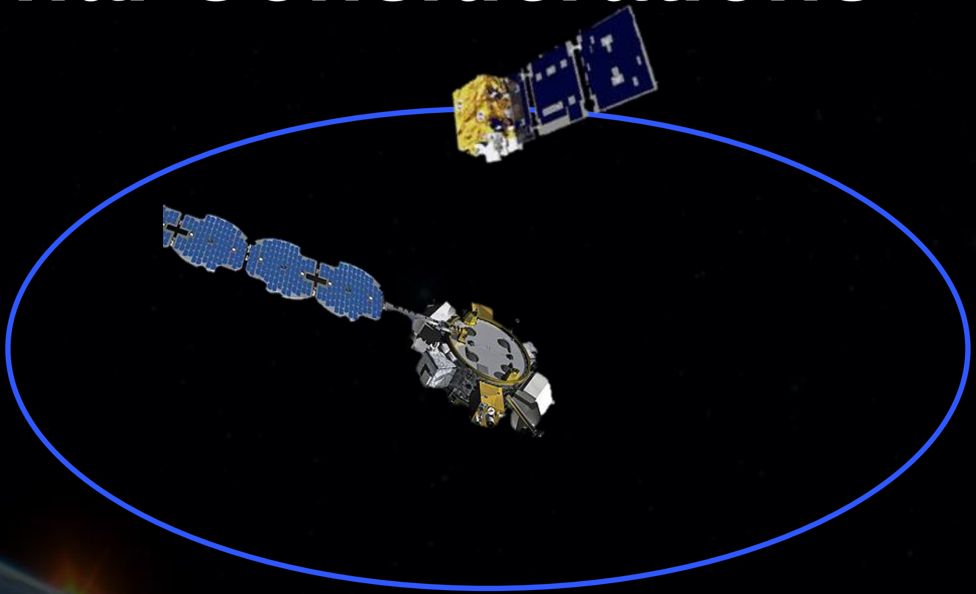
Enhancing Spacecraft Autonomy and Mission Success Via Mode Switching and Computational Considerations



Mr. Faraz Abed Azad
PhD Student

Dr. Christopher “Chrispy” Petersen
Assistant Professor

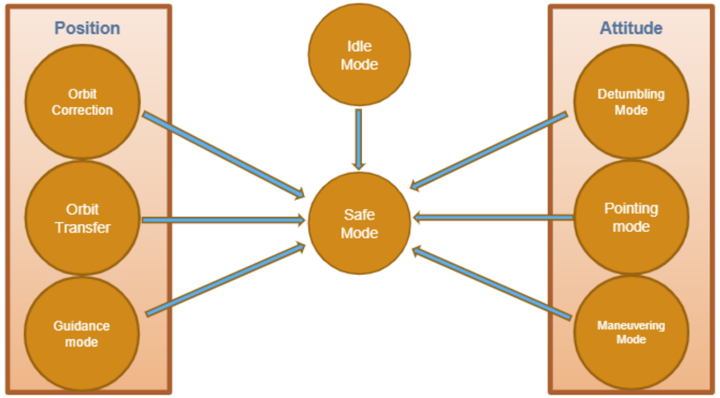
In Collaboration with:
Dr. Alicia K. Petersen
Assistant Professor

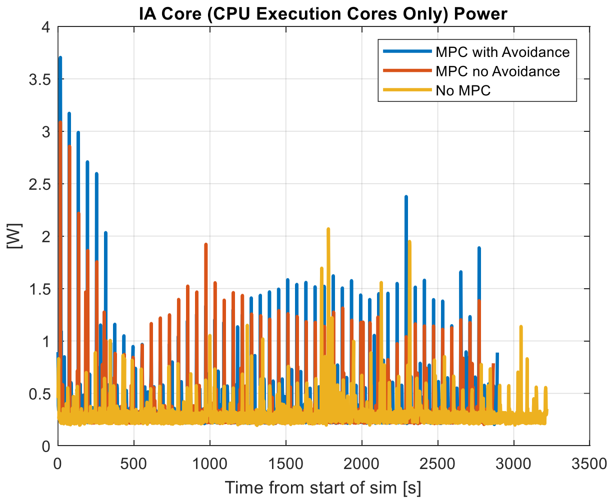
UF Herbert Wertheim
College of Engineering
UNIVERSITY *of* FLORIDA

Outline

- Enhancing via Safe Mode Switching
  - Relevance
  - MDP For Switching Spacecraft Modes
  - Next Steps Towards Informed Mitigation

- Enhancing via Computation Considerations
  - Relevance
  - Computational Metrics and Informed Dynamics
  - Next Steps Towards Computation-In-The-Loop Control

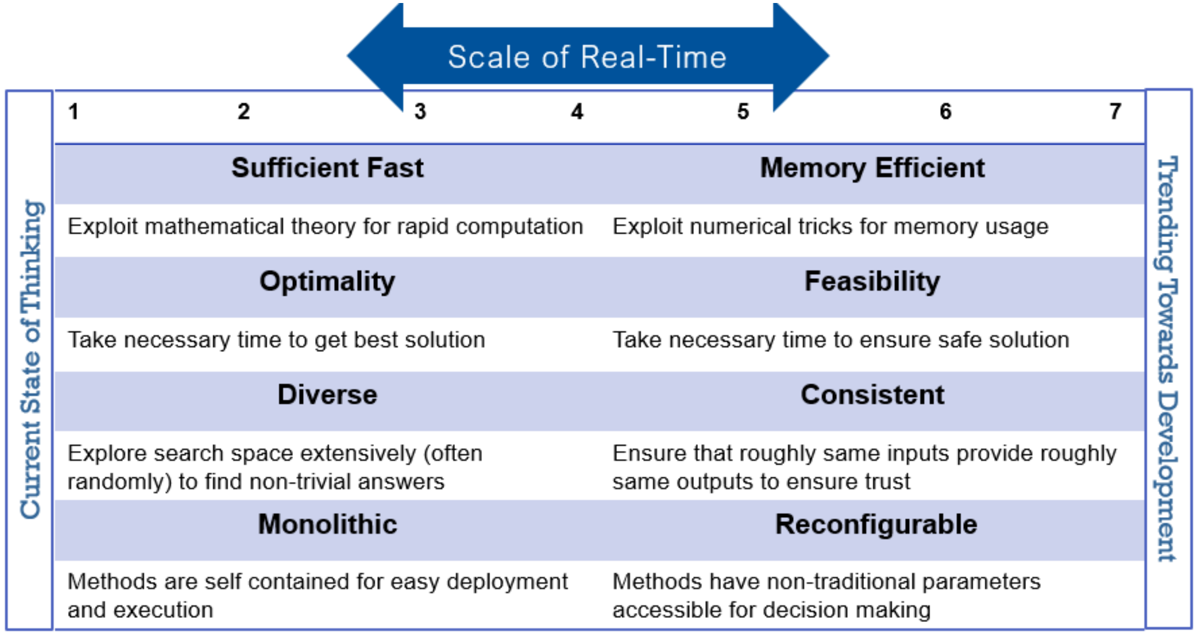




Aspects of Time for Assured Autonomy

Real-time – The ability for a vehicle to make decisions with the allocated computational resources on time fames necessary to complete the mission

- This is mission and vehicle dependent
- Does not imply sufficiently fast decisions at constant rate, real-time can imply decisions made asynchronously



