



PhD Defense

Relative-Motion Trajectory Generation and Maintenance for Multi-Spacecraft Swarms

Rahul Rughani



Overview

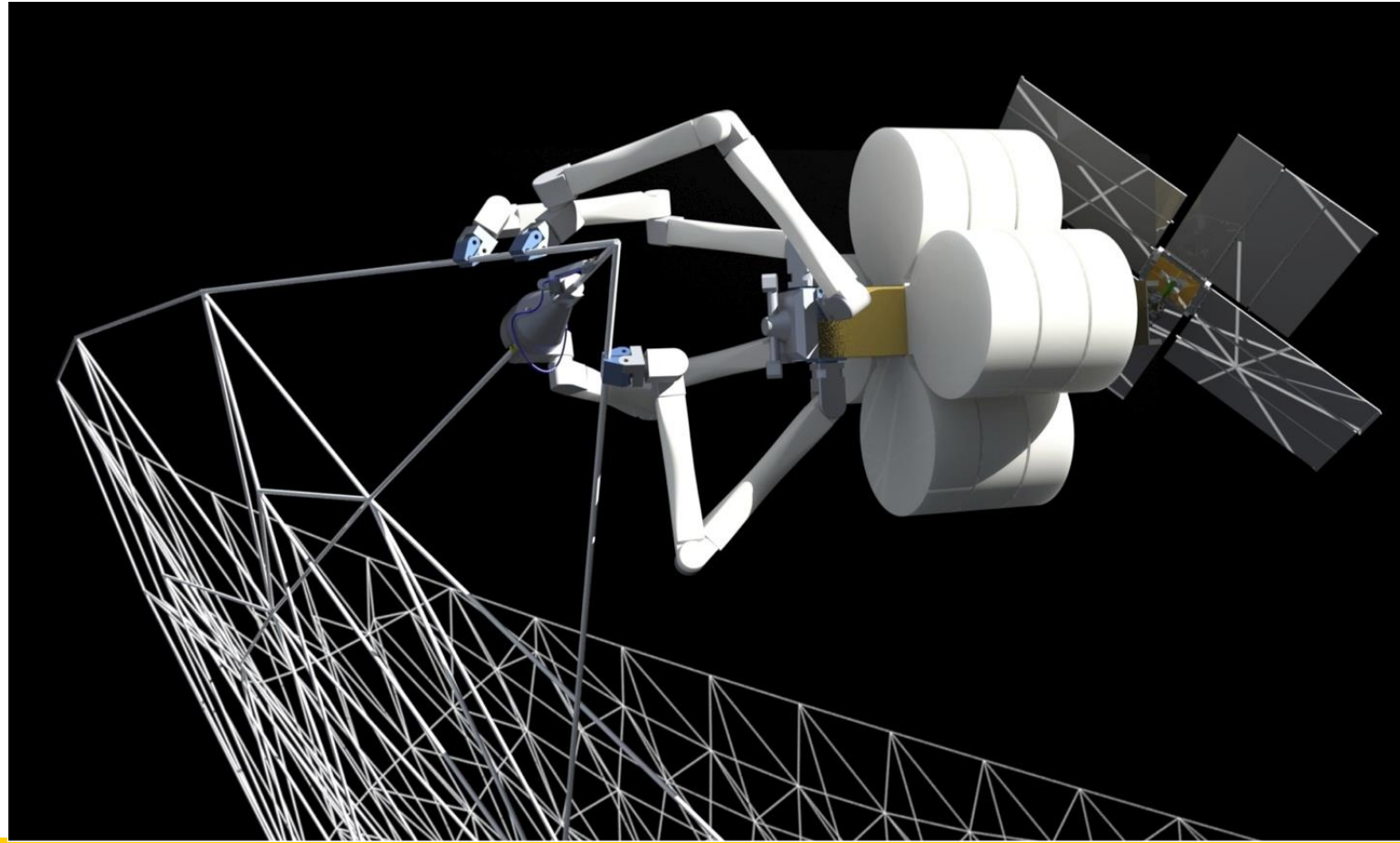
Research area: **Relative-Motion Trajectory Generation and Maintenance for Multi-Spacecraft Swarms**

1. Introduction
2. Ground Based Analogs
3. Rendezvous & Proximity Operations
4. Genetic Algorithms
5. Sensor Fusion Kalman Filtering
6. Assumptions & Requirements
7. Example Scenarios & Results
8. Behavioral Stresses of the System
9. Application to GEO Slot Sharing
10. Future Research
11. Summary & Conclusions

Presentation: 1h20min

Questions: 20 minutes

Committee Discussion: 20 minutes



Conference Publications



1. Barnhart, D., Rughani, R., Allam, Barnhart, D., Rughani, R., Allam, J., Weeden, B., Slane, F., and Christensen, I., “Using Historical Practices to Develop Safety Standards for Cooperative On-Orbit Rendezvous and Proximity Operations”, 69th International Astronautical Congress (IAC) Bremen Germany 1-5 October 2018, IAC-18,D1,5,8,x45161
2. Barnhart, D., Rughani, R., Allam, J., “Design-Based Safe Operable Metrics for Earth Regime RPO”, 10th Annual IAASS Space Safety Conference, May 15-17, 2019 (El Segundo, CA)
3. Barnhart, D.A., Rughani, R., “On-Orbit Servicing Ontology applied to Recommended Standards for Satellites in Earth Orbit”, 70th Annual International Astronautical Congress, 21-25 October 2019 (Washington, D.C.)
4. Rughani, R., Villafana, L., & Barnhart, D. A., “Swarm RPO and Docking Simulation on a 3DOF Air Bearing Platform”, 70th International Astronautical Congress (IAC). Washington D.C., United States, 21-25 October 2019.
5. Rughani R., Barnhart, D.A. “Using Genetic Algorithms for Safe Swarm Trajectory Optimization.” 30th AIAA/AAS Space Flight Mechanics Meeting. Orlando Fl, 6-10 January 2020
6. Rughani, R., Barnhart, D.A., “Safe Construction in Space: Using Swarms of Small Satellites for In-Space Manufacturing”, 34th Annual Small Satellite Conference. Logan UT, August 1-6, 2020 (Virtual)

Peer Reviewed Journal Publications



Published

1. Barnhart, D.A., and Rughani, R., “On-orbit servicing ontology applied to recommended standards for satellites in earth orbit”, Journal of Space Safety Engineering, <https://doi.org/10.1016/j.jsse.2020.02.002>
2. Barnhart, D.A., Rughani, R., Allam, J.J., Clarke, K.W., “Initial Safety Posture Investigations for Earth Regime Rendezvous and Proximity Operations”, Journal of Space Safety Engineering, <https://doi.org/10.1016/j.jsse.2020.06.010>

Submitted – Pending Review

1. Rughani, R., Barnhart, D.A., “Using Genetic Algorithms for Safe Spacecraft Swarm Trajectory Generation”, Acta Astronautica (in review).
2. Rughani, R., Presser, T., Barnhart, D.A., “The Use of Genetic Algorithms for Novel Geostationary Satellite Co-location Trajectories”, Journal of Guidance, Control, and Dynamics (in review)

Advances Made to State of the Art



- Created a novel method for swarm trajectory generation using Genetic Algorithms
- Created an algorithmically controlled method for optimized aggregation, in close proximity to a growing structure
- Implemented a Sensor Fusion Kalman Filter for swarm trajectory maintenance
- Created a novel method for slot-sharing stationkeeping of GEO spacecraft that reduces Delta-V vs existing methods (provisional patent)



Introduction

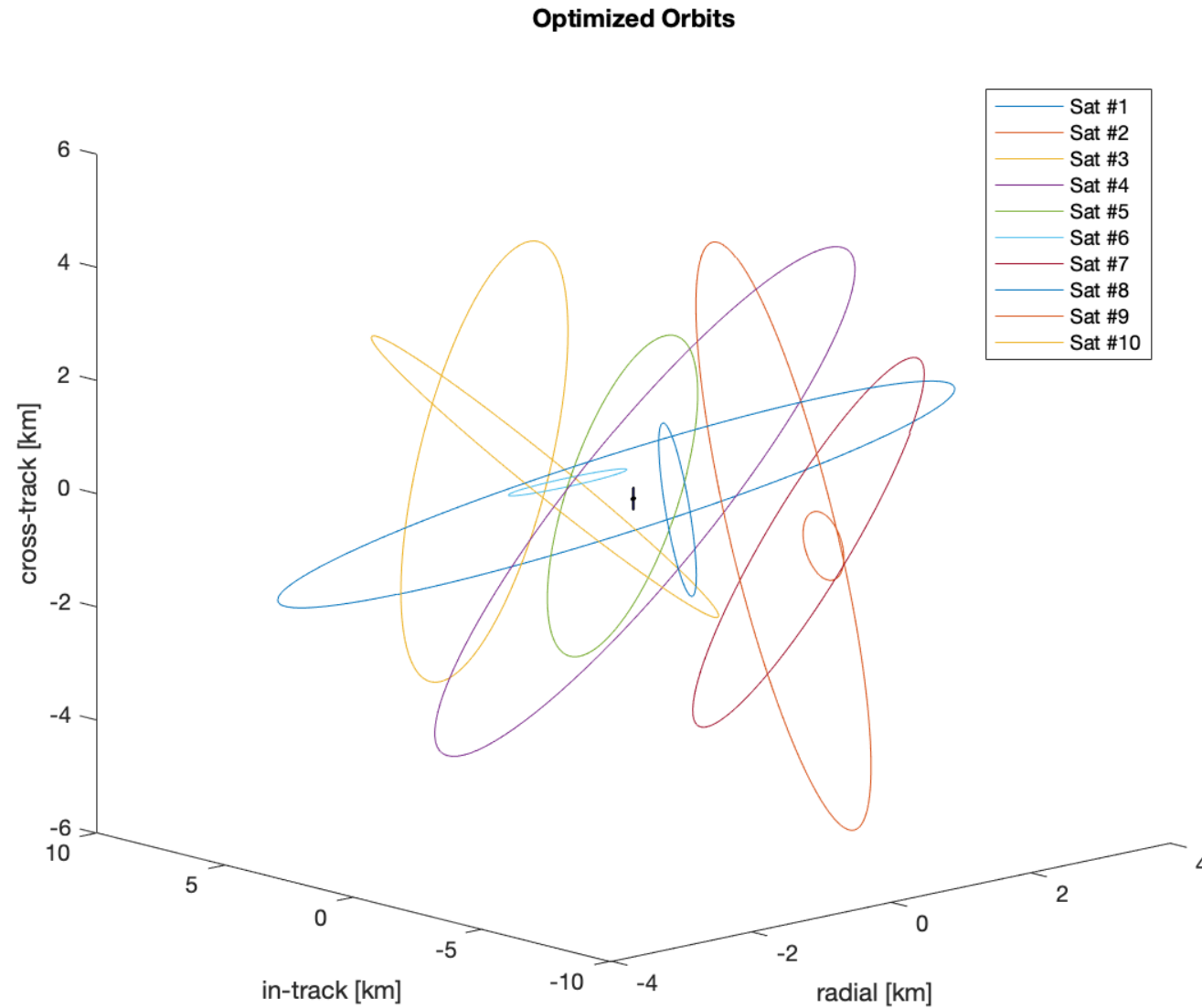
Research Goal



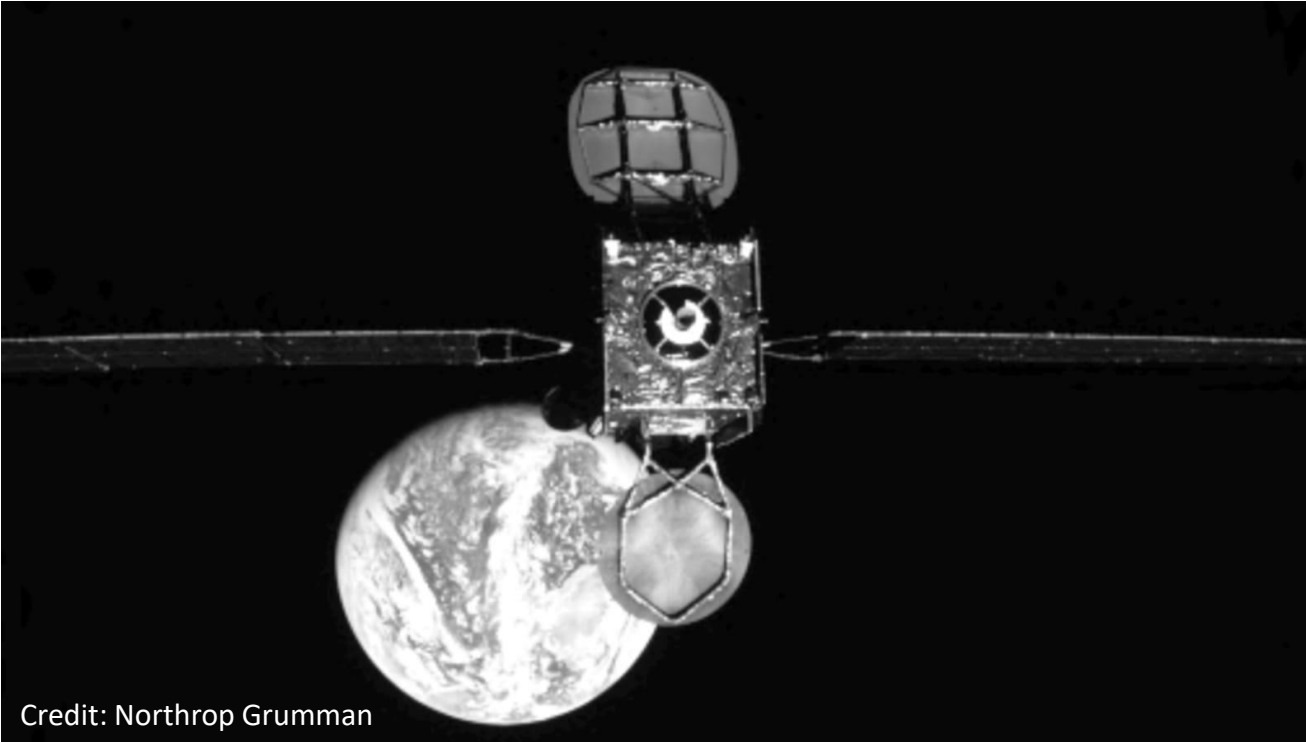
Advance the state of the art for spacecraft swarm operations to enable large-scale in-space manufacturing in the not-too-distant future



What is a Spacecraft Swarm?

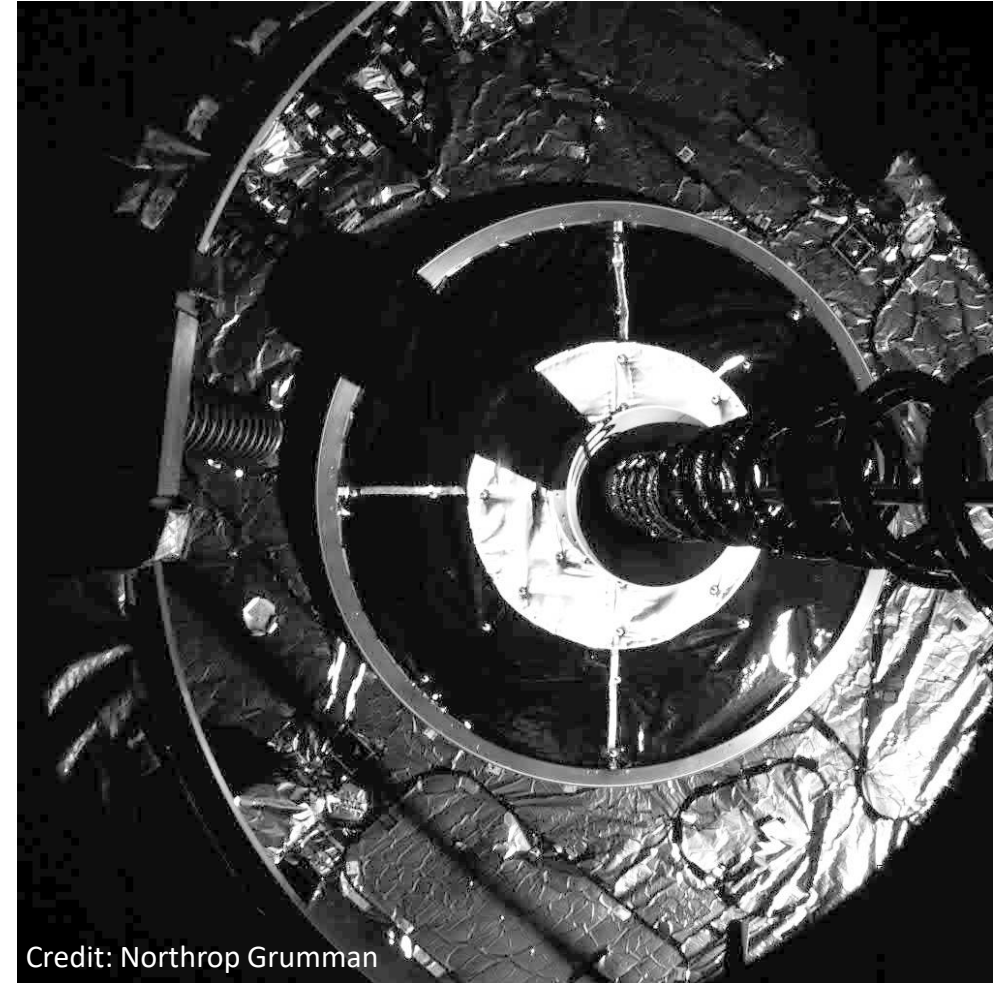


Current State of the Art



Credit: Northrop Grumman

- MEV-1 successfully docked to a retired GEO spacecraft to provide mission extension services

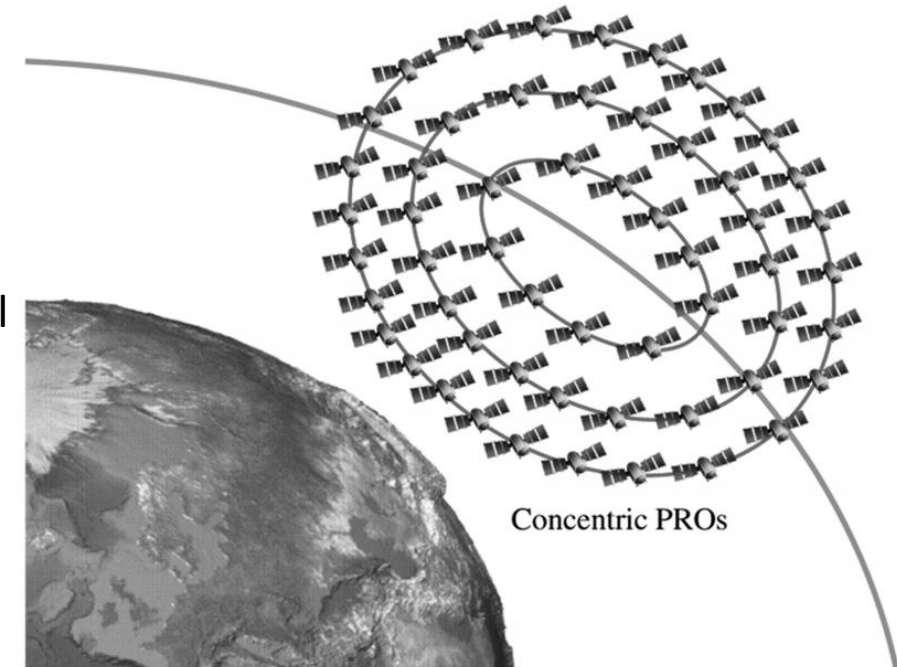


Credit: Northrop Grumman



Swarm RPO Operations – State of the Art

- **Specific energy-based optimization [1]**
 - Primarily static formation flying configs
- **Eccentricity/Inclination vector alignment (E/I) [2]**
 - Good for smaller swarm sizes. E/I method doesn't scale well
- **Relative Pose Estimation [3]**
 - Solves the problem of distributed attitude determination and control
- **Sliding-Mode Control Algorithms [4]**
 - Good for large swarms, but primarily reactionary rather than predictive, thus uses large amounts of dV
- **Convex Programming [5,6]**
 - Best for irregular gravity fields around asteroids



- [1] Morgan et. al. Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations. Journal of Guidance, Control, and Dynamics, 35(5):1492–1506, 2012.
- [2] Simone D’Amico. Autonomous formation flying in low earth orbit. 2010.
- [3] William Bezouska and David Barnhart. Decentralized cooperative localization with relative pose estimation for a spacecraft swarm. In 2019 IEEE Aerospace Conference, pages 1–13. IEEE, 2019
- [4] Chakravarthini M Saaj, Vaios Lappas, and Veysel Gazi. Spacecraft swarm navigation and control using artificial potential field and sliding mode control. In 2006 IEEE International Conference on Industrial Technology, pages 2646–2651. IEEE, 2006
- [5] Bandyopadhyay et. al. Distributed fast motion planning for spacecraft swarms in cluttered environments using spherical expansions and sequence of convex optimization problems. 2017.
- [6] Bandyopadhyay et. al. Distributed spatiotemporal motion planning for spacecraft swarms in cluttered environments. In AIAA SPACE and Astronautics Forum and Exposition, page 5323, 2017.



Five Classes of Swarms [7]

Class 0: no coordination either in movement, sensing, or communication

Class 1: Each spacecraft coordinates its movement, but there is no explicit communication coordination or sensing coordination.

Class 2: movement and communication coordination. Swarm has collective sensing capabilities, but is not optimized

Class 3: Each spacecraft coordinates sensing with communication and position, but is still not collectively optimized

Class 4: positioning, movement, communication, and sensing are coordinated to perform system level optimization. Computing is evenly distributed within the swarm

[7] Ravi Nallapu and Jekan Thangavelautham. Spacecraft swarm attitude control for small body surface observation. arXiv preprint arXiv:1902.02084, 2019



Five Classes of Swarms

- Current state-of-the-art swarm research is limited to Class 0, 1, & 2
- Higher swarm classes increase in operational complexity and interconnectivity, but offer greater returns in terms of safety of operations and propellant utilization
- The method described in this thesis uses GAs to create an overall set of trajectories that avoid collisions, however there is still a centralized computation authority for the initial trajectory generation
 - Somewhere between Class 3 and Class 4, thus a Class 3.5 Swarm

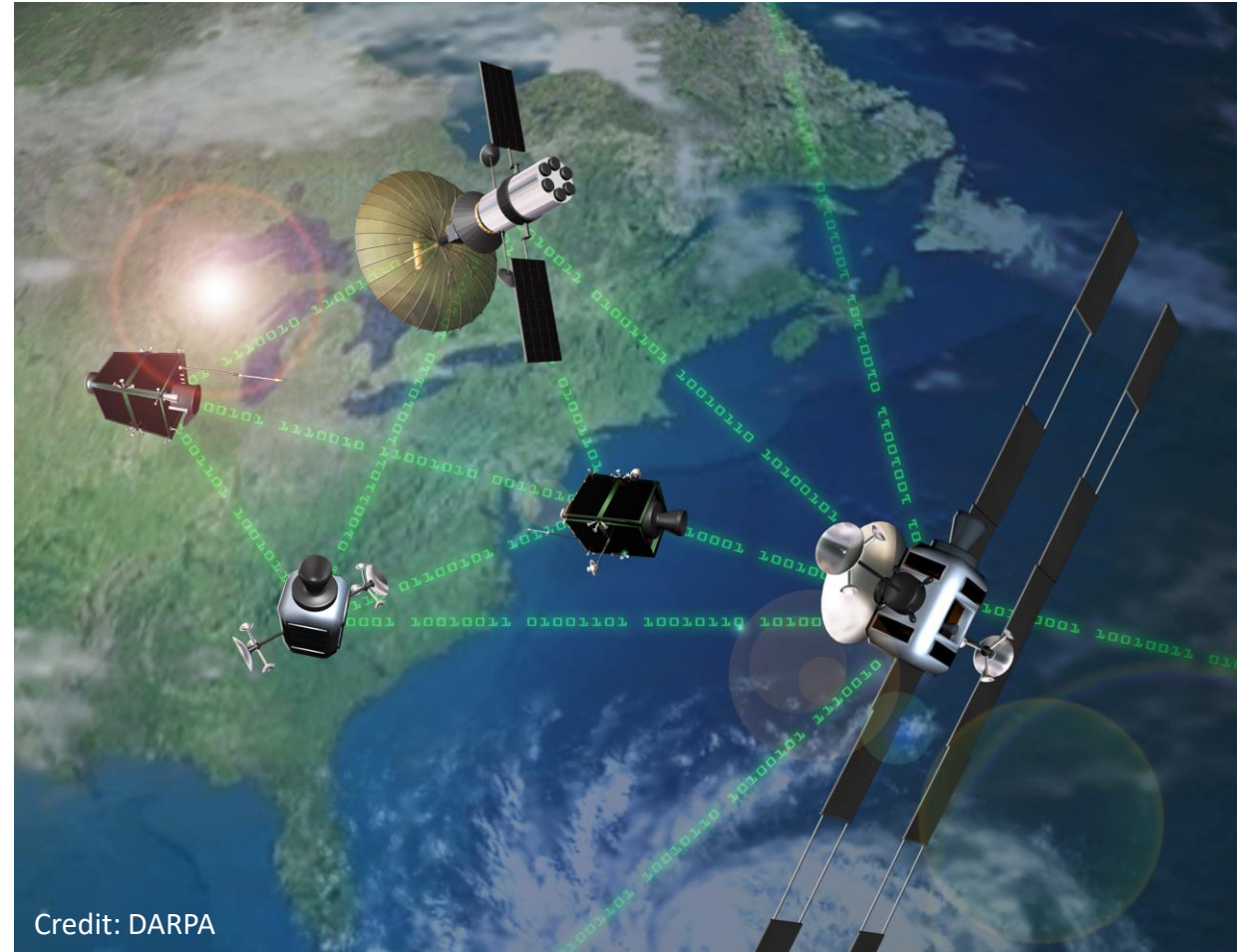




DARPA F6

DARPA's Future, Fast, Flexible, Fractionated Free-Flying Concept (F6) was one of the first attempts to formalize the concept of the spacecraft swarm

Although the project never reached flight, ground test showed that the system was a viable method to protect national security space infrastructure by distributing sensing and computing over an array of spacecraft in close proximity, preventing single-node failures from crippling the system.



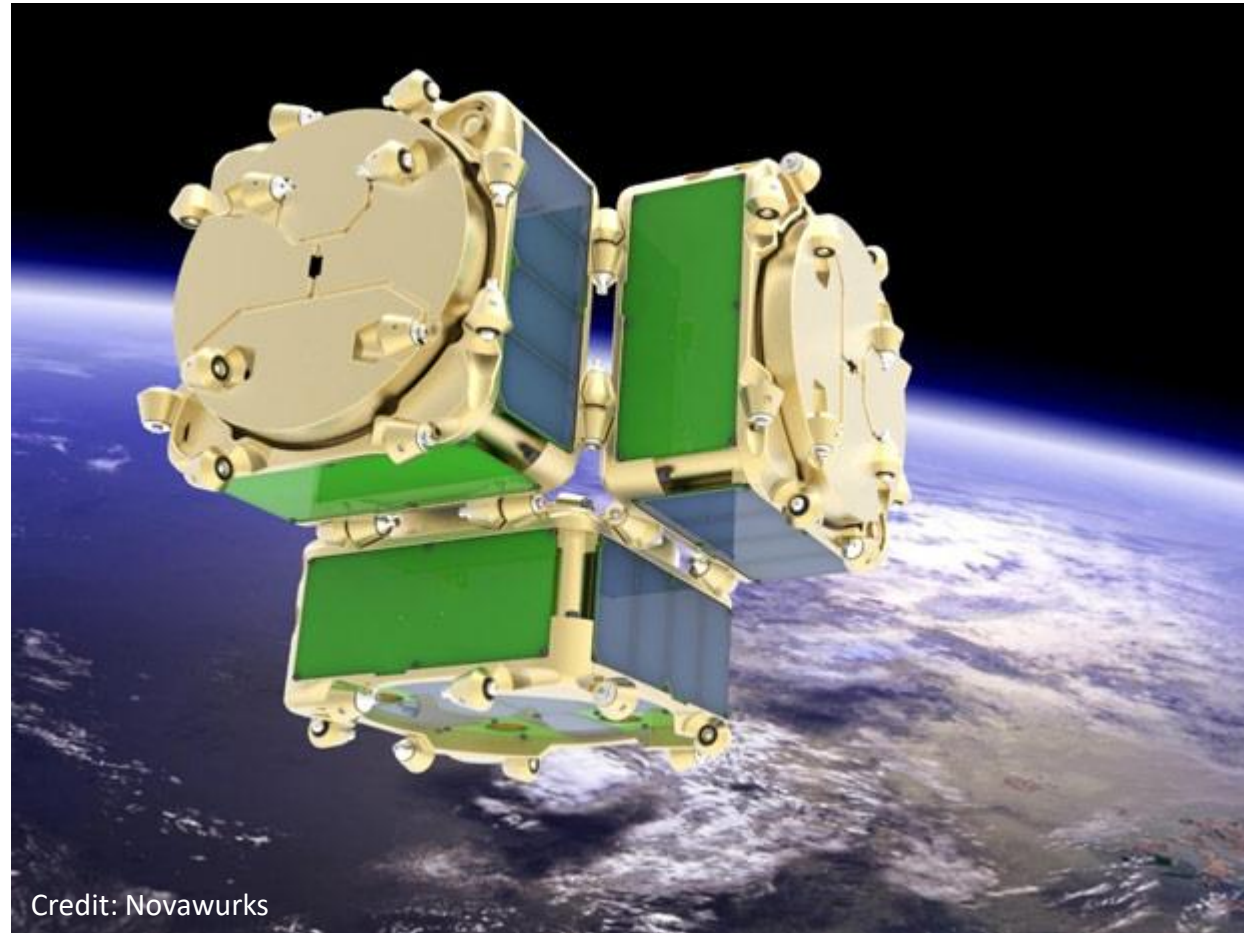
Credit: DARPA



DARPA Phoenix – Satlets

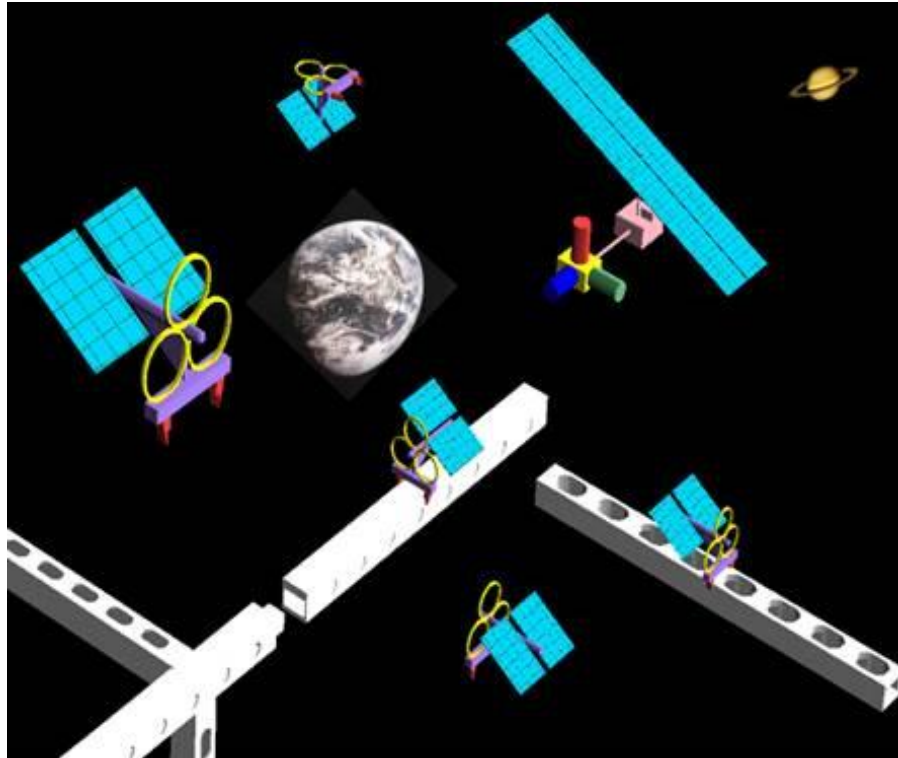
DARPA's Phoenix satellite servicing project included *Satlets*, small free-flying spacecraft capable of aggregated together to form larger objects

These were essentially building blocks for spacecraft that can be assembled and reconfigured to match the mission requirements and can change with the needs of the mission.

