

Ensuring Safety via Computation Regulation and Reference Switching



Mr. Faraz Abed Azad

PhD Student

Mr. Channing Ludden

PhD Student

Dr. Christopher “Chrispy” Petersen

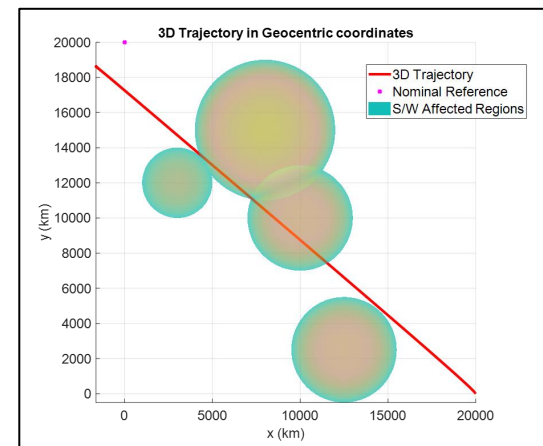
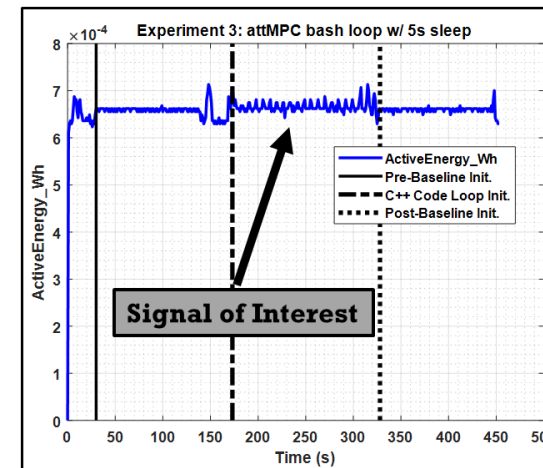
Assistant Professor





Outline

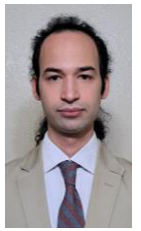
- Lab, Success, Recent Publications
- Real-time Refresher
- Enhancing Safety via Computation Regulation
- Enhancing Safety via Reference Switching





Spacecraft Technology And Research (STAR) Lab Students

Graduate Students



Faraz Abed Azad



**RY/
ACT3**

Channing Ludden

Diverse pool of idea and successful transition for workforce development in AFRL

Undergraduate Students:



RQR

Sarah Clees




Adam Sardouk



Trevor Evetts


ROTC



RV

Cannon Whitney

ROTC



Will Rummy



Matt Krininger

All progress here is due to them



Recent Publications and Success

Lab Successes

- Lab Processor Infrastructure Up and Running
- Work on COE enabled SFFP
- Student Engagement via Space Cyber Reading Group
- Multiple Space Autonomy Sessions
- Assured autonomy interface work made it to final rounds of AFRL hub



Publications:

- Ludden, C., Petersen, C., “Characterizing Computational Resources of GNC Algorithms,” *Space Mission Challenges for Information Technology/Space Computing Conference*, IEEE, Accepted, 2024
- Azad, F., Petersen, A., Petersen C., “Autonomous Satellite Operational Mode Switching for Anomalies and Sace Weather Effects Mitigation,” *Proceedings of the AIAA SciTech/Spaceflight Mechanics Meeting*, AIAA/AAS , 2024.
- Petersen C., “A Coupled Guidance & Navigation Optimization to Improve Rendezvous and Proximity Operations,” *Proceedings of the AIAA SciTech/Spaceflight Mechanics Meeting*, AIAA/AAS , 2024.

A dark space background featuring a view of Earth's horizon in the bottom left corner and a crescent moon in the top right corner. The text "Refresher on Real-time" is centered in white.

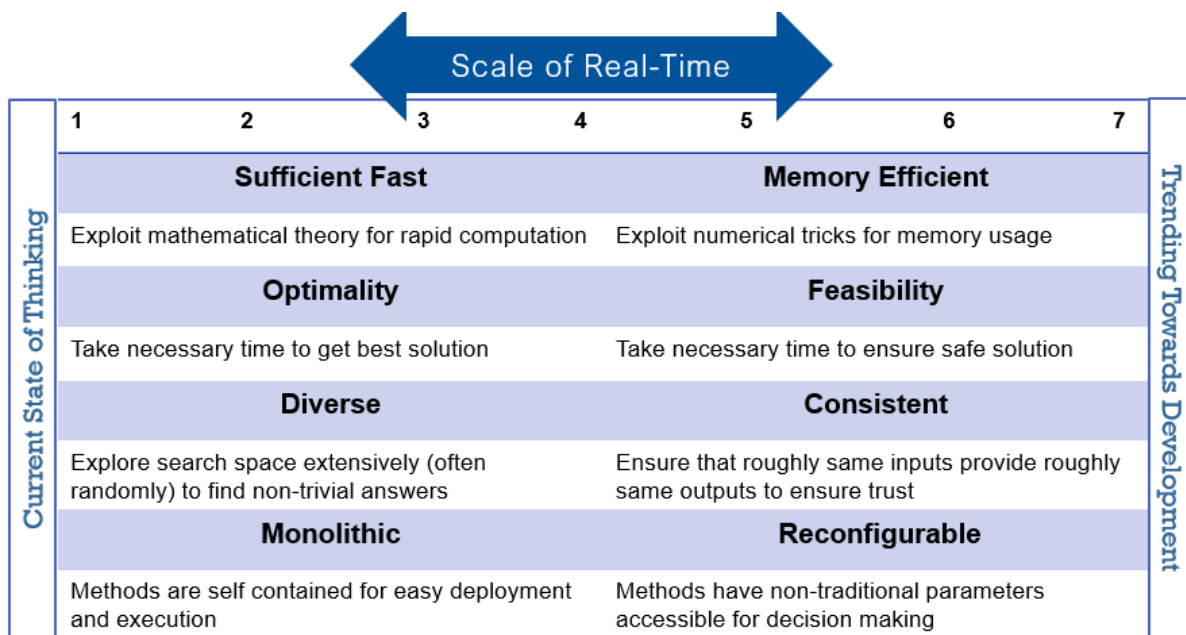
Refresher on Real-time



Aspects of Time for Assured Autonomy

Real-time – The ability for a vehicle to make decisions with the allocated computational resources on time frames 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



A dark space background featuring a view of Earth's horizon in the bottom left corner and a crescent moon in the top right corner. The text is centered in the middle of the frame.

