

Adapting New Processes to Support Improved Space Based Surveillance Ground Operations

Shawn W. Abernethy Jr.
Stratagem Group
Weston Faber
L3Harris

Emily Gerber
L3Harris
Thomas Kelecy
L3Harris

William Delude
Stratagem Group
Taylor Nave
L3Harris

Abstract

Lumos is a suite of software-based capabilities that enable more agile space-based surveillance operations. It reduces the labor involved in SDA anomaly detection and sensor tasking. A near real-time data anomaly filtering process, along with a coordinated multi-constellation optimized tasking scheme, are being established to automate processes for ground operators. The Lumos data anomaly alerts will be provided to a ground operator in near real-time, and the coordinated tasking will provide a “recommendation” for optimal combined collections that minimizes space object uncertainty. This paper will walk through the problem framework and provide example use cases by leveraging simulated data with notional SDA satellite sensors. With this dataset, Lumos efficiently detects a data anomaly from degraded sensor state information by employing a USKF, and it recommends effective coordinated collection using either a Genetic Algorithm or Monte Carlo Markov Chain to maintain custody of an RSO. The operational architecture and Concept of Operations (CONOPS) for Lumos will also be described.

1. Introduction

Lumos consists of two automated processes for the Tasking and Exploitation steps of the Tasking, Collection, Processing, Exploitation, and Dissemination (TCPED) loop. The tasking capability, the Lumos **Collection Recommender**, provides recommended coordinated collections of a Resident Space Object (RSO) of interest across multiple sensors of a Space Domain Awareness (SDA) collection system in order to meet the desired uncertainty thresholds specified by a user. Additionally, if the tracking data from each system is processed and validated independent of the others then there could be cases where the data quality assessment could be inconsistent, and the processes may not be fully effective in characterizing electro-optical (EO) sensor biases and weights (measurement noise) for any one or more of the systems. The exploitation capability, the Lumos **Anomaly Detector** provides a consolidated data anomaly and quality assessment process using a multi-state Unscented Schmidt Kalman Filter (USKF). This USKF is designed to leverage shared information between the sensors when data from the multiple collection sensors are fused in the filter.

In this paper, we will first define the problem framework of the Collection Recommender and present results of a simulated collection scenario. We will then present the definitions and metrics captured from the USKF used within the Anomaly Detector, walk through the Concept of Operations (CONOPS) and how an operator would interact with the capability, and present results of detecting a sensor anomaly in a simulated scenario. Both simulated scenarios were generated from publicly available Two-line Element set (TLE) files via www.space-track.org and fictitious SDA satellite sensor parameters.

2. Collection Recommender

One of the jobs of an operator within ground operations of a space-based surveillance system is to maintain custody of RSOs. To do this, they must generate tasking of a system of SDA sensors to schedule the creation of observations and tracks of RSOs of interest. This data is then used to generate estimated states and covariances for where this RSO will be at a time in the future, where the covariance represents uncertainty in the estimated state. Furthermore, it has been shown observations and tracks from multiple sensors with varying collection geometries closely spaced in time will result in reduced uncertainty of the estimated RSO state. However, with a huge catalog of RSOs whose states must also be maintained with collects, they need to derive the most accurate information from the least amount of data for a given RSO.

When each constellation operator is independently performing this trade-off of scheduling enough collects to meet certain uncertainty thresholds while trying not to overly burden their system, they are likely not tasking in a globally

optimal fashion. Coordinating the tasking will enable optimum utilization of tracking resources. Let's say the operator was given an initial covariance for an RSO they were interested in, pictorially shown as a brown bubble in Fig. 1., and they wanted to schedule collects such that at some time of interest in the future, the uncertainty of the RSOs state was below some threshold. When looking at the collection schedule shown in Fig. 1., they would then be balancing the effort of taking the collects and the resulting covariance meeting their needs in the time of interest.

