CASE STUDY

Proliferated LEO Satellite Optimization

MemComputing, Inc.

MemComputing worked with the U.S. Air Force to develop a satellite design tool that optimizes the tracking & processing of high-speed aerial targets & delivers that information to the warfighter in a timely fashion. Learn about this challenging problem and how our solution is poised to transform current and future satellite capabilities.

Introduction

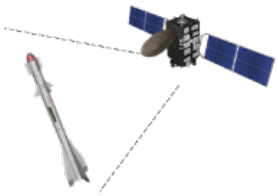
The US Air Force relies heavily on satellite imagery for a large range of combat operations. It is imperative that the processing, accuracy, and dissemination of this data is both rapid and reliable in order to maximize the performance of the warfighter. Although today's satellites are very sophisticated, there are still great challenges when tracking high-speed aerial targets. This is due to varying satellite orbits, communication networks, processing techniques, and time constraints. Ultimately, this hinders the DoD's ability to efficiently react to potential aerial threats.

In this case study, we discuss MemComputing's solution for a Phase II SBIR contract commissioned by AFWERX and performed with the USAF Space Directorate/Space Force. For this project, MemComputing developed a unique software-based solution, the MemLEO-sat Design Tool, for the automated design and optimization of proliferated Low Earth Orbit (LEO) satellites. In this platform, we deploy MemComputing's bleeding-edge optimization methods as well as advanced simulation models for the tracking, detection, and classification of flying targets using state of the art sensors, hardware, and algorithms. This solution also includes a high-fidelity communication architecture that utilizes an efficient MemComputing algorithm to orchestrate a network of satellites to perform optimized distributed communication.

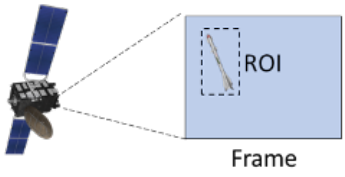
While this tool is specifically optimized for tracking moving targets, with minor modifications it can be used for almost any optimization application required by p-LEO Satellite constellations. A commercial example could be deploying satellite-based internet communications where the client's goal is to optimize the coverage over land masses.

Problem Description

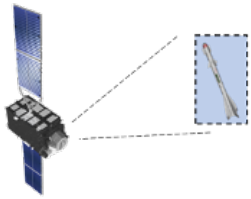
The objective of this project was to develop a tool to optimize the design of a proliferation of hundreds to thousands of satellites operating in a network of LEO (under 2,000 km) constellations to track flying targets, such as hypersonic aircraft or ballistic missiles. A requirement was that the tool must optimize how best to share the image processing methods and computational efforts among/between the satellites and ground computing resources to get information to the warfighter in a timely manner. It must also consider the size of the data communicated by different methods, as well as the communication speeds and bandwidths among satellites and to ground stations. Put another way, the goal was to maximize the probability to successfully detect and track a target, while minimizing the latency of the image processing and communication speed. The tool considers models at different levels of sophistication for data acquisition, detection, classification, and communications as depicted in the figure.



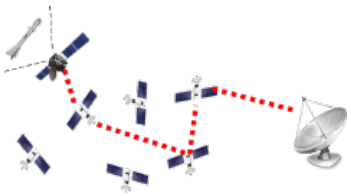
Acquisition



Detection



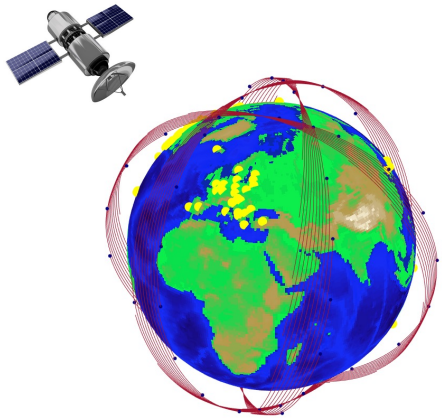
Classification



Communication

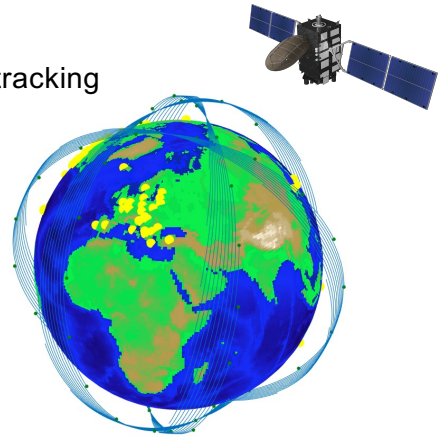
Satellite Roles

Satellites may perform specific roles, and the tool must optimize the number and configuration of satellites in each role. For this project we considered, but are not limited to, three satellite roles based on the tasks they serve: Relay, Primary and Secondary satellites. The primary and secondary satellites communicate directly or through the relay satellite network. The specifications are as follows:



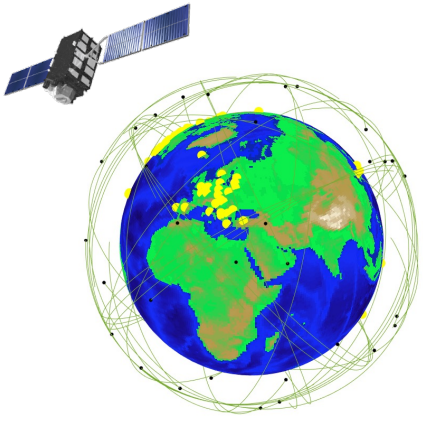
Relay Satellites

- Space-based Communication Backbone
  - Sat to Sat & Sat to Ground Communications
  - Hardware:
    - Laser Communications (Sat to Sat)
    - RF Communications (Sat to Ground)
    - CPU for communications



Primary Satellites

- Low-resolution target imaging and optional target detection & tracking
  - Hardware:
    - Low-re EO/IR steerable sensor
    - GPU (optional)
    - Laser comm (Sat to Sat)
    - CPU for comm



Secondary Satellites

- High-resolution target imaging and optional target classification
  - Hardware:
    - High-re EO/IR steerable sensor
    - GPU (optional)
    - Laser comm (Sat to Sat)
    - CPU for comm

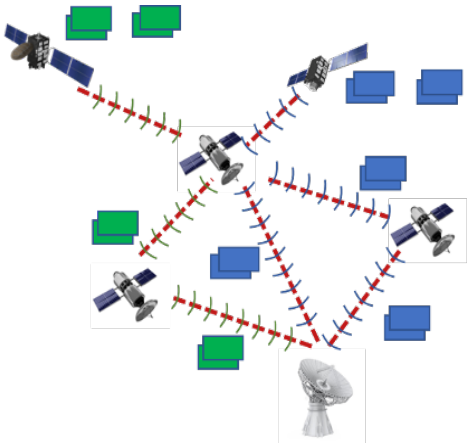
Communication Modes

The following modes of communication are the options that are considered during an optimization run based on the parameters, constraints and objectives chosen.

Tip & Cue

Tip & Cue is where the Satellites will perform most of the image processing and only exchange a small amount of data to ground receivers. This data includes information such as target velocity and direction, an image chip representing the target as well as an update to the catalog of the flying targets and satellite position.

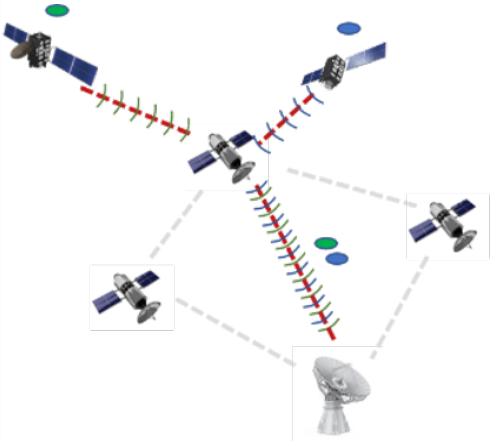
- Pro: Data exchange is small and will get to the ground fast.
- Con: Satellites have limited processing power, so the time to develop the chips can be long.



Full Imaging

Full imaging communications is where the satellites perform almost no image procession and instead send all the captured image data to ground receivers for ground-based processing. Only one channel is used for communications.