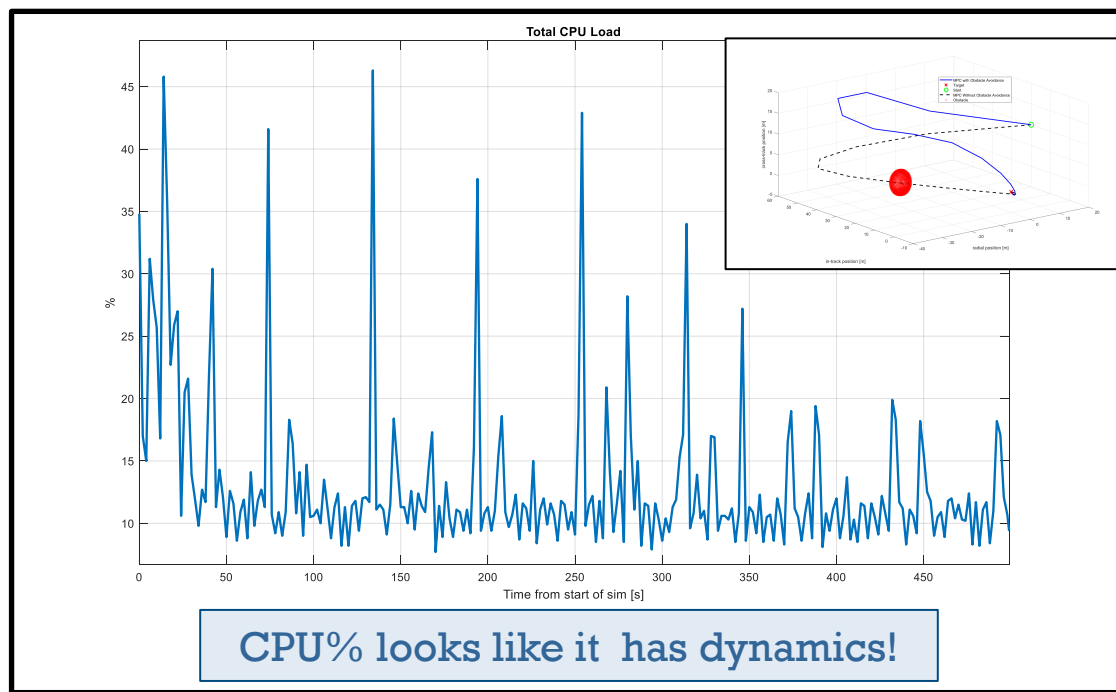
**Real-time Spacecraft Autonomy Enabled via
Computation Regulation**



Enhancement via Computation Considerations

Credit: Channing Ludden(G) , Sarah Clees (UG), Cannon Whitney (UG), Matt Krininger (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	Sufficient Fast			Memory Efficient			Trending Towards Development	
	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		

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



STAR Labs Computation Set-up

Objective:

- Assess computationally limited HW vs. computationally expensive SW
- Deployment to Space GNC separate with separate satellite digital twin
- Measure & assess metrics: Power consumption, memory, CPU%, etc.

Expected Outcomes:

- Can we generate a predictive model of computation?
- Can we freely adjust computation?
- Can we impose optimization constraints on computation?

Satellite "Bus"



Separate Satellite "Physics Engine"/Digital Twin



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 RAD 750



- True Flight Rad Processor
- BAE development harness (VXWorks-like)
- Used on a several long-duration flight missions (25+ years)

← Off the Shelf, Simple
Computationally Limited

→ Rad Tolerant & Hardened, Custom
Computationally Limited

Wide range of processors that are not just for spacecraft



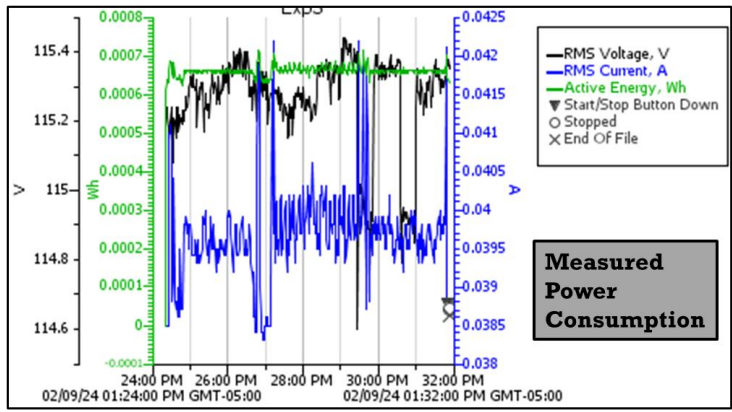
Test Scenario

- Two satellites within RPO regime
- One satellite is trying to optimally point & inspect (LQR cost) another satellite
- There are safety constraints
- Real-time MPC controller based on accelerated gradient
 - Code flew previously with <24 kb but was not executed (sat did not turn on)

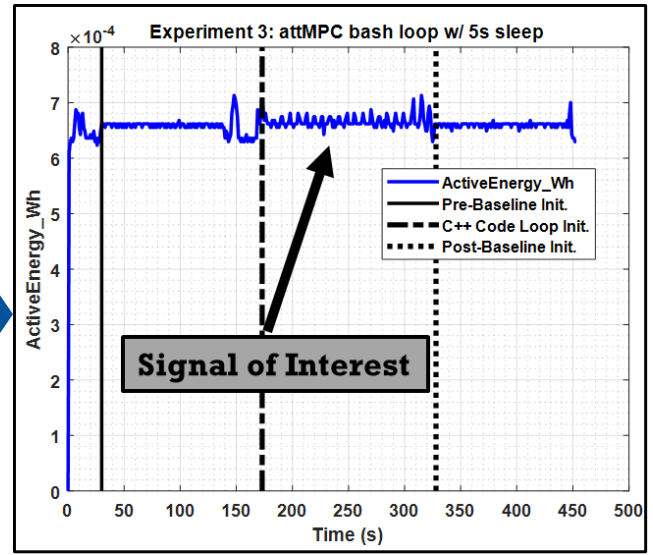


Objective: Observe computational metrics on other processors like on computer

Initial Raspberry-Pi B+ running looped MPC algorithm on Raspbian OS



Raw UX120-018 Data (HOBO SW .hobo file)



