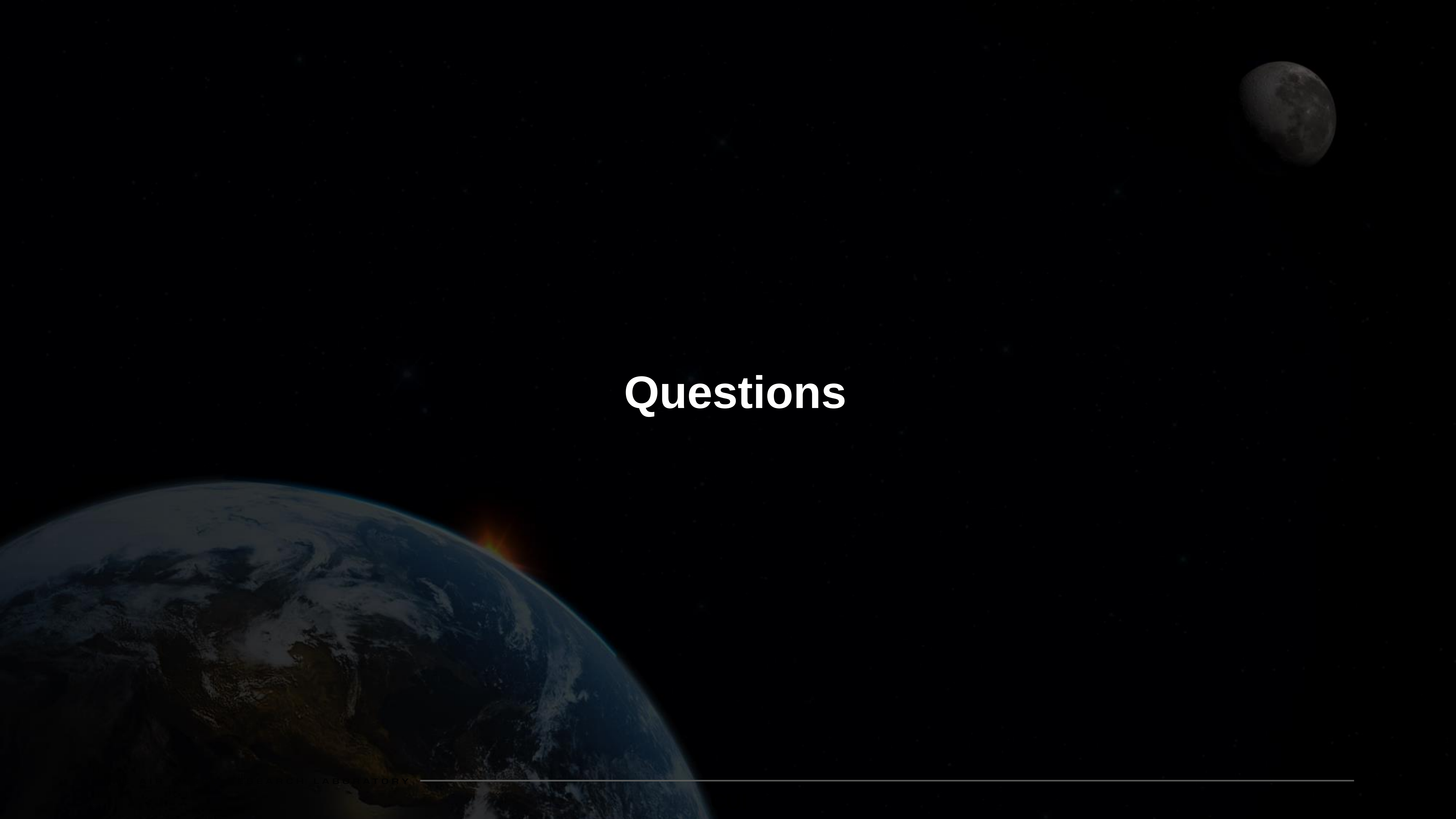
$$G(\chi_p) = \begin{bmatrix} y_p^+ \in H(\eta) \\ \phi_{\epsilon,p}(y) \in V(y)+\epsilon \\ \tau_p^+ \in [\Delta t_{\min}, \Delta t_p^*] \end{bmatrix},$$

$$D = \{y_p \in \mathbb{R}^n, \phi_\epsilon \in \psi_\epsilon, \tau_p = 0\},$$

$$F(\chi_p) = [\dot{\tau}_p = -1],$$

$$C = \{\tau_p \in \mathbb{R} |\ \tau_p \in [0, \Delta t_p^*]\},$$

Hybrid, Computation-R2O Model



Questions

STAR Lab Processor Range

Arduino

- Micro-Controller
- C/C++ Uploaded through IDE
- Reduced background noise
- "Isolated" processor / breadboard

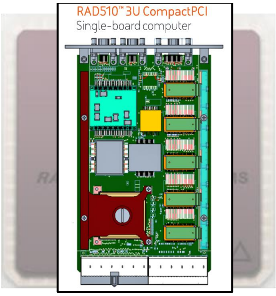
Raspberry-Pi

- Micro-Processor
- Raspbian-OS (Linux) or Command Line Interface (CLI)
- Complex Background Environment
- Interactive OS environment

NVIDIA Jetson

- Development Board
  - CPU, GPU, RAM, etc.
- Stated to be a COTS option that has some RADHAZ-like features (1-5 years, though not tested)

BAE RAD510



- True Flight Rad Processor
- BAE development harness (VXWorks-like)

Off the Shelf, Simple
Computationally Limited

Rad Tolerant & Hardened, Custom
Computationally Limited

Wide range of processors that span the complexity of spacecraft development