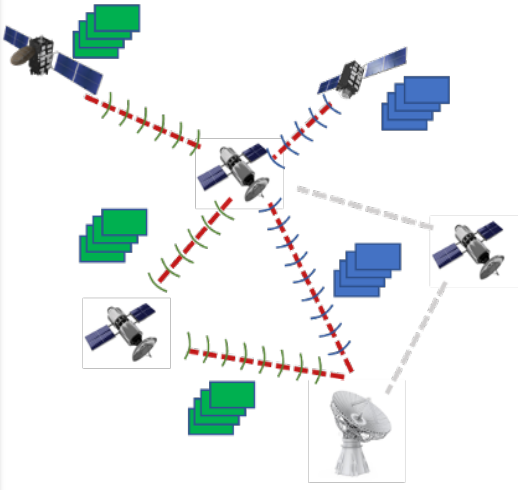
- Pro: There is ample processing power on the ground and thus image processing is fast.
- Con: Sat to ground communications is slow and there could be latencies getting the data to a Relay Satellite within range for ground communications.



Distributed Imaging:

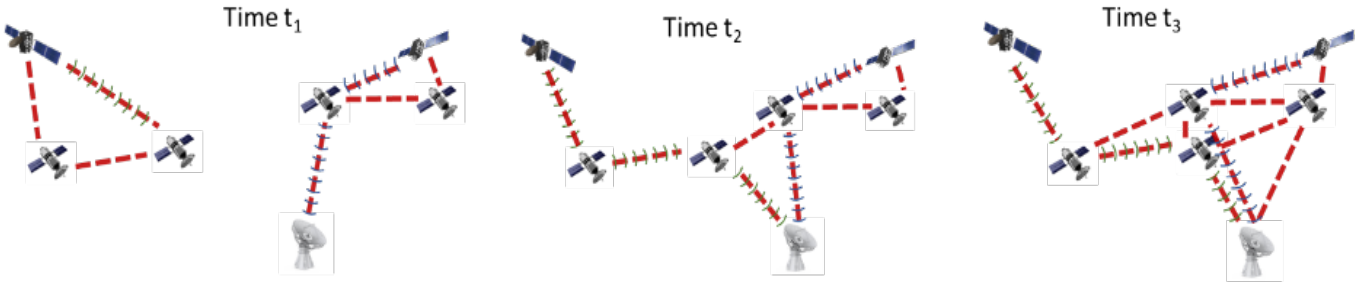
With Distributed Imaging, satellites perform almost no image processing. The satellites send the full image data to the ground. However, the data is broken up and sent over multiple communication channels within the Relay Satellite Communications backbone.

- Pro:
  - There is ample processing power on the ground and thus image processing is fast.
  - Breaking up the data decreases the time required to get all image data to the ground
- Con: Distributed communication needs additional computation to optimally distribute the data packets over the network considering bandwidth, busy channels and temporary links.



Dynamic Communications Network

The MemLEO-sat Design Tool also considers the dynamic aspect of the communication network among and between satellites and with the ground. Satellites are in constant movement. Optimal communications must include the knowledge of available links within the windows of time needed to transmit the data. This ultimately depends on the mutual distance and angle between transmitter and receiver that are in constant reciprocal motion. For example, in the figure below we show a system of 6 satellites (1 primary, 1 secondary and 4 relays) belonging to different constellations and their positions at three different time instants.



The figure highlights how the communication links among the satellites, and between satellites and ground stations change in time, so each link exists only during a limited time window. An efficient communication system must consider this and find the optimal communication channel with temporary links. Our communication model incorporates dynamic links, finite bandwidths (different for each link type), and the implementation of a specialized algorithm to find the optimal channel or set of channels "on the fly" when considering distributed communication. This algorithm also considers when links are already in use or reserved by other satellites. This solution is novel and, to the best of our knowledge, not yet implemented in any current satellite system, where only static links are used. We did not consider static links in this work because they would limit the communication among satellite constellations, which is crucial for proliferated LEO satellites.

Earth-based Communication Receivers

The locations of Earth-based communication receivers must be provided to the MemLEO-sat Design Tool. The design tool assumes that the Ground Stations are distributed within the continental United States, Alaska, Hawaii, United States' possessions, NATO and other allied countries (e.g., Japan, Australia, South Korea). The application also assumes that Ground Stations are interconnected within high performance computing facilities and use high-speed communications securely through the internet, or alternative military networks.

Targets

The MemLEO-sat Design Tool allows for the identification of one, some or many targets for optimization/simulation runs. Targets are currently limited to hypersonic airborne objects such as ballistic missiles and jet aircraft. The starting location of each target, the target's speed, and trajectory are required to fully define a target for an optimization/simulation run.

MemComputing's Satellite Optimization Design & Simulation Tool

MemComputing developed a prototype application that serves three main functions.

MemComputing Blackbox Optimizer

A Blackbox optimizer allows the structure of the objective function and/or the constraints defining the problem to be unknown or unexploitable. MemComputing's Blackbox Optimizer is built with the following objectives:

- Minimize the overall latency related to image processing and data communications.
- Maximize the probability to successfully detect and track a target or targets.
- A future feature would include minimization of costs which considers the cost of different satellites, the cost of deployment of these satellites, etc.

Core to the technology is an electronic circuit emulated by standard architecture capable of mimicking any function. This circuit has parameters that adjust depending on the sampling coming from the Blackbox. The more samples we feed to the circuit the more accurately it can forecast the behavior of the Blackbox. Additionally, we can use this circuit to provide the next input for the Blackbox. This is because we can use the circuit to forecast where the minimum of the Blackbox should be located in the space of the inputs. We use these characteristics to build a sequence of inputs that quickly converges to the minimum of the Blackbox.