Scale of Real-Time

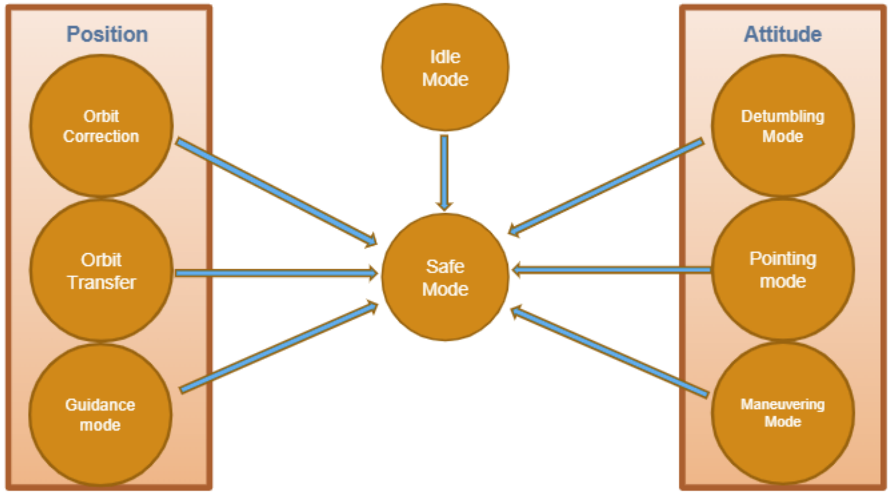
1 2 3 4 5 6 7

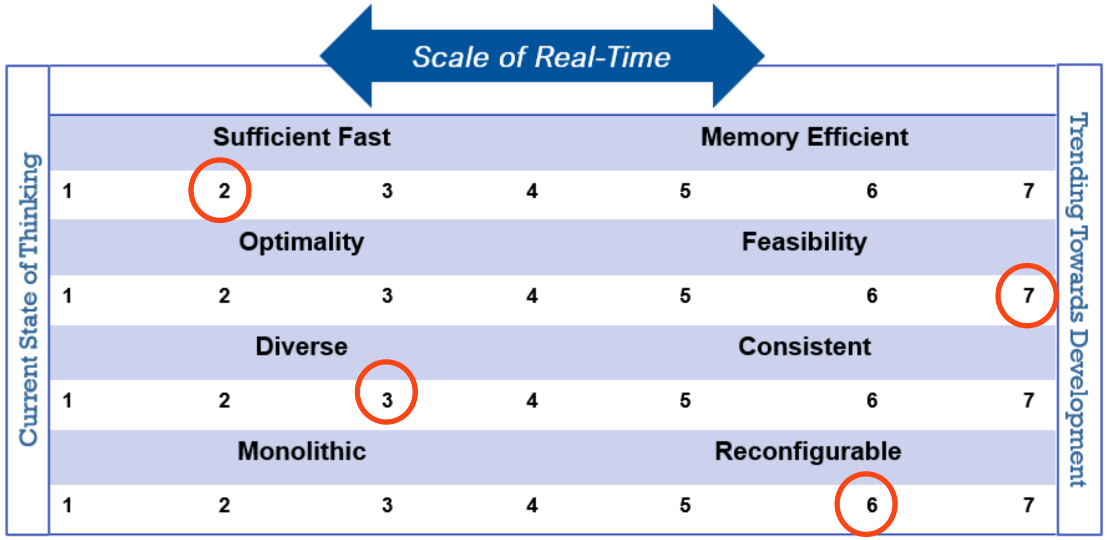
Current State of Thinking	Trending Towards Development
Sufficient Fast	**Memory Efficient**
Exploit mathematical theory for rapid computation	Exploit numerical tricks for memory usage
Optimality	**Feasibility**
Take necessary time to get best solution	Take necessary time to ensure safe solution
Diverse	**Consistent**
Explore search space extensively (often randomly) to find non-trivial answers	Ensure that roughly same inputs provide roughly same outputs to ensure trust
Monolithic	**Reconfigurable**
Methods are self contained for easy deployment and execution	Methods have non-traditional parameters accessible for decision making

Enhancement via Mode Switching

- *Credit: Mr. Faraz Abed Azad (G) , Xander Blunt (UG)*

Problem: External effects such as space weather can force a satellite into safe mode, keeping it safe but destroying the mission





Current State of Thinking	Scale of Real-Time							Trending Towards Development
Sufficient Fast	1	[✓ 2]	3	4	5	6	7	Memory Efficient
Optimality	1	2	3	4	5	6	[✓ 7]	Feasibility
Diverse	1	2	[✓ 3]	4	5	6	7	Consistent
Monolithic	1	2	3	4	5	[✓ 6]	7	Reconfigurable

Hypothesis: A mode switching paradigm can be used to keep the satellite safe while achieving mission objectives *"as much as possible"*

Failure Happens, Especially Due to Space Weather (SW)

Most Failures Happen Soon After Launch

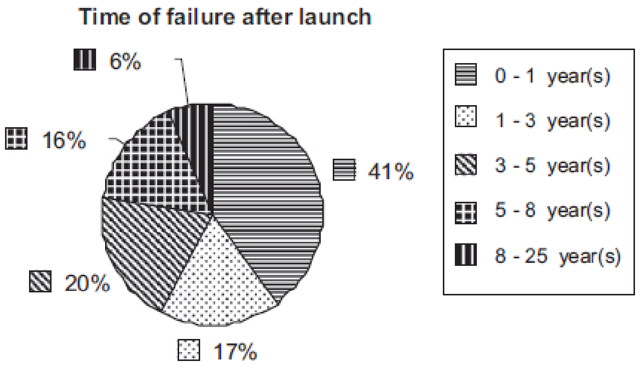


Fig. 3. Time of failure after launch.

M. Tafazoli/ActaAstronautica64(2009)195–205

Most Failures Are Due To Electrical

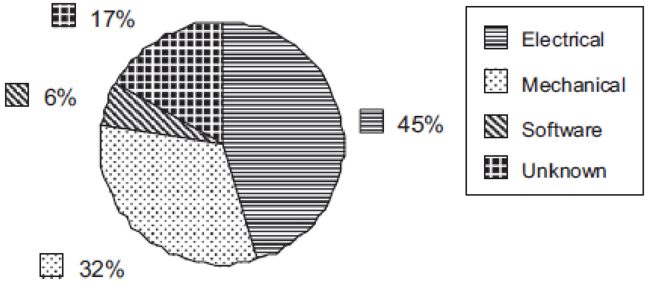


Fig. 2. Spacecraft failure type.

M. Tafazoli/ActaAstronautica64(2009)195–205

A number of electrical failures induced by SW

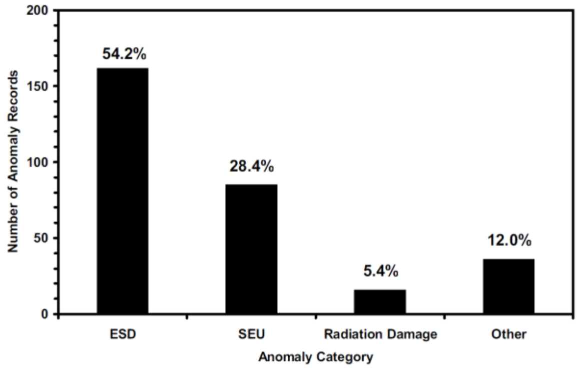


Figure 1. Spacecraft anomalies as a function of the space environment effect, where ESD is electrostatic discharge, SEU is single event upsets, Radiation Damage is total ionizing or non-ionizing dose, and Other represents other causes or unknown sources, from Koons et al., Aerospace Technical Report, 1999.