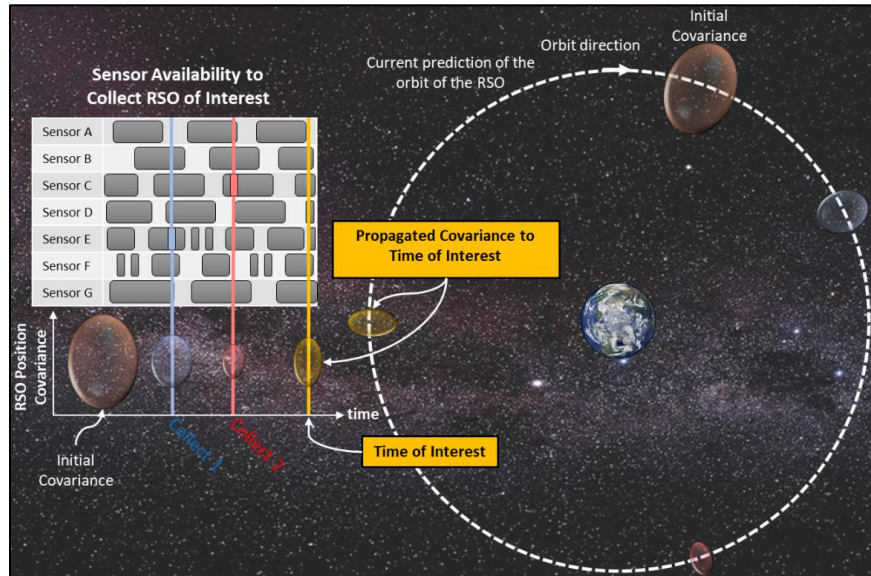


Fig. 1. Lumos Collection Recommender Overview

The Lumos Collection Recommender was designed to solve this problem for the user in an automated manner to reduce the guesswork at scheduling these collects. The Collection Recommender significantly reduces the uncertainty of a RSO's state by creating a set of coordinated collections across multiple space sensors. On the left we can see a schedule of when each sensor in the various constellations have accesses to a RSO of interest. Given an initial position and covariance of an RSO, the Lumos Collection Recommender will determine the optimal collection schedule across the sensors that maximizes an underlying reward function subject to the constraints of the constellation and operator input. As we can see in Fig. 1., the Collection Recommender recommended 2 collects, one on Sensor E that resulted in the Blue covariance bubble, and one on Sensor C that then resulted in the covariance bubble in Red. When this covariance is propagated to the time of interest, the yellow covariance bubble, it is within a defined threshold for position uncertainty as defined by the operator.

2.1 Problem Framework

The Collection Recommender algorithm first defines the orbital states of the RSOs to be tasked within the information space, and the set of possible actions at each state within the underlying Markov Decision Process (MDP). The orbital state information gain is calculated via a Kalman filter to estimate target state and covariances and information gain from potential collects. The information gain is translated to form a reward resulting from an action at the current state. Finally, we leverage search algorithms to find the optimal set of coordinated collections to satisfy user defined objectives and solve this MDP. We will first define our objective function, then define the MDP, and finally show the results using both a Monte Carlo Markov Chain (MCMC) and a Genetic Algorithm (GA) to solve the MDP of a simulated scenario. For a more detailed derivation of the underlying problem, see Appendix A.

The goal of this project is to determine *Sensor Configurations*, $A(t)$, that:

- Maximizes **Probability of Success**, $P_S(A(t))$, that is defined as a function of sensor configuration, $A(t)$ and probability of detect, P_D .
 - The goal of this metric is to choose taskings with a high **Probability of Detection**, P_D , and configurations that avoid observations with a low P_D

- Minimizes the amount of *Effort*, a metric that is a function of number of sensors required, number of observations from each system, and time/duration the systems are operating
- Minimizes *Radial/In-Track/Cross-Track (R/I/C) error* at the time of the completion of the last observation, t_F , subject to inequality constraints provided by the user
- Minimizes *R/I/C error* at a desired time after collects, T , subject to inequality constraints provided by the user