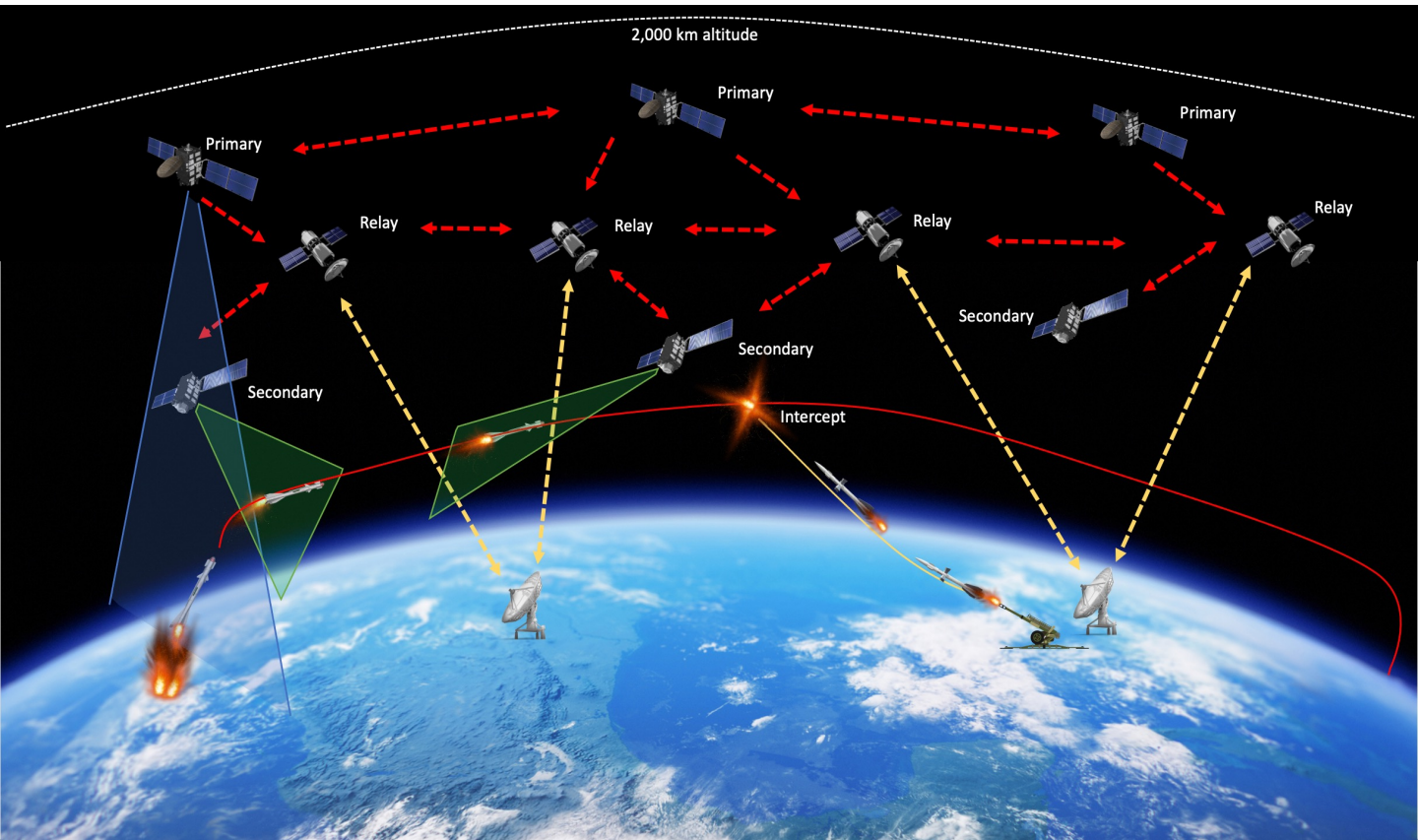
MemComputing Communication Optimizer

The MemComputing Communication Optimizer finds the optimal routing of data communications between and among satellites and to the ground stations. It considers the location of all satellites at any point in time, the relation of the Relay Communication Satellites to ground stations [1] as well as the static locations of ground stations and dynamically optimizes the routing. The routing is computed considering relevant dynamic communication links, finite transmission bandwidths, large data sets, temporarily busy or unavailable satellites and distributed communication.

The tool finds optimal communication channels with dynamic links for multiple constellations supporting hundreds of satellites in an order of milliseconds, even for the most complex computations involving distributed communication. This can be implemented separately, running on common CPUs installed on the satellites and would run in near real time. This solution represents a perfect communication orchestrator that can be deployed on proliferated LEO satellites to greatly enhance current communications system.

[1] Future considerations will allow for airborne communication receivers.

The graphic below depicts the communication architecture between the satellites and ground stations when tracking high-speed aerial targets, such as hypersonic or ballistic missiles. This approach enables the warfighter to react to such threats in a timely fashion.



MemComputing Proliferated LEO Satellite Constellation Simulator

The P-LEO Satellite Constellation Simulator is an environment which simulate the satellite orbits, target trajectories, satellite sensors to detect and track targets, classification systems, and communication systems. The simulation is performed considering all of these aspects at once and for hundreds to thousands of satellites and targets. The workflow is as follows. The Simulator calls the communication optimizer to compute the communication channels. The Simulator is invoked by the Blackbox optimizer to test optimization parameters to find optimal ones. Therefore, the Simulator represents the Blackbox for the Blackbox optimizer. Input parameters for the Simulator include:

- Satellite Configurations
  - Types, Sensors, Hardware, Communications, …
- Constellation Configurations
  - # Planes, Inclination, Altitude, Spacing, sun-synchronous, …
- Computing Strategies
  - A mix and match between Satellite-based processing, ground-based processing or a hybrid model related to image processing and tracking instructions (tip & cue).
- Communication Strategies
  - Tip & Cue, Large Data Transmission – Single Channel, Distributed Data Communication over multiple channels.

A video demonstration of the prototype P-Leo Satellite Constellation Simulator can be found here.

Optimization Parameters

- The MemLEO-sat Design Tool is extremely flexible in terms of tunable parameters. The parameters provided to the simulator can be either tuned and fed by the user or have the blackbox optimizer feed them to test and find optimal ones. The blackbox optimizer can tune parameters like the number and type of constellations to use.
- The configuration of each constellation including: # planes, inclination, altitude, spacing,..
- The number and distribution of satellites for each constellation.
- The configuration of each satellite including: Field of Regard (FOR), sensors, hardware, communications, …

The MemLEO-sat Design Tool can optimize all of them at once or the user can set some of the parameters and let the optimizer tune the rest of them. A detailed set of parameters can be found in Appendix 1.

Case Study

Ground-based Computing vs Satellite-based Computing

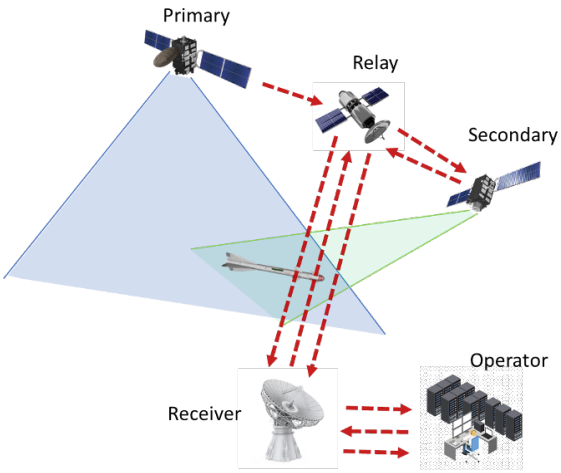
One of the primary objectives of this project was to determine the most effective way to perform detection and classifications of moving targets with proliferated LEO satellites. This helps the AF determine the viability of using LEO satellites for detection, tracking and classification of hypersonic targets. The MemLEO-sat Design Tool has been developed precisely for this reason.

With minor modifications, the tool can be modified for purposes of earth coverage without necessarily identifying and tracking objects. This could be useful for commercial applications such as optimizing land-based coverage to supply internet and satellite television transmissions.

The following sections describe the workflow and performance of the MemLEO-sat Design Tool for detecting, tracking and classifying moving targets while determining the optimal blend (or not) of performing the image processing on the satellites and/or on the ground.

Ground-based Computing Flow Chart

The flow chart reported here summarizes the tasks performed by the satellites and the chain of events when a flying target is detected. In this case the intensive part of the computation is performed on the ground. The satellites stream imaging to the ground and the ground performs target classification as well as satellite orchestration.



1. Wide Area Search - Primary

The Primary satellite has a "wide" FOV sensor with low spatial resolution and the sensor is nadir fixed (possible model for steering sensor is also included)

2. Primary data acquisition

A target moves into the FOV of a Primary satellite, and this initiates the entire tracking sequence

Compression algorithms run

3. Primary sends imaging to the ground

Primary uses the network of relay satellites to send imaging to the ground receivers

4. Processing to track happens on the ground

This initial estimate could be 2D or 3D vector information to be able to project where the target will be by the time the cue gets to a Secondary satellite

5. Ground send cue to secondary

A cue is set through the relay network to the secondary satellites that have the UFP withing their FOR

6. Maintain high-res tracking Secondary sends imaging to the ground

The Secondary satellite senses the target and maintains custody

At the same time, it sends imaging to the ground

Decision?

7p. ground-pass tracks to relay withing the reach

Primaries receiving the info update the "catalog"

Target identified?

Yes

Stop

For optimization purpose this never stops. In fact, not stopping this process gives the provides better estimates of the coverage, comm time and classification time with less simulation recourses.

8. Iterate steps 4 – 7

7r. Relay networks update a "relay catalog" Pass tracks to Primary

8r. Relay catalog is shared with primaries. Discrepancies are sent to the ground to be resolved

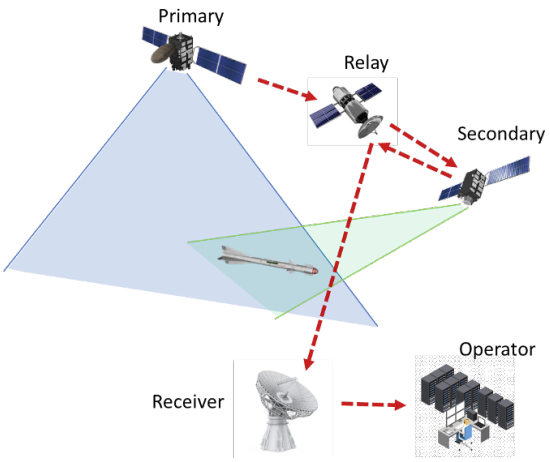
Relay catalog

- The relay catalog is the reference one that periodically is shared with primaries
- It takes info from the secondaries and the ground
- Relays satellites check inconsistencies that can be resolved by communicating with the ground
- The relay catalog is synchronized with all relay satellites
- This is the reference catalog that dictates the stop criterion to classify a Target through a Secondary satellite.