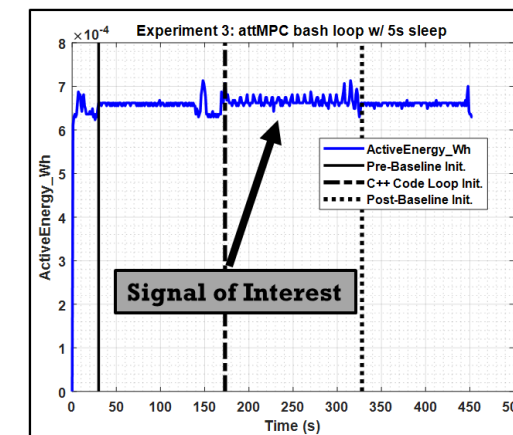
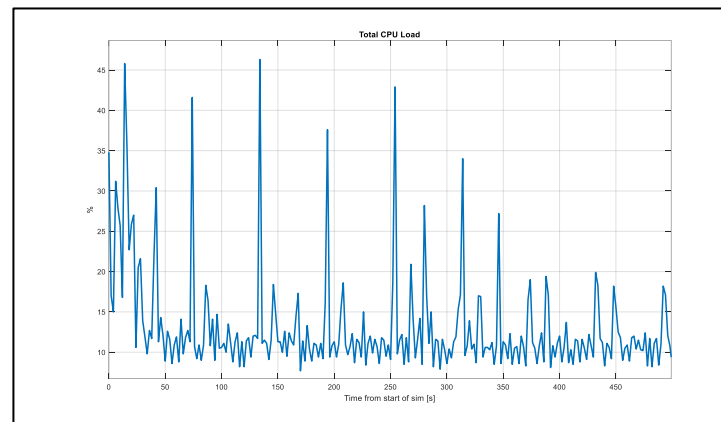
Processed Data

Data shows trends on Raspberry Pi similar to those we see on the computer



Next Steps

- Continue testing on variety of boards
- Determine a preliminary integrator-like system
- Investigate control of positive systems for computational control



- Other AF/USSF Synergies > SFFP
 - “Enabling Spacecraft Autonomy through Metrics and Computation-In-The-Loop”
 - Will assess NN/ML methods for spacecraft real-time (AFRL/Ry ACT3)
 - Will see if “computational throttling” can provide a software solution for implementation



End Goal: Computational Modeling and Throttling

A dark space background featuring a view of Earth's horizon in the bottom left corner and a crescent moon in the top right corner. The text is centered in the middle of the frame.

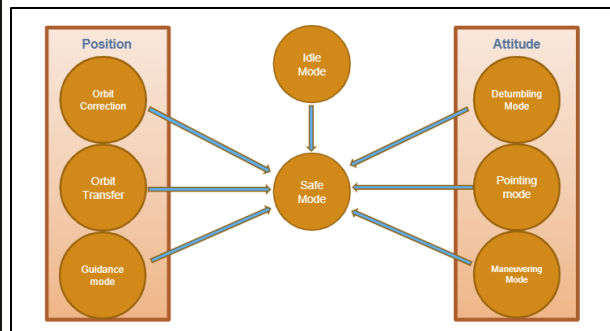
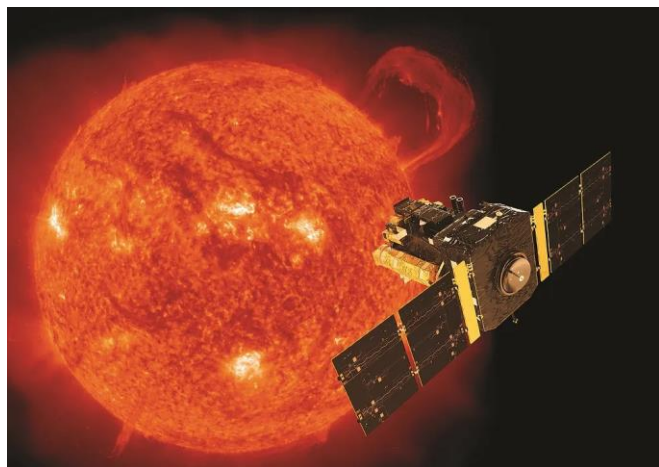
**Real-time Spacecraft Autonomy Enabled via
Regulation Switching**



Enhancement via Mode Switching

▪ Credit: Faraz Abed Azad (G), Nick Furiso (G), Dr. Alica Petersen (AP)

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



← Scale of Real-Time →

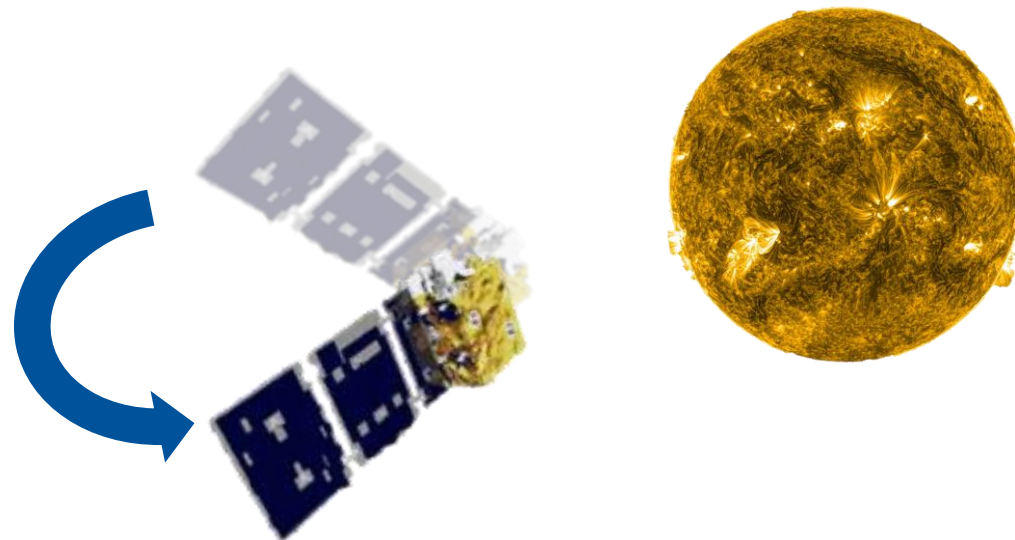
Current State of Thinking	Sufficient Fast			Memory Efficient			Trending Towards Development	
	1	2	3	4	5	6		7
	Optimality			Feasibility				
	1	2	3	4	5	6		7
	Diverse			Consistent				
1	2	3	4	5	6	7		
Monolithic			Reconfigurable					
1	2	3	4	5	6	7		