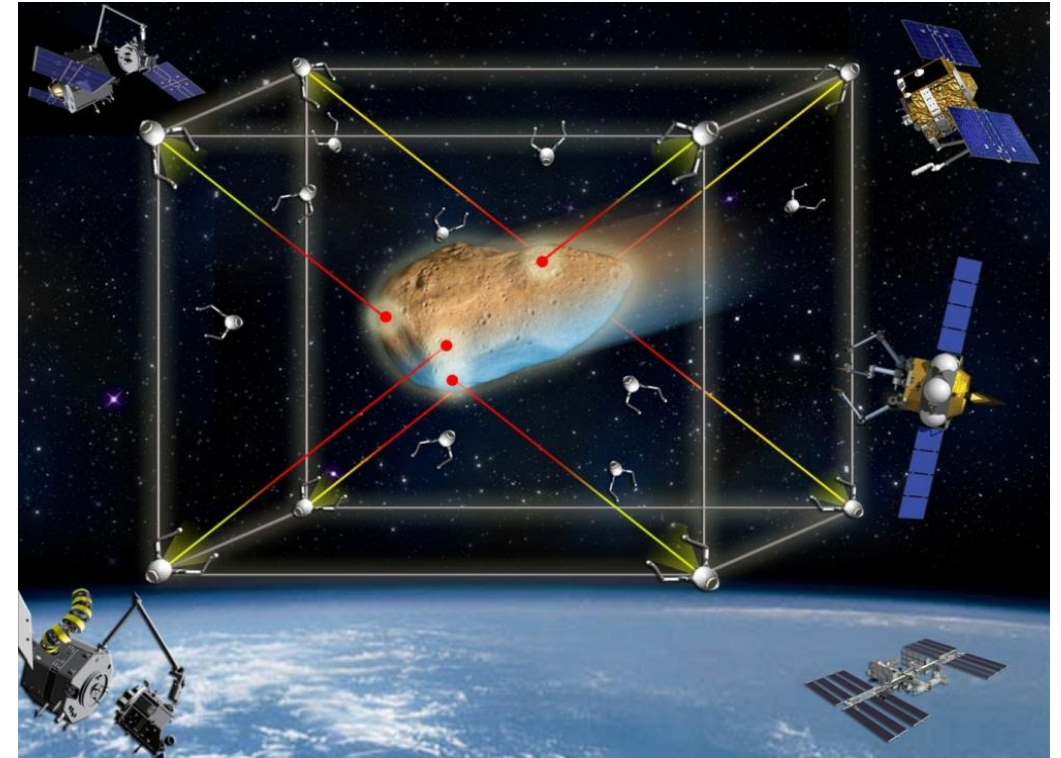




Motivation & Use Cases



On-orbit assembly

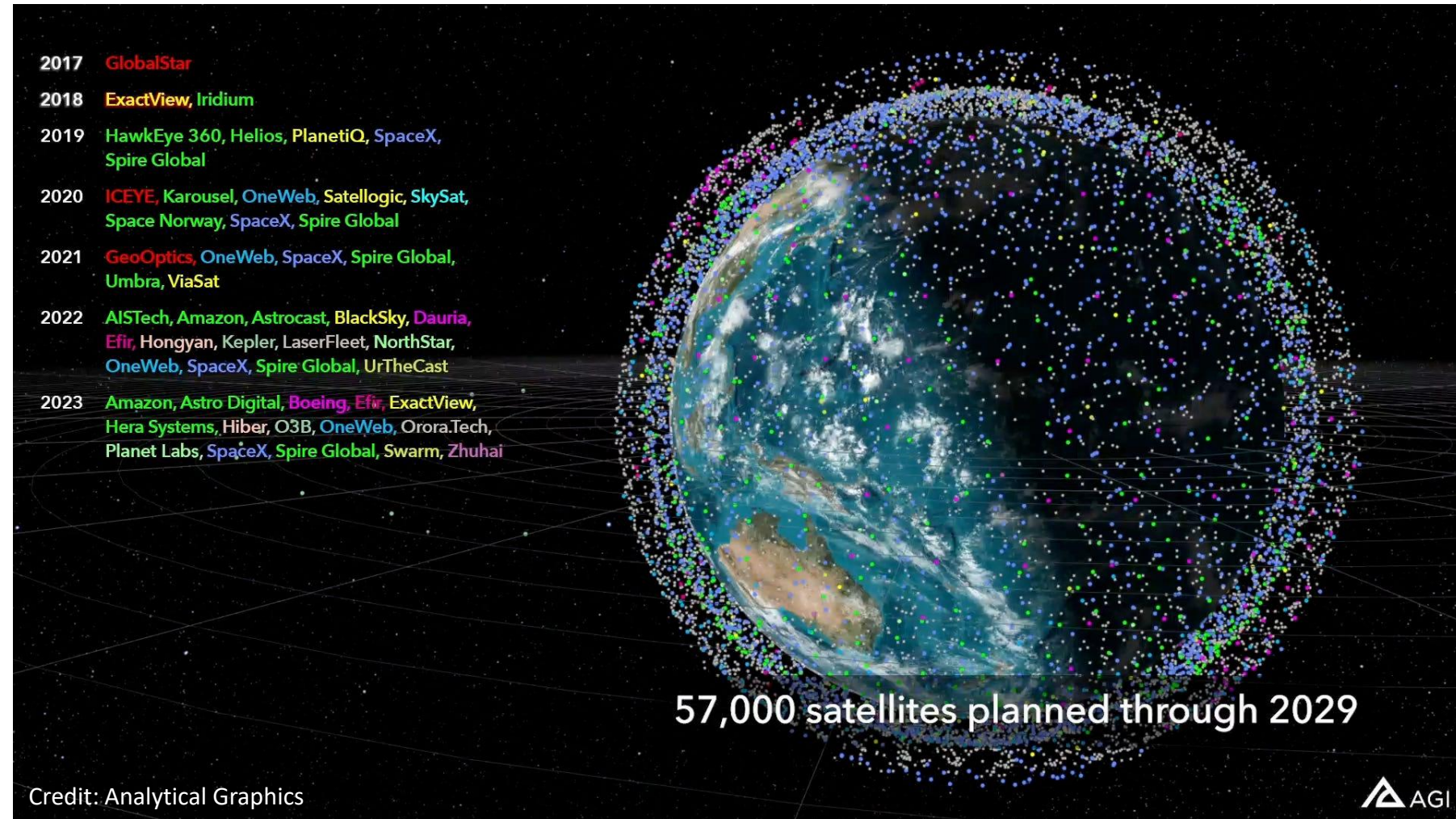


Cooperative Proximity Operations

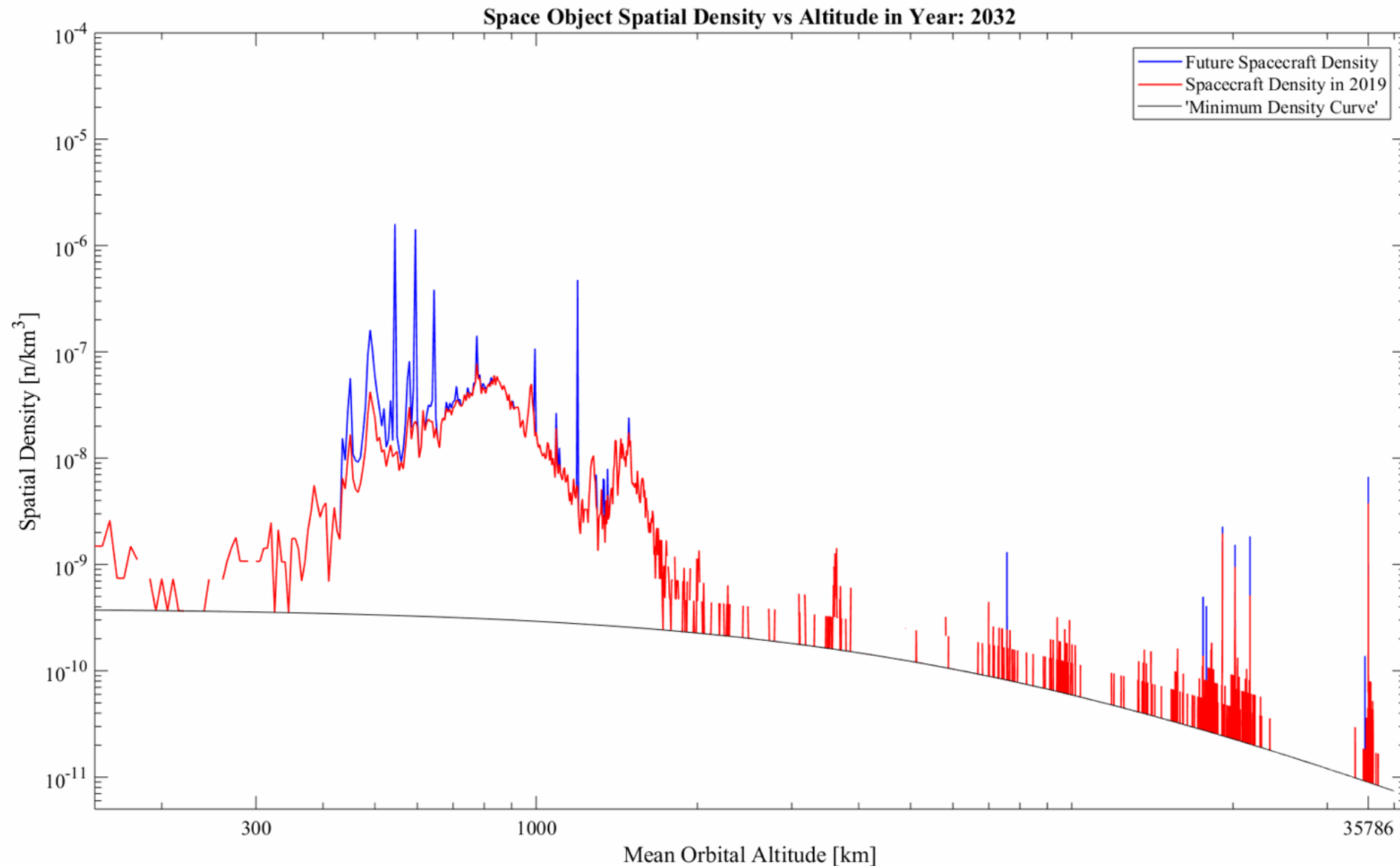
Increasing Density of Spacecraft in LEO



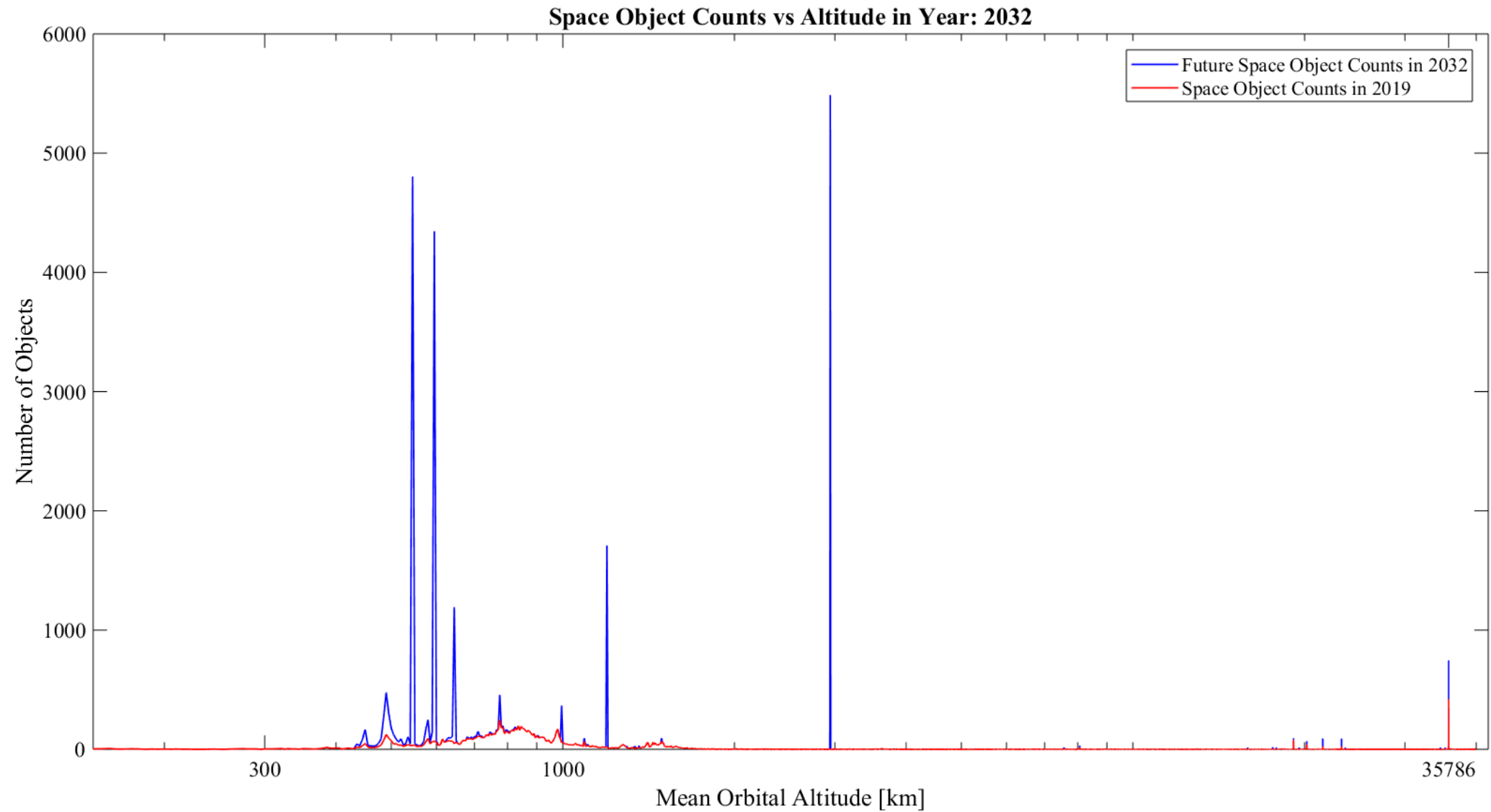
- Thousands of satellites are planned for LEO over the coming decade
- Crowding of LEO will require new and novel methods for collision detection and avoidance in close proximity



Increasing Density of Spacecraft in LEO



Credit: Kyle Clarke





Problem Statement & Solution Approach

- 1. Given a co-located swarm of N free-flying spacecraft capable of relative position, velocity, and orientation determination, generate a set of trajectories that enable these spacecraft to complete their individual tasks within their ΔV budgets, while mitigating collision risks over a minimum 24hr period**
 - Can be solved using Genetic Algorithms, an evolutionary optimization scheme
- 2. Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, maintain these trajectories in real-time, accounting for deviations due to injection errors, unaccounted for higher-order or non-gravitational perturbations, sensor errors, or system noise**
 - Can be solved using a Sensor Fusion Kalman Filter, combining shared sensor data from all spacecraft in the swarm
- 3. Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, generate a new set of trajectories for a modified swarm, with some spacecraft either added or removed, while minimizing the ΔV required to re-position the existing swarm spacecraft to accommodate the new spacecraft**
 - Can be solved using Genetic Algorithms, similar to problem 1

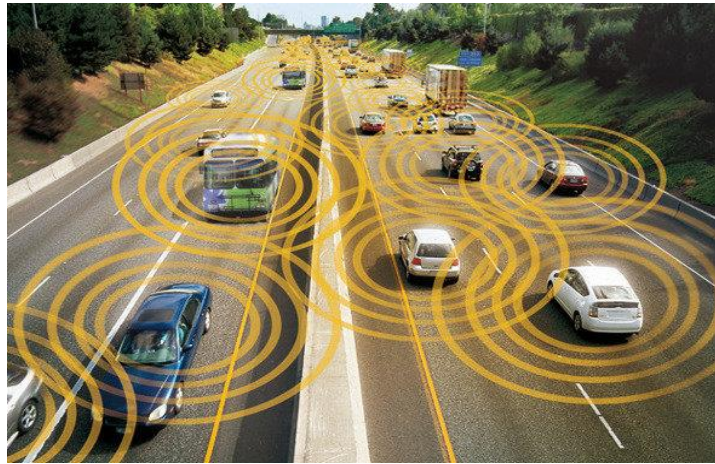
Research Path





Ground Based Analog

Examples of Swarm Operations



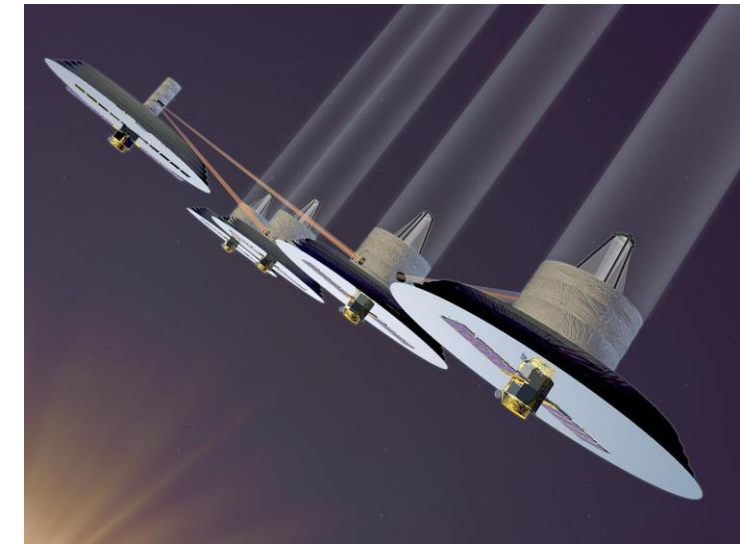
Vehicle-to-vehicle communications



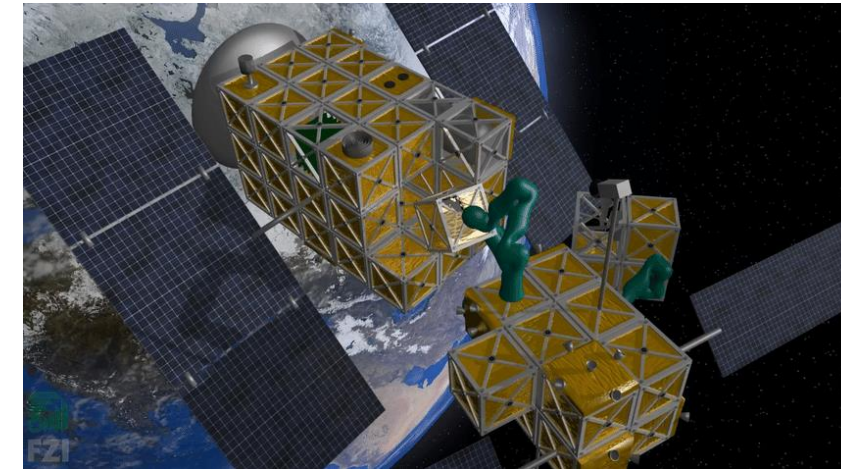
Space situational awareness



Drone cooperation



Formation flying



On-orbit manufacturing

Drone Swarms



UAV swarms provided an excellent analog for swarm operations in space, as demonstrated using ground-based equipment

The use of Kalman filtering and shared sensor nets can be directly ported to in-space swarm applications for collision avoidance and trajectory generation



Insect Analogs



- Bees were also considered as swarm analogs
- Bees display task coordination and division of labor, concepts ported into spacecraft swarm ops
 - Constant communication used to prevent collisions and transmit directions





Rendezvous and Proximity Operations



Local-Vertical Local-Horizontal (LVLH) Coordinate Frame

Definition:

X.ax = radial (green)

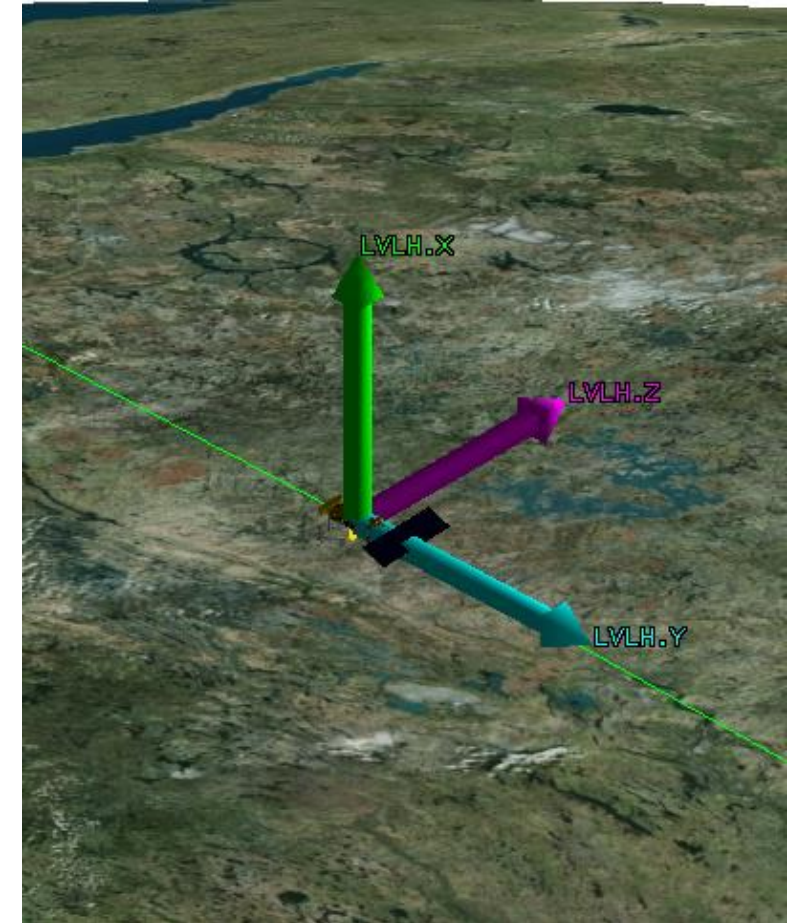
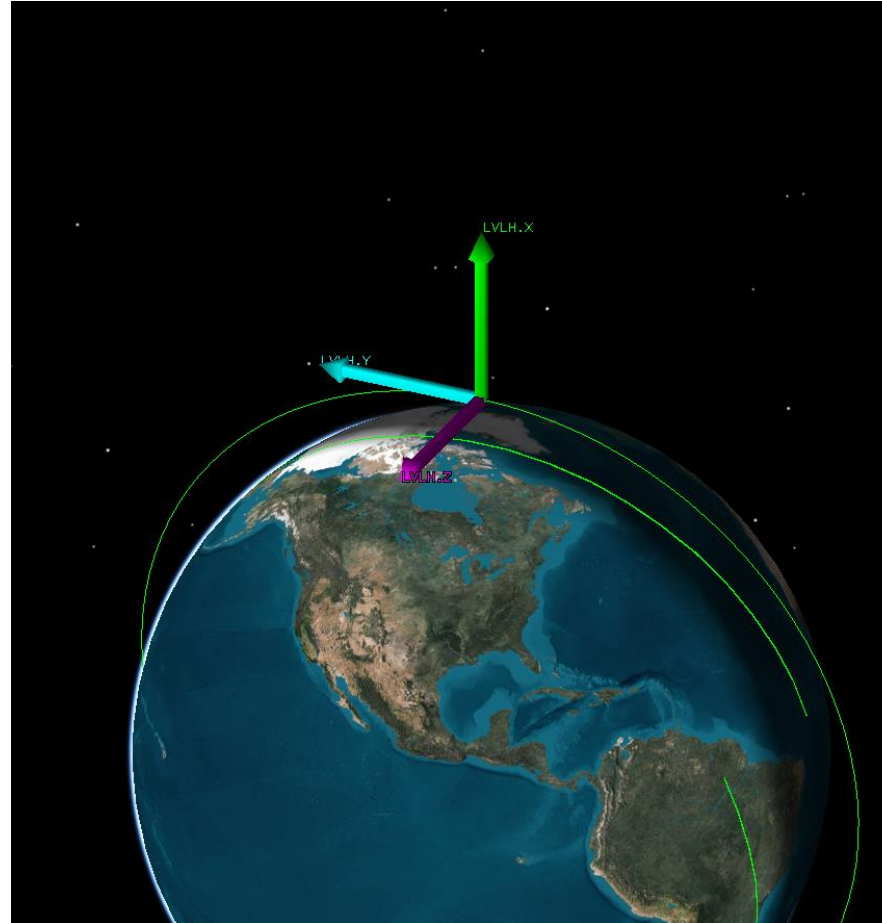
- outward radial vector from center of Earth to target

Y.ax = in-track (cyan)

- parallel to velocity vector for circular reference orbit. Otherwise forms triad with X&Z

Z.ax = cross-track (purple)

- normal to orbital plane (aligned with angular momentum vector)



Rendezvous and Proximity Operations (RPO)