In the figure above, the flow of information is depicted during detection and classification when the intensive part of computation is performed on the ground.

Satellite-Based Computing Flow Chart

The flow chart reported here summarizes the tasks performed by the satellites and the chain of events when a flying target is detected. In this case almost all the computation is performed on the satellites. The satellites tip & cue other satellites and perform tracking, detection and classification of targets and finally send image chips to the ground.



1. Wide Area Search - Primary

The Primary satellite has a "wide" FOV sensor with low spatial resolution and the sensor is nadir

2. Initial target detection - Primary

A target moves into the FOV of a Primary satellite, and this initiates the entire tracking sequence

Target detection algorithms are run to detect the target in the scene

3. Initial track estimate - Primary

For a moving target, obtaining an initial track estimate from the Primary satellite could be important for cueing the Secondary satellite

This initial estimate could be 2D or 3D vector information to be able to project where the target will be by the time the cue gets to a Secondary satellite

4. Follow-up w/ High spatial res sensor - Secondary

The Primary satellite determines which Secondary satellites can follow-up on the target

The task request is passed to the Secondary satellites

5. Maintain high-res tracking & classify/id - Secondary

The Secondary satellite senses the target and maintains custody

A track with low uncertainty and classification of the target is completed

6. Pass tracks to ground/end user – Secondary/Relay

Once a tracking uncertainty threshold is met and the target is classified, track/target information is passed to the ground and/or end user

Decision?

7p. Secondary pass tracks to Primaries withing the reach

Primaries receiving the info update the "catalog"

Target identified?

Yes

Stop

For optimization purpose this never stops. In fact, not stopping this process gives the provides better estimates of the coverage, comm time and classification time with less simulation recourses.

8. Iterate steps 4 – 7

7r. Relay networks update a "relay catalog" Pass tracks to Primary

8r. Relay catalog is shared with primaries. Discrepancies are sent to the ground to be resolved

Relay catalog

- The relay catalog is the reference one that periodically is shared with primaries
- It takes info from the secondaries and the ground
- Relays satellites check inconsistencies that can be resolved by communicating with the ground
- The relay catalog is synchronized with all relay satellites
- This is the reference catalog that dictates the stop criterion to classify a Target through a Secondary satellite.

In the figure above, the flow of information is depicted during detection and classification when the intensive part of computation is performed on the satellites.

Classification Time Models

The time it takes to classify a target depends on many factors, for example the algorithms and method used, the hardware employed, the resolution of the images, etc. In this project, we have focused on a generic model that implicitly considers (i.e., set as a parameter by the user) the dependency on the hardware, while we have developed a more self-consistent treatment of the dependency on the image resolution and algorithm timing. Specifically, if there is a certain type of sensor on a satellite, then the resolution of the image will depend on the altitude, weather, and lighting conditions, therefore the time to classify will be a function of the image resolution.

Basic Model

Ideally, the model can achieve higher fidelity and include additional features such as actual track managers, probability of detection, and estimation uncertainties. However, here we describe a more basic model and in the next section, an evolution considering the loss of information given by the definition of the images.

The idea is to allow time for the secondary satellites to provide a high-quality track and classify a target weighted by the Ground Sampling Distance (GSD). Ideally, this is a function of altitude to make it easier but the sensor is steerable so the GSD will be different off-nadir. To simplify the problem, the weighting factor can be the max of the horizontal GSD and vertical GSD.

The idea with this addition is to factor in the dependence of having more pixels on target to enable better tracking and classification, thus reducing latency. By weighting the problem to increase the number of pixels on target for the secondary satellite, this would drive the altitude lower in the optimization process.

A similar weighting can be applied to the primary satellite, but the scaling factor would be different because the primary satellite is not trying to resolve the target – there is still value for the primary satellite to have more pixels on target.

Advanced Model

The classification time depends linearly on max($horizGSD$, $vertGSD$) if the altitude range considered is small enough. However, for a large range, a different model should be considered. To define a more realistic dependence between the classification time and the GSD, let us consider the process that is used to classify. The problem is that the resolution decreases with the altitude, and the GSD is roughly proportional to the altitude for the ranges we are considering. When the resolution is low, it is a common practice to use a large number of images of the object and implicitly (using an artificial neural network) or explicitly (creating a super-resolution image from multiple images). However, to arrive at a given resolution, the number of images needed does not grow linearly with the resolution difference. In fact, there is a loss of information because the redundancy of images tries to reconstruct a single image from the correlations among images. The process is similar to reconstructing Shannon information loss because there are less pixels in the single image. In this scenario, the Shannon information loss is proportional to the ratio between the number of pixels used to cover the object at two different resolutions. The number of pixels is given by

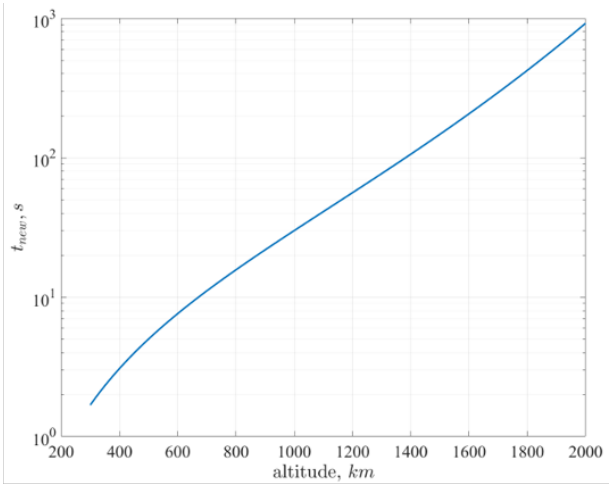
$$n\_pixel \sim A\_\text{Target}/(horizGSD \times vertGSD)$$

Where A target is the area of the target. To reconstruct the information loss in this context, one must consider a decoding process. A full analysis would be good for further studies, especially to decide the actual machine learning method for enhancing the resolution. To this end, the Shannon and the Nyquist-Shannon theorems can be used. However, for this project we just consider the worst case in which the mutual information between two image resolutions is exponentially decreasing with respect to the information loss. So, the Shannon theorem will imply that the decoding algorithm complexity would grow exponentially. In our case, using several image samplings grows exponentially. So, a good model for this scenario is

$$t\_new = t\_0 + t\_l\ (e\text{^}(k \times horizGSD \times vertGSD) - 1)$$

Where t_0 can be set as the ideal time to classify the object at distance 0, i.e., the ideal situation where the sensor is so close that we have infinite resolution, therefore $horizGSD \times vertGSD \sim 0$. If the frame rate of the sensor is for example 30 fps, then

$$t\_0 = 1/30\ s.$$



Advanced Model Continued

We can also numerically evaluate t_l and k requiring t_new for two different values. For example, an estimate can be given assuming t_new=5s at 500km (GSD=1m) and t_new=30s at 1000km (GSD=2m). These numbers are reasonable since at 500km we have GSD=1m, then t_new=5s means that we use 150 frames to recognize the TARGET which should be more than enough if the TARGET covers an area of 50-100m2 since 50-100 pixels would cover the area.

At 1000km, t_new=30s can be justified because a perfect code that recovers the loss of information would be quadratic in the GSD, and therefore at 1000km, t_new would ideally be 20s. However, we heuristically increase this by 50% to consider the non-ideality. A 50% increase is reasonable because it implies that with a GSD = 2m we are still able to make a reasonable classification of the TARGET even if this would take 900 frames.