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



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



Concept of Safety

- Safety commonly is divided into “hard” and “soft” constraints
 - Hard constraints: Those that must not be violated
 - Soft constraints: Those that can be violated a little
- Both above have notions of boundaries that “stop” at safety
- However, there is another type of safety which we are calling **risk accumulation** safety
 - You have to traverse an area, accumulating risk as you maneuver through space
 - This is similar to an integral gain in
- Such a philosophy is akin to space weather, where there are regions in orbit of high activity

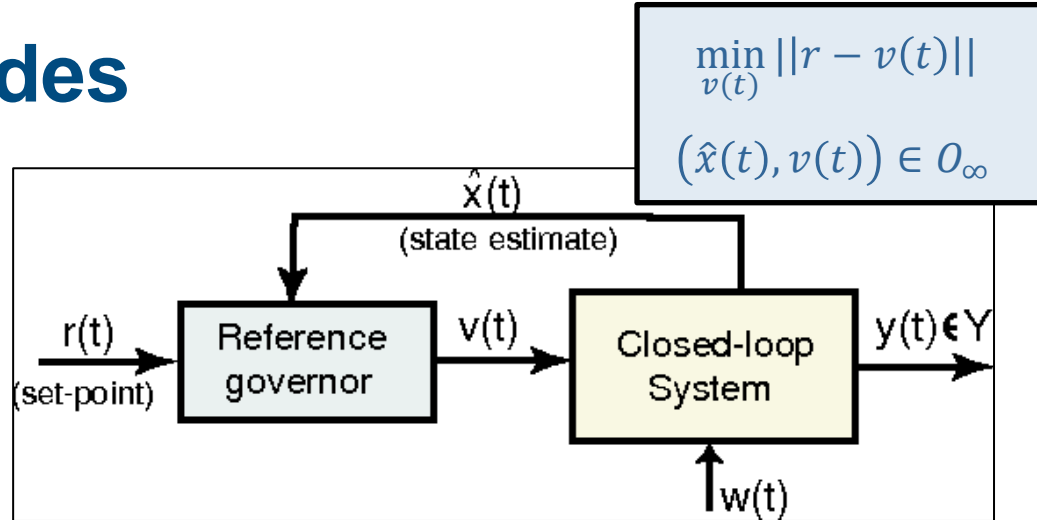


Avoiding these areas is not necessary, but durations of intensity can harm satellites (think sun burn)



Reference Governors in Safety Modes

- A reference governor acts upon a **closed-loop stable system**, continually modifying the reference to satisfy constraints.
- Tie to assured autonomy: the closed-loop system could be automated, NN/AI/ML, complex optimization, etc.
- Safety constraints are imposed via safe positive invariant sets (i. e., O_∞)
- We append the traditional reference governor with a weighting term on the risk-accumulation, $S(v(k))$, which maps **reference to risk**



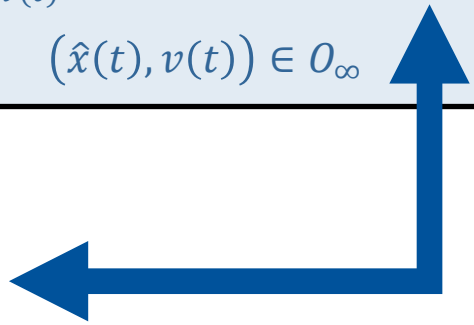
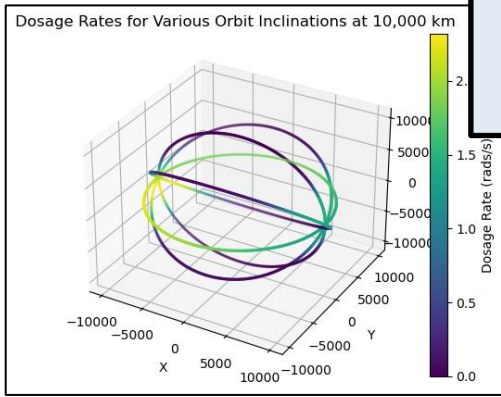
$$\min_{v(t)} ||r - v(t)||$$

$$(\hat{x}(t), v(t)) \in O_\infty$$



$$\min_{v(t)} ||r - v(t)|| + \alpha \int S(v(t)) dx$$

$$(\hat{x}(t), v(t)) \in O_\infty$$



Applying a governor scheme enables “as close as possible” to the objective while weighting risk due to space weather



Intuitive for military operators to understand the autonomy



Mathematical Hypothesis

Hypothesis 1

Given a closed-loop asymptotically system with a risk-accumulation reference governor scheme using a closed and bounded function S , there exists a sufficiently large α to have the actual reference $v(t)$ converge arbitrarily close to the desired reference asymptotically.

$$\forall \delta > 0, \exists \alpha > 0 \mid \lim_{t \rightarrow \infty} \|r - v(t)\| \leq \delta$$

I can get close enough if I want to

Hypothesis 2

Given a closed-loop asymptotically system with a risk-accumulation reference governor scheme. Let S be closed, bounded, and have a global minima at r . Then there exists a sufficiently large α to have the actual reference $v(t)$ converge to the desired reference asymptotically.

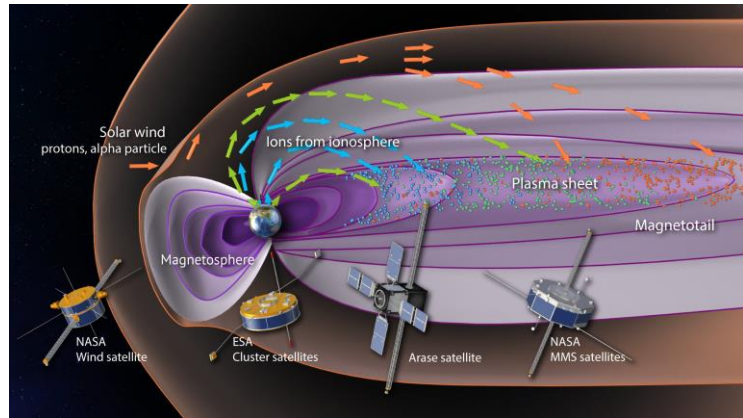
$$\exists \alpha > 0 \mid \lim_{t \rightarrow \infty} \|r - v(t)\| = 0$$