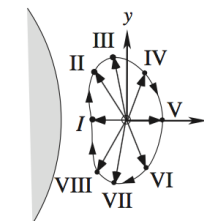
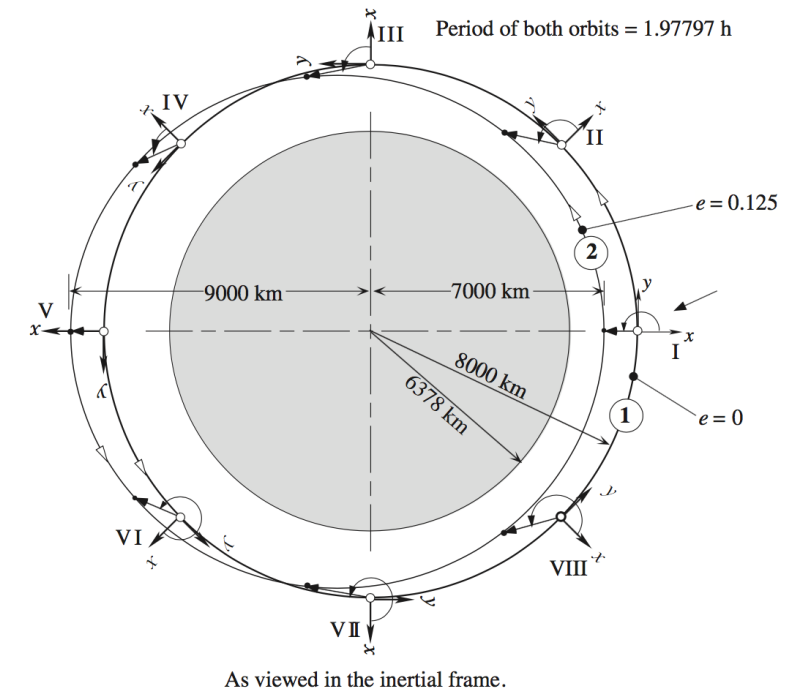


$$d\ddot{\mathbf{r}} = -\ddot{\mathbf{R}} - \mu \frac{\mathbf{R} + d\mathbf{r}}{\|\mathbf{R} + d\mathbf{r}\|^3} \quad (1)$$

$$d\ddot{x} - 3n^2 dx - 2nd\dot{y} = 0 \quad (2)$$

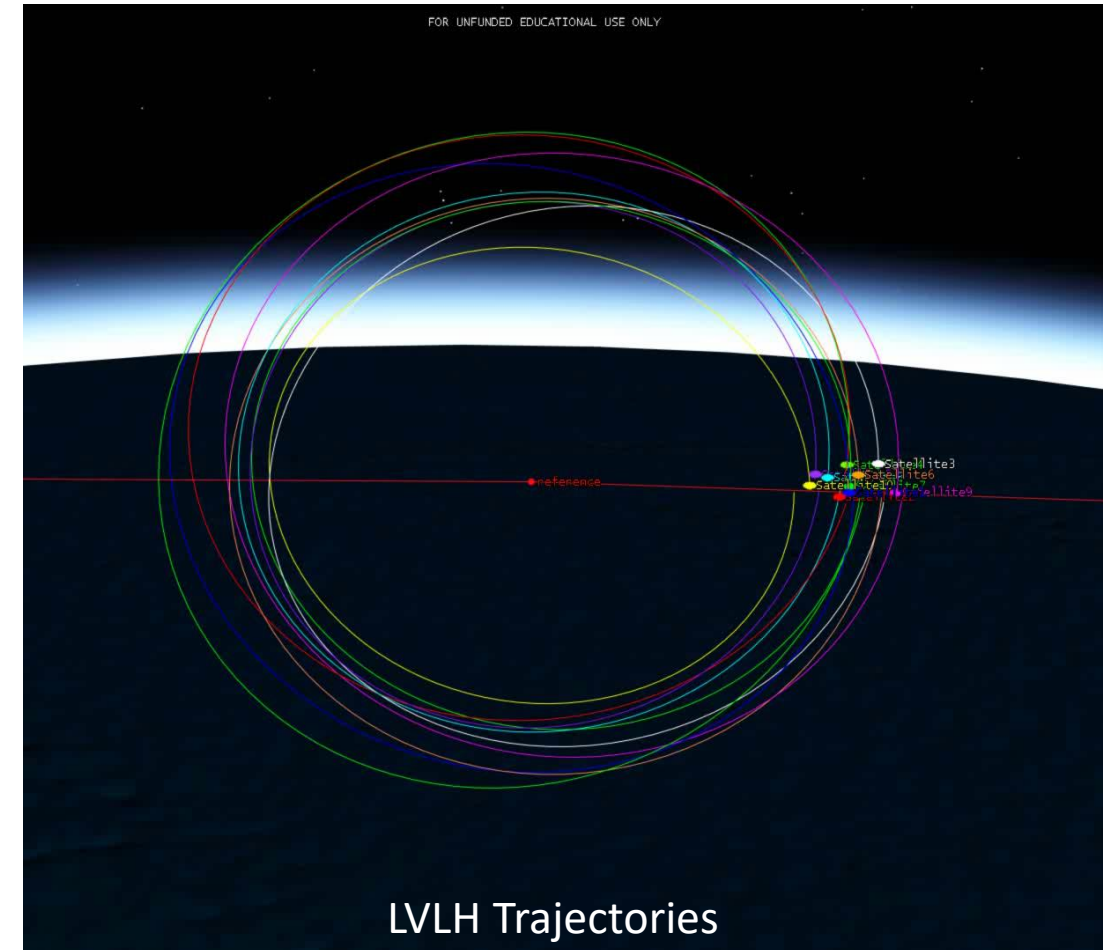
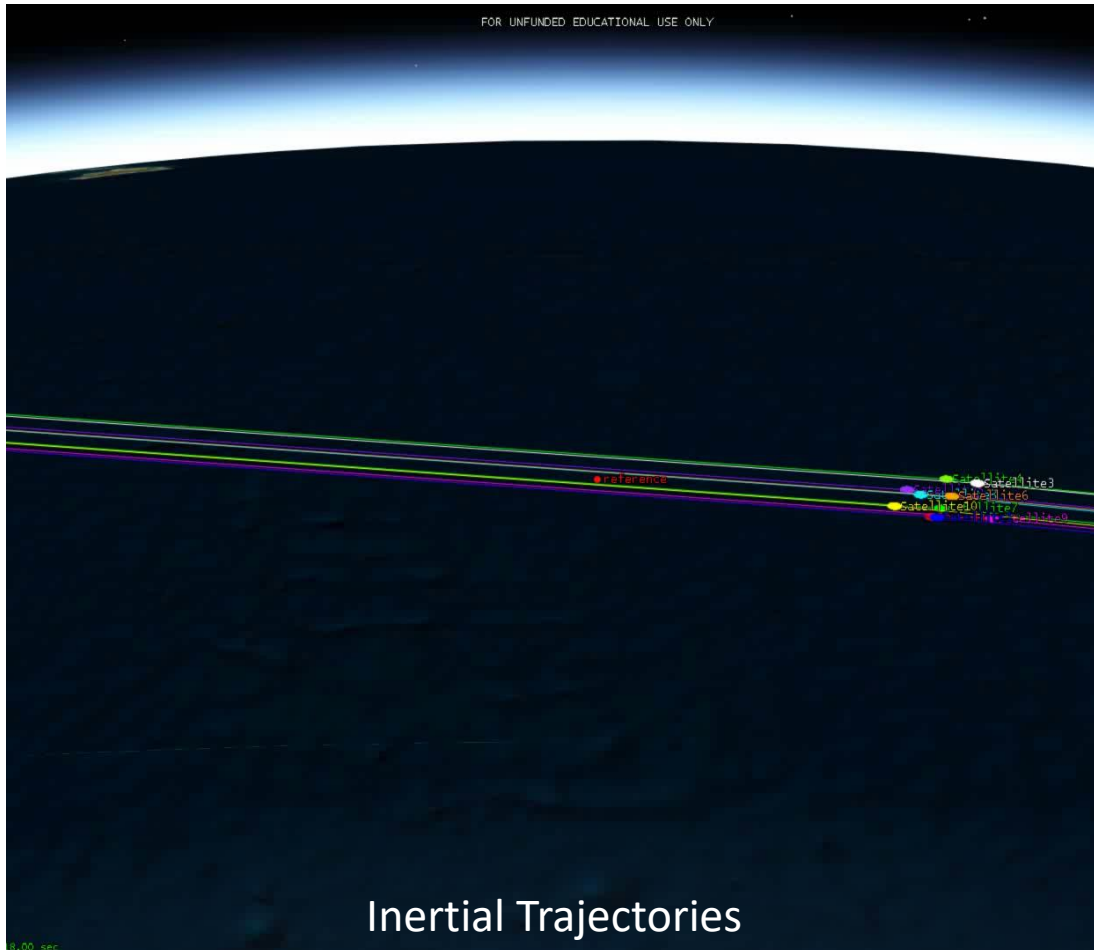
$$d\ddot{y} + 2nd\dot{x} = 0 \quad (3)$$

$$d\ddot{z} + n^2 dz = 0 \quad (4)$$





Swarm of Spacecraft in Relative Motion

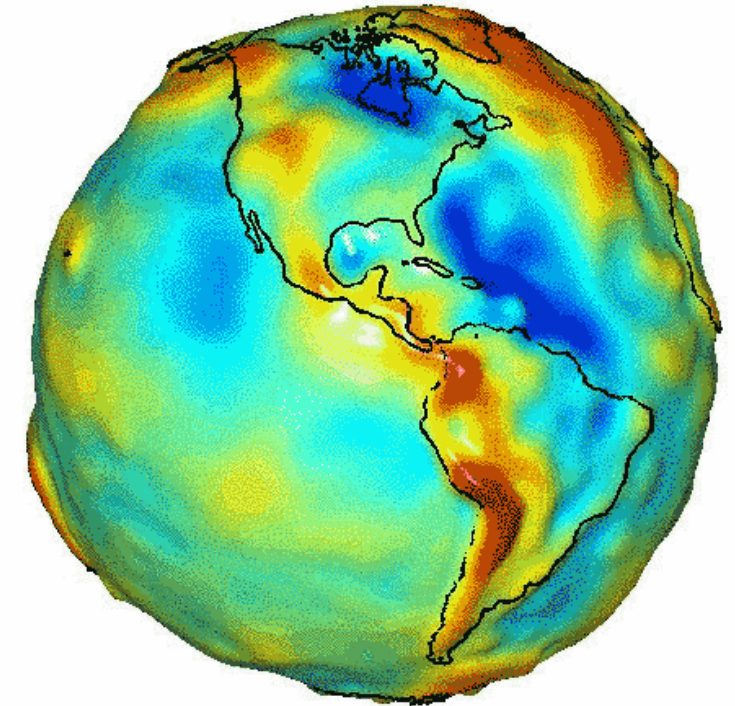


Gravitational Perturbations



Although the C-W equations are a good approximation of relative motion trajectories, they fail to account for perturbations of the orbit and thus are only a first order solution

GA solver was thus implemented using a two-stage process, with the perturbed gravitational potential. This enables more accurate predictions farther in the future



Irregularities of Earth's gravitational field (GRACE data)

$$U = \frac{\mu}{r} \left[1 - \sum_{l=2}^{\infty} J_l \left(\frac{R_{\oplus}}{r} \right)^l P_l[\sin(\phi_{gc_{sat}})] + \sum_{l=2}^{\infty} \sum_{m=1}^l \left(\frac{R_{\oplus}}{r} \right)^l P_{l,m}[\sin(\phi_{gc_{sat}})] \{C_{l,m} \cos(m\lambda_{sat}) + S_{l,m} \sin(m\lambda_{sat})\} \right]$$

Gravitational Perturbations



Sun-Moon: perturbations from the Sun's and Moon's gravitational fields are significant at GEO

$$\vec{a}_{sun-moon} = -GM_{\odot} \frac{\vec{r}_{sun-earth} + \vec{r}}{\|\vec{r}_{sun-earth} + \vec{r}\|^3} - GM_{\text{Moon}} \frac{\vec{r}_{moon-earth} + \vec{r}}{\|\vec{r}_{moon-earth} + \vec{r}\|^3} \quad (2.22)$$

SRP: Solar Radiation Pressure perturbations depend on the area-to-mass ratio of the spacecraft, and are significant in GEO, or for large spacecraft in LEO

$$\vec{a}_{SRP} = -(1 - \beta) \frac{F_s}{c} \frac{A_c}{m} \frac{\vec{r}_{sun-earth} + \vec{r}}{\|\vec{r}_{sun-earth} + \vec{r}\|^2} \quad (2.23)$$



Genetic Algorithms

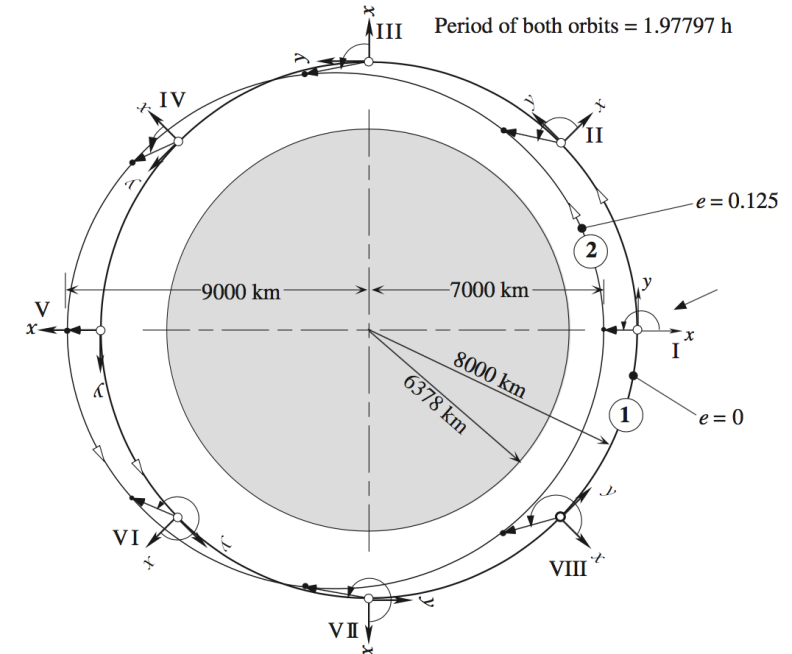
Relative Motion Trajectory Determination



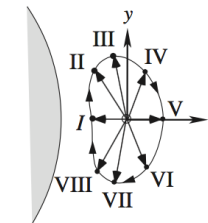
Problem: How to quickly and efficiently determine a set of trajectories for a swarm of spacecraft, in relative motion, that don't collide

- Also minimize delta-V and perform mission-specific tasks

Solution: Genetic Algorithms



As viewed in the inertial frame.

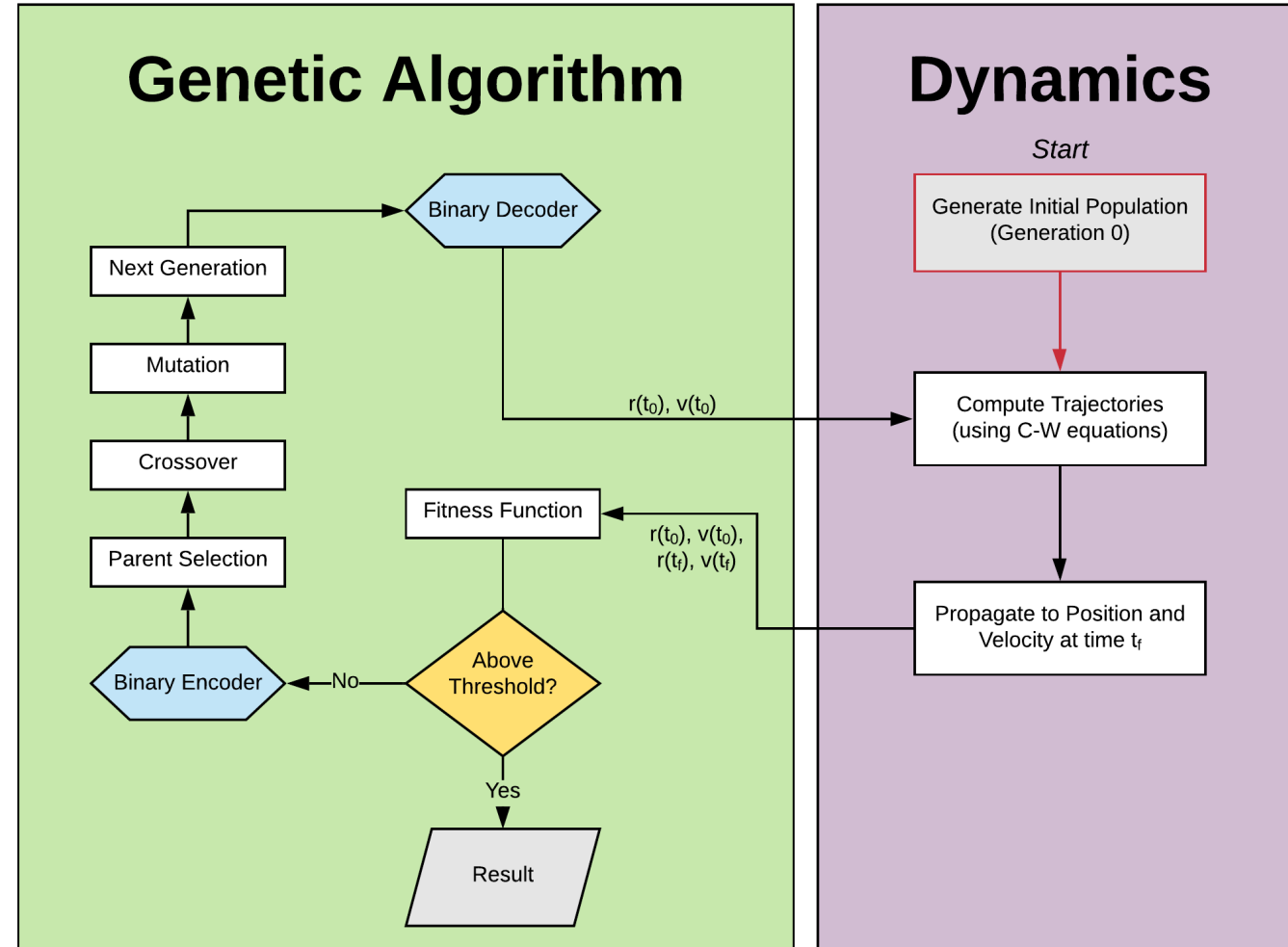


As viewed from the comoving frame in circular orbit 1.

Genetic Algorithms – How Do They Work?



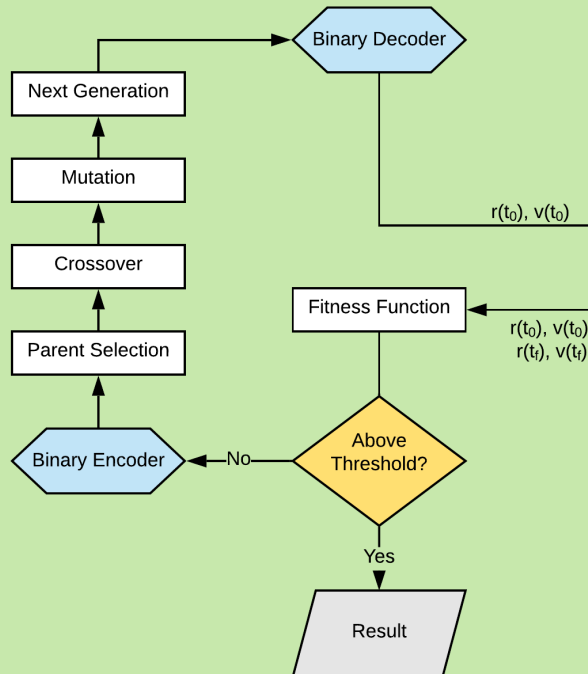
- GAs use principles of Darwinian evolution to progress a set of initial conditions towards a solution
- A fitness function is used to rank solutions to determine which are the *fittest* that should proceed to the next generation



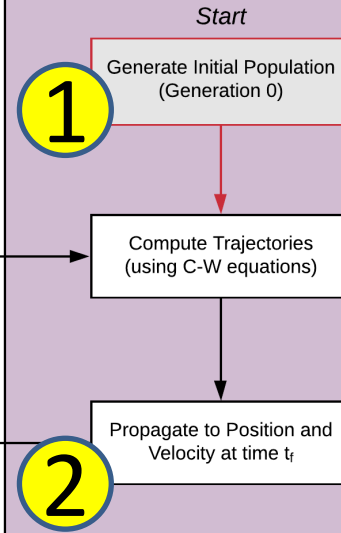
GA – Generate Population



Genetic Algorithm



Dynamics

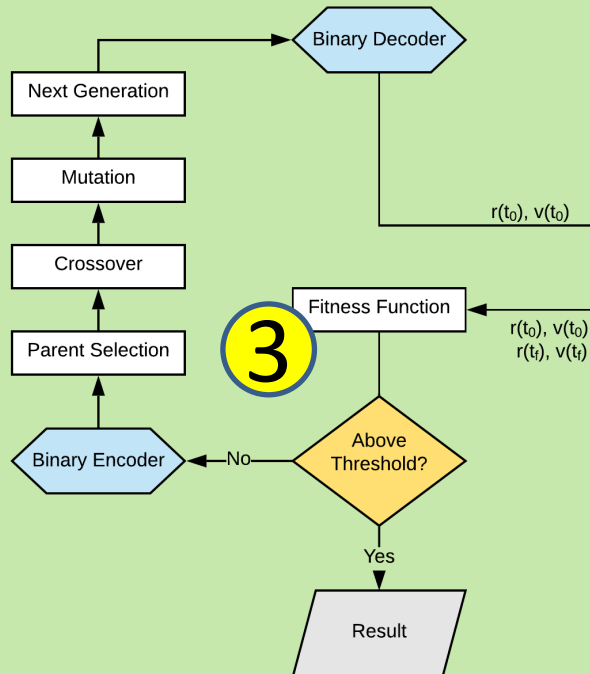


1. Begin with a set of random or distributed starting points
 - Goal is to solve for a closed trajectory over fixed time interval
 - Initial conditions are position and velocity
 - Larger populations are more diverse but require more computation. Test cases use 200
2. Propagate to find pos/vel at time T
 - Want to see if trajectory is closed or not

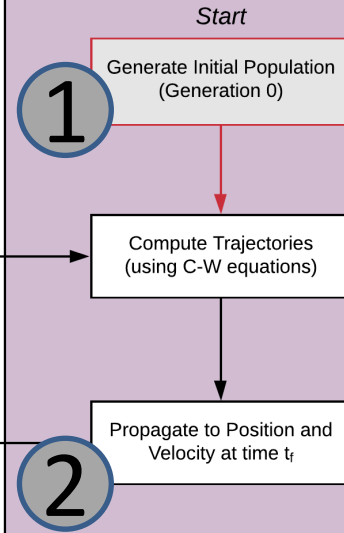
GA – Fitness Function



Genetic Algorithm



Dynamics



3. Apply fitness function to population

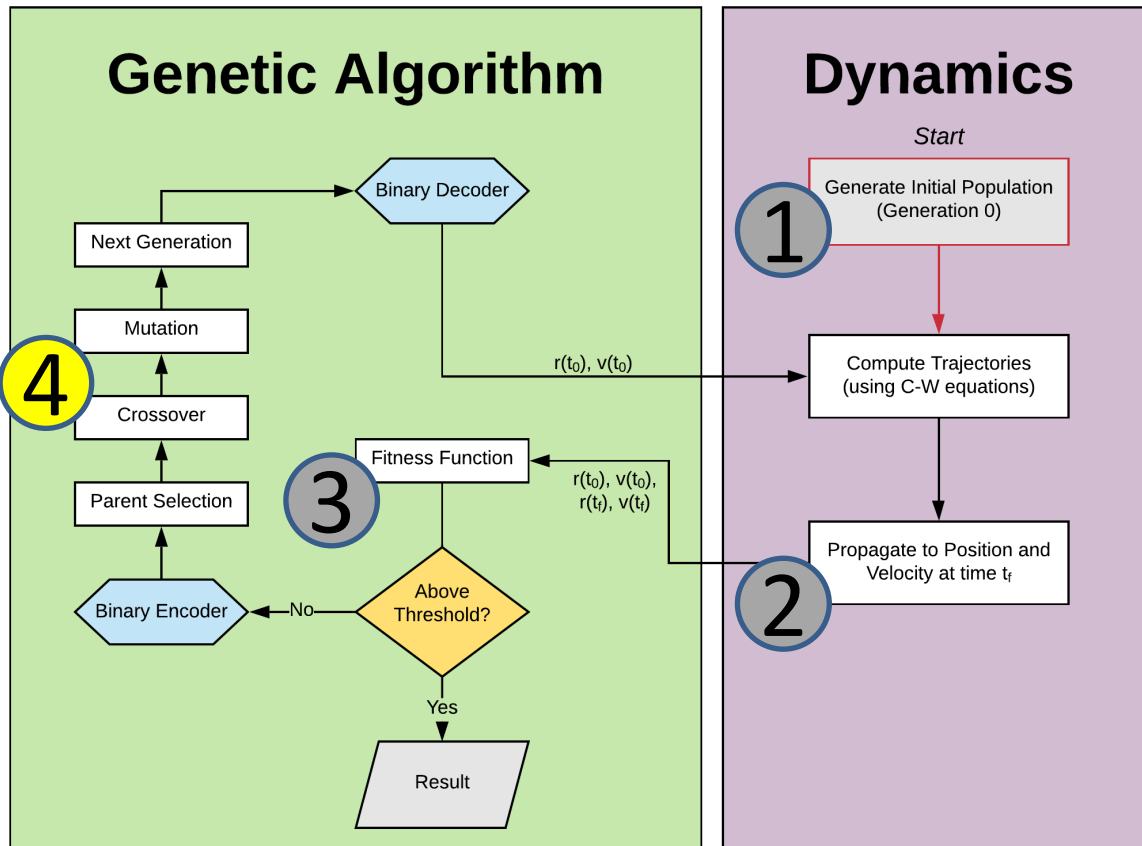
- Fitness function quantifies the acceptability of a given solution
- Enforces the criteria specified by the swarm system architect

$$F = \left(1 + C_r \left\| \vec{r}(t_f) - \vec{r}(t_0) \right\| + C_v \left\| \vec{v}(t_f) - \vec{v}(t_0) \right\| \right)^{-1}$$

C_r : coefficient of position

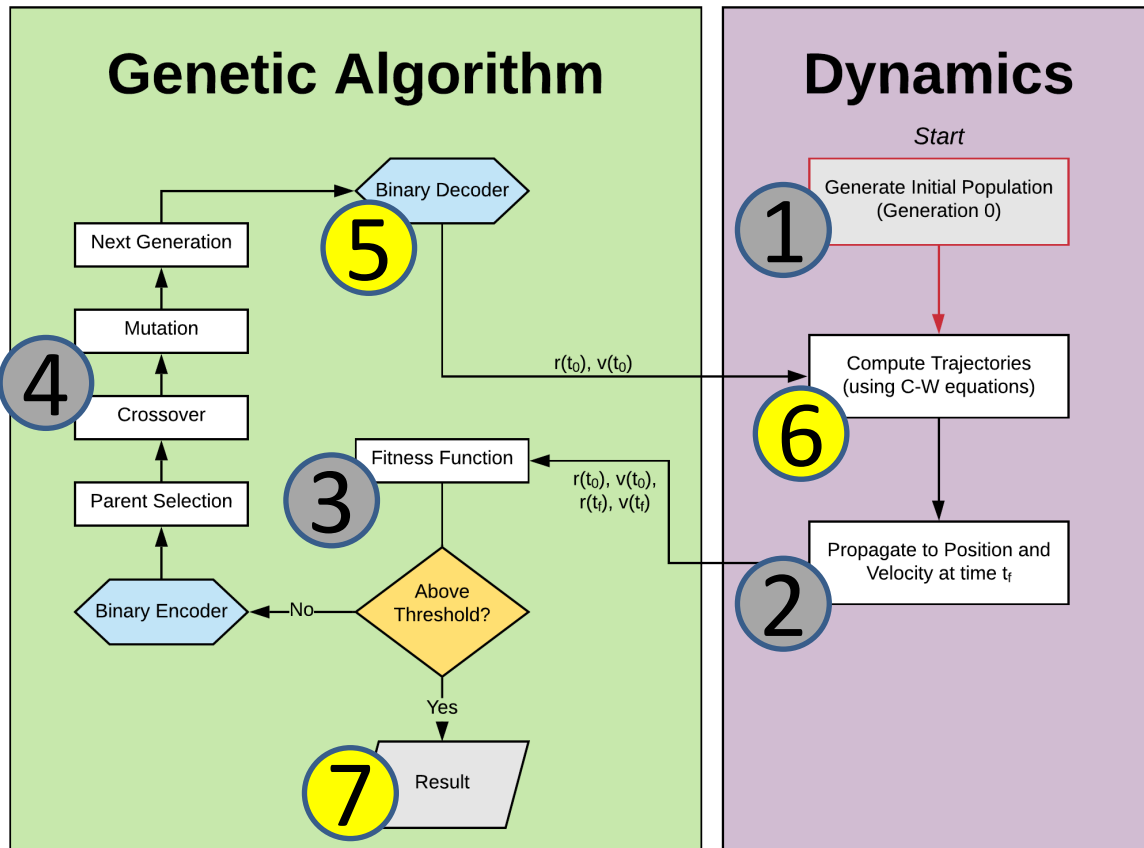
C_v : coefficient of velocity

GA – Crossover and Mutation



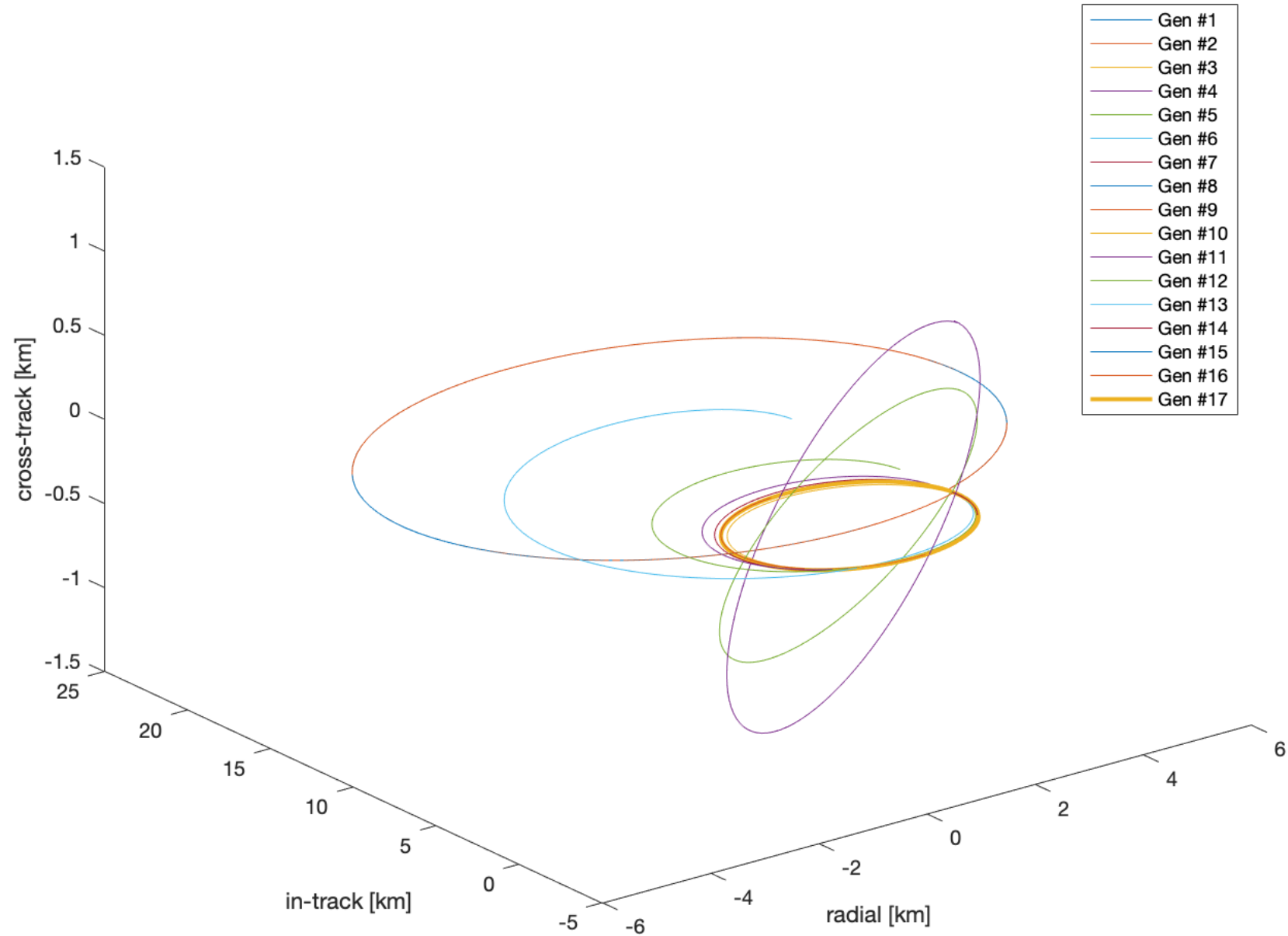
4. Rebuild population using crossover
 - Drop solutions with undesirable fitness values
 - Encode values in binary strings
 - Use binary string crossover to promote genetic diversity in solution sets
 - Higher fitness values have a higher rate of crossover (roulette method)
 - Mutation then creates random bit flips to enable new traits to emerge

GA – Achieving the Desired Results

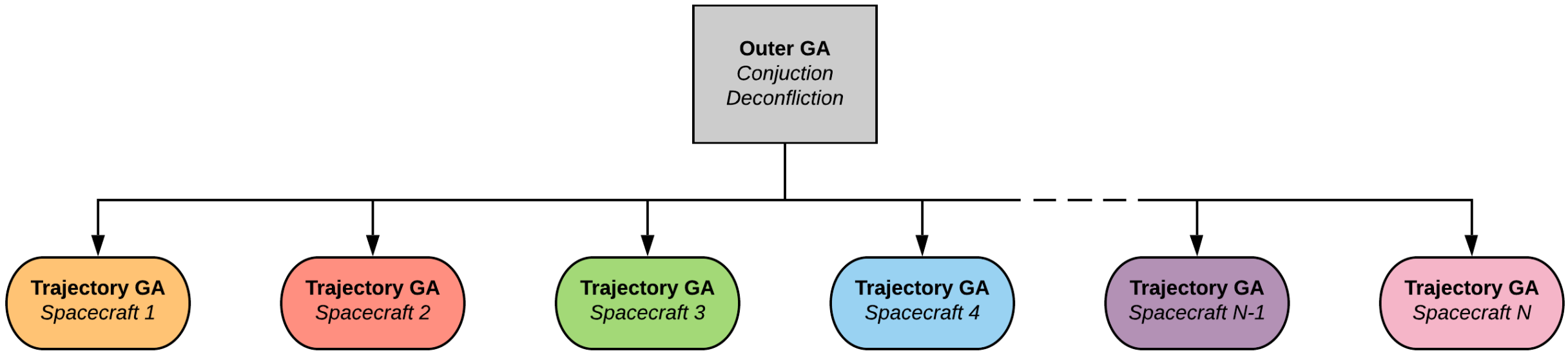


5. Decode population into their respective variables from binary
6. Propagate trajectories of new initial conditions
7. Continue this process until the desired results are achieved
 - Typically a fitness value of 1 (or within a threshold)
 - Use a generation counter to limit generations if no solution is found