Therefore, solving numerically (we used the Newton algorithm to find zeros of the nonlinear vector field) for t_l and k assuming t_0=1/30s, t_new=5s at 500km and t_new=30s at 1000km we have

k=0.24852
t_l=17.605

Primary Satellite – FOV Model

The Field of Regard (FOR) vs Field of View (FOV) is a sensitive parameter that strongly impacts the coverage of the satellites. However, the larger the ratio of FOR / FOV, the longer the primary satellites will take to track everything in the FOR. This does not happen for secondary satellites as they just track and classify one object at time, so they do not scan all FOR but just the FOV. For this reason, it is best to have a model that considers this optimization for the primary satellites.

A basic but effective model is to consider the time t_ROI to define a region of interest on the image of the primary satellite depending on the FOR/FOV ratio, and in particular

$$t\_ROI = k\text{" FOR/FOV"}$$

With both FOR and FOV evaluated in steradians. A reasonable value for k is order of 5 to 10 seconds.

Objective Function

The objective function is a function of the outcome of the simulation of the satellite constellations. In this case study, we optimize:

a) the average time to transmit information (imaging, target position, catalog, etc.)

b) the average time to classify TARGETs and

c) the probability of successfully detecting, tracking, and classifying targets for a given scenario

We optimize all parameters characterizing the constellations (# constellations, # planes, # satellites per plane, plane inclination, altitude, periapsis) and the variable primary FOR.

These parameters are constrained (for example the total number of satellites is 150 for this case study) and bounded (each parameter has its own ranges) and they can be integers and continuous parameters. The objective function reads:

$$obj = (t\_comm + t\_class + t\_ref) / (p\_success\text{^}\alpha + p\_ref\text{^}\alpha)$$

where t_comm is the time to communicate, t_class is the time to classify, t_ref is a threshold time (i.e., any $t\_comm + t\_class$ below t_ref is already good enough and therefore more importance is given to the denominator of obj); p_success is the probability to successfully detect, track and classify a target and α>1 is a parameter to weight $p\_success$ vs $t\_comm + t\_class$ in case the time does not go under t_ref. Finally, p_ref is a threshold probability (i.e., any $p\_success$ below p_ref is already good enough and therefore more importance is given to the nominator of obj if not below t_ref) that also prevents a vanishing denominator.

This objective function is well posed based on definition of probability of successfully detecting, tracking, and classifying p_comm, classification time t_class and communication time t_comm.

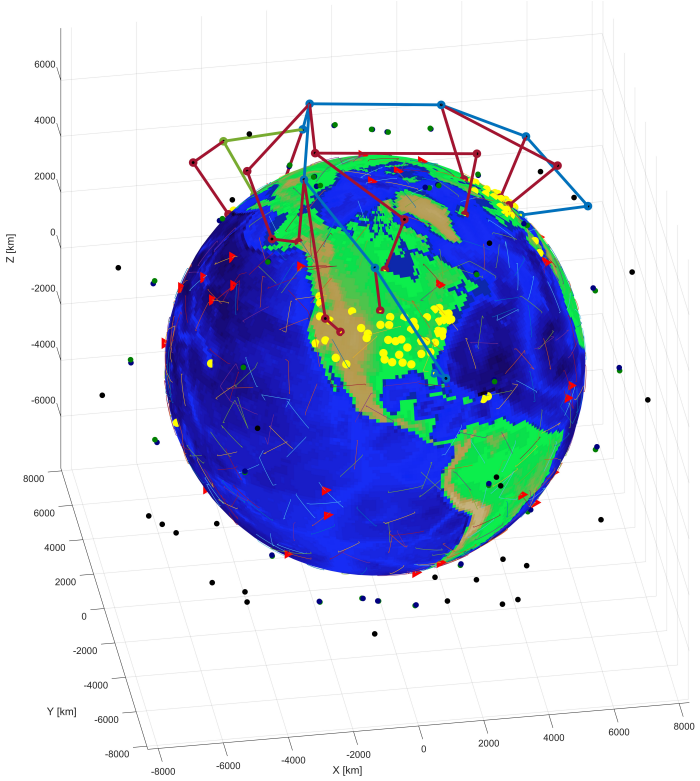
Results & Discussions

The MemLEO-sat Design Tool supports a limitless number of possible scenarios. During the Phase II SBIR process, several different scenarios were considered. These included a mix of the number of satellites, different types of satellites and hierarchies, parameters to optimize, different models for the simulator and much more. For this case study, we consider the use case where we compare figures of merit to assess the efficiency and pros and cons of LEO constellations including Primary, Secondary and Relay satellites. The configurations we report here consist of studying the cases of a) intensive computations performed on the satellites, b) intensive computation performed on the ground and communication through a single channel per satellite c) computation performed on the ground and communication distributed through multiple channels. These scenarios have been discussed in the previous sections.

MemLEO-sat Design Tool Outcome

The MemLEO-sat Design Tool was set up to find optimal parameters reported in the tables below, for the three different scenarios discussed in the previous paragraph. Further constraints that were used are:

- The total number of satellites is limited to 150
- At most, we allow two constellations for each type of satellite (if the outcome of the optimizer returns 0 satellites in a constellation, it means that constellation was not used in the optimal solution)
- The constellations are walker type, and one has sun synchronous orbits, and the other does not.
- The parameters have been optimized using a scenario in which 40 ballistic hypersonic (e.g., Mach 10) missiles distributed at random fly for 500 seconds in random directions and then restart from another random location with another random direction. They do this repeatedly for the simulated time, which is set at 2 hours.



This example represents 150 satellites tracking 40 hypersonic targets.

- Red Triangles Represent the Targets
- Black Dots represent communication Relay satellites
- Red Dots are the Primary Satellites that first identify the targets (and continue to monitor)
- Green Dots are the Secondary Satellites that track the targets
- As the satellites orbit, control is tipped and cued between satellites of the same type to maintain tracking.
- The blue lines show the communication path among satellites getting information to the closest satellite over a ground link, with that communication to the ground.

Results Continued

The outcome of the MemLEO-sat Design Tool can be visualized as reported in the figure on the next page. The figure is generated using the outcome of the case b) (intensive computation performed on the ground and communication through a single channel per satellite). For the sake of conciseness, we report and discuss this visualization only for the case b). For the case a) and c) the final results (i.e., the optimal parameters returned by the MemLEO-sat Design Tool) are reported in the subsequent table. The figure reports the distributions of the objective function discussed in the previous section versus each of the parameters tuned by the MemLEO-sat Design Tool. It is worth remarking that the parameters are all tuned at the same time, so the distributions are not independently evaluated, but are the projection of the tuned parameter of the distribution with the space of all tuned parameters as support. In other words, $obj = obj(x_1,\ldots,x_{37})$ where $x_1,\ldots,x_{37}$ are the 37 tuned parameters, and the projection of each one of them is reported in the figure. To generate this distribution, the MemLEO-sat Design Tool works in the following way:

1. It starts from a random configuration of the parameters and evaluates the Blackbox (i.e. it calls the proliferated LEO satellite constellation simulator)

2. The MemLEO-sat Design Tool updates the circuit parameters of the Blackbox optimizer

3. The Blackbox Optimizer adapts to fit the results from the Blackbox and it provides the next input for the Blackbox

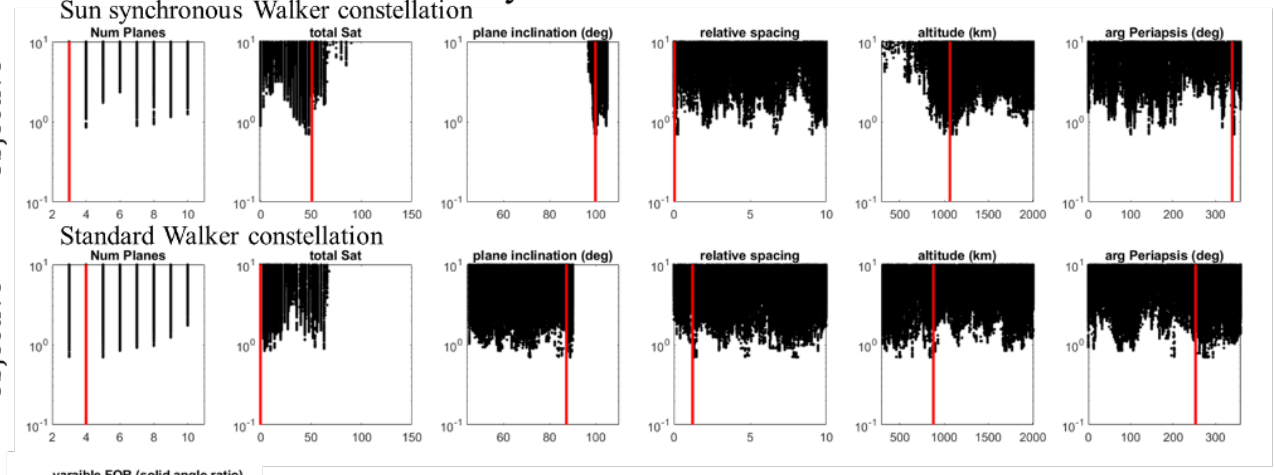
4. The BlackBox is evaluated using this new input