Effects of Space Weather on Technology Infrastructure

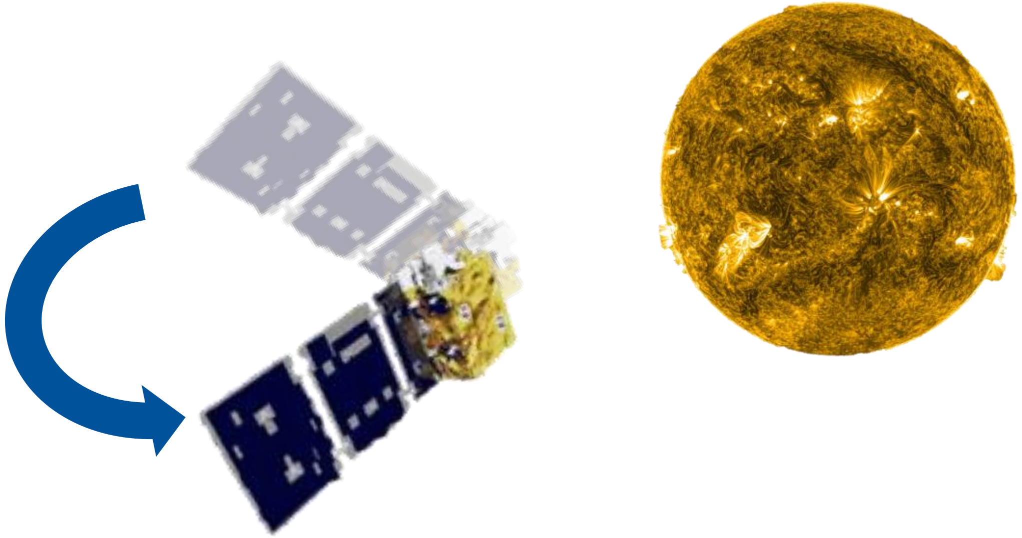
Most errors will place a satellite into “SAFE MODE”, stops the mission and puts satellite into minimal safe operations

Safe but not resilient

Resiliency via Prevention, Degradation & Recovery Set-up

- Safe mode will take you off mission
  - Safe but not resilient
  - Drifting can take satellites tremendously off mission

- When preparing or experiencing a fault, what is the best decision to make for
  - Preventing – Ensuring an impending impact does not hurt the satellite in the future
  - Safe degradation – Fail so that the mission can be achieved back to X% > 0%
  - Recovery Set-up – Fail so that you can get back on mission easier



Example – Changing satellite rotation and reducing on-board computations to reduce spacecraft potential.

Can save majority of the mission, even if unknowns are known or otherwise

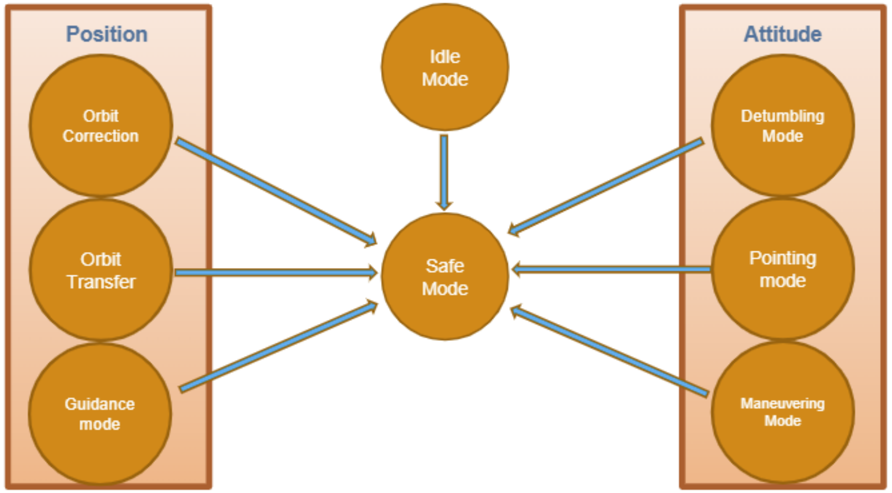
Markov Decision Process for Switching

Reward for Position

$$R^{pos}_{pointing} = -\sqrt{\frac{1}{3}(\mathbf{r}-\mathbf{r}_{des})^2}$$

$$R^{pos}_{maneuver} = -\frac{R^{pos}_0 - 1}{R^{pos}_0}\sqrt{\frac{1}{3}(\mathbf{r}-\mathbf{r}_{des})^2} - 1$$

Translational modes use mix of sliding mode and artificial potential functions



Reward for Attitude

$$R^{att}_{pointing} = -||\delta\hat{q}||$$

$$R^{att}_{maneuver} = -\frac{R^{att}_0 - 1}{R^{att}_0}||\delta\hat{q}|| - 1$$

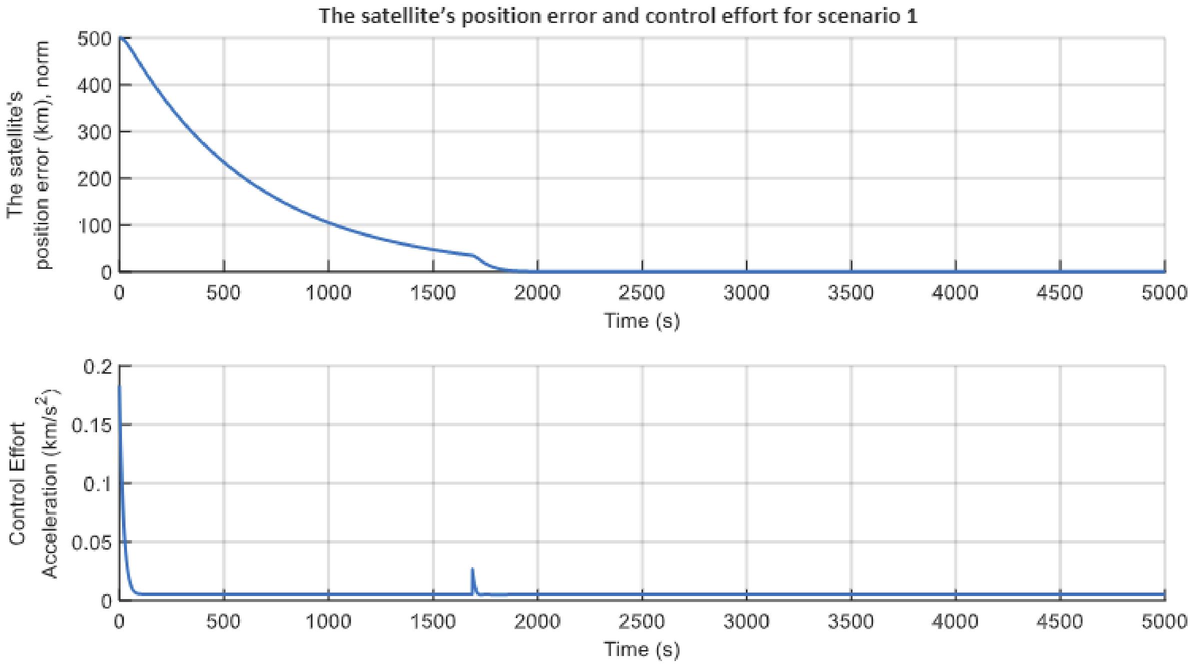
Rotational modes based on control Lyapunov functions

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	$\alpha < \frac{\hat{n}\vec{d}^T}{\|\|d\|\|_2} \leq 1$	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

Enables encoding of complexity to discrete nodes that are used in mission scheduling

Mode Switching With SW and Density Change

- CONOP 1
  - Satellite is in LEO when an alarm for a solar flare is to hit
  - Satellite moves to higher altitude to mitigate flux on drag
    - Solar flares affect the density of air in the atmosphere
  - Satellite switches modes at around approx. 1700 seconds
    - Goes from coarse correction mode to fine correction mode