I can get exactly where I need if conditions are correct



Test Scenario

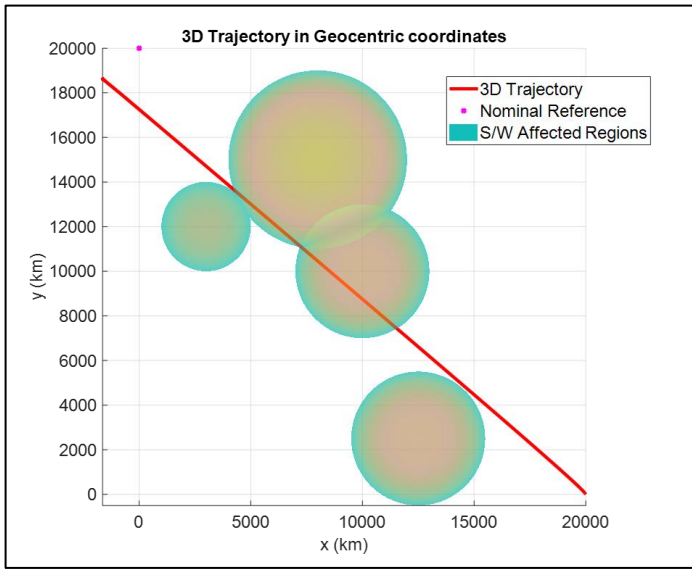
- Satellites are maneuvering in an area where there are pockets of high solar activity
- Satellite is trying to get to its desired orbit for its mission
- Only use a single mode



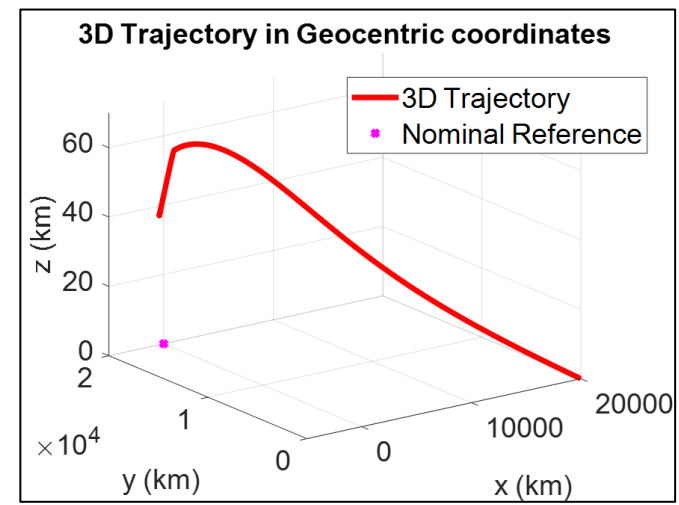
Objective: Get “as close as possible” to goal while mitigating risk accumulation

Initial Raspberry-Pi B+ running looped MPC

algorithm on Raspbian OS



Trajectory with pockets of solar activity



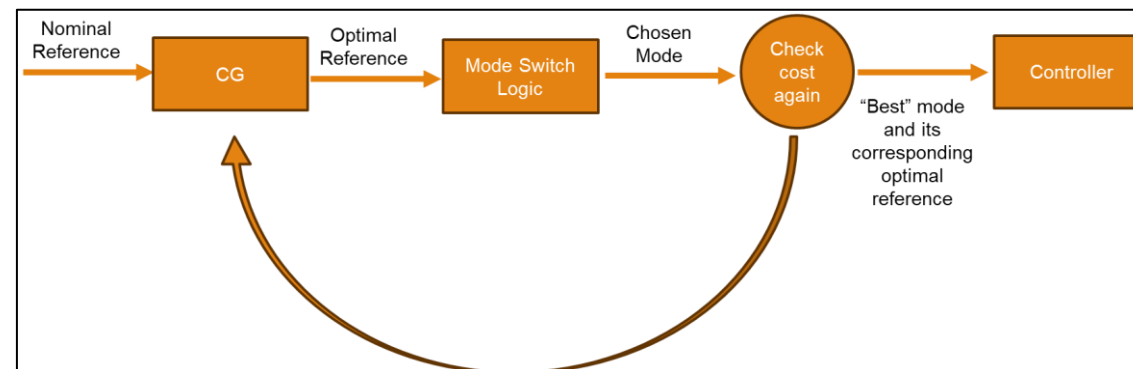
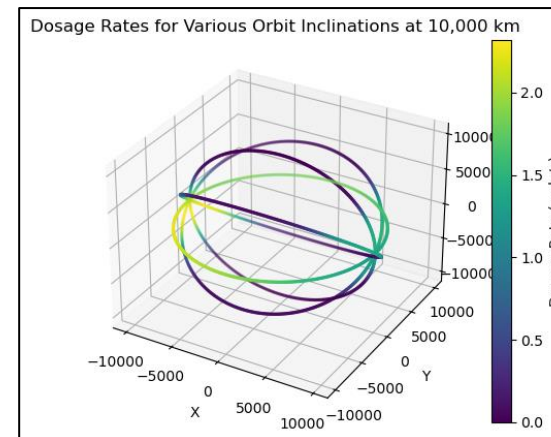
3D plot with just reference

Results show that the spacecraft will modify the reference to get “close” but is not afraid to skirt through regions of activity