Next, we will define our objective (cost) function as $J(\chi(t), A(t), T)$,

$$J(\chi(t), A(t), T) = \omega_S P_S(\chi(t), A(t)) + \omega_E \sum_{t_0 \leq s \leq t} \frac{1}{E(A(s))} + \sum_{t_0 \leq s \leq t} \Delta I(\chi(s), A(s), \omega_t) + \sum_{t_0 \leq s \leq T} \Delta I(\chi(T), A(s), \omega_T), \quad (2-1)$$

where $P_S(\chi(t), A(t))$ is the probability of success, $E(A(t))$ is the count of number of collections used in order to execute $A(t)$, $\Delta I(\chi(t), A(t))$, $\Delta I(\chi(T), A(T))$ are the information gain in the desired R/I/C frame at times t and T , and $\omega_S, \omega_E, \omega_t, \omega_T$ are a set of weights for each component of the objective function. These are user configurable to increase or decrease the importance of each of the individual pieces of the objective function. Probability of success is defined as follows:

$$P_S(\chi(t), A(t)) = \frac{1}{|A(t)|} \sum_{z_k(t) \in A(t)} P_D(\chi(t), z_k(t)). \quad (2-2)$$

The effort is defined as follows

$$E(A(t)) = 1 + |A(t) \cap Z(t)| \quad (2-3)$$

Finally, the information gain terms will be defined as follows

$$\Delta I(\chi(t), A(t), \omega_t) = \omega_t^T \left(\frac{g(\chi(t^-)) - g(\chi(t))}{g(\chi(t^-))} \right) \quad (2-4)$$

$$\Delta I(\chi(T), A(t), \omega_T) = \omega_T^T \left(\frac{g(\chi(T^-)) - g(\chi(T))}{g(\chi(T^-))} \right) \quad (2-5)$$

where $\chi(t^-)$ is the propagated prior PDF, $\chi(T^-)$ is the propagated PDF from the previous instance, T^- to time T , $h(\chi(t))$ and $h(\chi(T))$ are the entropies at times t and T , respectively. This is depicted in Fig. 2. Also note we are dividing by $g(\chi(t^-))$ and $g(\chi(T^-))$ in equations (3-12) and (3-13), respectively. This is to normalize the measured information gain as a ratio of the total information gain over the measured covariance before the measurement was taken. Finally, we will then define the function g as follows

$$g(\chi(t)) = \text{diag} \left(P_{\text{position}}(t) \right) \quad (2-6)$$

where $P_{\text{position}}(t)$ is the R/I/C position covariance matrix of the PDF of the RSO at time t .

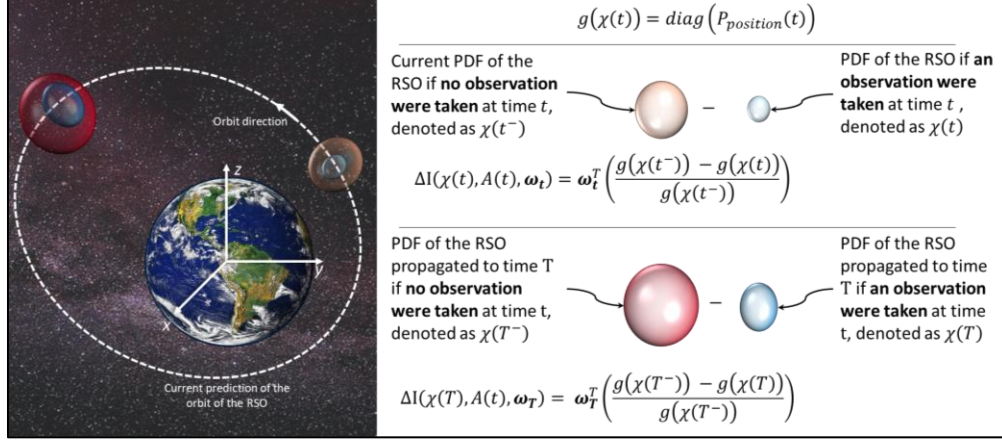


Fig. 2. Illustration of the Computed Differential Entropies at Time t and Time T

We will now define the constrained optimization problem for finding the optimal sensor configuration $A^*(t_f) \approx \underset{A(t)}{\operatorname{argmax}} J(\chi(t), A(t), T)$ s. t. (2-7)