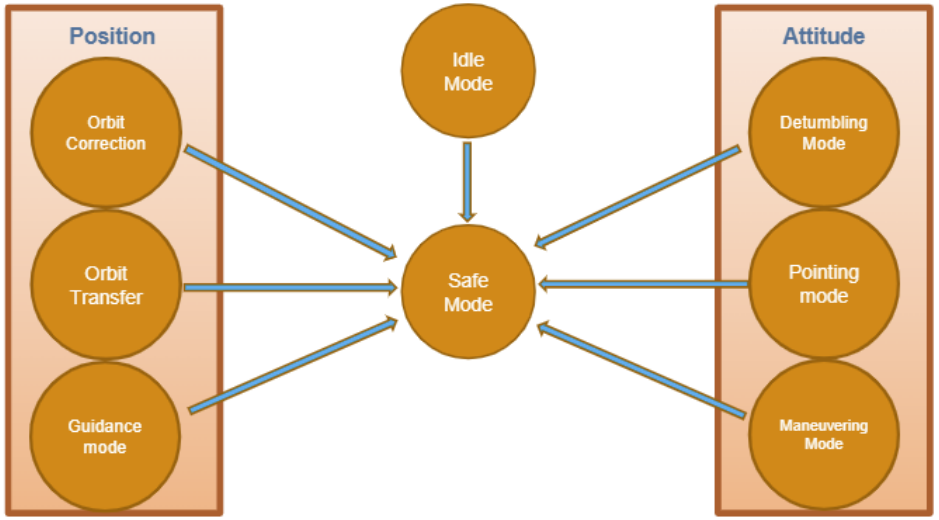

The satellite's position error and control effort for scenario 1

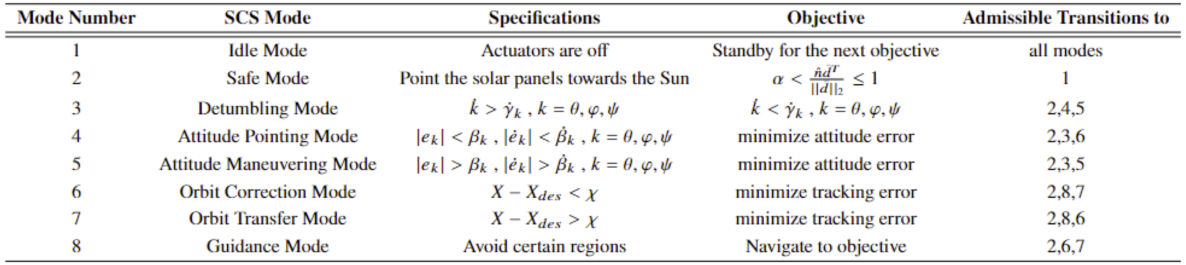
Satellite modes encoded and can reject small amounts of perturbations

Azad, F., Petersen, A., and Petersen, C. "Autonomous Satellite Operational Mode Switching for Anomalies and Space Weather Effects Mitigation" 2024 AIAA SCITECH. To Appear

Next Steps

- Incorporate optimization approaches in the nodes
- Each mode in the spacecraft system is a function of tunable parameters
  - Some are control focused like error gains
  - Some are computational based, like how far ahead to look and counter act
- Create a process that modifies tunable parameters so that satellite does not go into safe mode





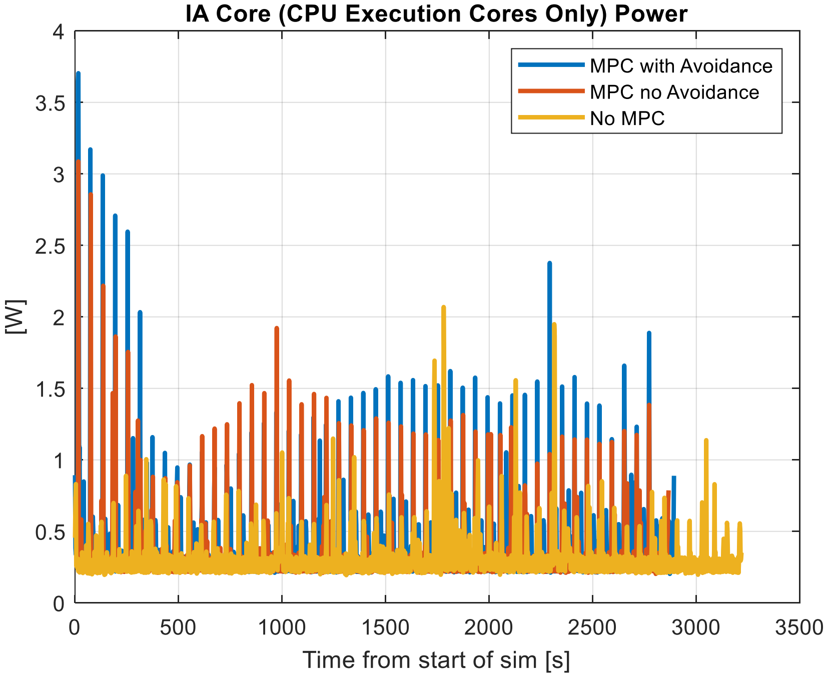
Mode Number	SCS Mode	Specifications	Objective	Admissible Transitions to
1	Idle Mode	Actuators are off	Standby for the next objective	all modes
2	Safe Mode	Point the solar panels towards the Sun	$\alpha < \frac{\hat{n}\vec{d}^T}{\|\|d\|\|_2} \leq 1$	1
3	Detumbling Mode	$\dot{k} > \dot{\gamma}_k$, $k = \theta, \varphi, \psi$	$\dot{k} < \dot{\gamma}_k$, $k = \theta, \varphi, \psi$	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

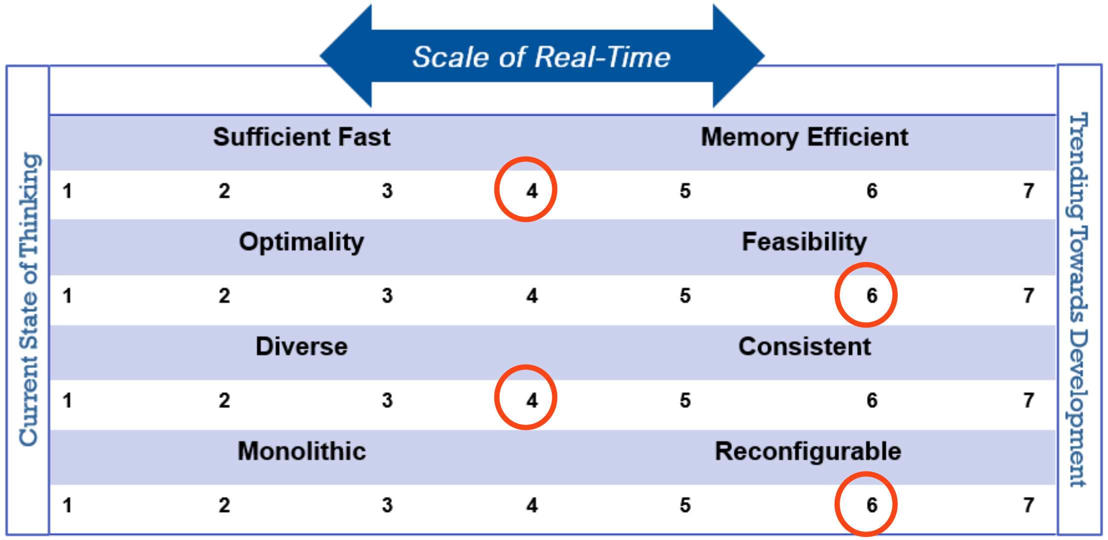
End Goal: Smart Fault Mitigation

Enhancement via Computation Considerations

- *Credit: Channing Ludden(G) , Sarah Clees (UG)*

Problem: One large barrier to implementation of autonomy is complexity, yet only one metric (computation time) is ever assessed and always treated as if it cannot be fixed in situ




Scale of Real-Time

Current State of Thinking								Trending Towards Development
Sufficient Fast	1	2	3	[✓ 4]	5	6	7	Memory Efficient
Optimality	1	2	3	4	5	[✓ 6]	7	Feasibility
Diverse	1	2	3	[✓ 4]	5	6	7	Consistent
Monolithic	1	2	3	4	5	[✓ 6]	7	Reconfigurable