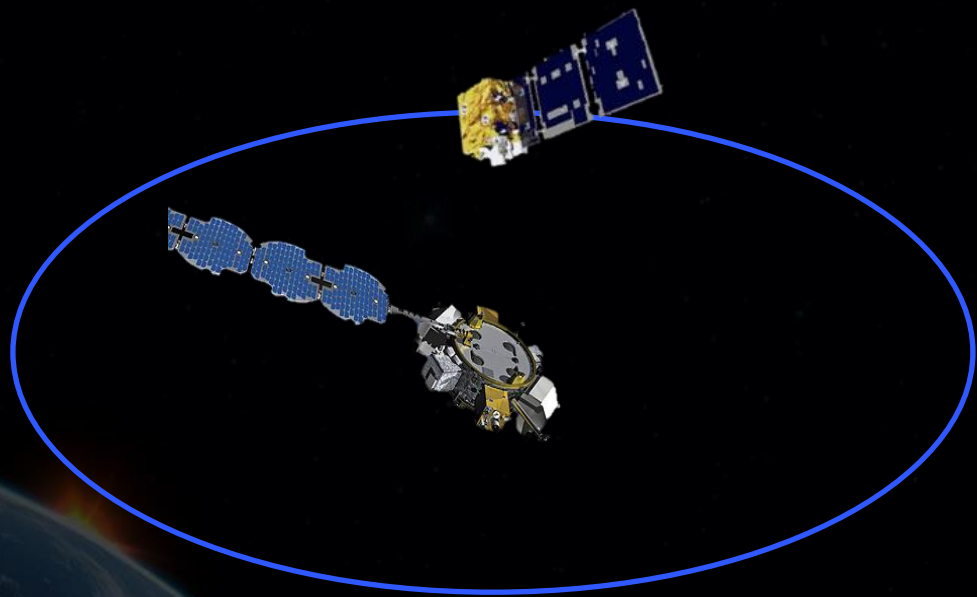
Next Steps

- Incorporate the solar model from SWIFT Lab
 - Hot off the presses results
- Use this idea of reference switching when multiple satellite modes are in the mix
- Demonstrate “stability-like” properties,
 - Convergence to the “best” solution when accumulating risk
 - Convergence to exact solution under nominal properties
- Look at synergies in other domain
 - Traversing in domains with RF risk (air/ocean)



End Goal: Smart Fault Mitigation

Questions



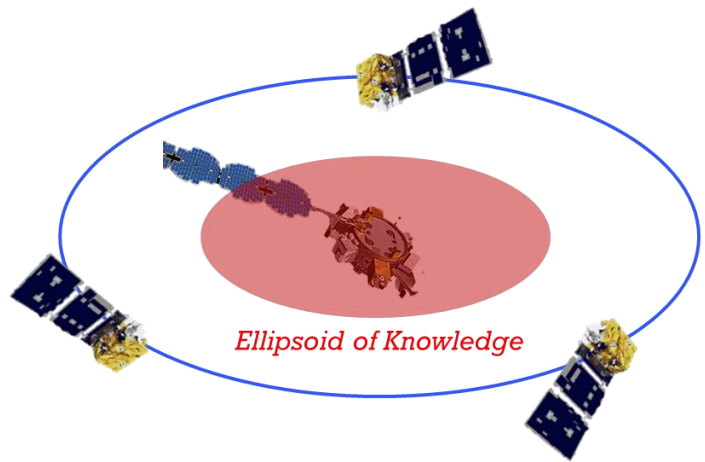
A dark space background featuring a view of Earth from space in the bottom left corner, showing the Americas. In the top right corner, the Moon is visible in a crescent phase. A thin white horizontal line is located at the bottom of the image.

Other Side Projects



Decision-Making Under Ignorance

▪ Credit: Joseph Derienzo (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

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



Optimize over Expectation to Ensure Consistency

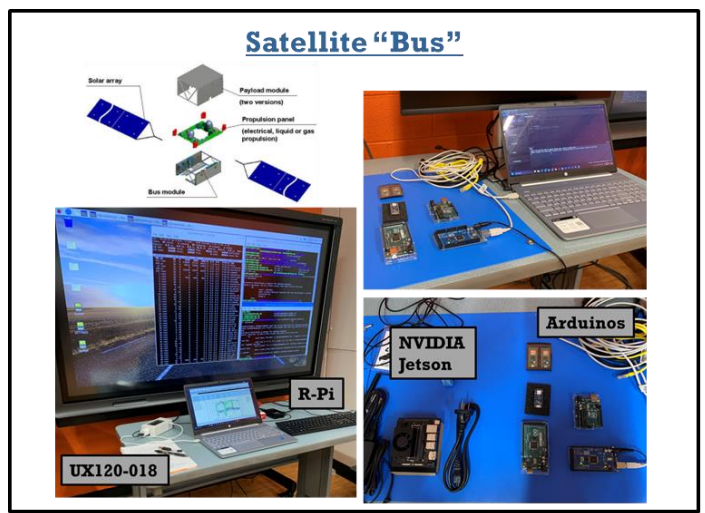
$$\min_{\mathcal{V}} \mathbb{E}(J_j(\mathcal{V})) + \lambda^T \mathbb{E}(G_j(\mathcal{V})) + \kappa [\mathbb{V}(J_j(\mathcal{V})) + \lambda^T \mathbb{V}(G_j(\mathcal{V}))]$$

← Scale of Real-Time →

	Sufficient Fast			Memory Efficient			Trending Towards Development
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	



Digital Twinning of HW/SW using Category Theory ▪ Collab: Dr. James Fairbank



Problem: USSF satellites will need to be able to verify, validate, and mission plan using as-close-as-possible realistic systems in as close to their operational environment on real-time time frames

Solution: Leverage category theory (CT), we will describe a) a physics engine for satellites and its hardware and b) the bus digital twin that is a software simulation of the software.

Focus Areas

- CT theory for appropriate modeling of physics and software
- Memory safe language coding for rapid transition of verifiable code given WH directives
- Integration with data and systems at AFRL

Challenges

- Security of data and IP of realistic systems
- How close to “real-time” can the simulation get



Looking at NSF Digital Twin Call

		← Scale of Real-Time →							
		Sufficient Fast			Memory Efficient				
Current State of Thinking	1	2	3	4	5	6	7	Trending Towards Development	
		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		

A dark space background featuring a view of Earth's horizon in the bottom left corner, showing the Americas. The Moon is visible in the top right corner. A thin white horizontal line is located at the bottom of the image.

Backup



Implementation of Autonomy

$$\min_u \sum_{k=0}^{N-1} (\hat{x}_{k|t}^T Q \hat{x}_{k|t} + u_{k|t}^T R u_{k|t} + \alpha \text{tr}(L_{k|t})) + \hat{x}_{N|t}^T Q_f \hat{x}_{N|t} + \alpha_f \text{tr}(L_{N|t}) \quad (23)$$

subject to

$$\hat{x}_{k|t} = F_1(x_{k|t}, u_{k|t}, y_{k|t}), \quad (24)$$

$$L_{k|t} = F_2(x_{k|t}, u_{k|t}, y_{k|t}), \quad (25)$$

$$x_{k+1|t} = A\hat{x}_{k|t} + Bu_{k|t}, \quad (26)$$

$$u_{min} \leq u_{k|t} \leq u_{max}, \quad (27)$$

When implementing MPC on a vehicle, an embedded system will receive a function command
`[output]=MPC_execute(input)`