Solution Convergence



Solving for Spacecraft Swarms



- Nested GAs are used: one for each spacecraft, all nested within a larger GA to de-conflict for collisions
- Conjunction is defined as two spacecraft being closer than a given buffer threshold (50m in test cases)

Specifying Individual Spacecraft Requirements



$$F = (1 + C_r \|\vec{r}(t_f) - \vec{r}(t_0)\| + C_v \|\vec{v}(t_f) - \vec{v}(t_0)\| + C_d \delta_{dist})^{-1}$$

$$\delta_{dist} = \begin{cases} d_{min} - r_{min} & \text{if } r_{min} < d_{min} \\ r_{max} - d_{max} & \text{if } r_{max} > d_{max} \\ 0 & \text{otherwise} \end{cases}$$

C_r	: coefficient of position
C_v	: coefficient of velocity
C_d	: coefficient of distance
r_{min}	: closest range to Client spacecraft [km]
r_{max}	: farthest range to Client spacecraft [km]
d_{min}	: closest permissible distance to Client spacecraft [km]
d_{max}	: farthest permissible distance to Client spacecraft [km]

- Individual requirements are enforced using the fitness function, tailored to the swarm member's role
- Each spacecraft is assigned its own fitness function, defined by the mission requirements for that spacecraft

Swarm Solution – 10 Spacecraft



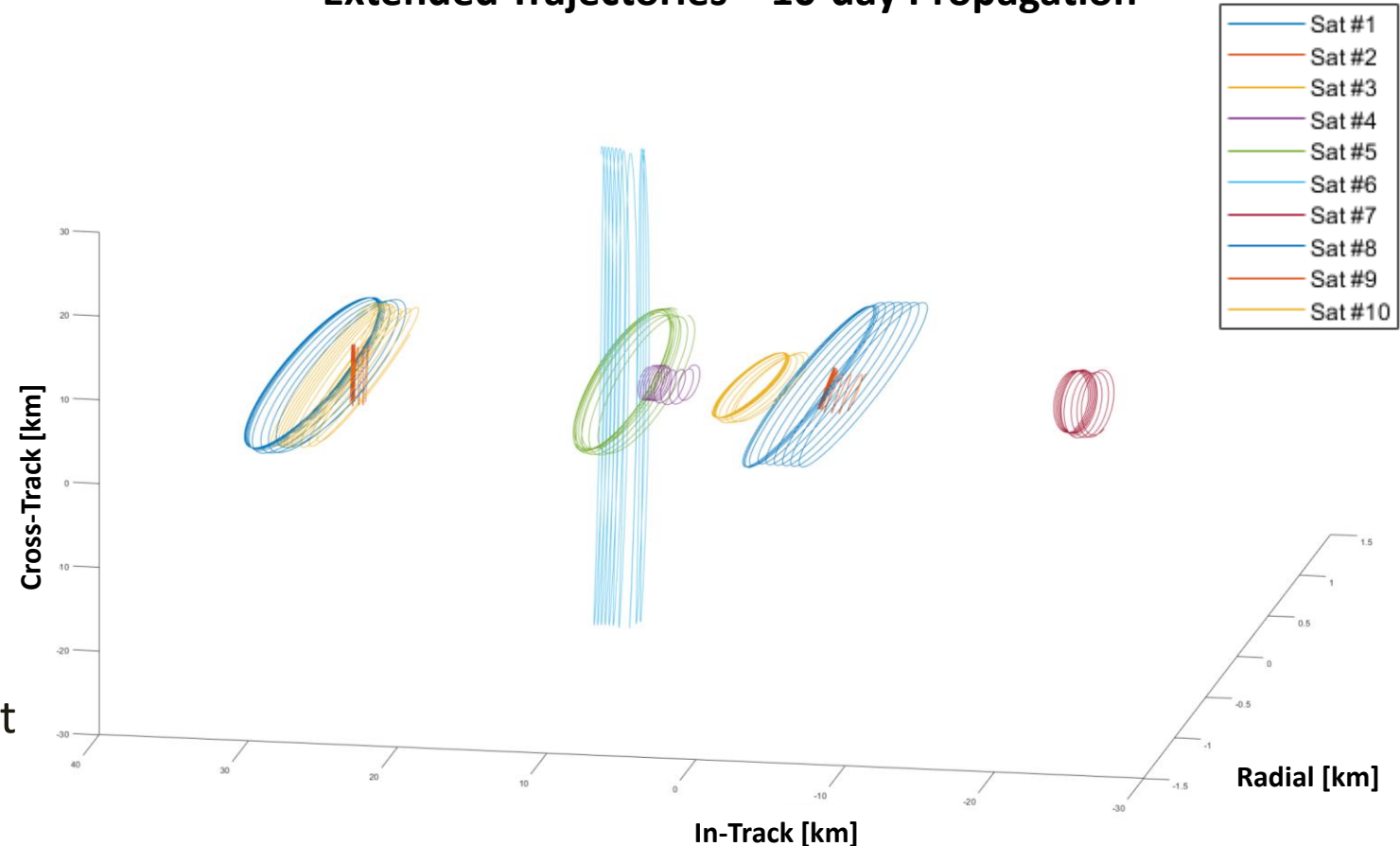
Swarm Inputs

Swarm Member	Constraint	Constraint
	Ellipsoid Dimensions [km]	Ellipsoid Center [km]
Spacecraft 1	[2,2,2]	[0,0,0]
Spacecraft 2	[2,2,2]	[0,-5,0]
Spacecraft 3	[3,3,3]	[0,2,0]
Spacecraft 4	[5,5,5]	[0,7,3]
Spacecraft 5	[2,2,4]	[0,10,0]
Spacecraft 6	[2,2,4]	[0,10,0]
Spacecraft 7	[2,2,4]	[0,-20,0]
Spacecraft 8	[5,5,5]	[0,30,0]
Spacecraft 9	[5,5,5]	[0,30,0]
Spacecraft 10	[5,5,5]	[0,30,0]

Not a unique solution

- Family of infinite solutions exist that satisfy criteria above
- We only want one

Extended Trajectories – 10-day Propagation



Dynamic Trajectory Modification



- What happens when a new spacecraft is inserted into the swarm?
 - Use modified GA scheme to build on previous N-spacecraft solution to generate N+1 solution
- New fitness function is required to minimize delta-v of transition
- Dynamic nature of the swarm must be considered
 - Can also be used to account for a spacecraft that is unresponsive and must be avoided

$$F_{insert} = \left(1 + C_r \left\| \vec{r}(t_f) - \vec{r}(t_0) \right\| + C_v \left\| \vec{v}(t_f) - \vec{v}(t_0) \right\| + C_d \delta_{dist} + \Delta v \right)^{-1}$$

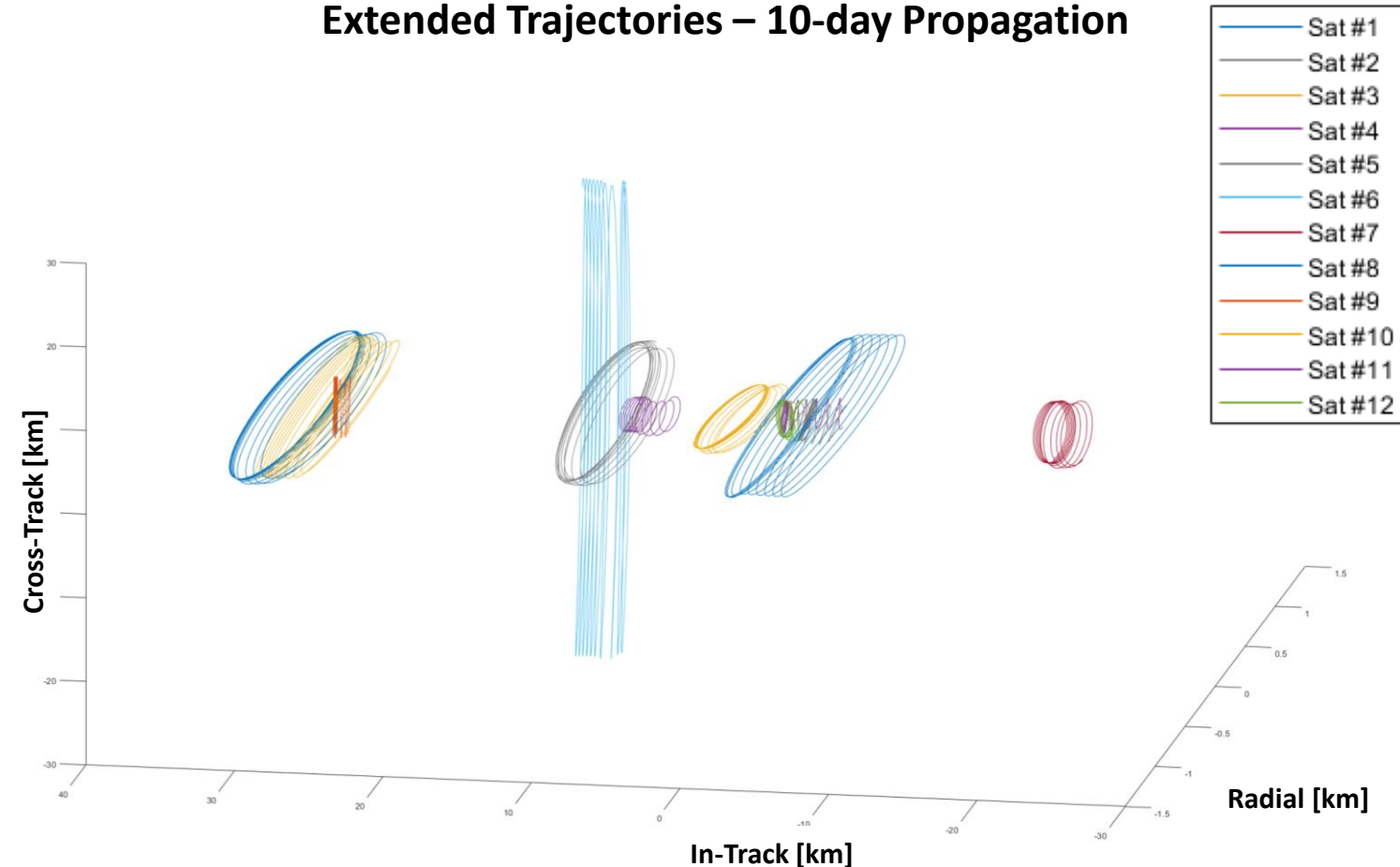
Swarm Solution – Added Spacecraft



Two new spacecraft added

- Other spacecraft trajectories modified to allow additional spacecraft into the swarm
- Algorithm minimizes DeltaV for operation
 - Results in approx. 5m/s for entire swarm (reference orbit altitude at 600km)

Extended Trajectories – 10-day Propagation





Problems 1 & 3 Have a Solution

1. **Given a co-located swarm of N free-flying spacecraft capable of relative position, velocity, and orientation determination, generate a set of trajectories that enable these spacecraft to complete their individual tasks within their ΔV budgets, while mitigating collision risks over a minimum 24hr period**
 - Can be solved using Genetic Algorithms, an evolutionary optimization scheme
2. **Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, maintain these trajectories in real-time, accounting for deviations due to injection errors, unaccounted for higher-order or non-gravitational perturbations, sensor errors, or system noise**
 - Can be solved using a Sensor Fusion Kalman Filter, combining shared sensor data from all spacecraft in the swarm
3. **Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, generate a new set of trajectories for a modified swarm, with some spacecraft either added or removed, while minimizing the ΔV required to re-position the existing swarm spacecraft to accommodate the new spacecraft**
 - Can be solved using Genetic Algorithms, similar to problem 1



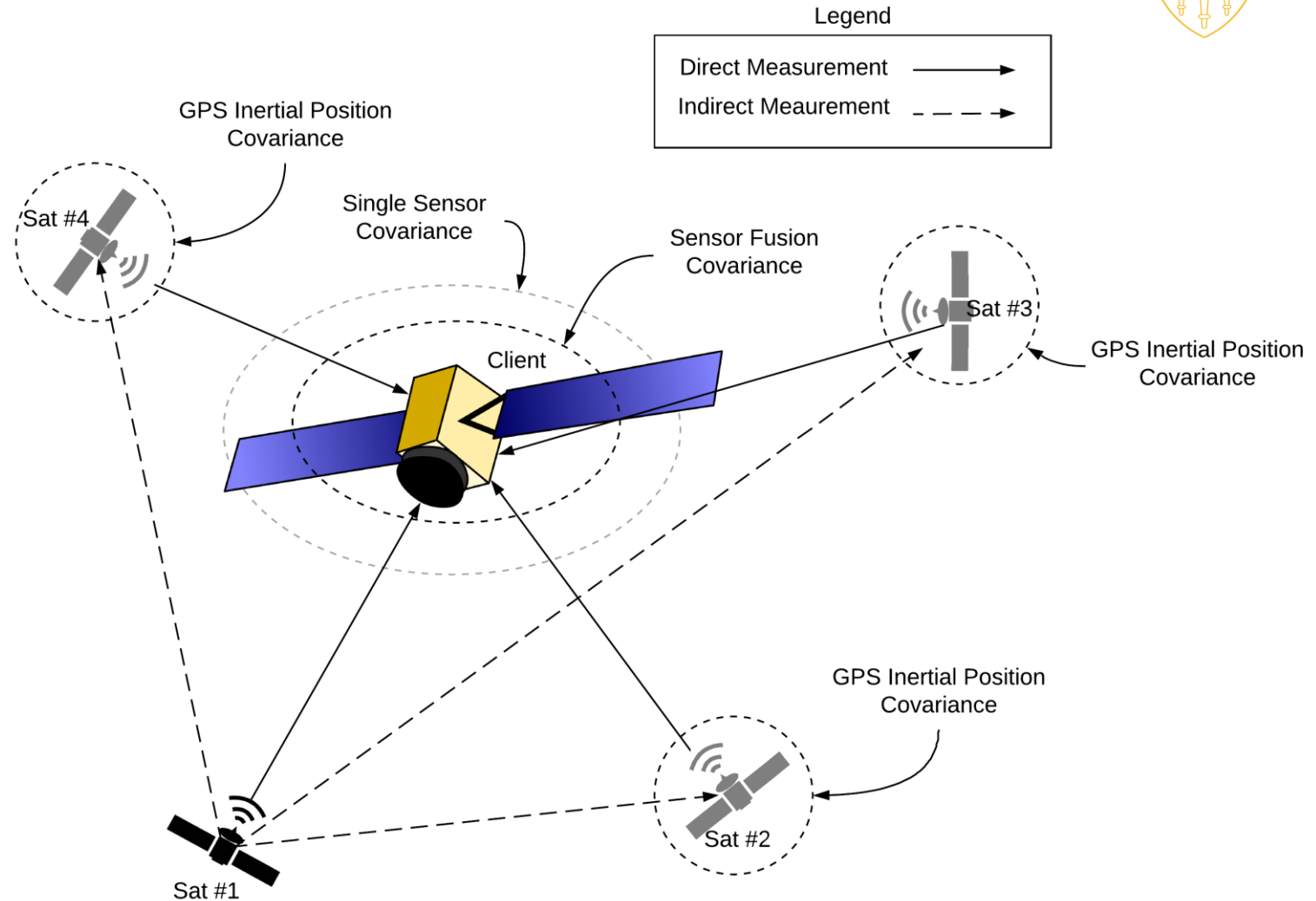
Sensor Fusion

Kalman Filtering

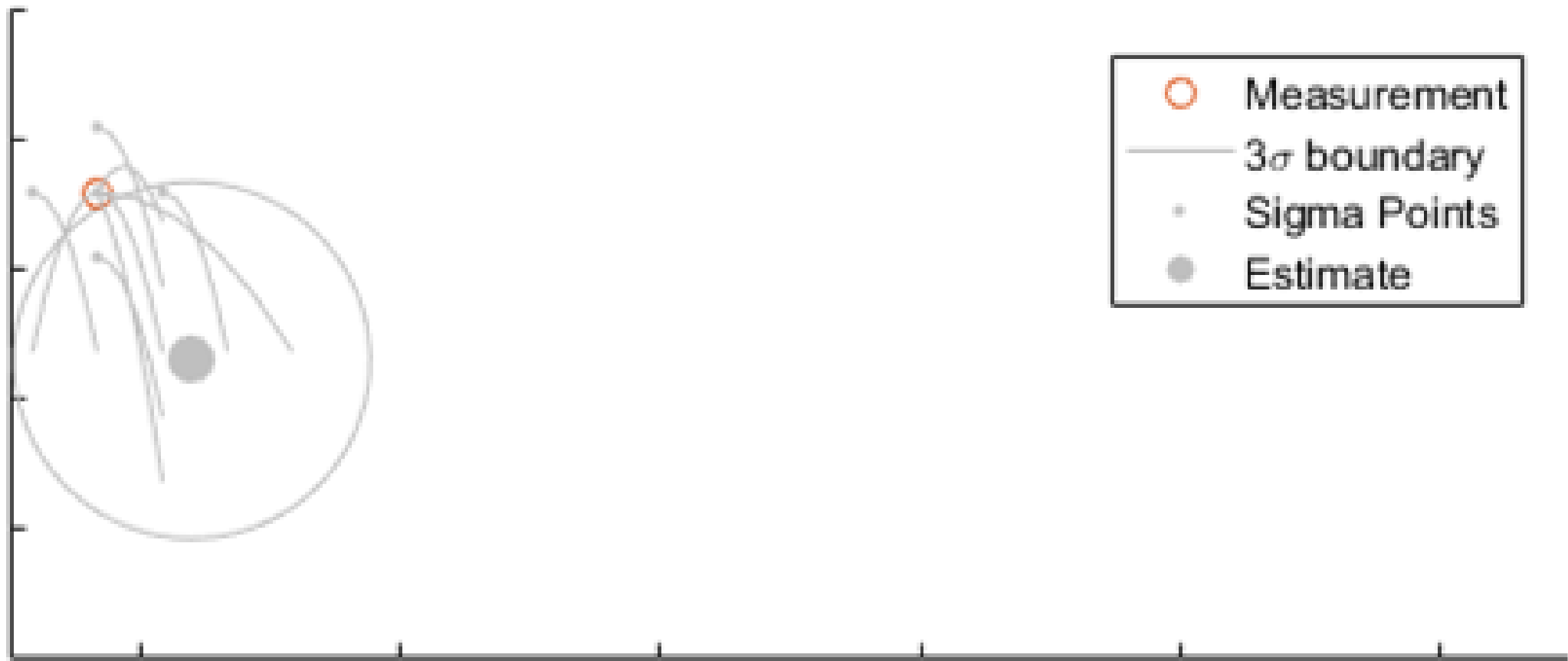
Swarm Sensor Fusion



- Sensor Fusion combines inputs from multiple sensors, spread across the swarm
- Using a Kalman filter, this shared data can be used to pinpoint the relative positions of each spacecraft more accurately, reducing their covariances

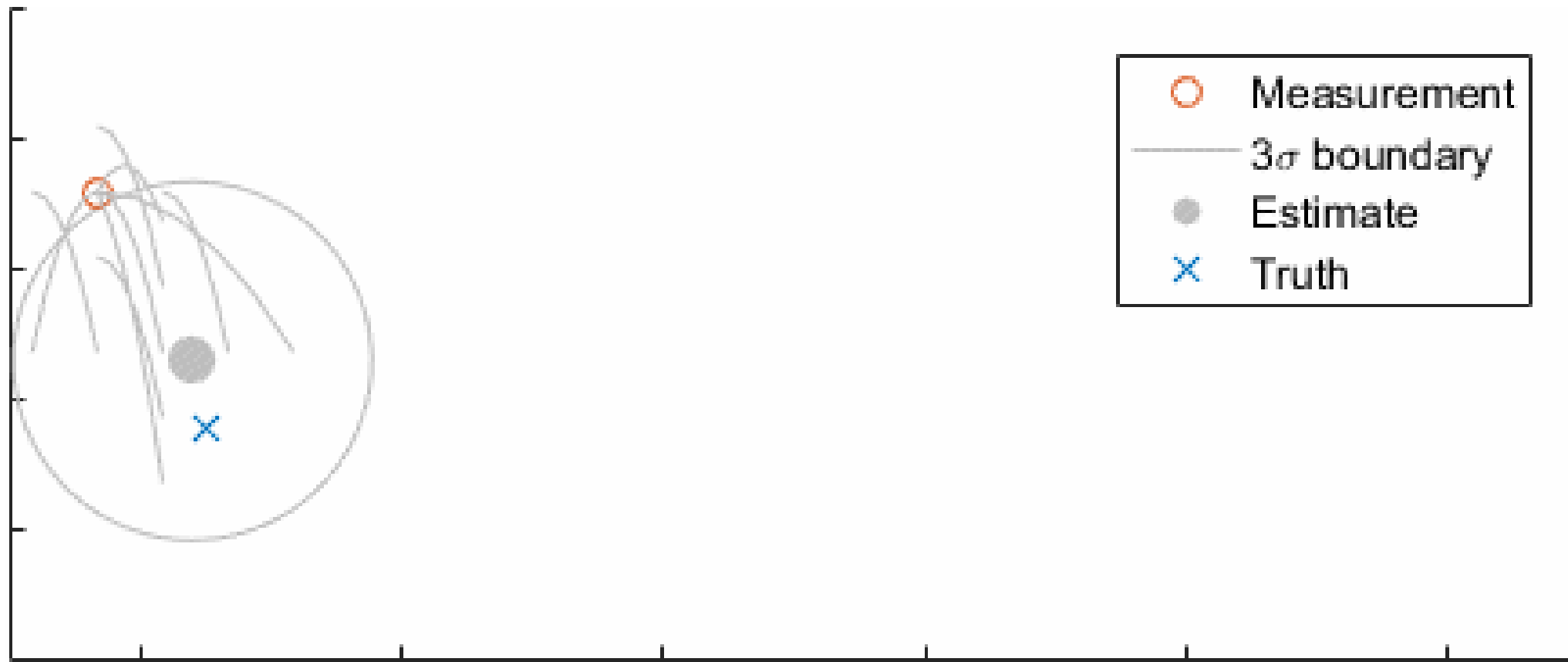


What is a Kalman Filter?



Credit: Tucker McClure, An Uncommon Lab

What is a Kalman Filter?

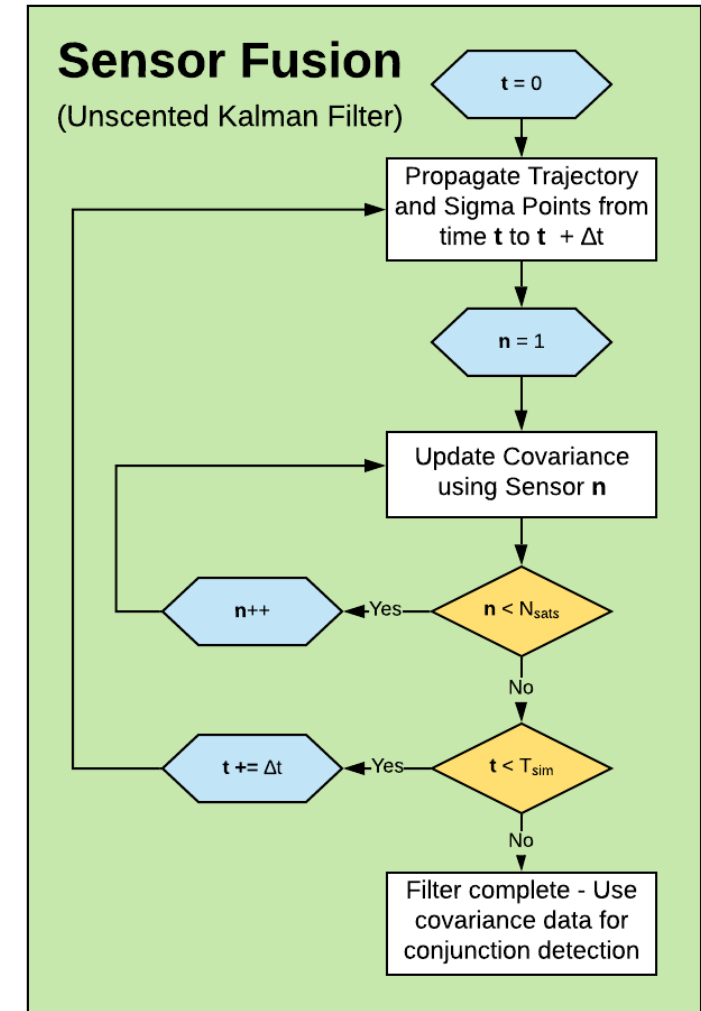


Credit: Tucker McClure, An Uncommon Lab

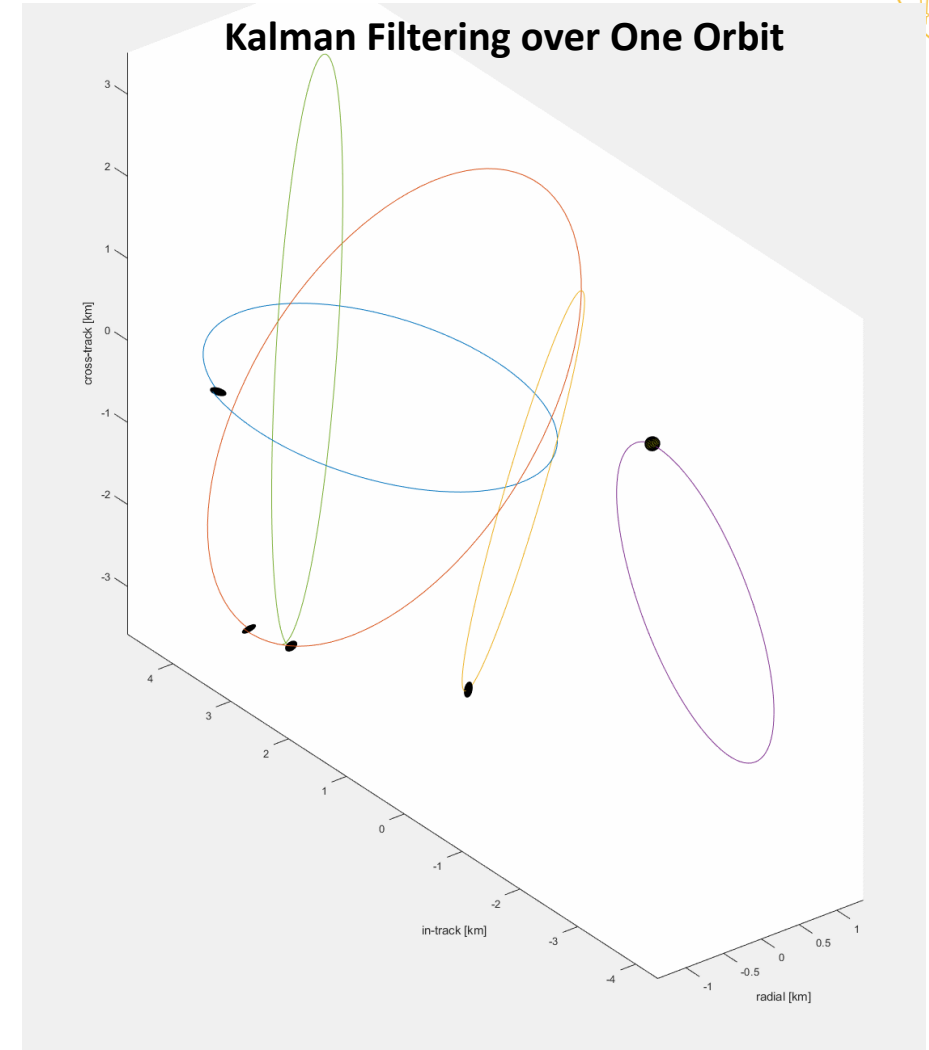
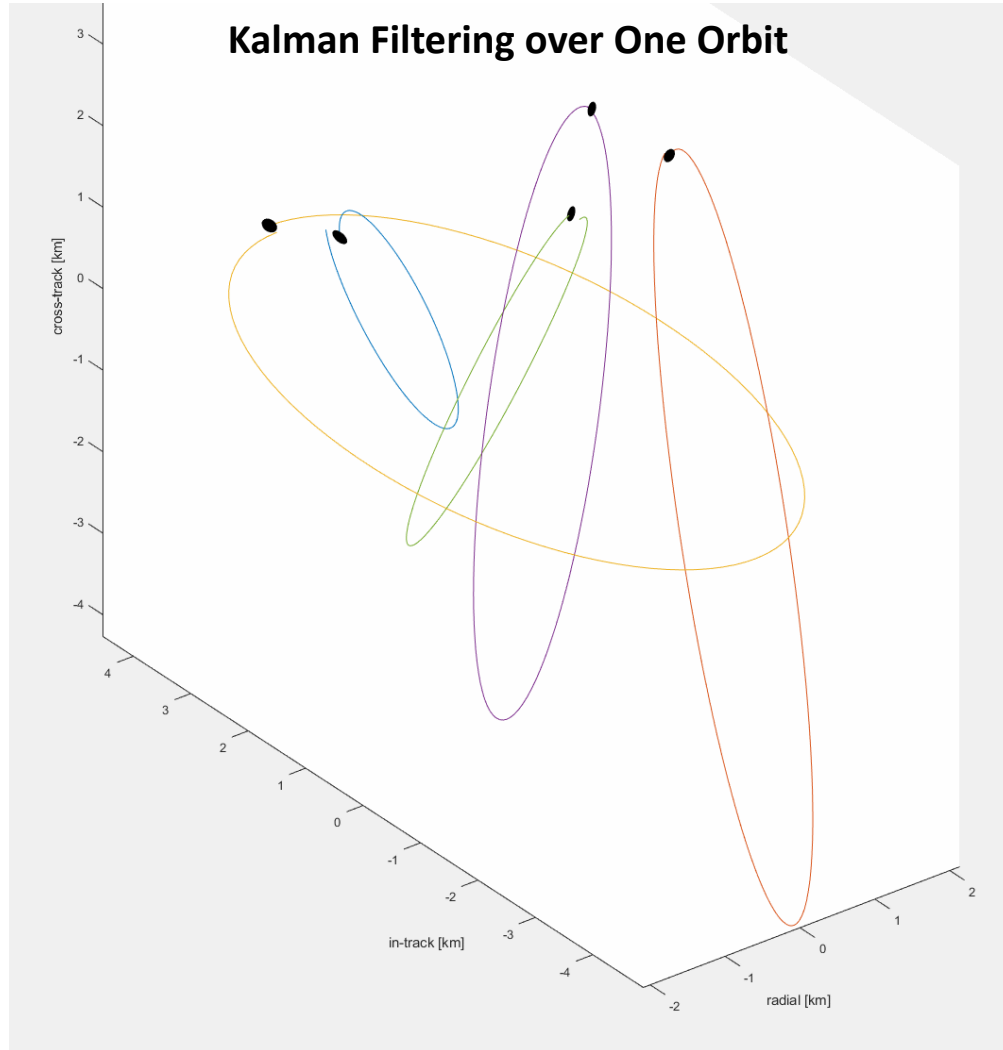
Sensor-Fusion Kalman Filtering