5. We repeat the sequence 2-4 until there are no sensitive improvements to the objective function

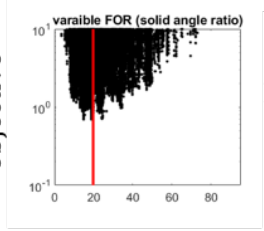
This process creates the distribution reported in the figure. From the distribution we can extract one or more parameter configurations. Because our Blackbox is affected by noise (the ballistic hypersonic targets are generated using random locations that change with each evaluation). A good strategy is selecting a set of good parameters and testing them further to determine the one that is least affected by the noise. In this way we have selected the parameter set highlighted with the red line and then reported in the table on the next page.

Primary satellite constellations

Sun synchronous Walker constellation



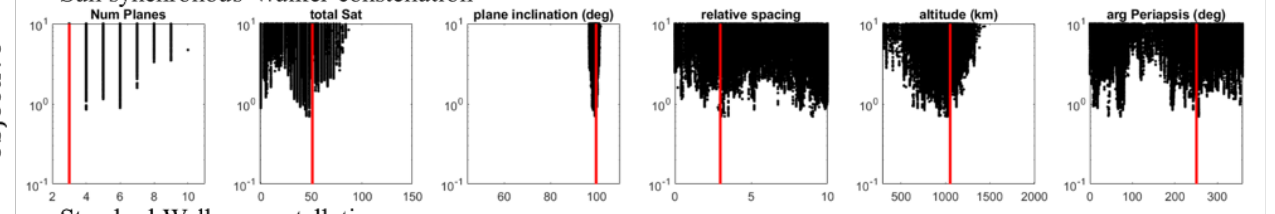
Standard Walker constellation



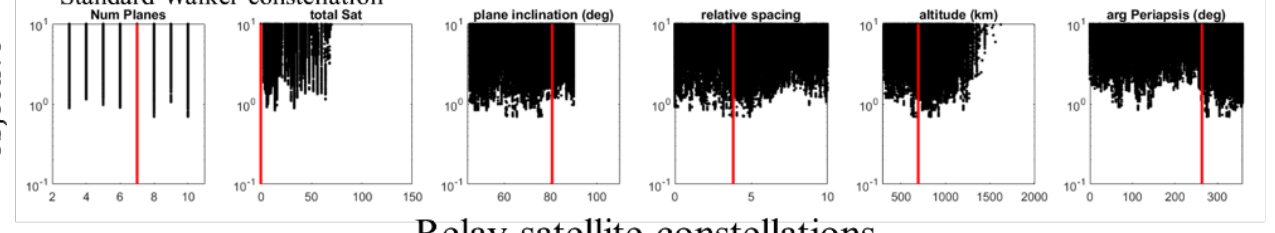
FOR common to
all primary satellites

Secondary satellite constellations

Sun synchronous Walker constellation

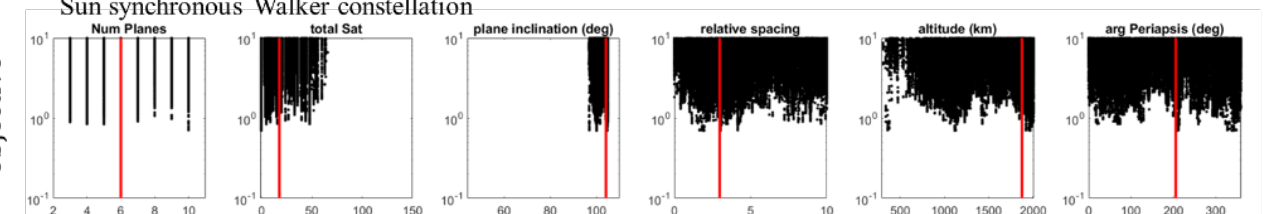


Standard Walker constellation

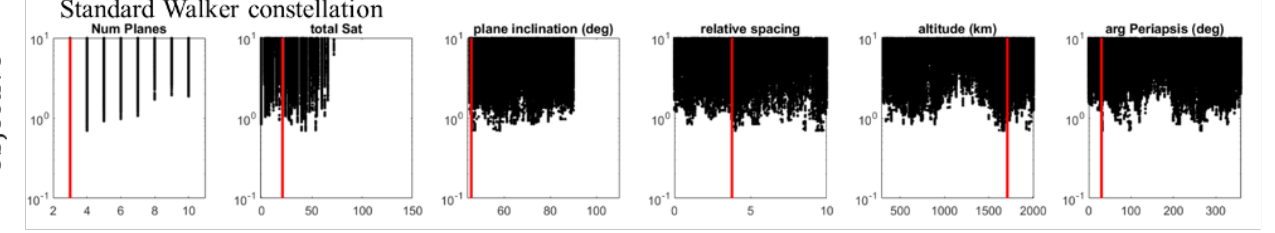


Relay satellite constellations

Sun synchronous Walker constellation



Standard Walker constellation



Primary Constellations

	a) Satellite Compute		b) Ground Compute		c) Distribute Ground Compute	
Parameter	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**
# Satellites	45	0	51	0	42	0
# Planes	3	n/a	3	n/a	3	n/a
Inclination	104.04 °	n/a	99.7°	n/a	99.61 °	n/a
Rel Spacing	4.3998	n/a	0	n/a	6	n/a
Altitude	1863km	n/a	1060 Km	n/a	1028 Km	n/a
Arg Periapsis	275 °	n/a	339°	n/a	176 °	n/a
FOR	42.94 °		20.4 °		33.47	

Secondary Constellations

	a) Satellite Compute		b) Ground Compute		c) Distribute Ground Compute	
Parameter	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**
# Satellites	69	0	51	0	42	0
# Planes	3	n/a	3	n/a	3	n/a
Inclination	100.07 °	n/a	99.7°	n/a	99.66 °	n/a
Rel Spacing	2	n/a	0	n/a	3	n/a
Altitude	1128km	n/a	1060 Km	n/a	1038 Km	n/a
Arg Periapsis	301 °	n/a	251°	n/a	99 °	n/a

Relay Constellations

	a) Satellite Compute		b) Ground Compute		c) Distribute Ground Compute	
Parameter	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**	**Sun sync Walker**	**Walker**
# Satellites	0	28	18	21	16	39
# Planes	n/a	4	6	3	4	3
Inclination	n/a	50.07 °	104°	45°	102.97 °	70.29 °
Rel Spacing	n/a	6	3	3.7	5	10
Altitude	n/a	1105 km	1880 Km	1710 Km	1682 Km	1955 Km
Arg Periapsis	n/a	295 °	205°	30°	299 °	170 °

Results Explained

From the tables and the figures reported above and below a few results are explained here:

- In all situations only the sun synchronous constellation is used for both Primary and Secondary satellites. This is a consequence of the limited (150) number of satellites leading to an optimal allocation concentrated on the sun synchronous orbits.