$$\min_{y,v,w} \sum_{i=1}^N D_G(y_i, v_i) + D_E(W_i) + E_G(y_N) + E_E(w_N)$$

Sub. To :

$$y_{i+1} = F_1(y_i, v_i, W_i), \quad i = 0, \dots, N-1$$

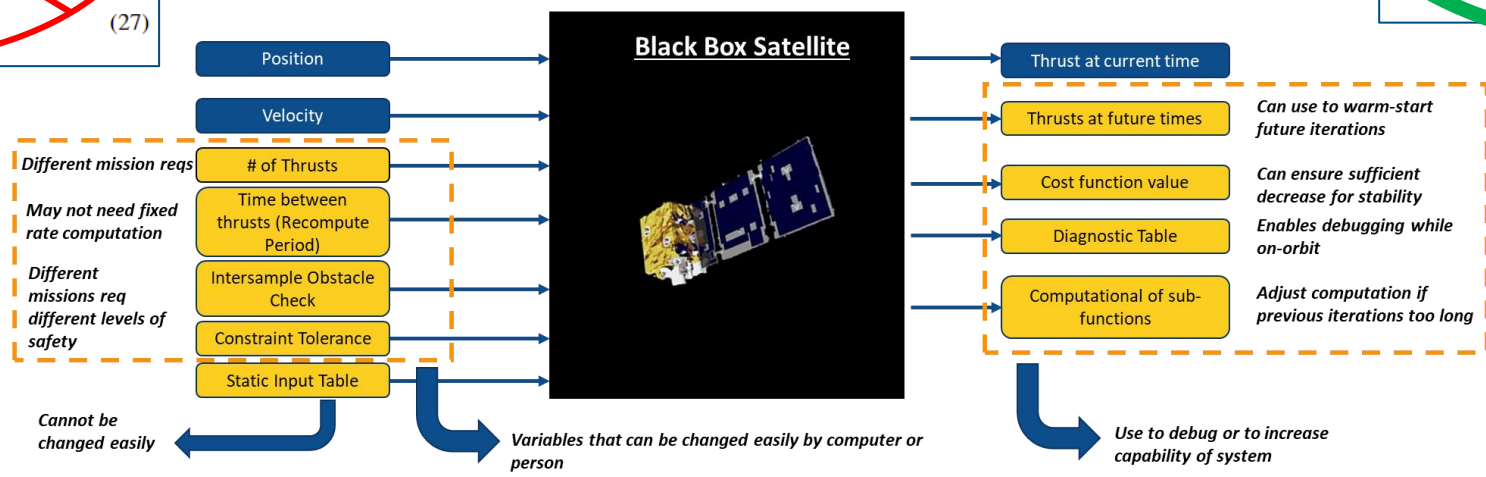
$$W_{i+1} = F_2(y_i, v_i, W_i), \quad i = 0, \dots, N-1$$

$$u_{min} \leq \|v_i\|_{\infty} \leq u_{max}, \quad i = 0, \dots, N-1$$

$$y_0 = \hat{X}(t)$$

$$W_0 = \hat{P}(t)$$

Question: What should that command contain?



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

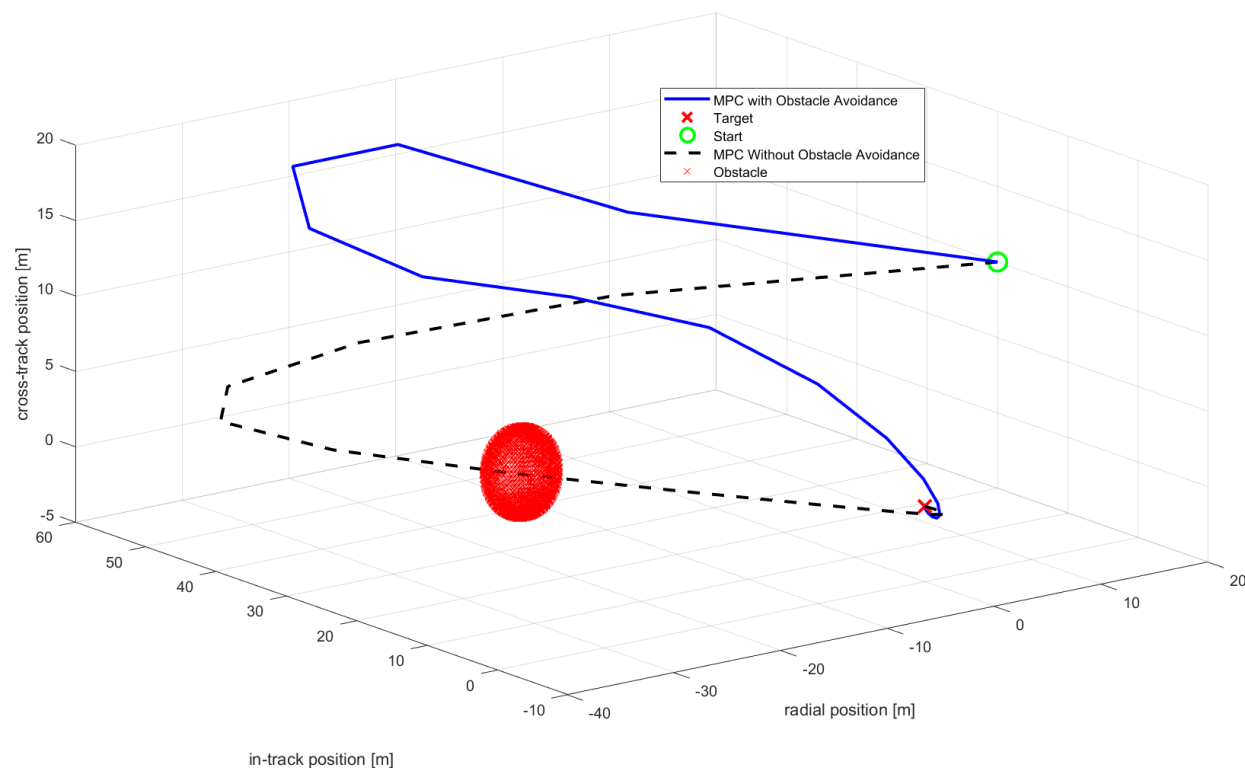
Understanding of what is easily accessible enables full system exploitation in unique ways by standard algorithms