- Sensor fusion can be applied to Kalman filters
 - Simulation uses the Unscented Kalman Filter since the perturbed 2-body problem is a non-linear problem
- Similar to a standard Kalman filter, except the update step is repeated for each sensor in the shared swarm sensor net
 - Adds very little computational overhead, as most of the wall-time is spent on the propagation step of the UKF



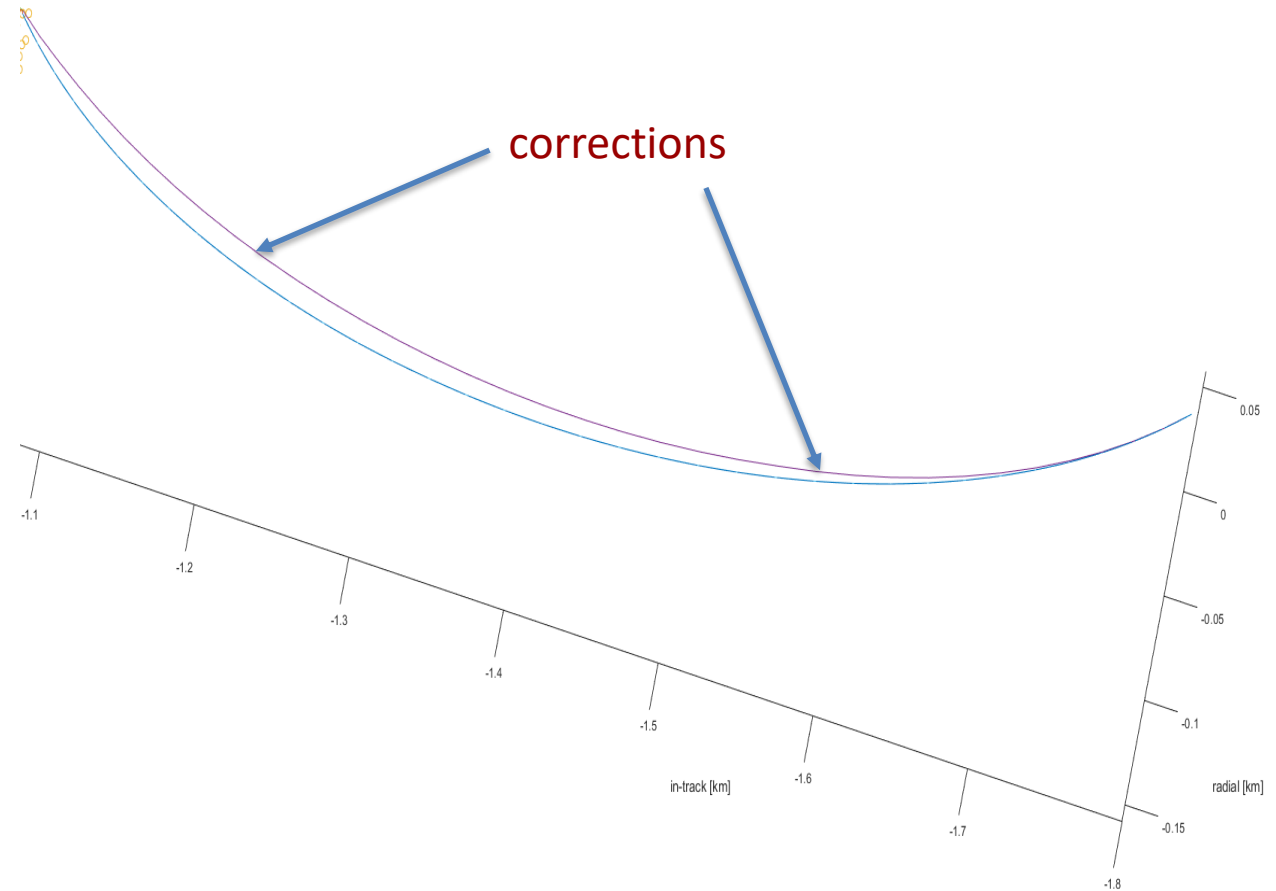
Example of Real-Time Kalman Filtering



Trajectory Maintenance



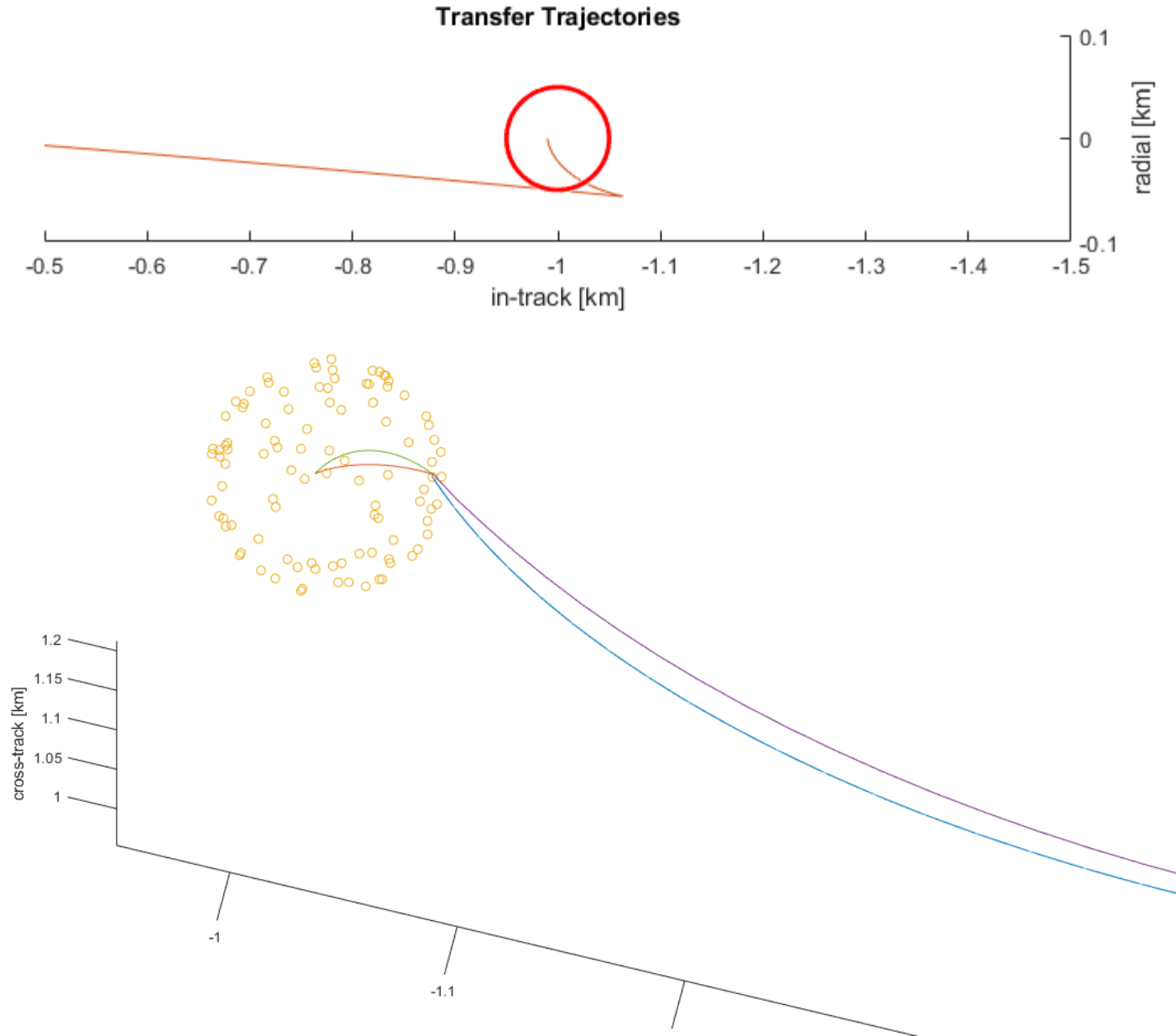
- A Sensor Fusion Kalman Filter can be used to maintain GA generated swarm trajectories over time by tracking deviations from the initial trajectory and compensating with periodic correction burns
 - Each spacecraft is assigned a corridor and can deviate within that corridor without risk of collision with other SC (50m in these simulations)



Patched RPO Transfers



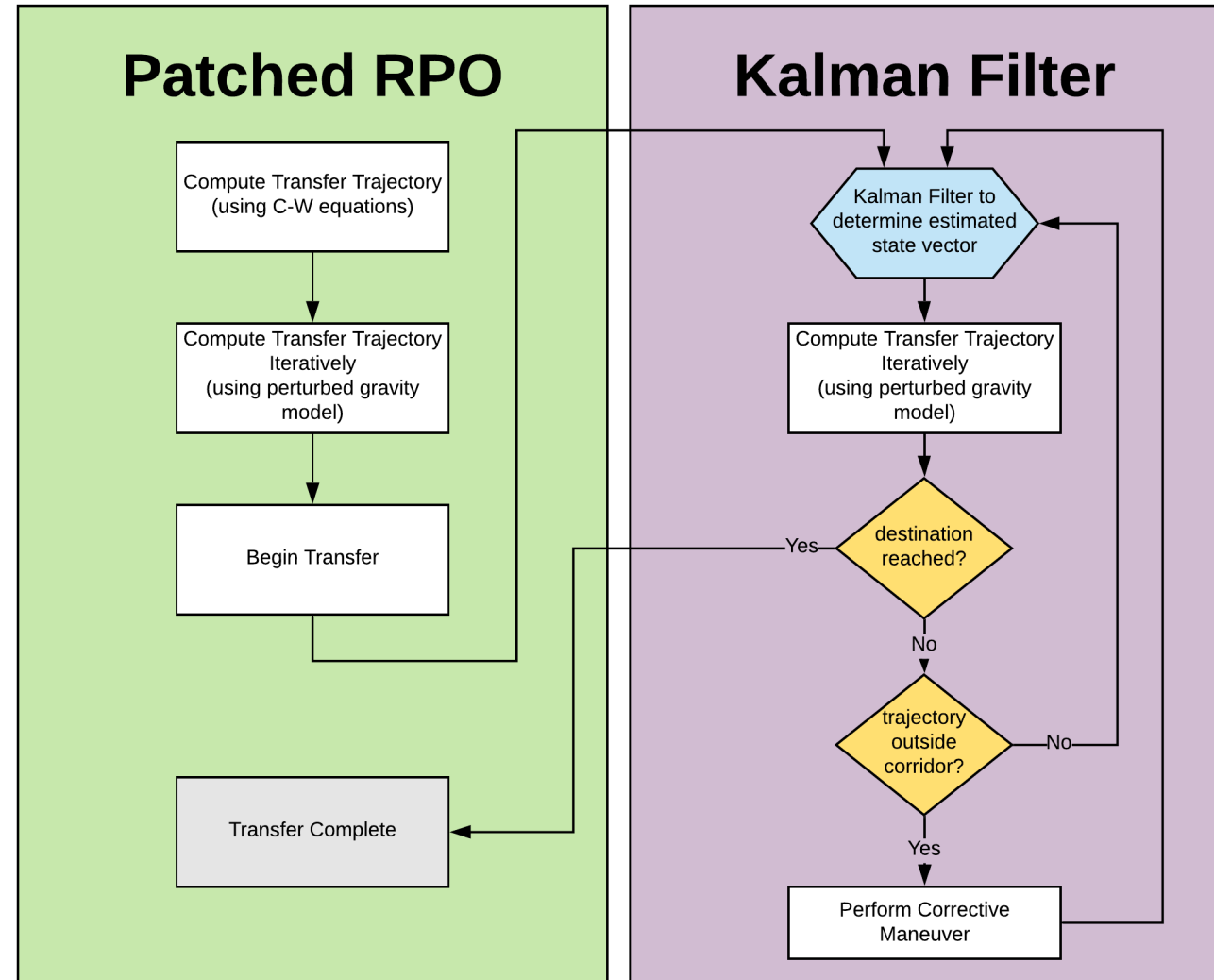
- Use a two-stage method for trajectory transfers
 - Much safer than single stage
- First stage gets vehicle close to the target, but remains entirely outside a keep-out zone around target
 - Enables a passively safe transfer
- Second stage then closes the small gap from the edge of the keep-out zone to the target



Patched RPO Transfers



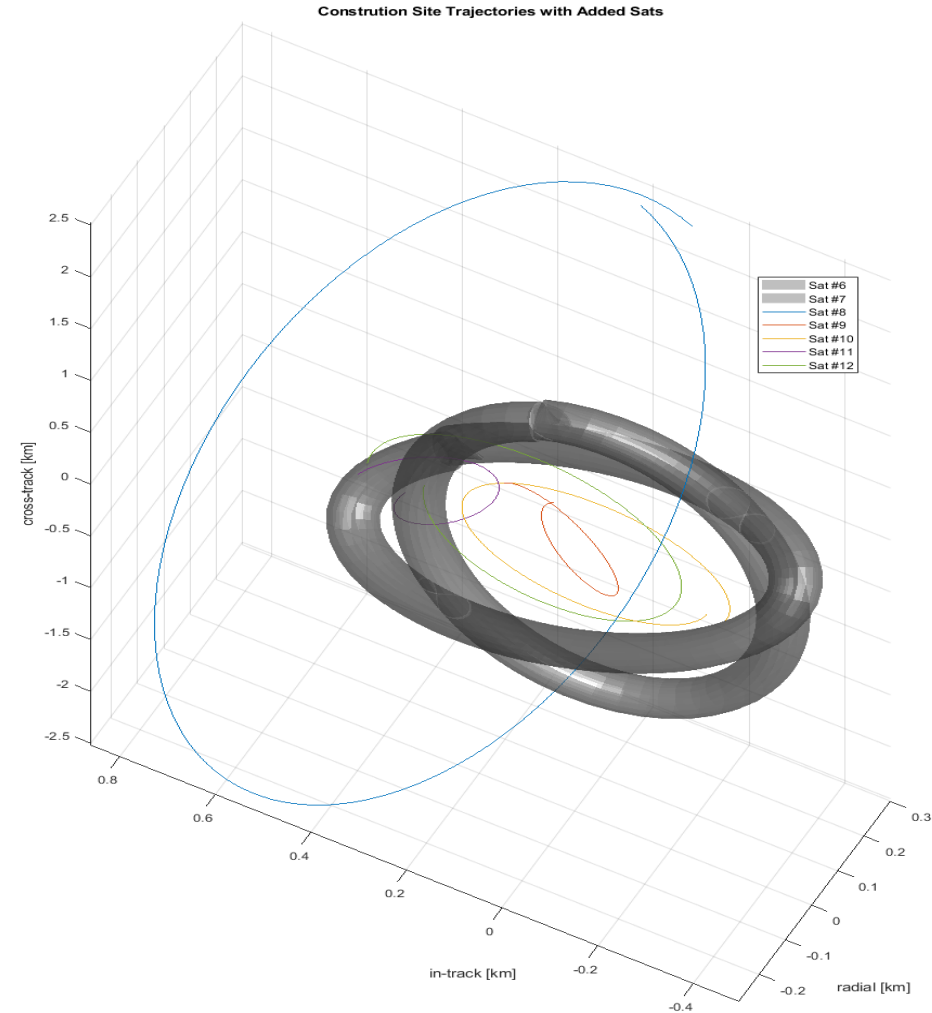
- When combined with Kalman filtering, trajectories can be corrected in real-time to account for injection and attitude errors
 - Define a safe corridor for transfer
- Kalman filter continuously updates the estimated position of the spacecraft, and re-computes the transfer trajectory in case of deviation



Death of a Spacecraft



- What happens when a spacecraft becomes unresponsive?
 - Designate as a *zombie* spacecraft with restricted zone
- All further trajectory modifications will consider this *zombie* spacecraft as immutable





Problem 2 Has a Solution

1. **Given a co-located swarm of N free-flying spacecraft capable of relative position, velocity, and orientation determination, generate a set of trajectories that enable these spacecraft to complete their individual tasks within their ΔV budgets, while mitigating collision risks over a minimum 24hr period**
 - Can be solved using Genetic Algorithms, an evolutionary optimization scheme
2. **Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, maintain these trajectories in real-time, accounting for deviations due to injection errors, unaccounted for higher-order or non-gravitational perturbations, sensor errors, or system noise**
 - Can be solved using a Sensor Fusion Kalman Filter, combining shared sensor data from all spacecraft in the swarm
3. **Given an existing set of co-located swarm trajectories as generated by the solution to the first problem, generate a new set of trajectories for a modified swarm, with some spacecraft either added or removed, while minimizing the ΔV required to re-position the existing swarm spacecraft to accommodate the new spacecraft**
 - Can be solved using Genetic Algorithms, similar to problem 1



Assumptions & Requirements



Initial Constraints

- 1. All spacecraft in the swarm have a known mass, moment of inertia, and center of gravity**
 - Any changes to these values are tracked by the system as fuel is consumed or replenished, and as the spacecraft are aggregated or dis-aggregated
- 2. The number of satellites, N , in the swarm is known and finite**
- 3. The relative-motion trajectory's reference point is moving in a circular orbit**
 - RPO takes place in a non-inertial co-moving reference frame. The origin of this reference frame is assumed to be traveling in a circular orbit in inertial space
- 4. Central body's gravitational field can be modeled using spherical or zonal harmonics**
- 5. Communications delay is negligible**
- 6. All clocks are perfectly synchronized across the swarm**
- 7. Spacecraft that lose communication with the rest of the swarm will enter a passive mode**
 - Ensures predictable trajectories for unresponsive spacecraft
- 8. Each spacecraft will have a radio beacon signal with unique identifier (IFF)**



Spacecraft Requirements

- **On-board propulsion (chemical or electric) with minimum 200m/s ΔV**
- **3-axis ADACS with 0.5deg accuracy**
- **Relative position sensors with 5% error**
 - Conservative estimate for current RADAR and LIDAR capabilities
- **Relative speed sensors with 1% error**
 - Conservative estimate for current RADAR and RF doppler capabilities
- **Redundant communication systems for inter-swarm data transfer**



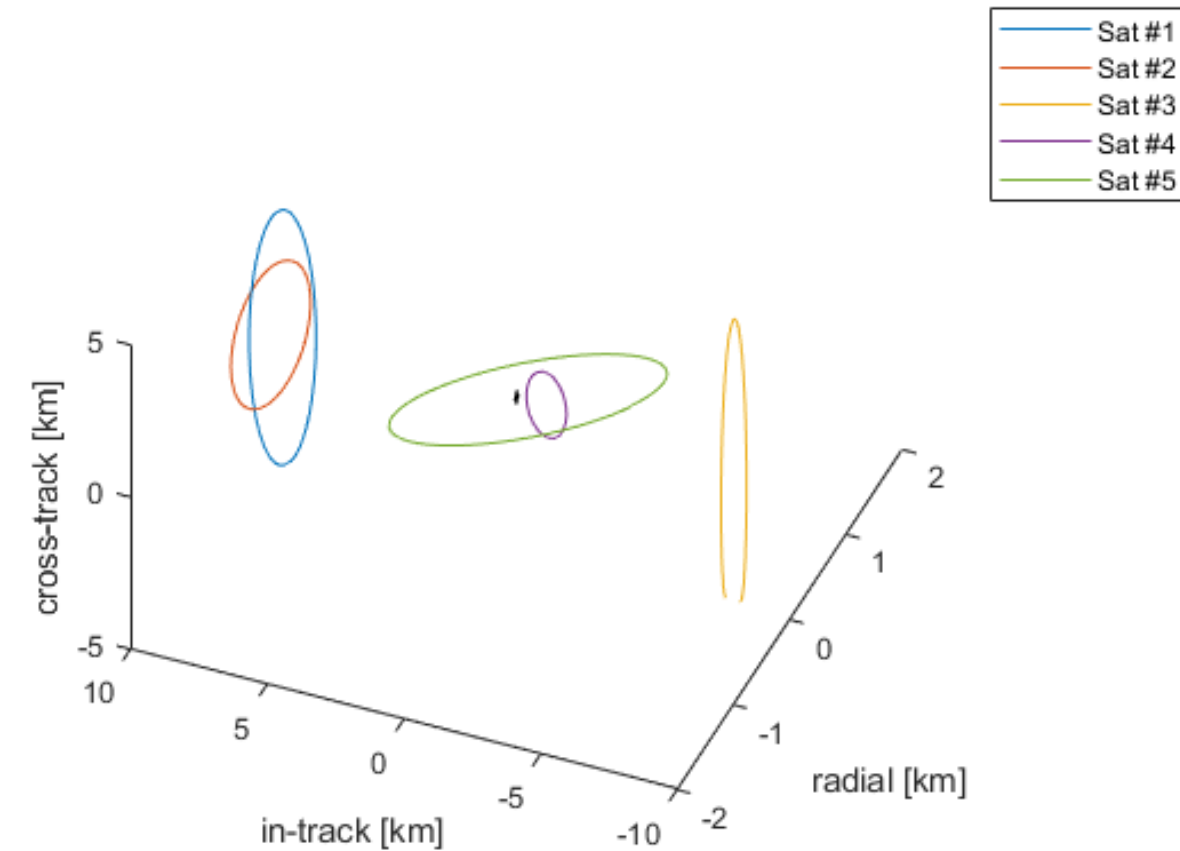


Examples & Results

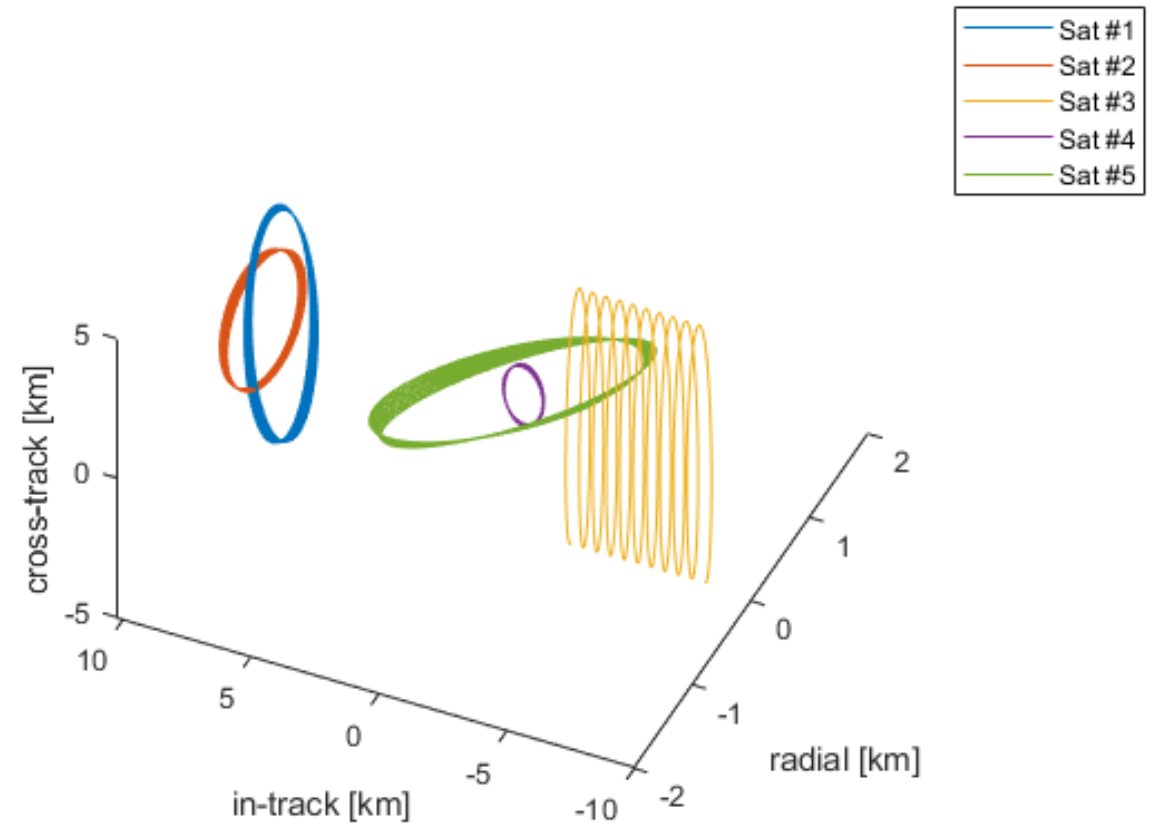
Trajectory Example



Optimized Trajectories



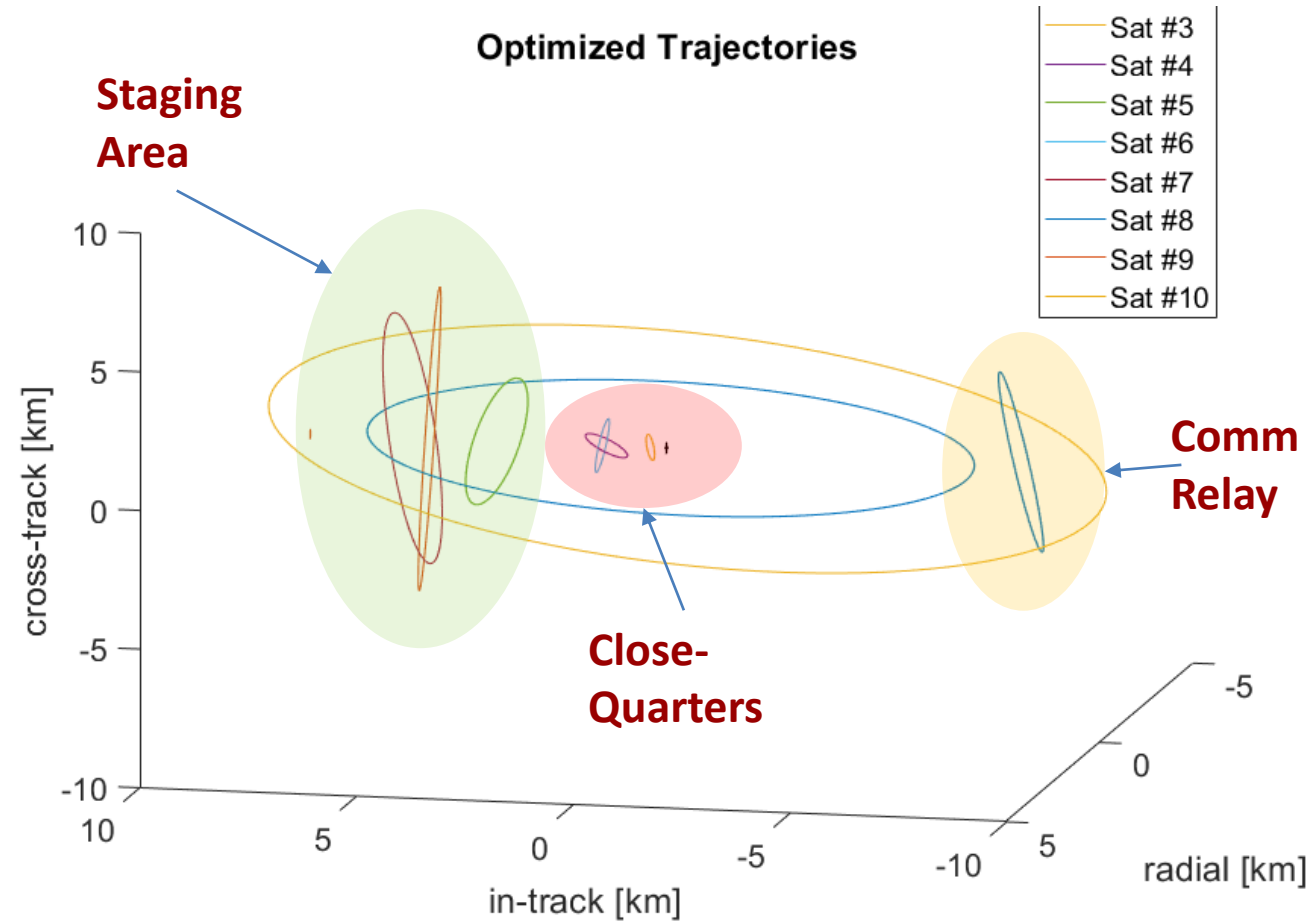
Extended Trajectories



In-Space Manufacturing



- Trajectory on right shows example of 10-spacecraft swarm for in-space manufacturing
- Swarm roles split up into the staging area, a comm relay, and close-quarters robotic operations



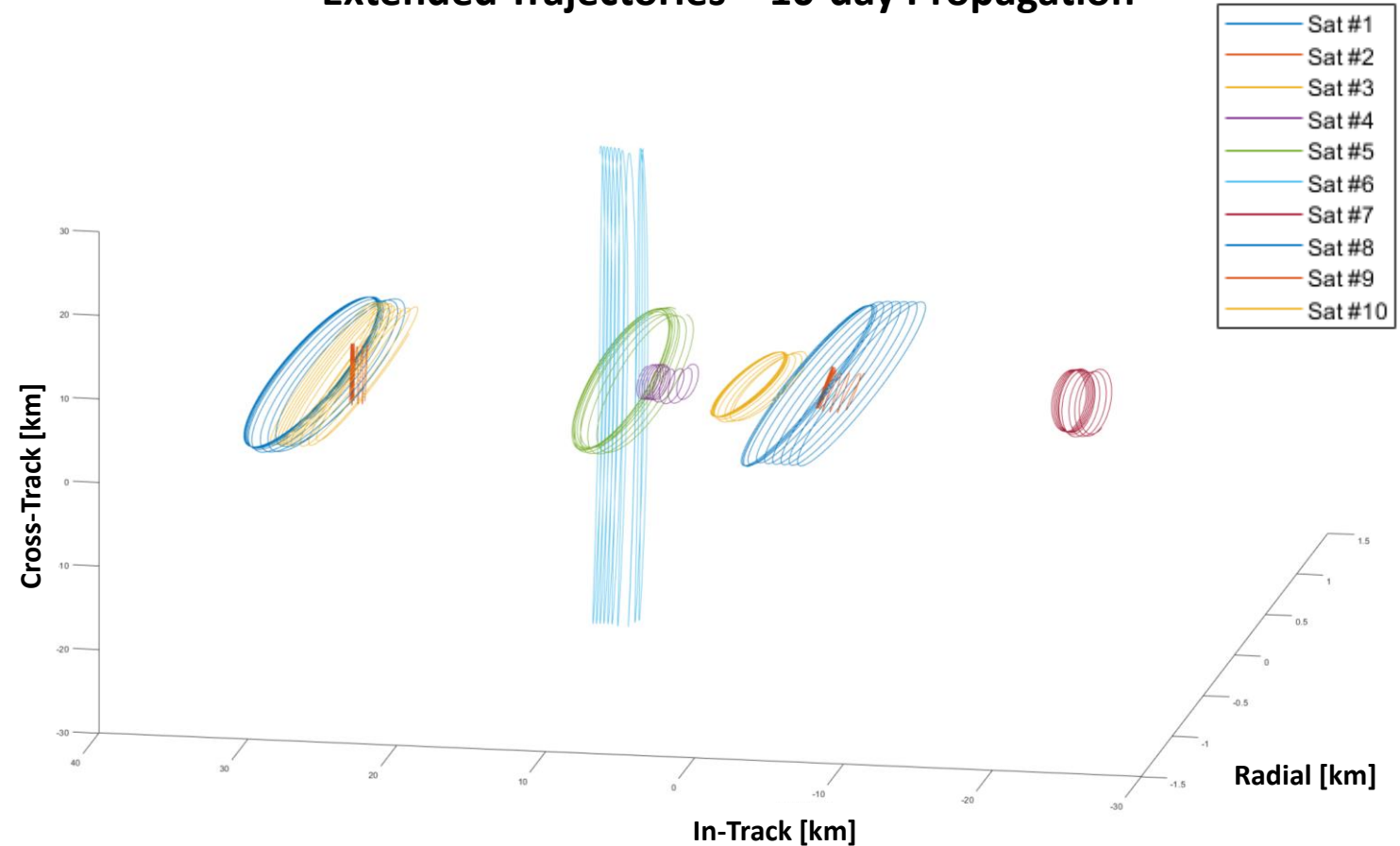
Swarm Solution – 10 Spacecraft



Swarm Inputs

Swarm Member	Constraint Ellipsoid Dimensions [km]	Constraint Ellipsoid Center [km]
Spacecraft 1	[2,2,2]	[0,0,0]
Spacecraft 2	[2,2,2]	[0,-5,0]
Spacecraft 3	[3,3,3]	[0,2,0]
Spacecraft 4	[5,5,5]	[0,7,3]
Spacecraft 5	[2,2,4]	[0,10,0]
Spacecraft 6	[2,2,4]	[0,10,0]
Spacecraft 7	[2,2,4]	[0,-20,0]
Spacecraft 8	[5,5,5]	[0,30,0]
Spacecraft 9	[5,5,5]	[0,30,0]
Spacecraft 10	[5,5,5]	[0,30,0]