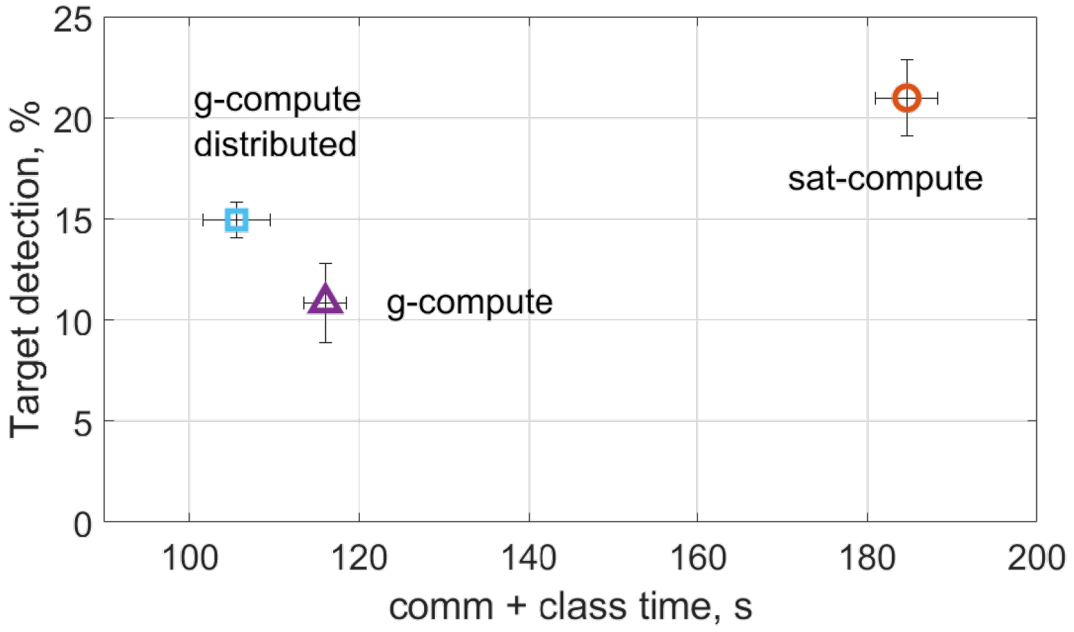
- The optimizers select the sun synchronous orbits because they provide more coverage, which includes over the poles.

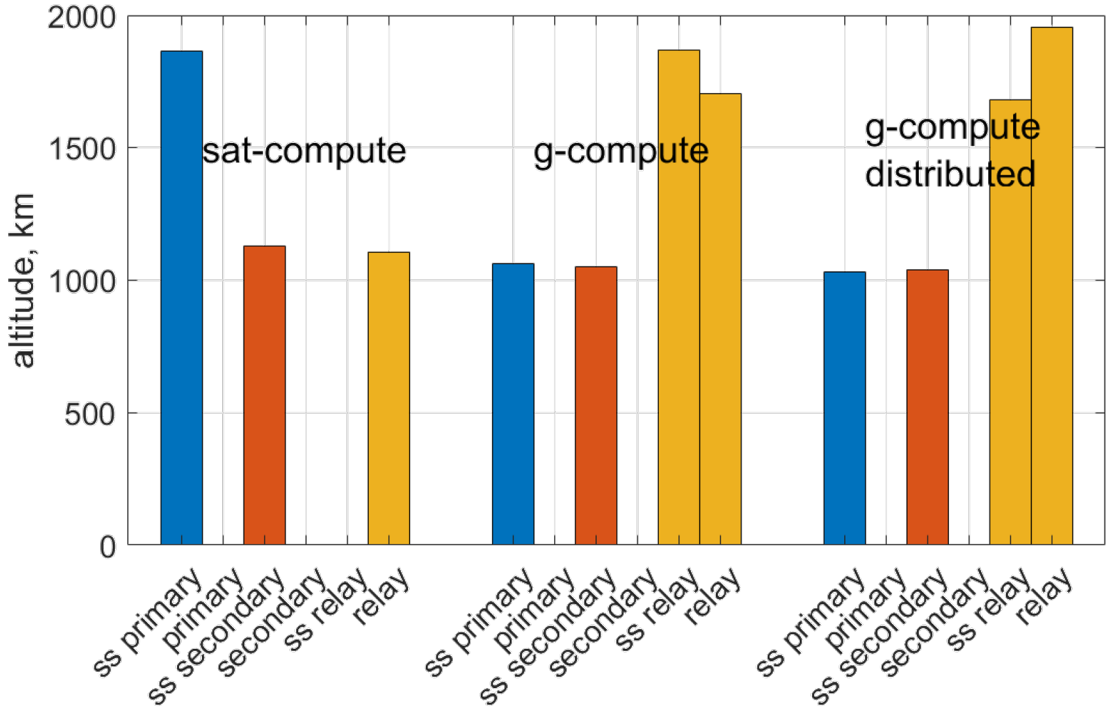
- Relay satellites are mainly distributed on normal orbits to enhance communication with ground stations. In case a) where satellites communicate a small amount of data (an image chip), no relay satellites occupy the sun synchronous orbit. On the contrary, the cases b) and c) require the streaming of large amounts of data, thus some satellites are also allocated on the sun synchronous orbit to enhance the communication with Primary and Secondary satellites.

- In cases b) and c), the primary and secondary satellites are perfectly synchronized to enhance the tracking of the target. Instead, case a) has more primary and secondary satellites that can be used because less relay satellites are necessary with respect to b) and c). This is because a much smaller amount of data needs to be transferred. Interestingly, in case a), the optimizer does not synchronize secondary and primary satellites, but rather it allocates them at very different altitudes, with more secondary than primary satellites. This is because the model using time classification strongly prevents secondary satellites from being allocated at high altitudes. Therefore, in case a), they remain at a lower altitude with respect to the primary. To compensate for the coverage, the optimizer allocates more secondaries than primary satellites and increases the FOR to the primaries.

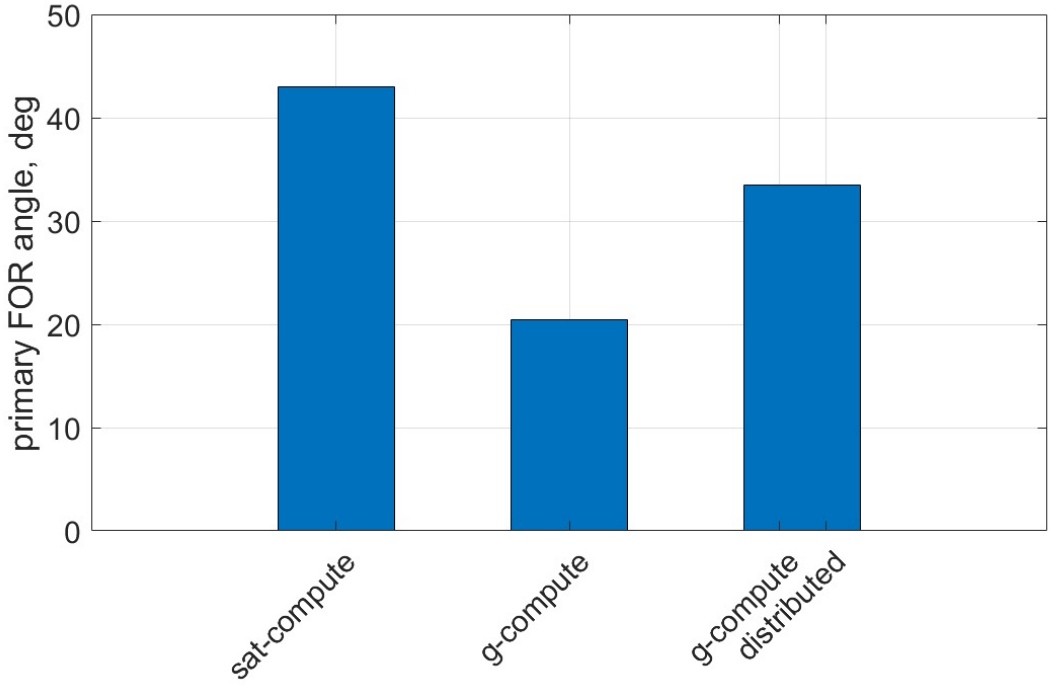
- The overall efficiency of the three different scenarios is synthesized in the figure reporting the percentage of successfully tracking and classifying a target vs the time to do it. This is estimated using the 40 hypersonic ballistic targets. The most reliable way to track and classify satellites under the conditions used here is case a). However, it is also the least efficient in terms of time as computing power on the satellites is limited. Instead, case b) is less reliable because it needs to employ more relay satellites for the intense data streaming, thus reducing the coverage. Overall it is faster, but less reliable. Interestingly, case c) demonstrates that using distributed communication sensibly increases the coverage because it allows a larger FOR for the primary satellites. When using a larger FOR, more data needs to be transferred at once, and distributed communications makes that very efficient. At the same time, it also reduces the overall time to detect and classify. This results in the most efficient compromise.