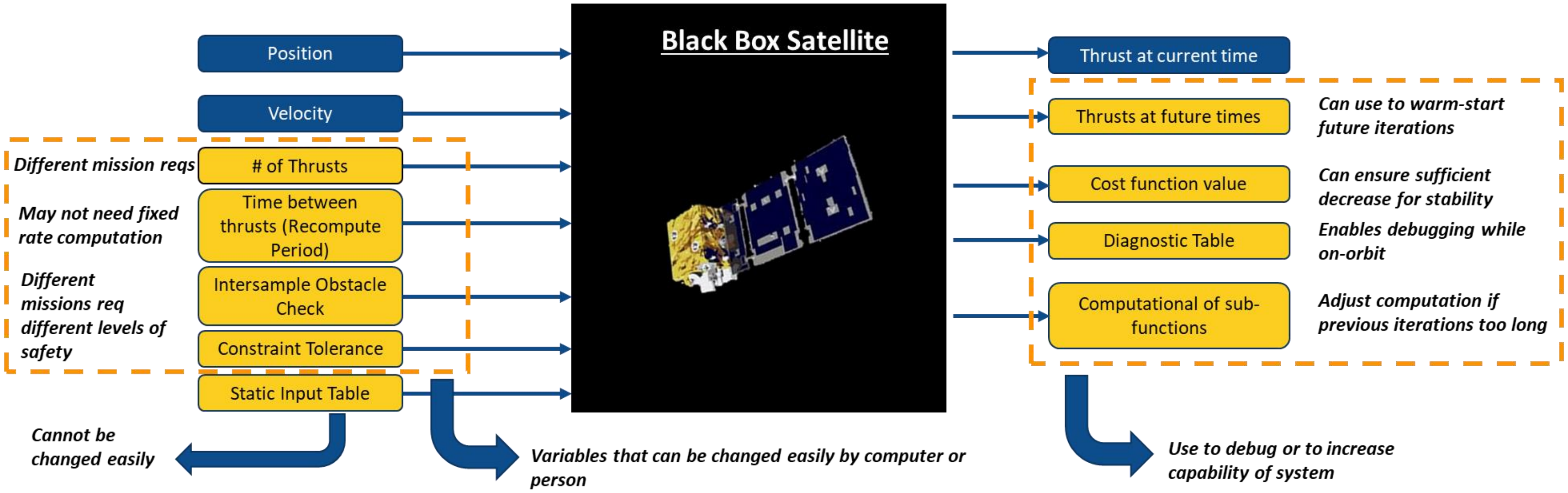
Hypothesis: Computation metrics can be quantified with their own "dynamics" which are functions of the complexity of the algorithm. These metrics can be adjusted in situ for real-time implementation

Implementation of Autonomy

When implementing MPC on a vehicle, an embedded system will receive a function command

`[output]=MPC_execute(input)`

Question: What should that command contain?



This understanding comes from understanding control and optimization are two coupled processes, not one

Understanding of what is easily accessible enables full system exploitation in unique ways by standard algorithms

Is there a correlation between what is accessible and "real-time"

Autonomy Metrics to Assess “Real-time”

Metric	Is it monitored	How it is accounted?	Algorithm impact known?
Computation Time	Yes	Worst case execution, time delay system	Mostly
Memory	Yes for system No for algorithm	Upper limit of static memory allocation	Naively
CPU Usage	Yes for system No for algorithm	Not	No
Power Usage	Yes for system No for algorithm	Not	No

If it reaches a certain threshold, system locks out (e.g., blue screen of death)

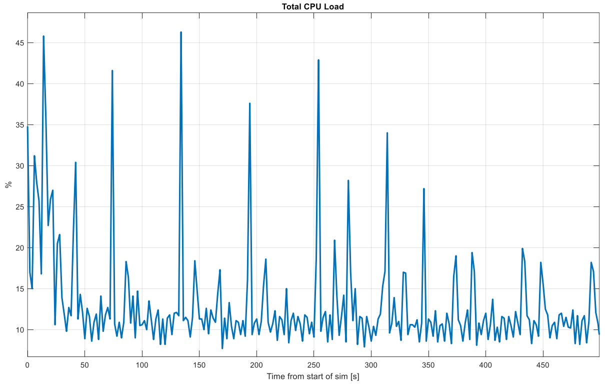
Question

Lack of understanding results in conservative designs and **unsure autonomy**

Question

Do the metrics have “dynamics” where the “inputs” are algorithm parameters?

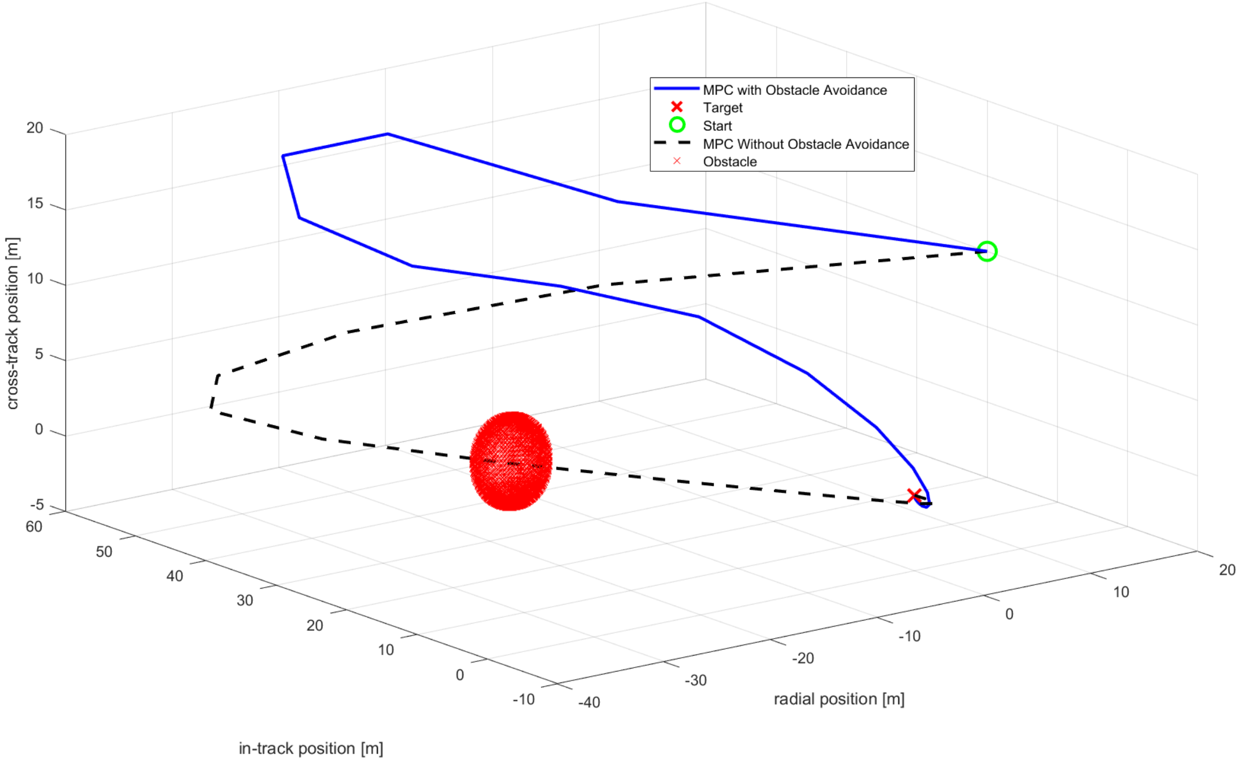
Computation-In-The-Loop



CPU% looks like it has dynamics!