Extended Trajectories – 10-day Propagation



Swarm Solution – Added Spacecraft

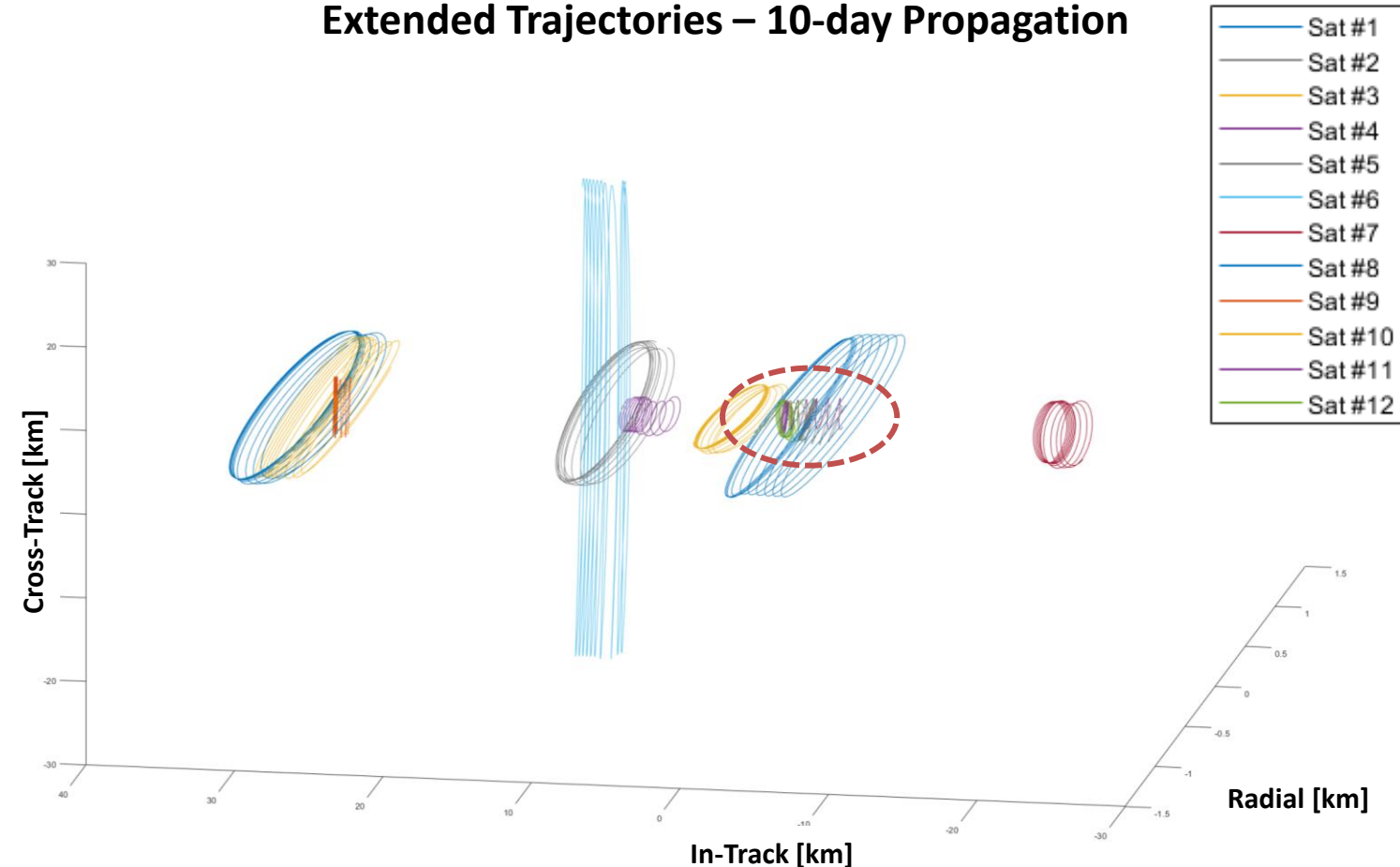


Two new spacecraft added

- Other spacecraft trajectories modified to allow additional spacecraft into the swarm
- Algorithm minimizes DeltaV for operation
 - Results in approx. 5m/s for entire swarm (reference orbit altitude at 600km)

SC #2 & #5 are zombie spacecraft, colored in gray

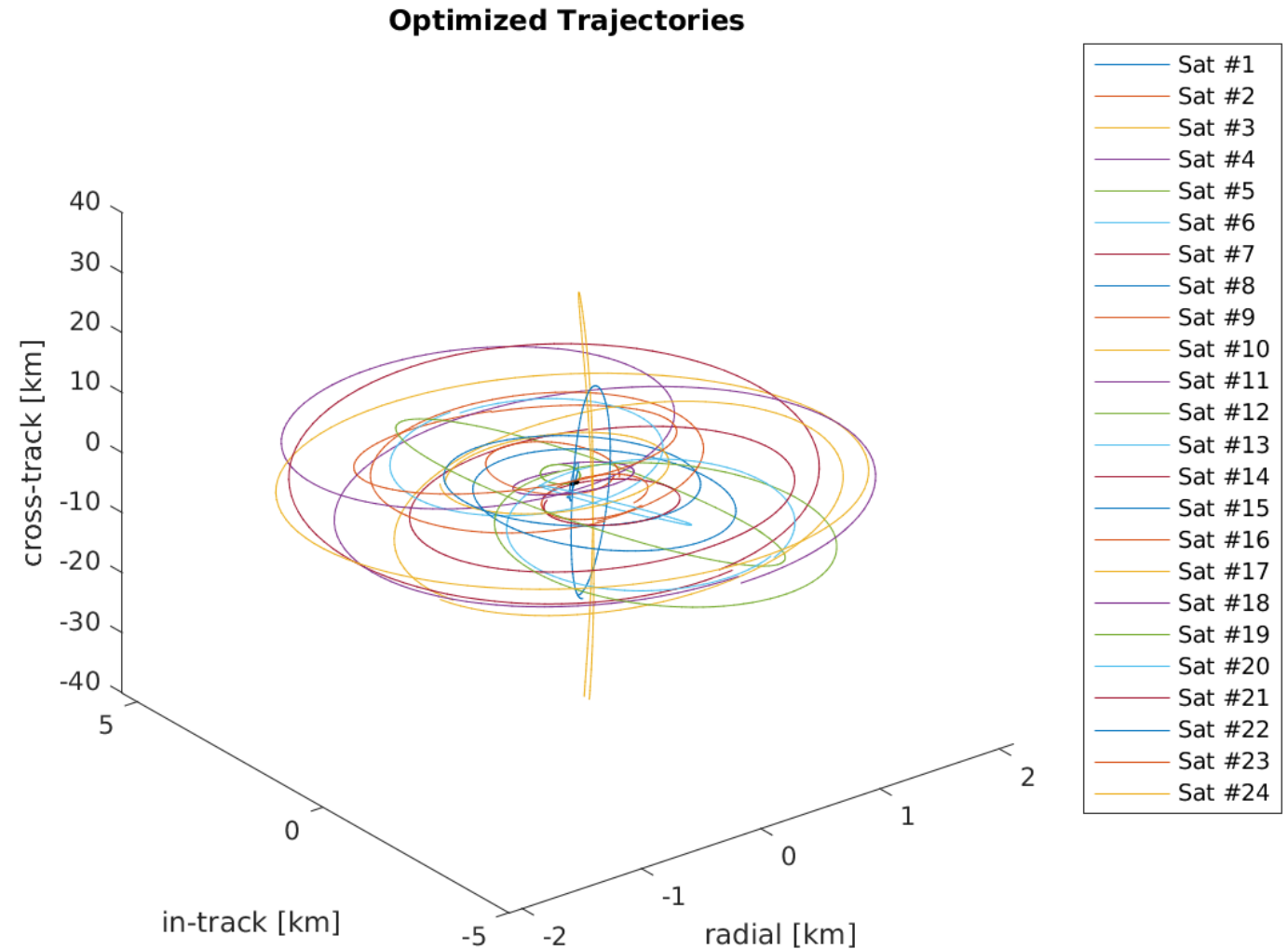
Extended Trajectories – 10-day Propagation



Large Swarms with HPC



- Large swarm simulations are run on the USC CARC supercomputing cluster
 - Parallel computing speeds up the process, as if each spacecraft were performing its own computations independently



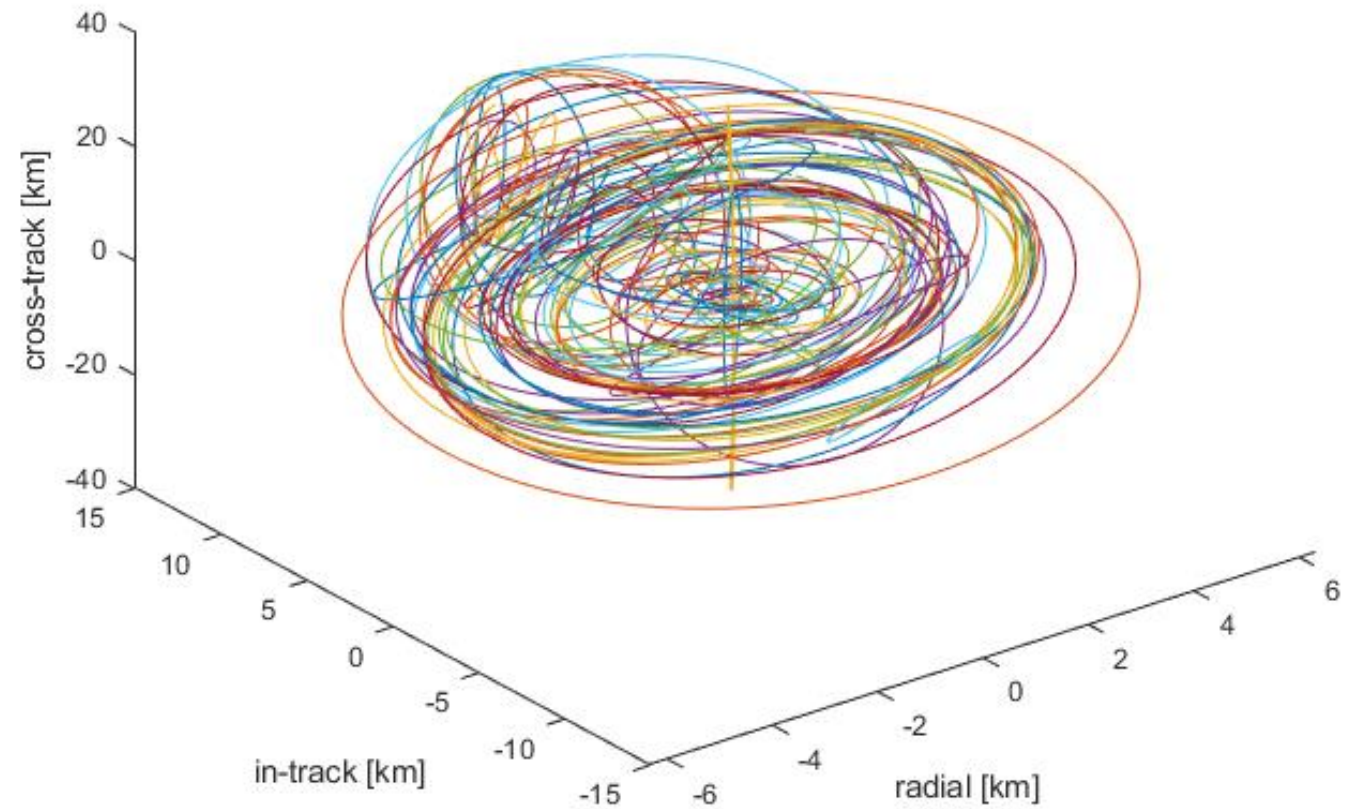
Supercomputing cluster access provided by the USC Center for Advanced Research Computing (CARC). <https://carc.usc.edu>

Large Swarms with HPC



- 100 SC simulation shows system complexity
- Semi-autonomous systems are required to effectively control large spacecraft swarms
 - The amount of data is too much for ground operators to control individually

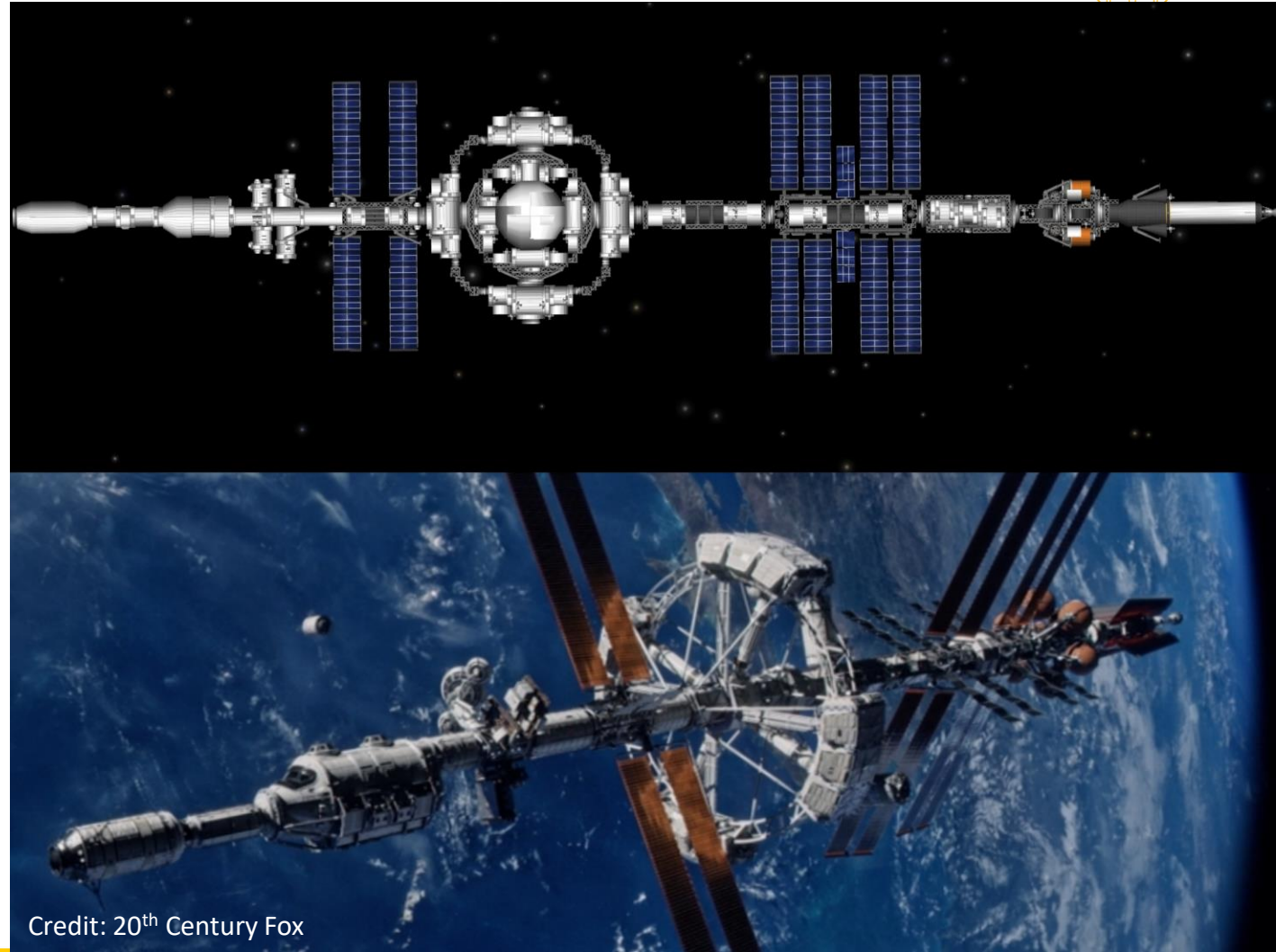
Optimized Trajectories - 100 SC Swarm



In-Space Assembly Example



- In-space assembly of the Hermes spaceship using a 10-spacecraft swarm
- Ship is composed of five equal mass segments, launched separately into orbit. These need to be retrieved and assembled



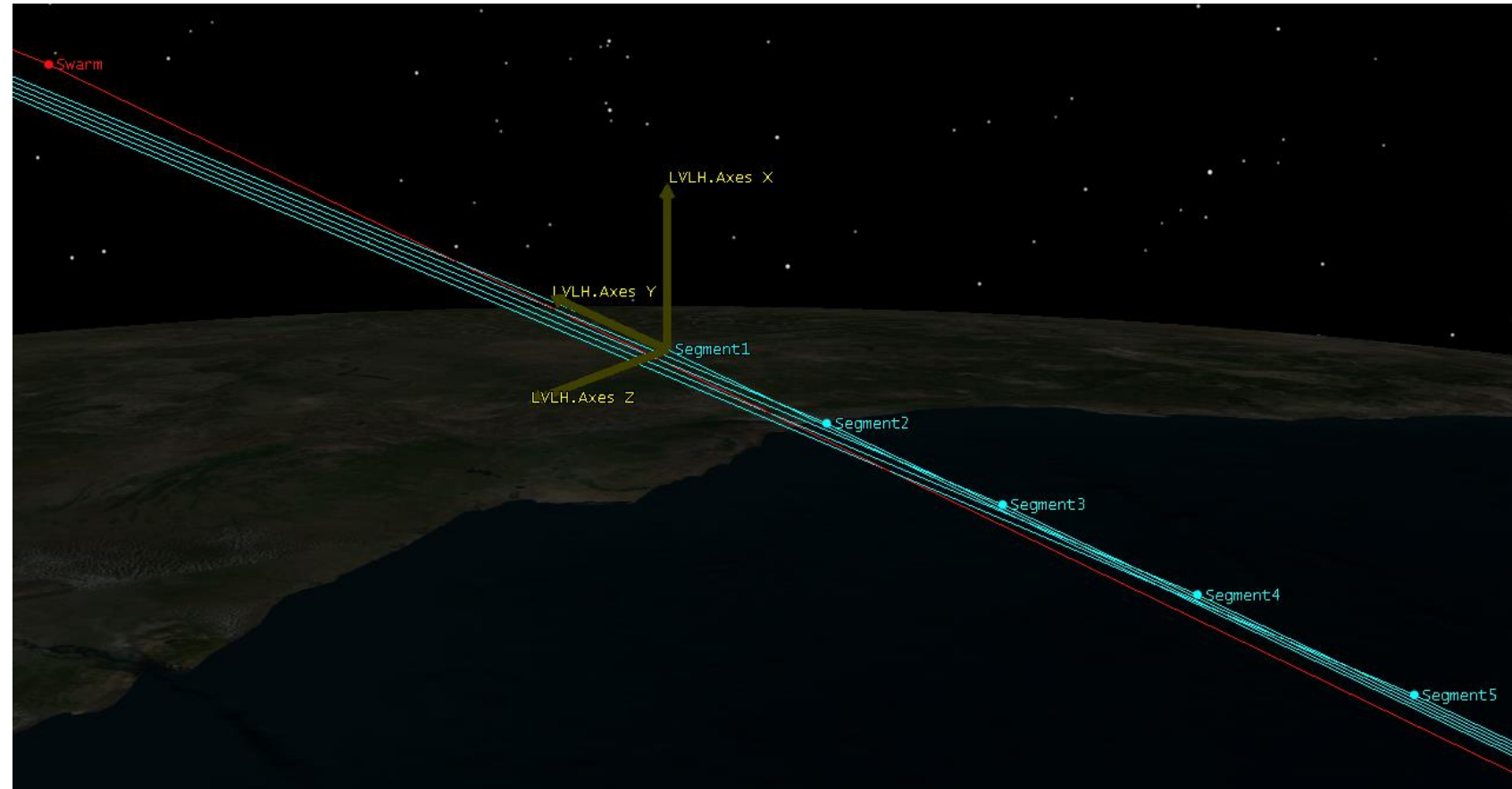
Credit: 20th Century Fox

Initial Conditions



Each ship segment is in a circular orbit at 600km altitude, with a 1km in-track separation between each

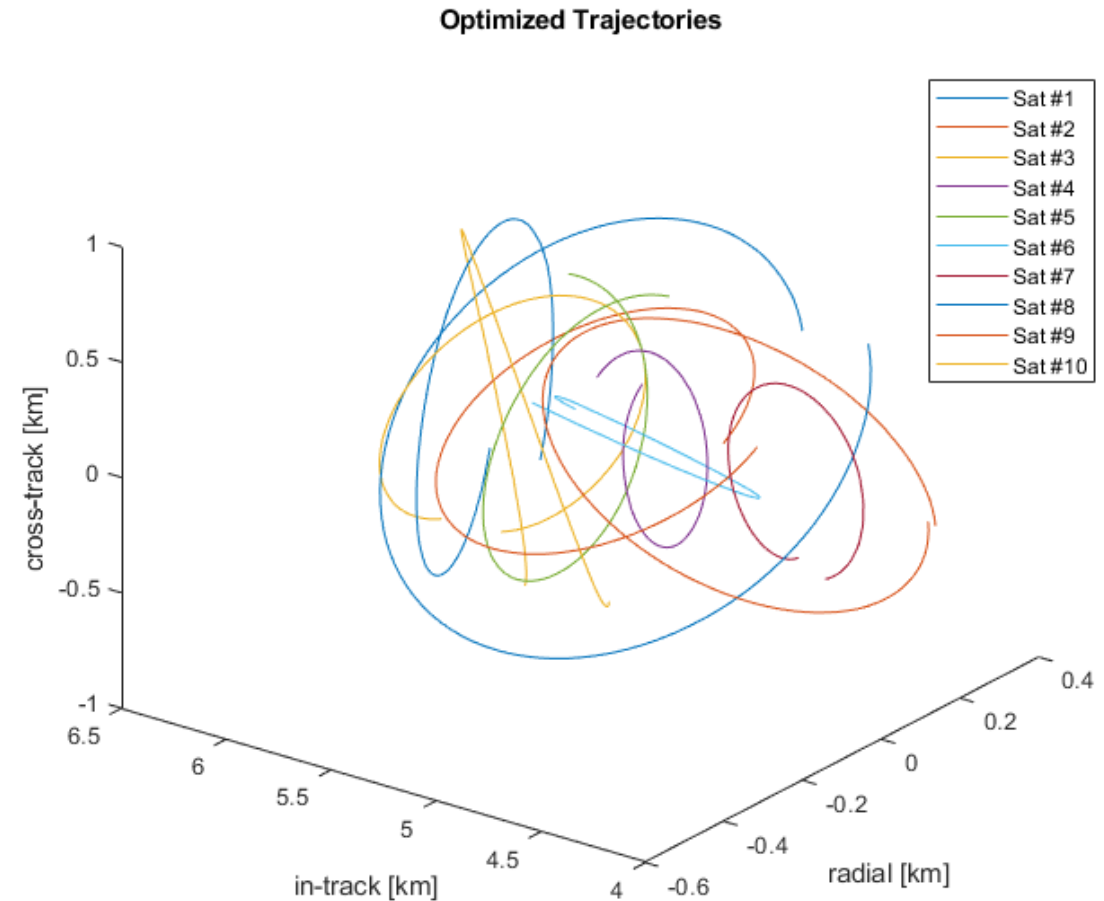
Assembly swarm is initially inserted 5km in-track from first ship segment



Initial Swarm Trajectories



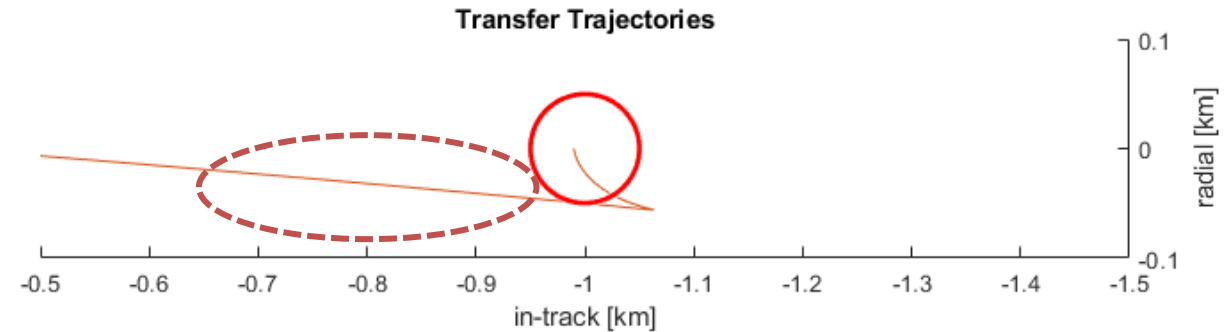
- Initial trajectories for the 10-spacecraft swarm, located 5km from the first ship segment



Transfer Trajectories



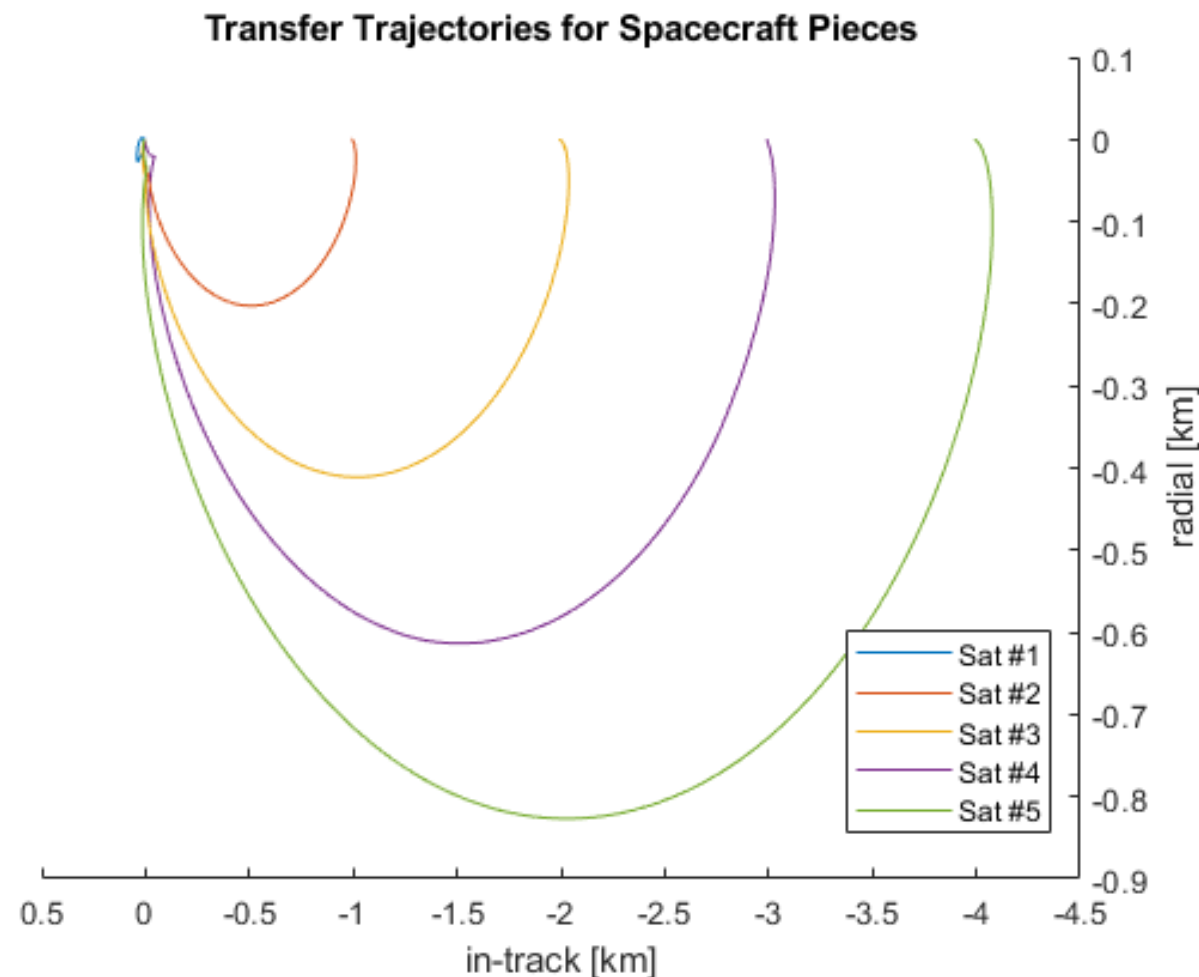
- Five spacecraft break off from swarm to go and rendezvous with the five ship segments.
- Trajectories computed using a two-stage RPO process, implementing a Kalman filter to account for drift and injection errors
- Uses 4.6m/s of ΔV