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



Scenario

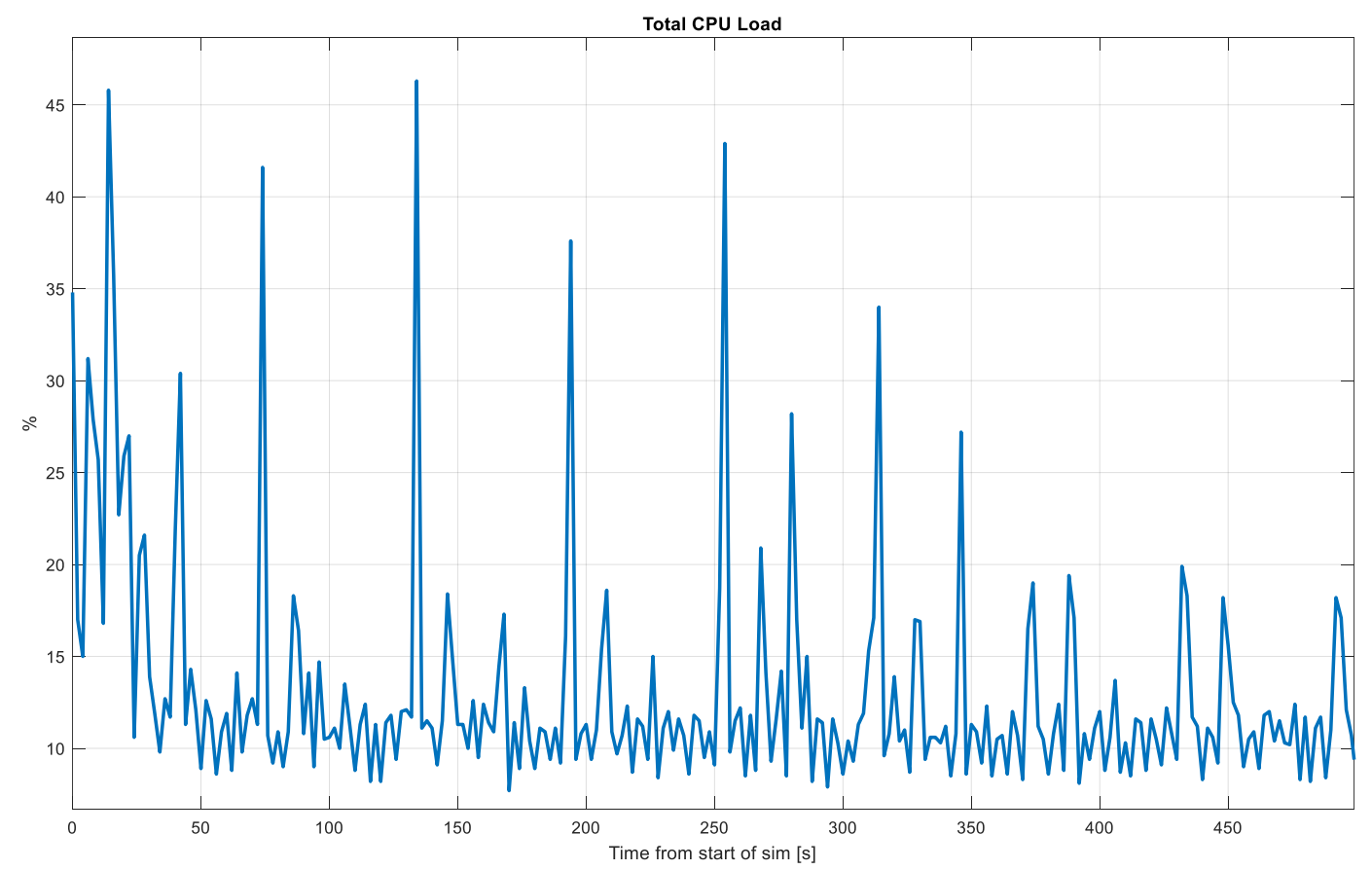
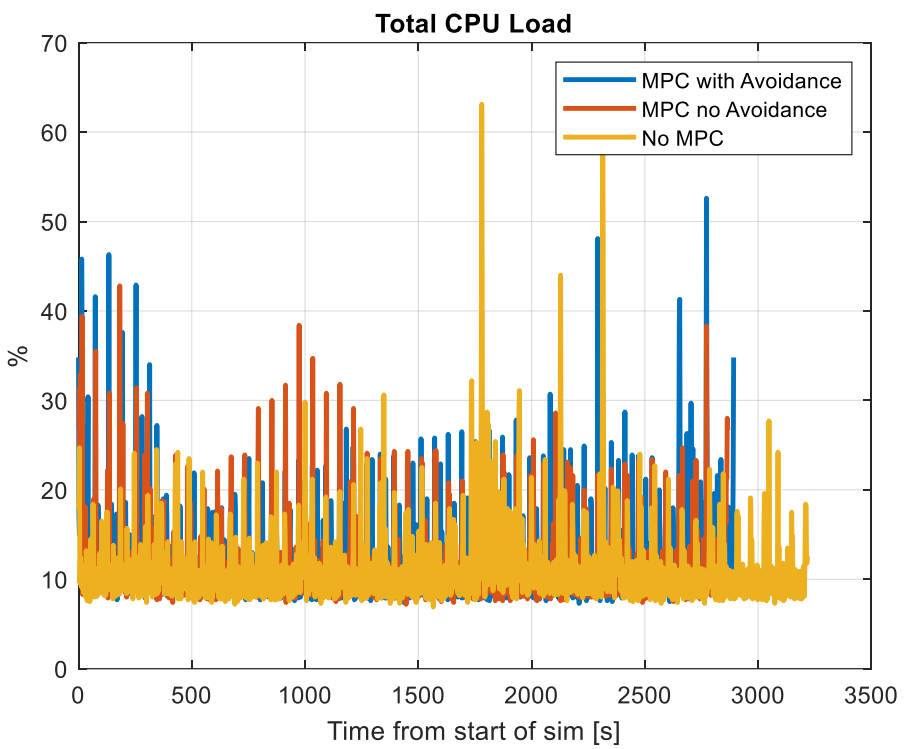
- Satellite is docking with another satellite
- Two algorithms are implemented
 - One with obstacle avoidance
 - One without obstacle avoidance
 - Both have control constraints
- Path is solved using QCLC formulation
 - Obstacle is dealt using convex hyperplane technique
 - Solver is custom made QP
- Useful parameters
 - Satellite ~ 30 m away, staged 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
It's a positive system as well