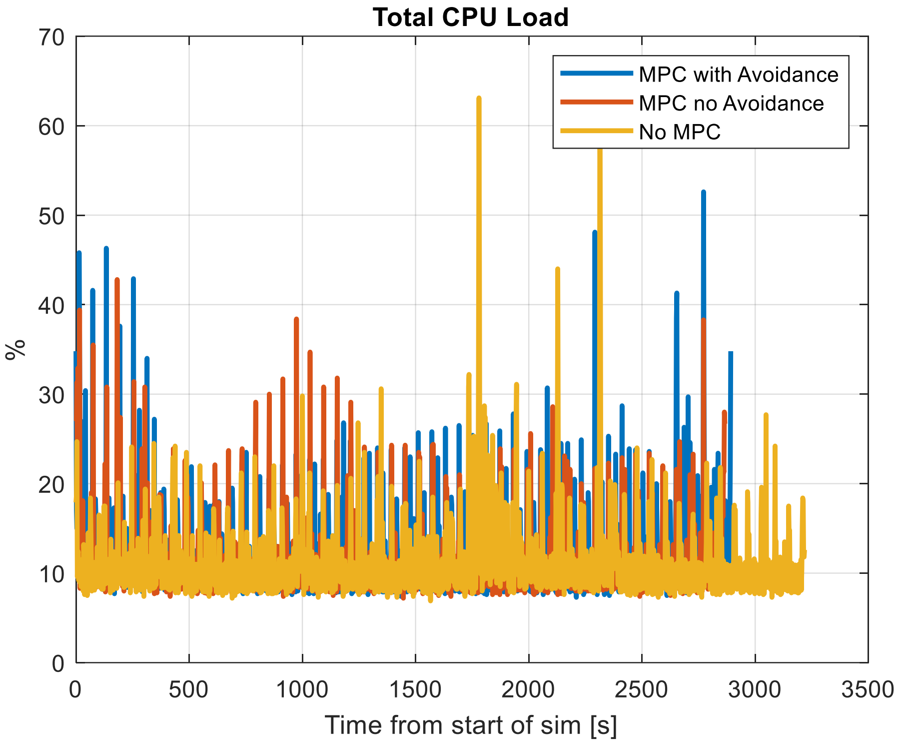
Scenario

- Satellite is docking with another satellite
- Two algorithms are implemented
  - One with obstacle avoidance
  - One without obstacle avoidance
  - Both have control constraints
- Path is solved using QCLC formulation
  - Obstacle is dealt using convex hyperplane technique
  - Solver is custom made QP
- Useful parameters
  - Satellite ~ 30 m away, stagged for docking
  - Control rate/discretization 60 seconds
  - Horizon length is 100 steps (~greater than 1 orbit)
- Computation metrics measure on Microsoft Surface 3, executed as if in "real time"
  - Computer measured with minimal processes too

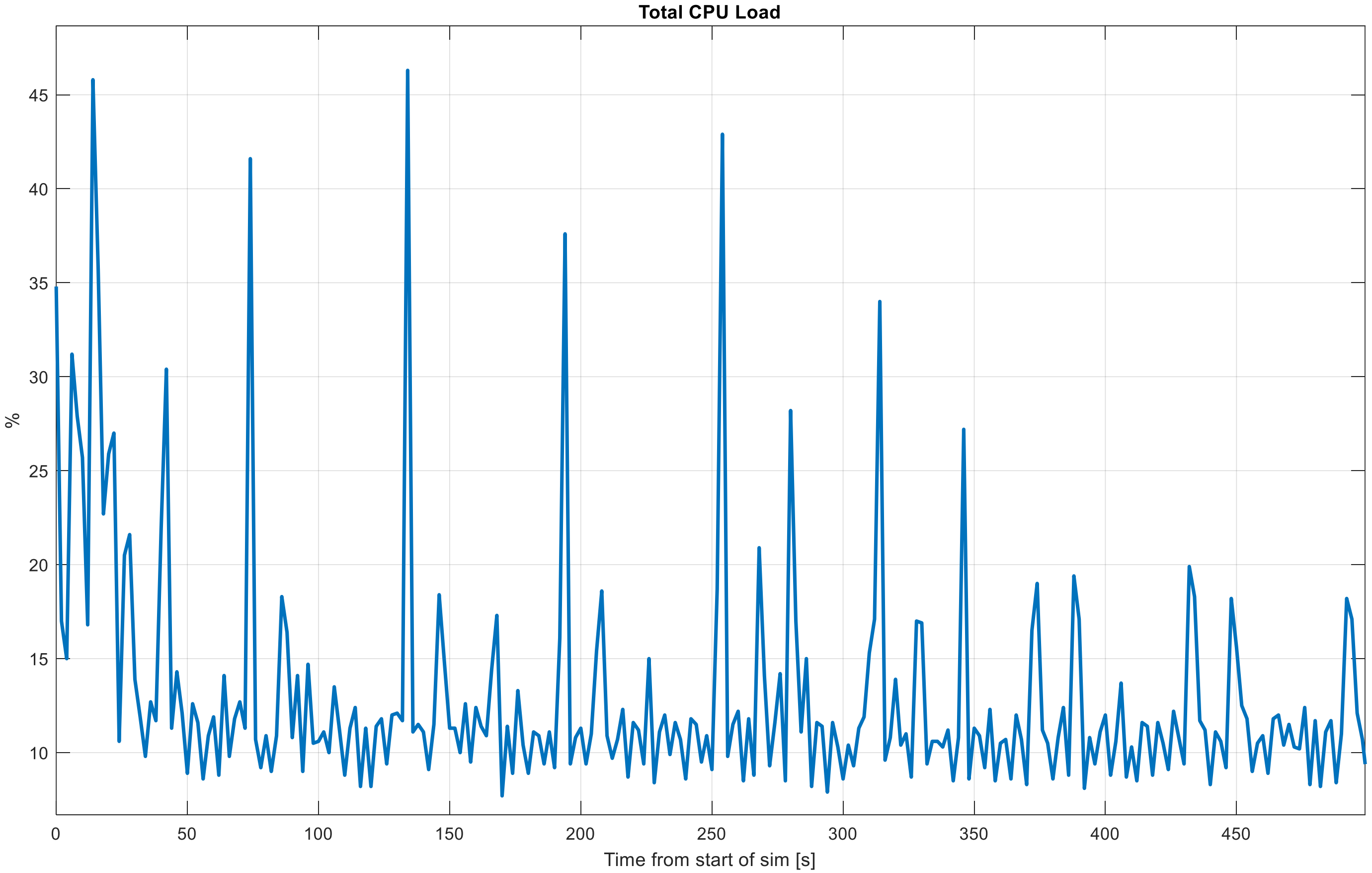


Objective: How do the computational metrics vary and evolve temporally?