Transfer Trajectories



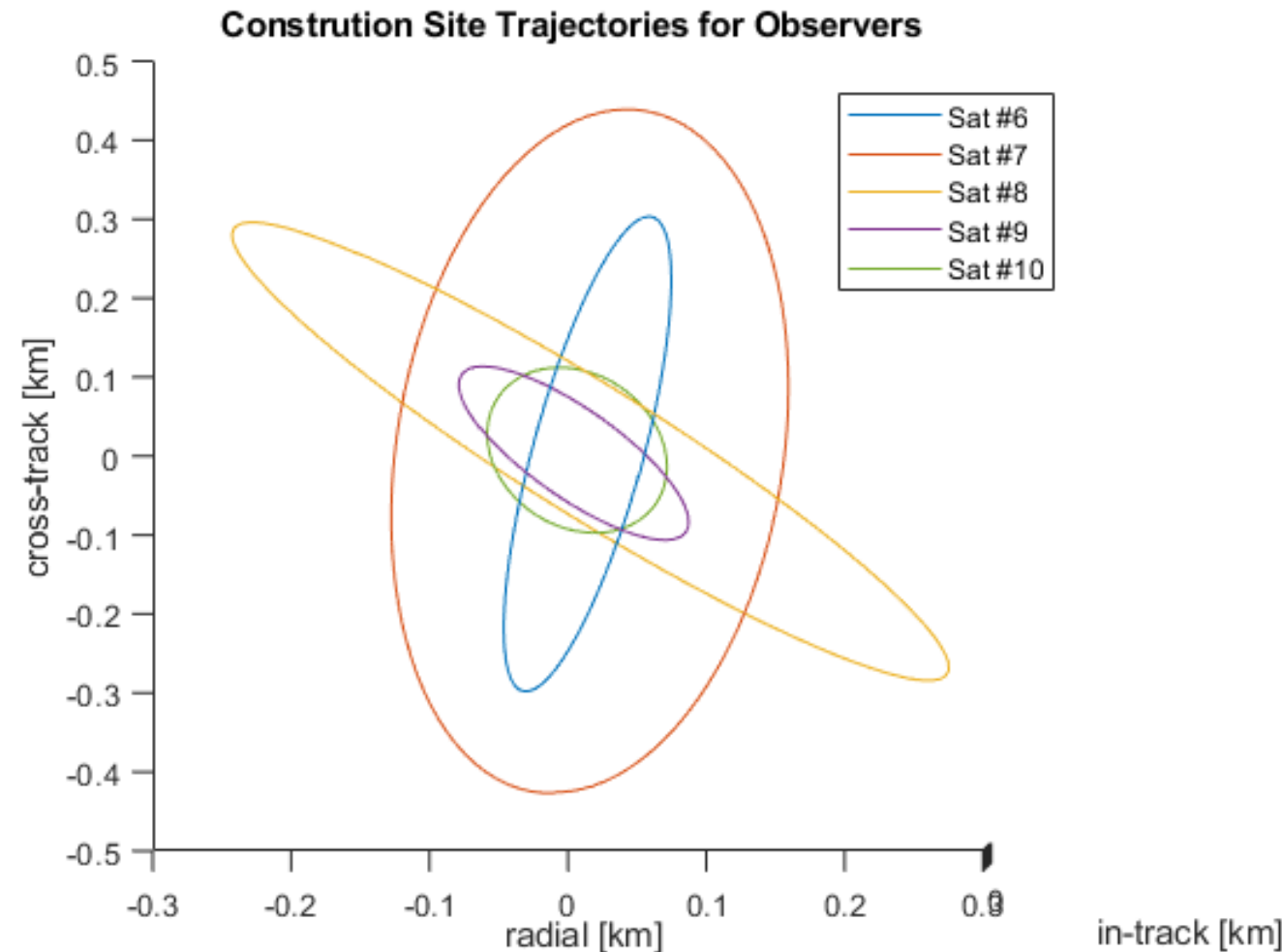
- Upon capturing the ship segments, the following trajectories were plotted to return all the segments to the construction site
- Uses 8.38m/s of ΔV



Construction Zone Trajectories



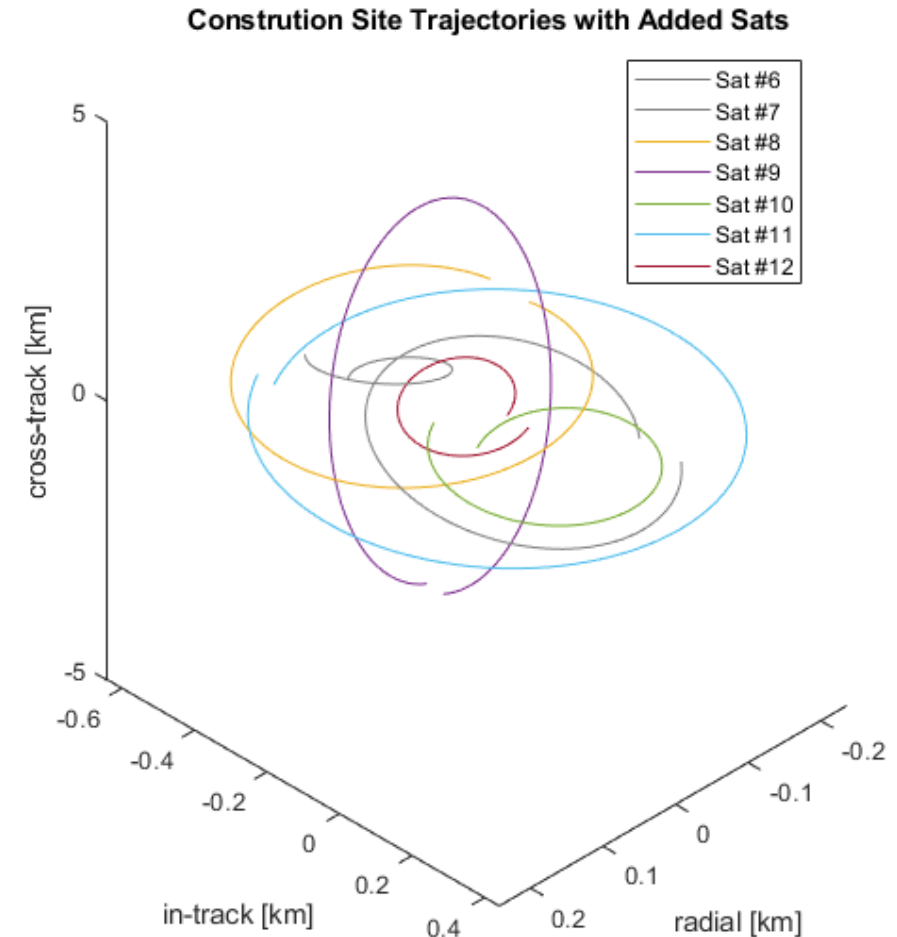
- Meanwhile, the remaining five spacecraft that did not go to retrieve the segments move into observer positions in the construction zone



Death of a Spacecraft



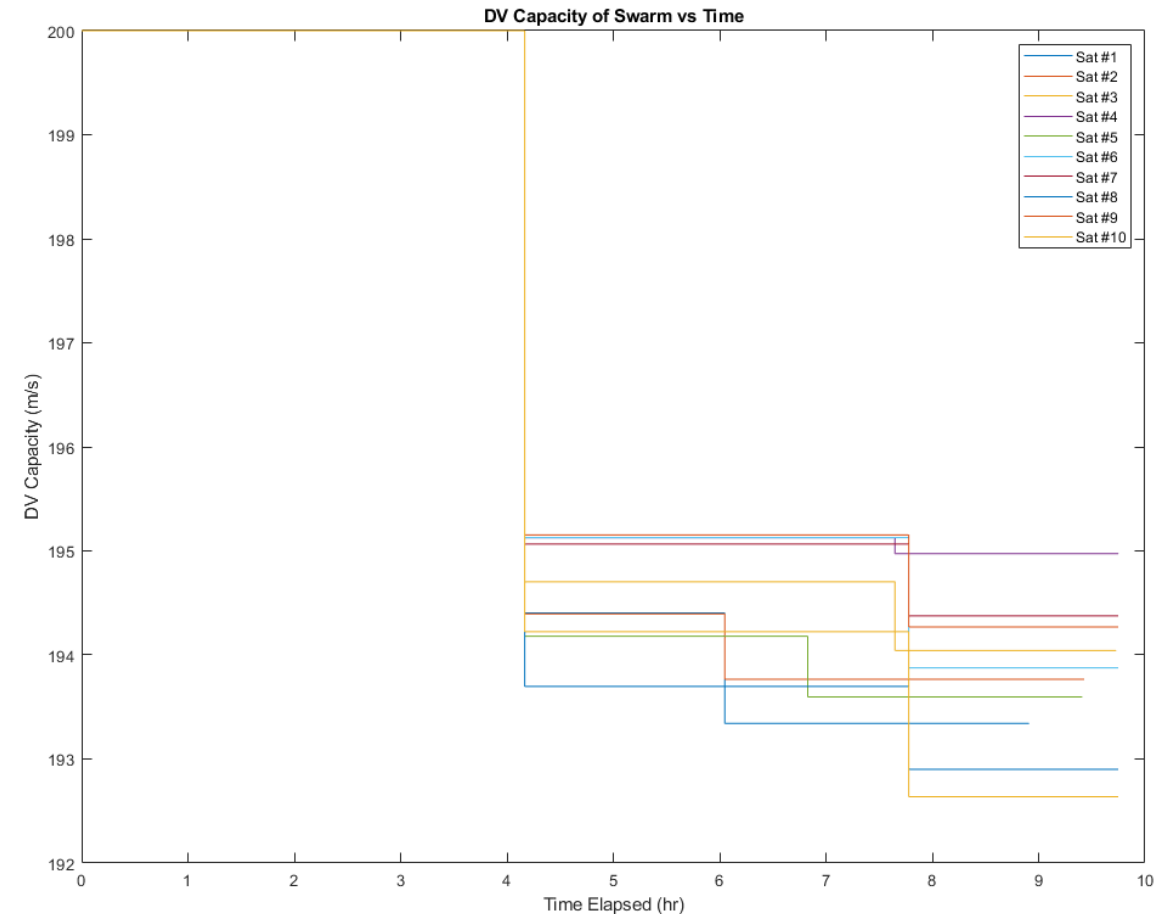
- In this scenario, observer spacecraft #6 and #7 fail unexpectedly. These become zombie spacecraft
- Two new spacecraft are added to the swarm from a holding point several km away
 - Performed using modified GA method, flagging *zombie sat* trajectories as no-fly zones



Trajectory Transfer Cost



- Delta-V capacity drops as the maneuvers are performed to transfer the spacecraft between the orbital trajectories.
- Varies depending on the spacecraft, but maneuvers are designed such that the Delta-V cost is spread out over the swarm



Trajectory Maintenance



- Stationkeeping maneuvers must be performed periodically to maintain these trajectories
 - Account for accumulated errors causing significant drift
- Uses 27.38m/s over 10-day period for this ten-spacecraft swarm
- Swarm GA method is able to account for either chemical or electric propulsion, though chemical is assumed for these simulations due to higher impulse.
 - GEO cases covered later will use EP, using method of patched splines to approximate constant thrust trajectories

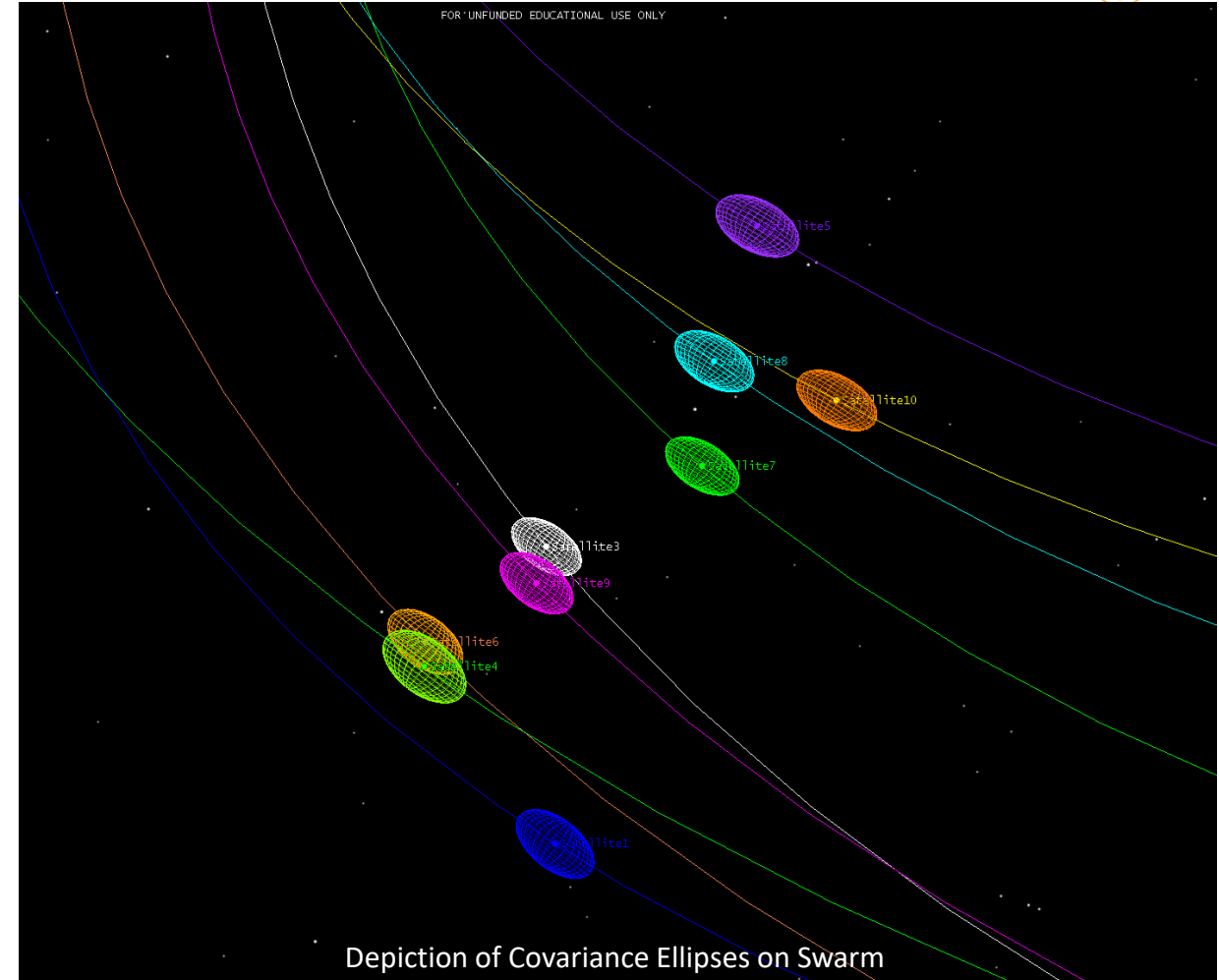


Behavioral Stresses of the System

Consider Edge Cases for Swarm Framework



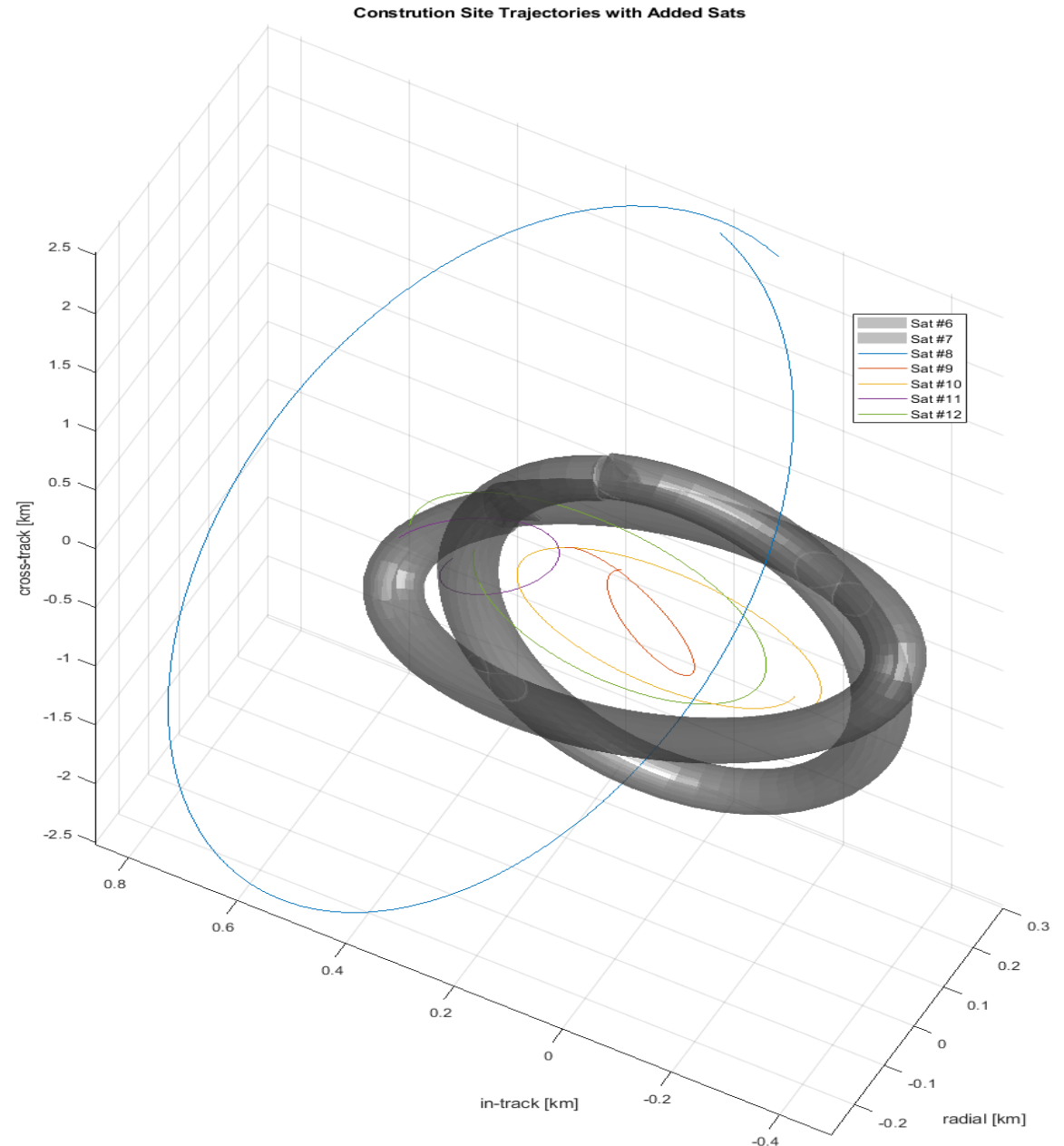
- Probing of edge cases is required to determine the operational regime of these swarm algorithms
- Considers:
 - Unexpected Loss of Vehicle
 - Responses to Dynamic Construction Environments
 - Collision Avoidance Schemes



Unexpected Loss of Vehicle



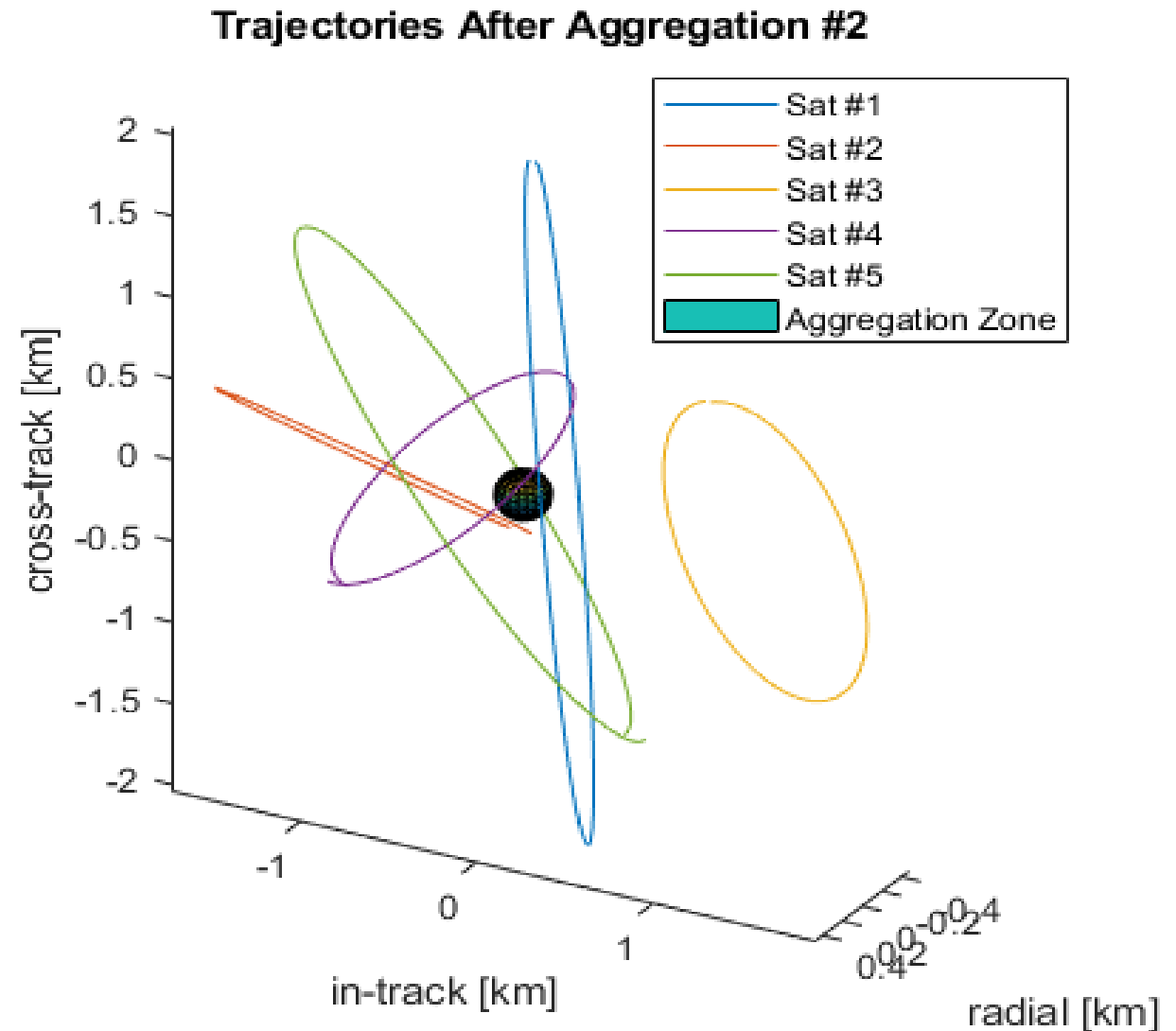
- Offline swarm member = *zombie satellite*
 - Assigned a 50m safety corridor as keep-out zone
- Trajectories are passively safe
 - Short term collision avoidance guaranteed
- Will hinder other swarm spacecraft until resolved or removed



Response to Dynamic Construction Environment



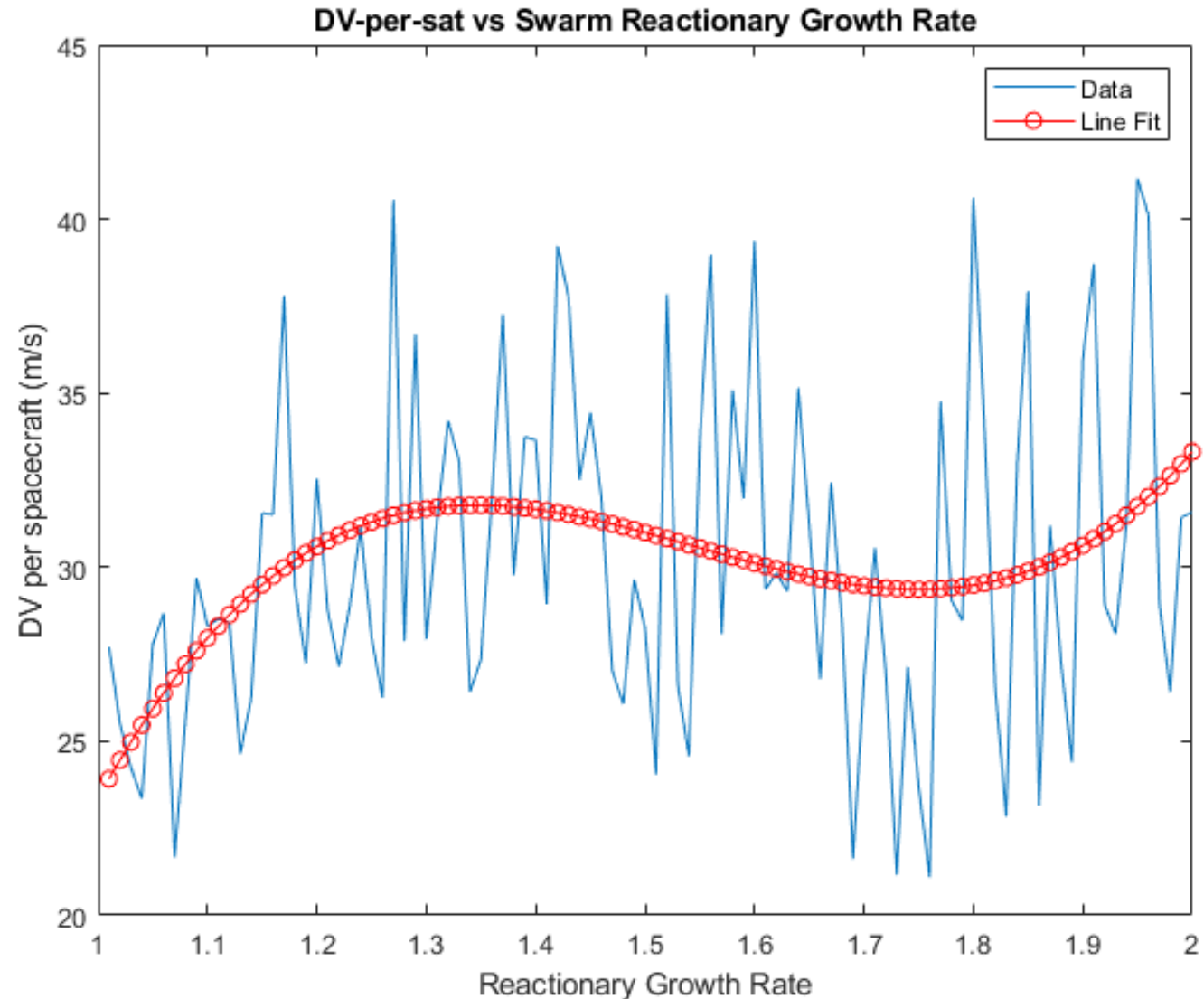
- As structure is aggregated, the swarm will need to steadily grow in size to match
 - Sphere shows keep-out zone for aggregation
 - If structure grows beyond the sphere, the keep-out zone and swarm must expand
- Problem is to determine what growth factor to apply to swarm when the structure exceeds the boundary



Response to Dynamic Construction Environment



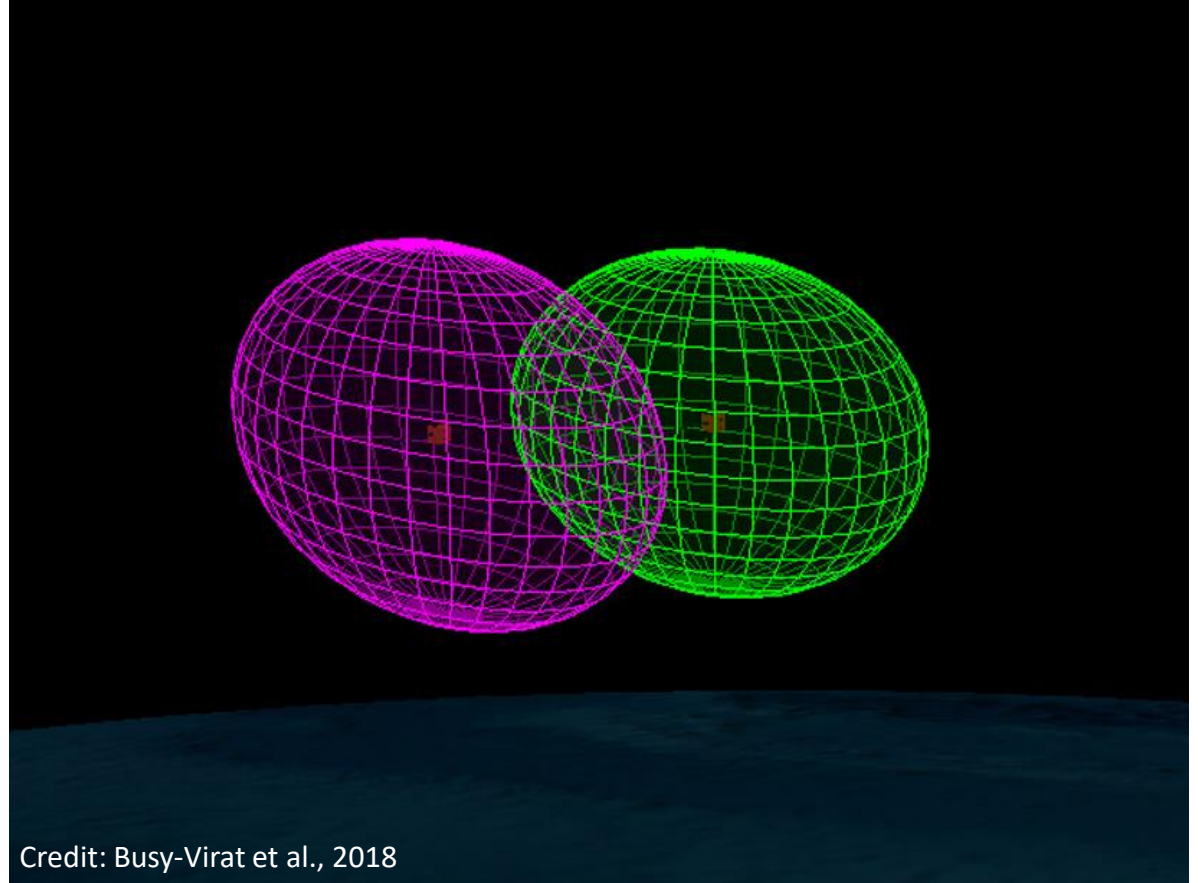
- Simulations were run for growth rates from 1.01 – 2.00 to determine the optimal value
 - Results show that DV can be minimized at 1.01 or 1.75 growth rate
- Discard anything less than 1.2, as this results in excessive computational and operational overhead.
 - $R = 1.75$ is optimal



Collision Avoidance Schemes



- SFKF is great for predicting collisions, but a system is needed to rectify the problem
- A hierarchical set of collision avoidance schemes were developed, expanding on existing systems and practices [9-12]



Credit: Busy-Virat et al., 2018

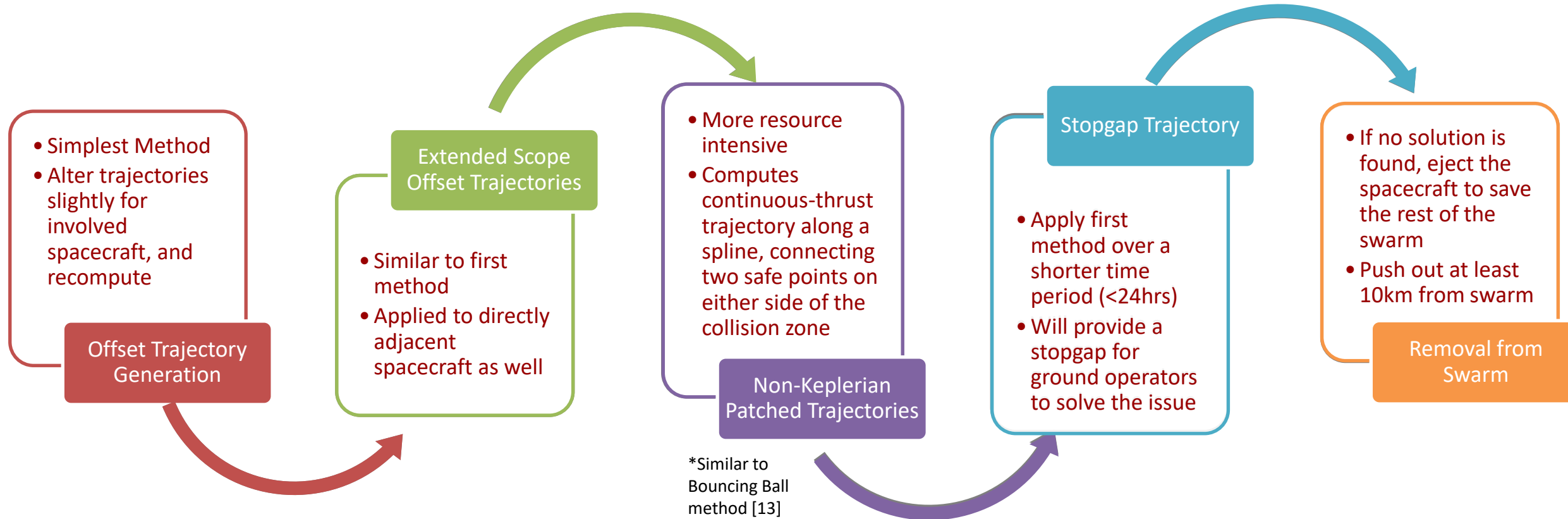
[9] GL Slater, SM Byram, and TW Williams. Collision Avoidance for Satellites in Formation Flight. *Journal of Guidance, Control, and Dynamics*, 29(5):1140–1146, 2006

[10] Uriot et. al. Spacecraft Collision Avoidance Challenge: Design and Results of a Machine Learning Competition. *arXivPreprint arXiv:2008.03069*, 2020.