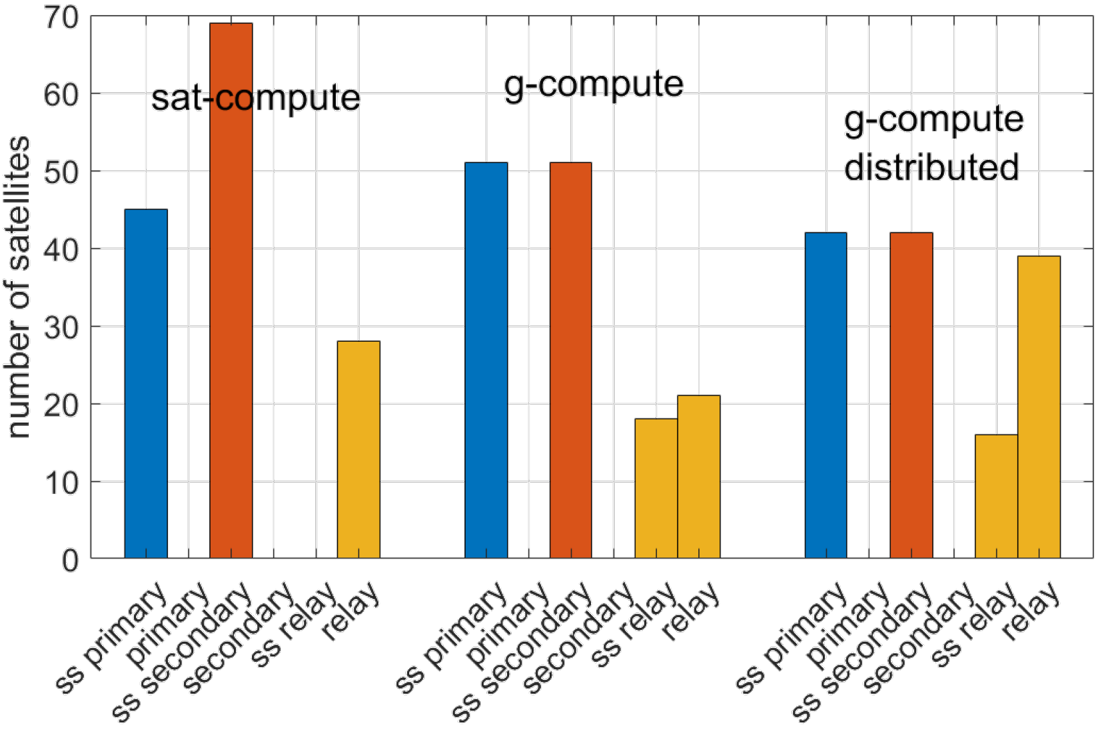
Results Continued





Results Continued





Conclusion

MemComputing, Inc. developed a unique platform, the MemLEO-sat Design Tool, for the optimal design of proliferated LEO satellite constellations. In this platform, MemComputing's bleeding-edge optimization methods and advanced models enable simulating the detection, classification, and tracking of hypersonic flying targets.

Furthermore, it includes a high-fidelity model of an optimized communication system using Memcomputing's technology to orchestrate distributed communications among satellites and between the Relay Satellites and Ground Stations. This is especially important when large amounts of data need to be transferred considering that ground links are temporary, and bandwidth is limited. This algorithm can be deployed directly on satellites.

MemComputing has demonstrated the working principles of the MemLEO-sat Design Tool by analyzing a case study consisting of three scenarios a) intensive computation performed on the satellites, b) intensive computation performed on the ground and communication through single channel per satellite c) computation performed on the ground and communication distributed through multiple channels. For these scenarios, different types of satellites were employed. The results show that case c) consisting of our distributed communication method is the best compromise under the constraints used for these three scenarios.

Conclusions Summary

- The MemLEO-sat Design Tool represents a unique platform for Automated P-LEO Constellation Design
- It employs MemComputing's bleeding-edge optimization methods.
- It minimizes latency while maximizing detection and tracking
- It includes a MemComputing-based, high-fidelity communication system and proprietary algorithm for optimal distributed communication.
- The study reveals pros and cons for processing EO/IR sensor data on Satellites, on the Ground, and using distributed communication.
- Considers effects of the communication protocols.

Appendix

Parameter Group	Parameter Name	Default/ Options	Comments
Orbit	LEO_min	300 km	
	LEO_max	2,000 km	
	constellation	Walker, % Walker, Random	
Comm Relay Satellites	N_relay_consteallations		
	relay_inclination	[60 90] degrees	
	relay_NumPlanes	[10 3]	
	relay_totalSat	[33 33]	
	relay_relSpacing	[1 1]	
	relay_altitude		[Earth_radius+500 Earth_radius+500]
	relay_eccentricity	0	
	relay_argPeriapsis	[350 350]	
Primary Satellites (Target Identification)	N_primary_consteallations		
	primary_inclination	[60 90] degrees	
	primary_NumPlanes	[10 3]	
	primary_totalSat		
	primary_relSpacing	[1 1]	
	primary_altitude		[Earth_radius+1800 Earth_radius+1800]
	primary_eccentricity	0	
	primary_argPeriapsis	[350 350]	
Secondary Satellites (Tracking)	N_secondary_consteallations		
	secondary_inclination	[60 90] degrees	
	secondary_NumPlanes	[10 3]	
	secondary_totalSat		
	secondary_relSpacing	[1 1]	
	secondary_altitude		[Earth_radius+1000 Earth_radius+1000]
	secondary_eccentricity	0	
	secondary_argPeriapsis	[350 350]	
Number of Satellites	NprimarySsat		
	NsecondarySat		
	NrelaySat		
	Nsat		NprimarySat + NsecondarySat + NrelaySat
	distribute		parameters.distribute; % comm strategy
	Nreceivers	95	For this a function is called to place receivers on the ground
	sat_angle_primary		parameters.FOR_primary; % radiants = 2.292 deg (FOV = 2*sat_angle_primary)% 0.4636/10; % radiants = 26.5623 deg
	sat_angle_secondary		45*pi/180/2; % radiants = 45/2 deg (FOV = 2*sat_angle_secondary)%$ 0.4636*2; % radiants = 26.5623 deg

Appendix Continued

Hardware and comm characteristics	primary_detec_time		4.*ones(N_primary_consteallations,1); (1-cos(parameters.FOR_primary))/(1-cos(0.04))*5.*ones(N_primary_consteallations,1); % to be updated
	primary_transfer_ratio		(1-cos(parameters.FOR_primary))/(1-cos(0.04));
	secondary_detec_time		1/30 +17.6047*(exp(0.24852 * ((secondary_altitude-Earth_radius)/500).^2) - 1); %25*(secondary_altitude-Earth_radius)/500/2;
	primary_detec_time		max(.5 * (1-cos(parameters.FOR_primary))/(1-cos(0.04)).* ones(N_primary_consteallations,1) ,2) + ... 1/30 + 17.6047*(exp(0.24852 * ((primary_altitude-Earth_radius)/500).^2) - 1); %25*(secondary_altitude-Earth_radius)/500/2;
	primary_sharing_time		1/30 + 17.6047*(exp(0.24852 * ((primary_altitude-Earth_radius)/500).^2) - 1); %25*(secondary_altitude-Earth_radius)/500/2;
	sat_sat_com_distance	5,000 km	
	relaysat_relaysat_com_distance	5,000 km	
	sat_earth_com_distance	3,000 km	
	sat_sat_min_com_distance	250 km	
	relaysat_min_relaysat_com_distance	250 km	
	sat_earth_min_com_distance	0 km	
	receiver_angle	0.7854	
	radiants	45 deg	
Targets	Ntargets		
	TargetType	Ballistic, aircrafts	