CPU Load

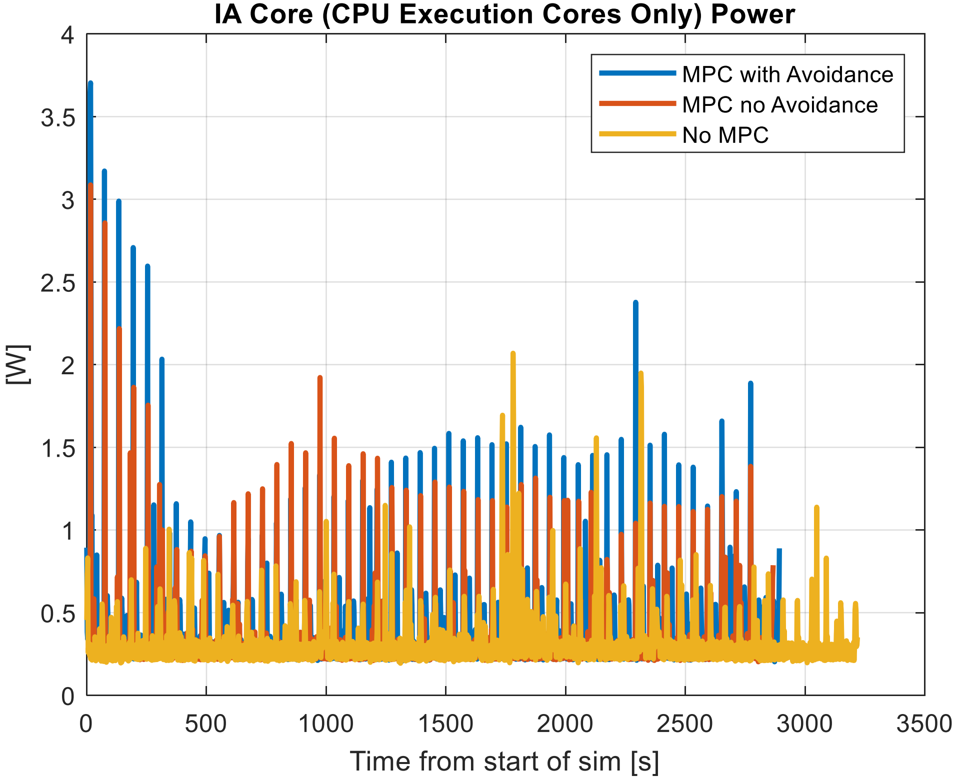


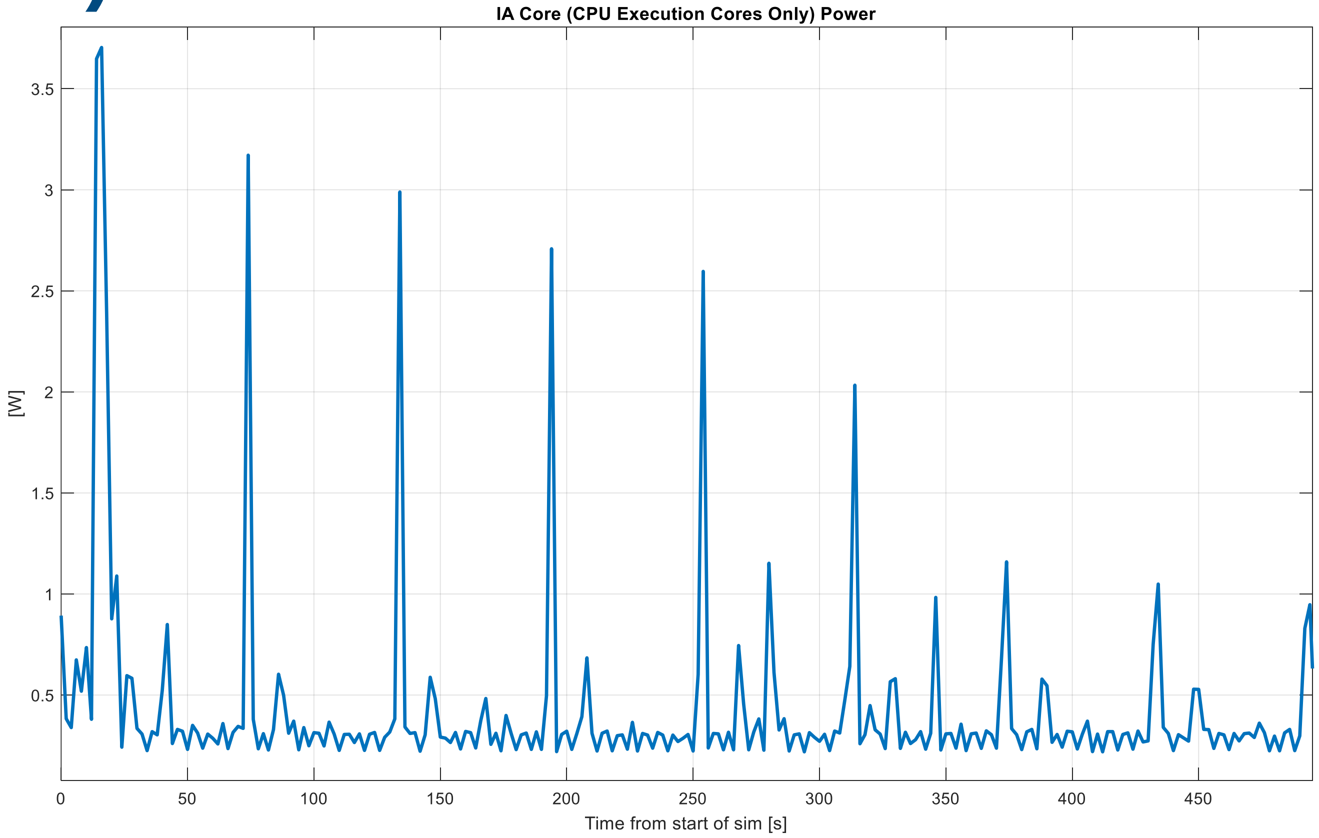
Total CPU Load > 2x most time



CPU appears as an asymptotically stable system with small disturbance and impulse input, some transients before

IA Cores (CPU Execution)

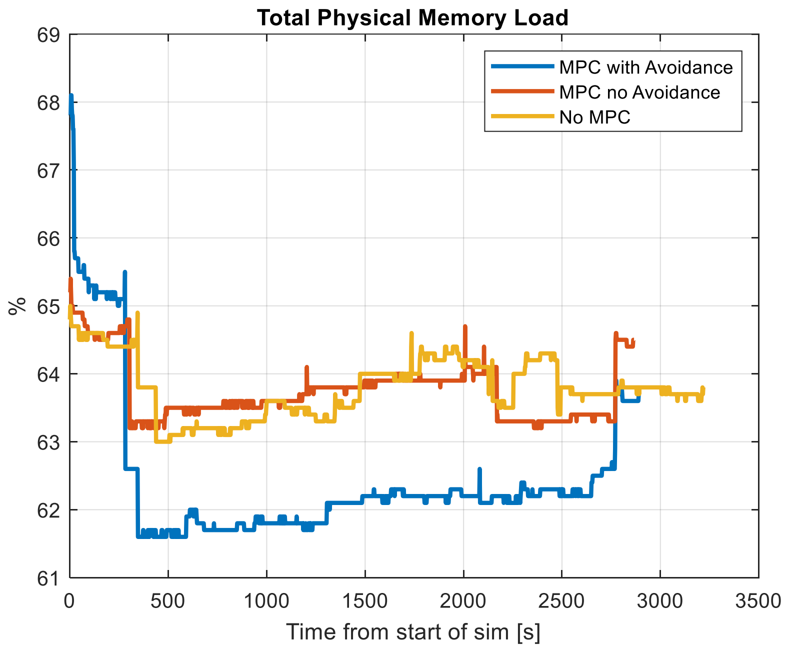


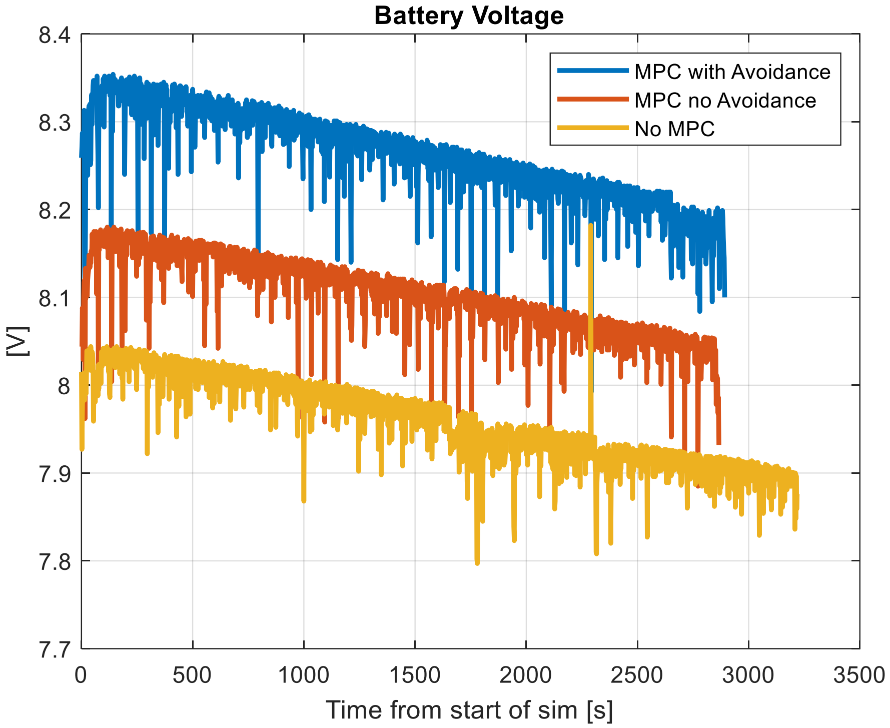


Power >4x most the time

Power consumption also appears as an asymptotically stable system with small disturbance and impulse input,