[11] Ti Chen, Hao Wen, Haiyan Hu, and Dongping Jin. Output Consensus and Collision Avoidance of a Team of Flexible Spacecraft for on-Orbit Autonomous Assembly. *Acta Astronautica*, 121:271–281, 2016.

[12] Noelia Sanchez-Ortiz, Miguel Bell o-Mora, and Heiner Klinkrad. Collision Avoidance Manoeuvres During Spacecraft Mission Lifetime: Risk Reduction and Required DV. *Advances in Space Research*

Collision Avoidance Schemes



[13] Yoonsoo Kim, Mehran Mesbahi, and Fred Y Hadaegh. Multiple-Spacecraft Reconfiguration Through Collision Avoidance, Bouncing, and Stalemate. Journal of Optimization Theory and Applications, 122(2):323–343, 2004



Application to GEO Slot Sharing

Application to GEO Co-location

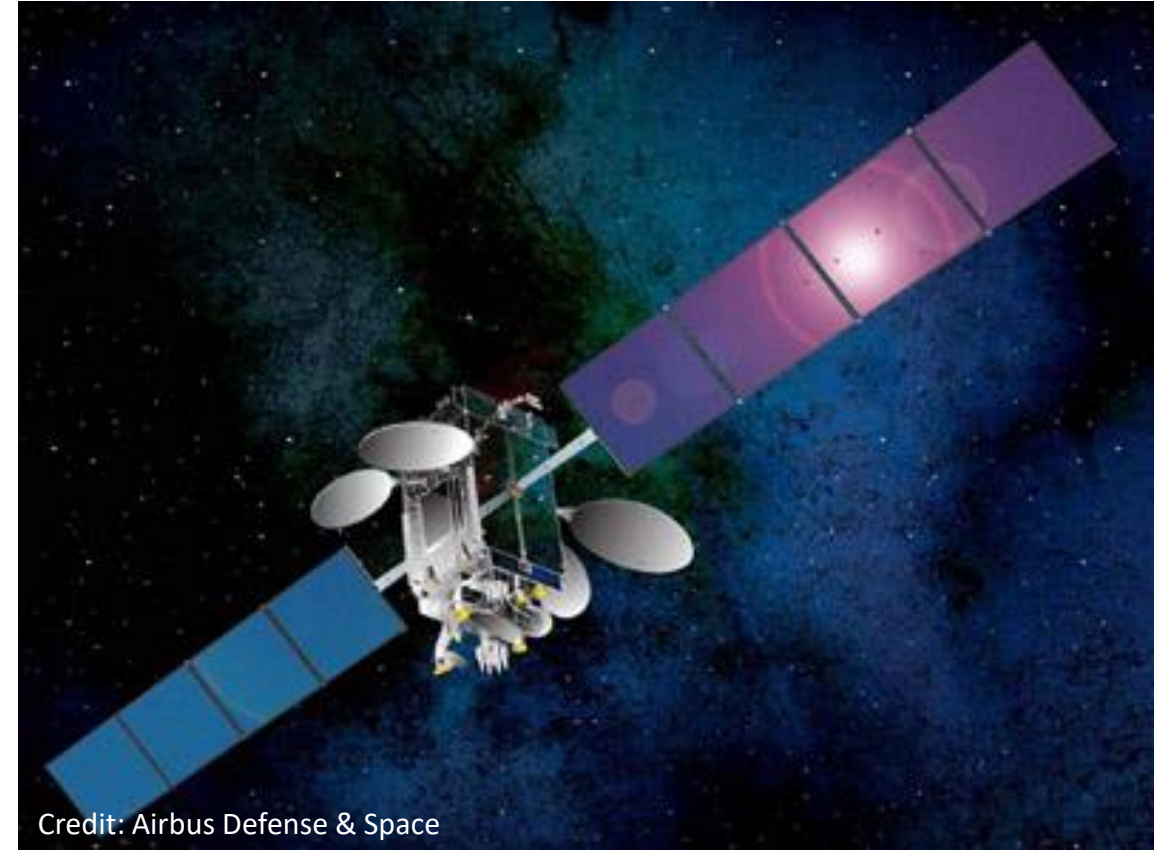


- Although the algorithms were created to solve the problem of in-space manufacturing using swarms of dozens or hundreds of spacecraft, an interesting subproblem was found to be applying this to Geostationary spacecraft sharing the same slot
- Results using 4-spacecraft co-location shows lower ΔV usage than existing literature values for GEO co-location
- Reached out to engineers in industry to get real-world data to test against.

Comparison to Existing Spacecraft



- Received data showing ΔV usage over 1yr of ops for four spacecraft co-located in GEO



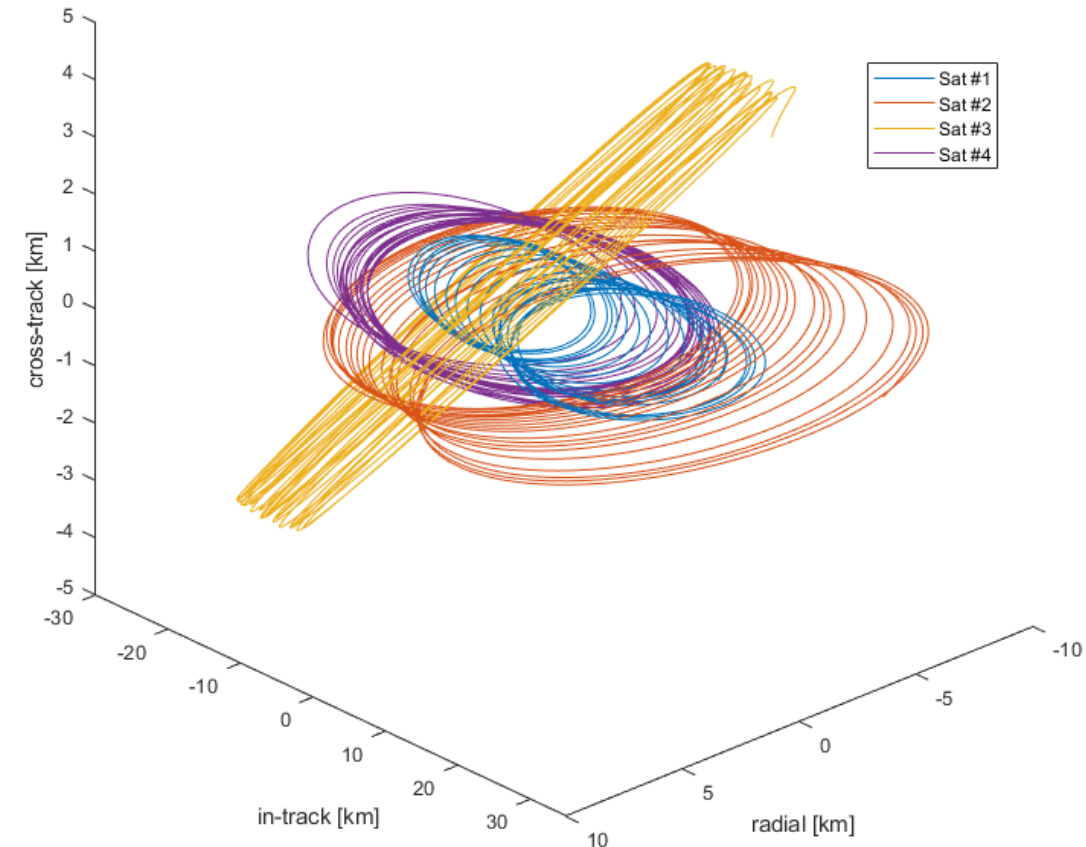
Test Case – 4 SC Sharing GEO Slot



- Comparing to actual spacecraft in same slot
- Real World $dV = 188.12(\text{m/s})/\text{yr}$
- GA method $dV = 166.69(\text{m/s})/\text{yr}$

SC drift slightly within the box, but do not stray outside of half the width/height.

Optimized Extended Trajectories for GEO Slot Sharing





Future Research Paths

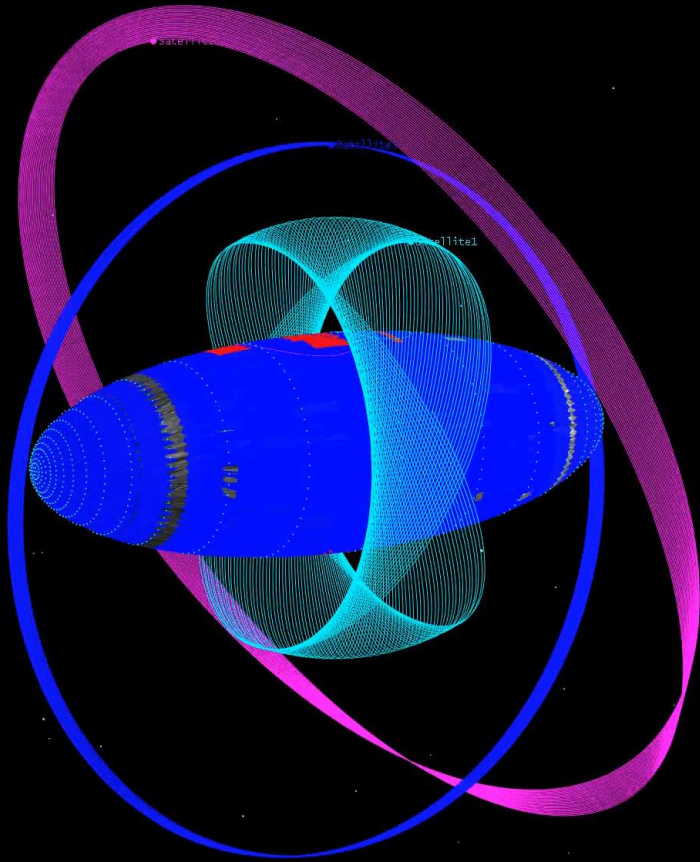
Future Research Topics – Outside of Scope



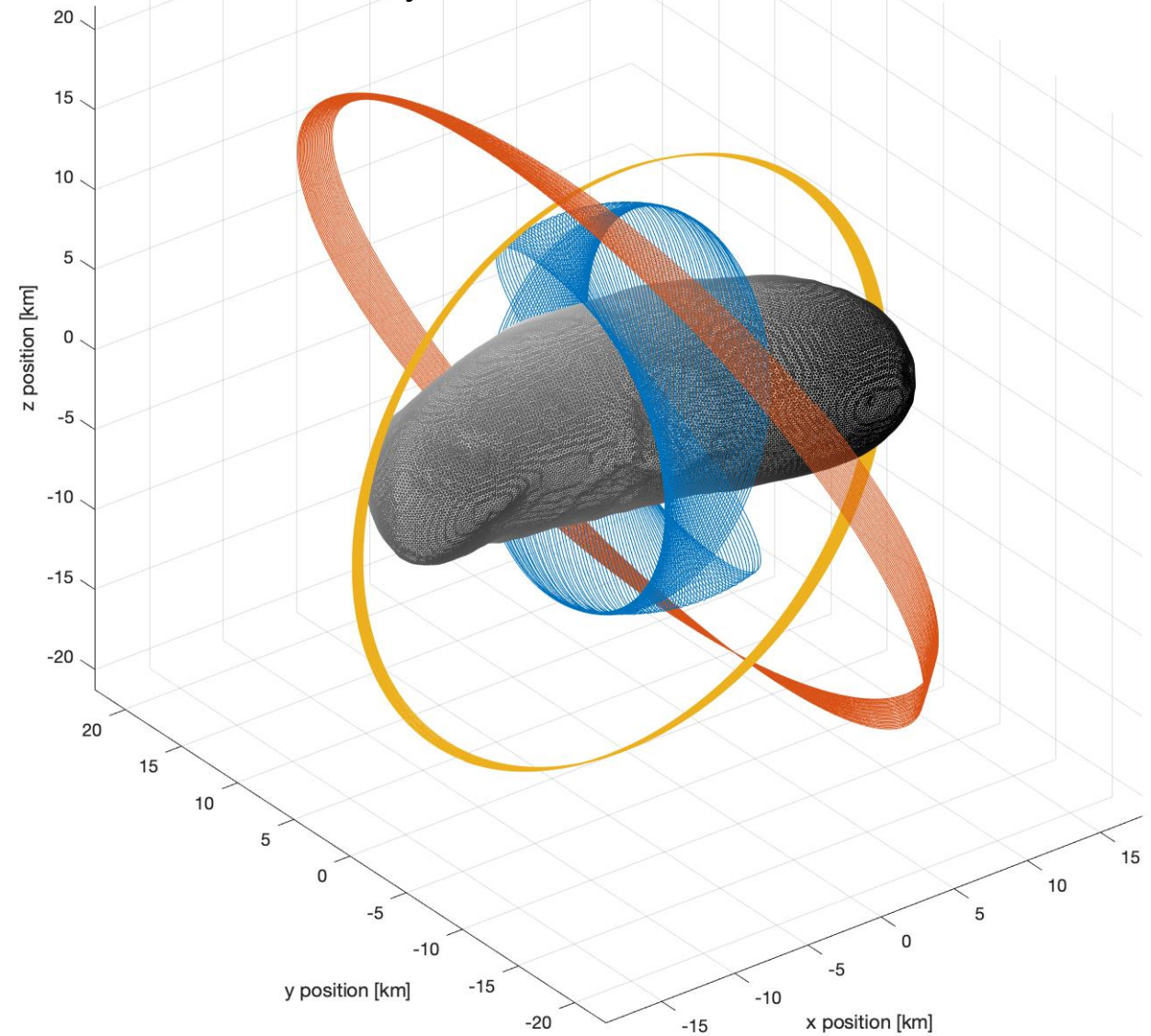
1. Applications to Orbital Reconnaissance
2. Self-Aggregating Swarms
3. Computational Distribution
4. Light-time delay for Autonomous Operations



Orbital Reconnaissance



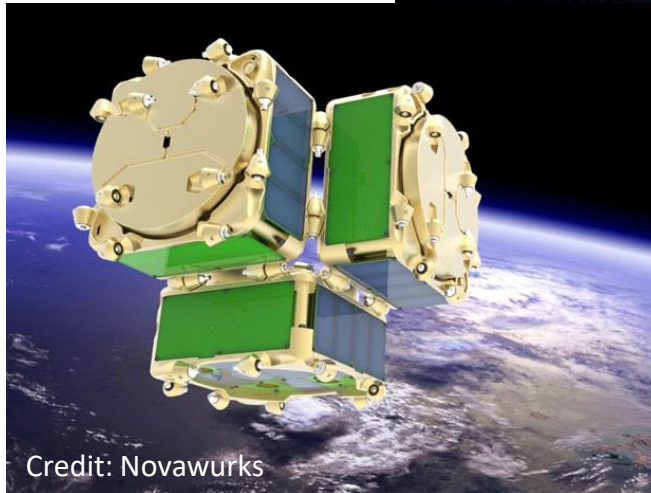
Swarm trajectories around asteroid 433 Eros



Self-Aggregating Swarms



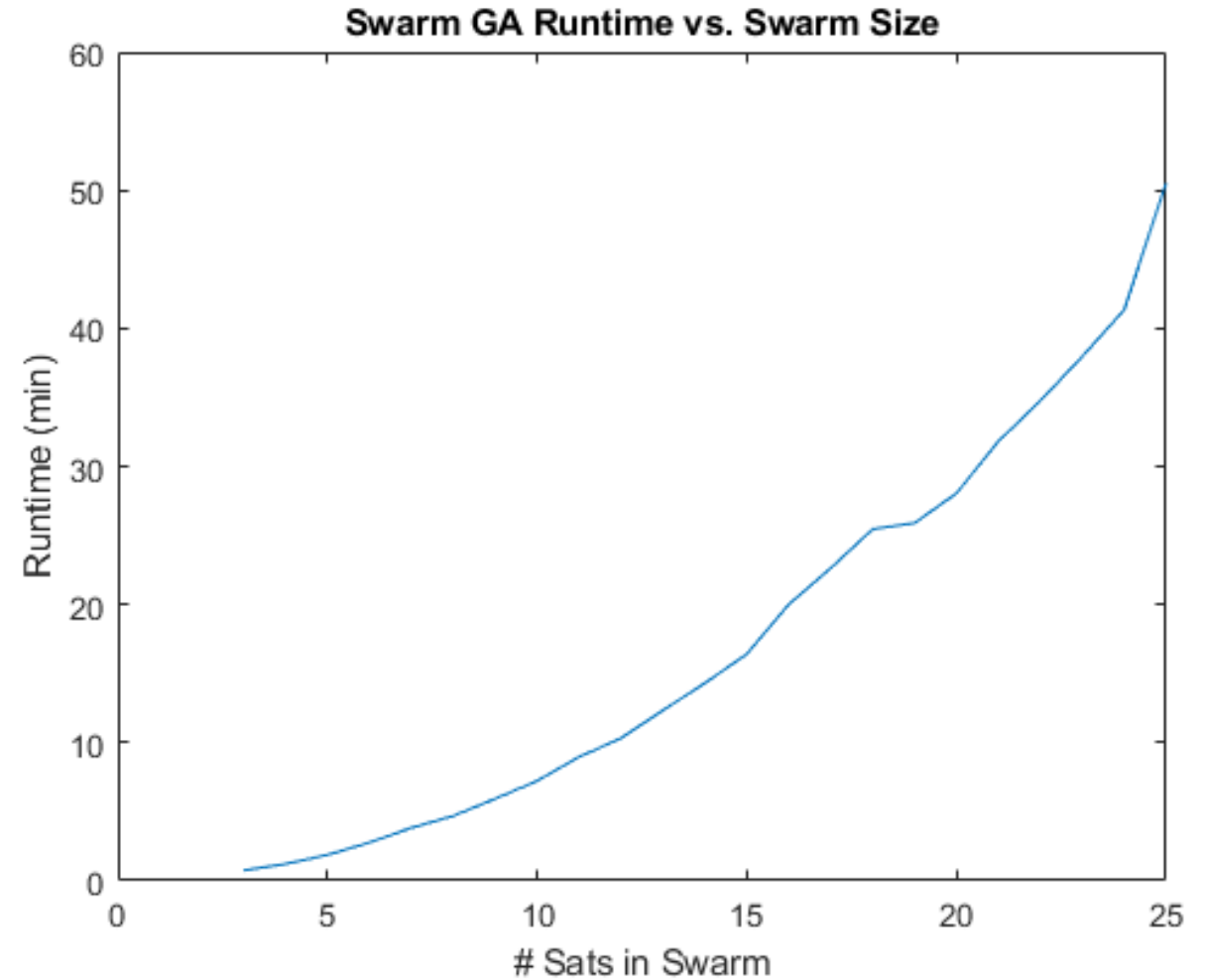
- Similar to swarm in-space manufacturing



Scaling with Number of Spacecraft



- Conjunction de-confliction takes the most wall-time
- Scales as $O(n^2)$
- Runtime also depends on pseudo-random initial conditions
 - Test cases use averages over 100 trial runs for each swarm size



Computational Distribution



- At first glance, computational complexity increases as n^2 (**Eq 1**)
 - Conjunction deconfliction is the most processor intensive task

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)\cancel{(n-2)!}}{2\cancel{(n-2)!}} = \mathcal{O}(n^2) \quad (1)$$

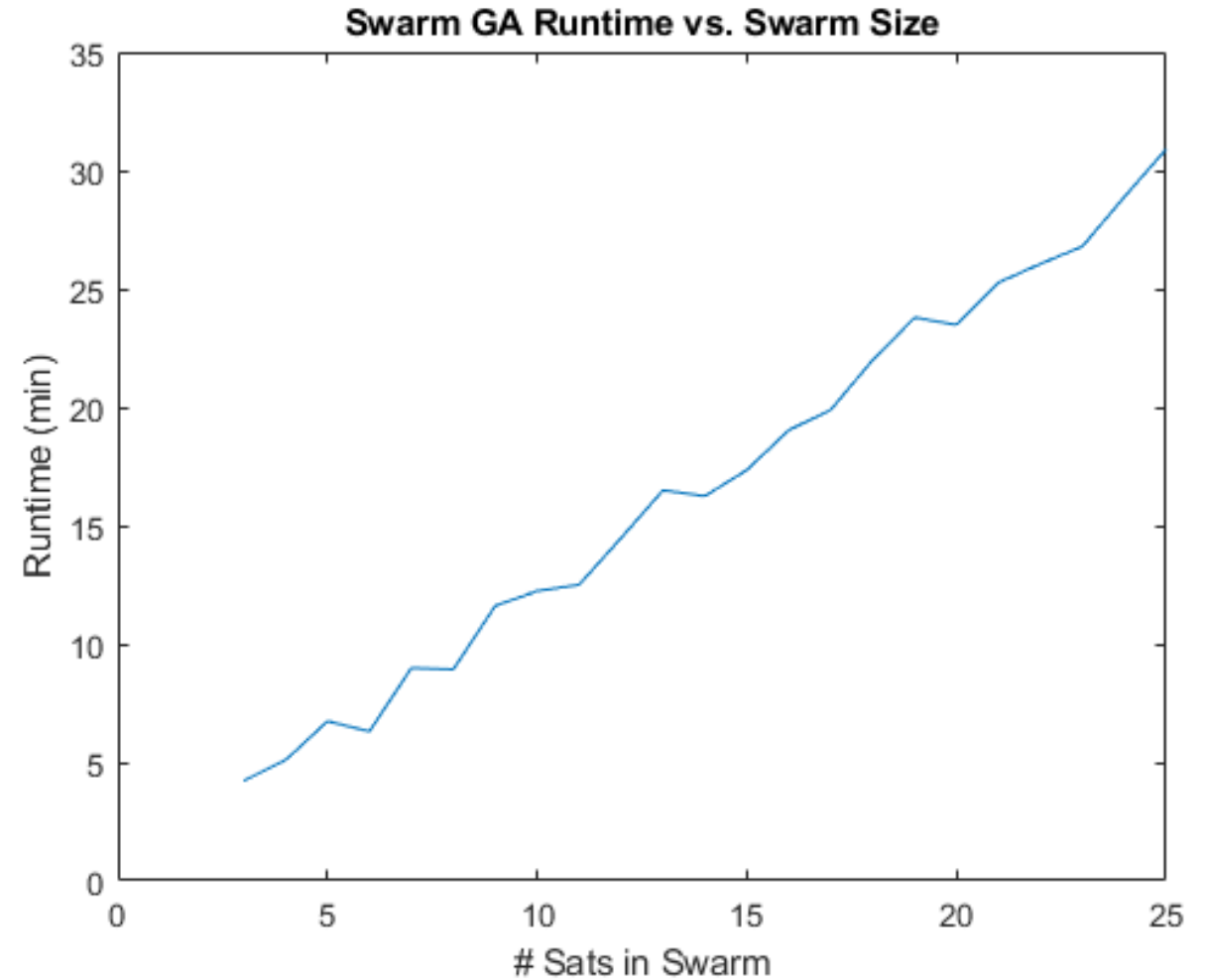
- If parallel computation is used, computing power increases as n as swarm grows (**Eq 2**)
 - Overall computational complexity increases linearly with swarm size (discounting overhead)
 - Requires further research into distribution of computational tasks, and what to do in the event of a node failure mid-compute

$$\frac{1}{n} \binom{n}{2} = \frac{n!}{2!n(n-2)!} = \frac{\cancel{n}(n-1)\cancel{(n-2)!}}{2\cancel{n}\cancel{(n-2)!}} = \mathcal{O}(n) \quad (2)$$

Parallel Computing



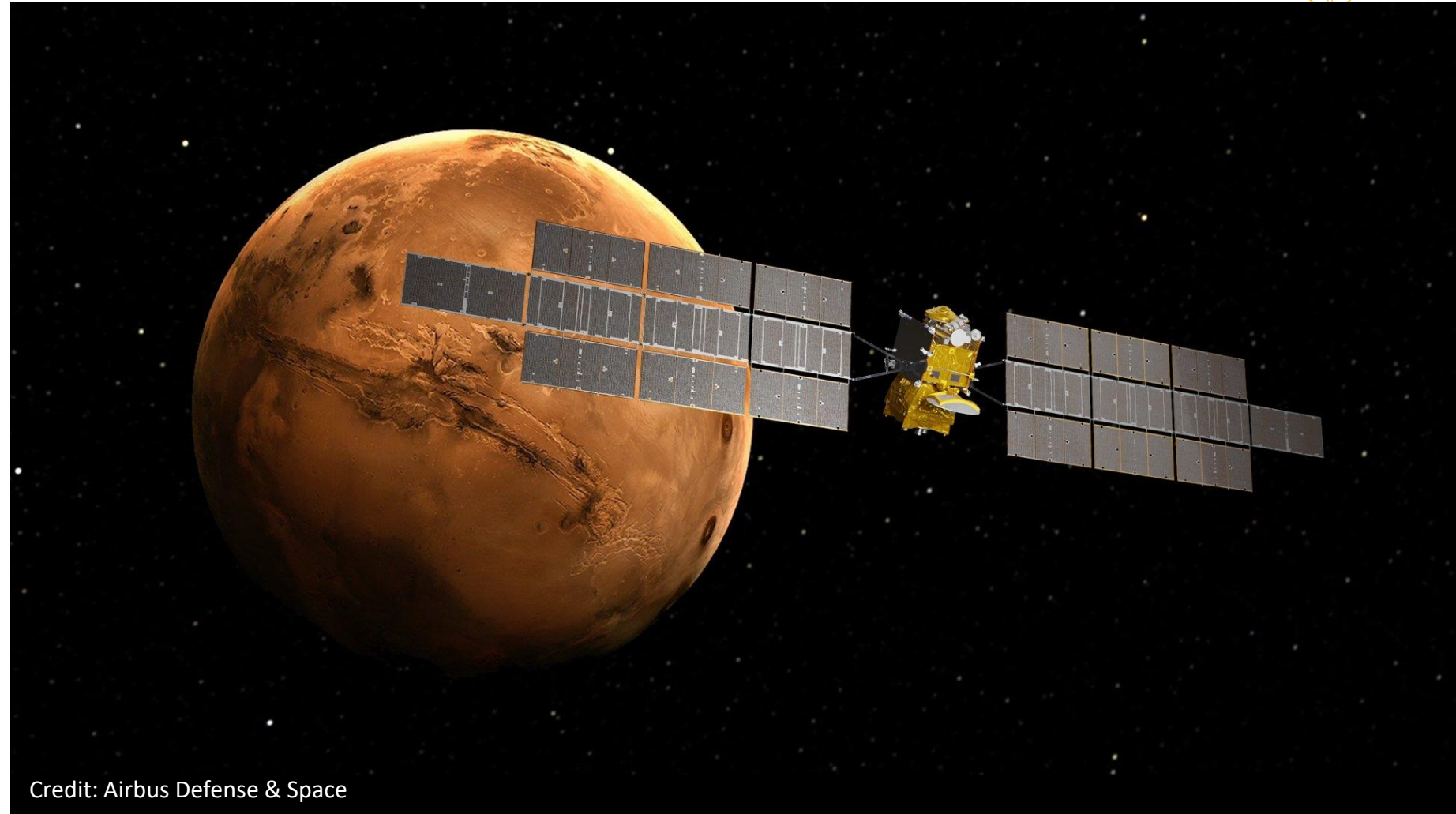
- Conjunction de-confliction takes the most wall-time
 - Sped up using parallel computing
- Slope is 1.15
- Data gathered using USC CARC resources



Light-Time Delays



- If light-delay plays a factor in communications, then the swarm cannot rely on ground intervention
 - Greater autonomy is needed to solve problems on-the-fly



Credit: Airbus Defense & Space



Summary & Conclusions

Summary of Dissertation



1. Created a novel method for swarm trajectory generation using Genetic Algorithms
2. Created a method for optimized aggregation, in close proximity to a growing structure
3. Implemented SFKF for swarm trajectory maintenance
4. Explored scenarios for in-space manufacturing
5. Explored applications to GEO slot-sharing
6. Demonstrated using parallel computing the benefits for on-board computational load balancing
7. Increased the state of the art of swarm ops from Class 2 to Class 3.5

Conclusions



- Swarm configurations can enable in-space manufacturing, while maintaining safe and collision-free trajectories
- SFKF, coupled with the Patched RPO method, can be used to maintain trajectories, accounting for unexpected anomalies
- Analysis of behavioral stresses identified the operational regime of the GA swarm framework
- Example scenarios demonstrated how the trajectory generation and maintenance can be combined to control swarms of spacecraft
 - A novel contribution to the field of Astronautical Engineering



Thank You!



THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

Credit: xkcd / Randall Munroe