Memory and Voltage



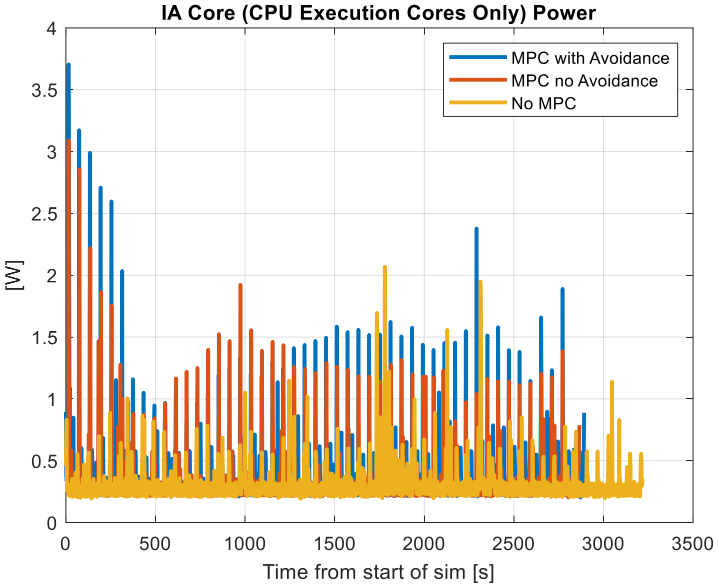


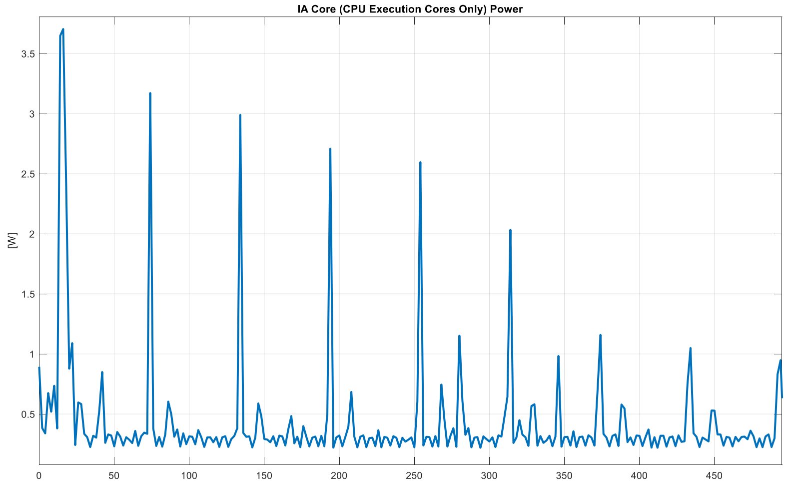
Memory for obstacle avoidance spikes greater.
Memory for no avoidance seems similar to no MPC at all

Voltage appear to have little change at this point though the beginning of transients are different

Next Steps

- Implement nonlinear optimization process
    - Lab has numerous processors
- Describe temporal computation mathematically
- Establish a list of "optimization" inputs and see how they drive computational dynamics
- Create a mechanism to dial-up and down computation





End Goal: Automated Process that Adjust Parameters In-situ Depending on Computation Resource

Decision-Making Under Ignorance

- *Credit: Joseph Direnzo (G)*

Problem: USSF satellites will need to act autonomous, optimizing over several objective, when information is not fully known

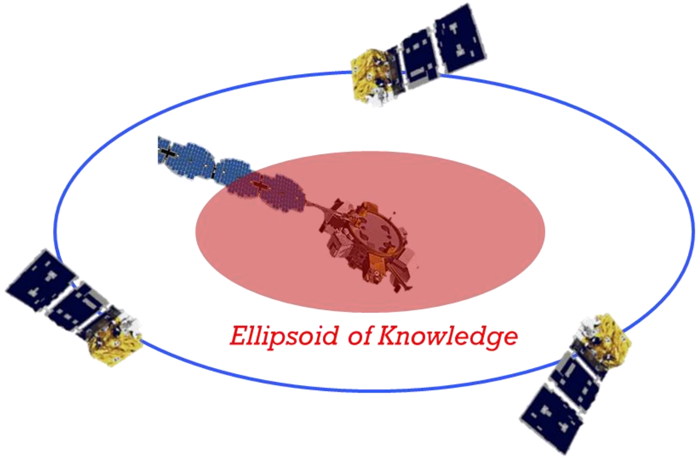
Solution: Multi-objective techniques that a) balance mission goals and objective b) retain constraint enforcement and consistency during operations to enforce safety (even if conservative), and c) gain information when not available

Focus Areas

- Develop stochastic optimization methods that are relatively quick (e.g. do not rely on extensive Monte Carlo) and provide consistent solutions
- Leverage lexicographic optimization to make decisions over multiple metrics
- Develop metrics to quantify obtaining information in order to act under ignorance

Challenges

- Stochastic optimizations are difficult to ensure consistent safety
- How to optimize over information when structure of ignorance is not exactly known

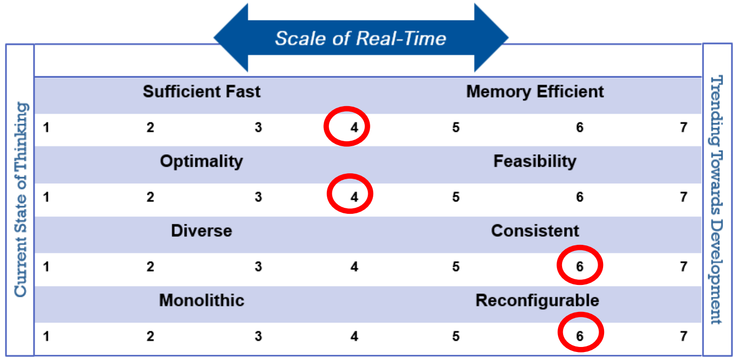


Multi-Objective Optimization that Quantifies Ignorance

$$
\begin{array}{llll}
 & \min_{\mathcal{V}} J_j(\mathcal{V}) & & \rightarrow \textit{Current Optimization Metric} \\
\text{subject to} & & & \\
 & \alpha_{i+1} = F_D(\alpha_i, \beta_i) & i = 0, ..., N-1, & \rightarrow \textit{State Variables} \\
 & \gamma_{i+1} = F_K(\alpha_i, \beta_i, \gamma_i), & i = 0, ..., N-1, & \rightarrow \textit{Variance Variables} \\
 & g_i(\alpha_i, \beta_i, \gamma_i) \leq 0, & i = 0, ..., N-1, & \rightarrow \textit{Safety Constraints} \\
 & W_i(\mathcal{V}) \in W_i^*, & i = 1, ..., k \leq N_o, & \rightarrow \textit{Mission Objectives} \\
 & J_i \leq J_i^* + \epsilon_i, & i = 1, ..., j, & \rightarrow \textit{Prev Optimization Metrics} \\
 & \alpha_0 = x(t), & & \rightarrow \textit{Connect State to Real Physics} \\
 & \gamma_0 = f(P_D(t), P_T(t)), & & \rightarrow \textit{Connect Variance to Real Physics}
\end{array}
$$

Optimize over Expectation to Ensure Consistency

$$\min_{\mathcal{V}} \mathbb{E}(J_j(\mathcal{V})) + \lambda^{\mathrm{T}}\mathbb{E}(G_j(\mathcal{V})) + \kappa\left[\mathbb{V}(J_j(\mathcal{V})) + \lambda^{\mathrm{T}}\mathbb{V}(G_j(\mathcal{V}))\right]$$



Sufficient Fast				Memory Efficient		
1	2	3	[✓ 4]	5	6	7
Optimality				Feasibility		
1	2	3	[✓ 4]	5	6	7
Diverse				Consistent		
1	2	3	4	5	[✓ 6]	7
Monolithic				Reconfigurable		
1	2	3	4	5	[✓ 6]	7

Questions