$$g(\chi(t_f)) \leq R_F,$$

$$g(\chi(T)) \leq R_T,$$

$$t \leq T$$

$$t_f \leq T - t_p$$

where t_f is the time the last set of observations are collected, t_p is a user defined constant representing time elapsed between when an observation is taken and when the operator receives the updated state for the RSO (processing time), and R_F and R_T user defined thresholds for R/I/C error at t_f and T , respectively.

2.2 Definition of Underlying Markov Decision Process (MDP)

From our problem framework defined above, we can see the environment for our MDP is the information state of the system where the states are the posterior PDFs in the J2000 reference frame of the RSO at time t , $\chi(t^-)$. In this environment, our agent is the set of space-based sensors, S . At each state $\chi(t^-)$ the agent can task the j^{th} sensor to collect an observation against the selected RSO, o if the RSO is accessible to the j^{th} sensor and the sensor is available for $j \in \{1, \dots, M\}$. For a given collection plan $A(t)$, we can see the actions the agent decided to take at time t are $A(t) \cap Z(t)$. We then can derive the probability the agent collects all tasked collects at a given time t as follows

$$T(\chi(t^-), A(t), \chi'(t)) = \prod_{z(t) \in A(t) \cap Z(t)} P_D(\chi(t), z(t)), \quad (2-8)$$

which we will define as the transition probability from our current state, $\chi(t^-) = P(x(t^-)|A(t^-))$, to $\chi(t) = P(x(t)|A(t))$.

Of note, for a given collect at time t , $z(t) \in A(t) \cap Z(t)$, the probability of the collect not occurring is $1 - P_D(\chi(t), z(t))$. So it is possible we could end up in a different state, $\tilde{\chi}(t) = P(x(t)|\tilde{A}(t))$ than intended when the collection plan $A(t)$ was derived. Here $\tilde{A}(t)$ represents the set of collections that were collected that resulted in an observation of RSO o .

2.3 OpenAI Gym Environment

OpenAI Gym is a python toolkit that is an industry standard for training Reinforcement Learning (RL) agents. For Lumos, we implemented a customized OpenAI Gym environment for our specific MDP. The details of this environment as well as interaction of this environment with the Scenario Setup microservice and optimizers / RL agents is shown in Fig. 3. At the core is the Information Environment is an Unscented Kalman Filter (UKF) used to propagate both the covariance before a measurement is taken at the current time (i.e., current covariance) and the resulting covariance if an observation were to be made of the RSO at the given time, as shown in [3]. Additionally, the Information Environment is designed to perform proper information fusion, resulting in a more realistic covariance and a smaller information gain for near-simultaneously collected observations. [1]

Since we used OpenAI gym and implemented its standard interfaces, we can swap optimizers, RL agents, and even human-machine-interfaces with ease. When the Information Environment's Step function is called, it takes an *action* from the caller. This action is just an array of 0's (no collection) and 1's (collection) of length M . The i^{th} entry indicates whether the caller wants to task a collection for the i^{th} sensor for the current time t . The Information Environment then takes the current covariance of the RSO (shown as the *state* in Fig. 3.), and computes the resultant covariance using the UKF and information fusion as necessary. We then use the reward function as derived above to compute the resulting reward from the user's action. The reward and resulting covariance are then sent back to the caller. Additionally, for some algorithms additional information can be provided back to the caller (e.g., time remaining, total number of collects taken so far, etc.).

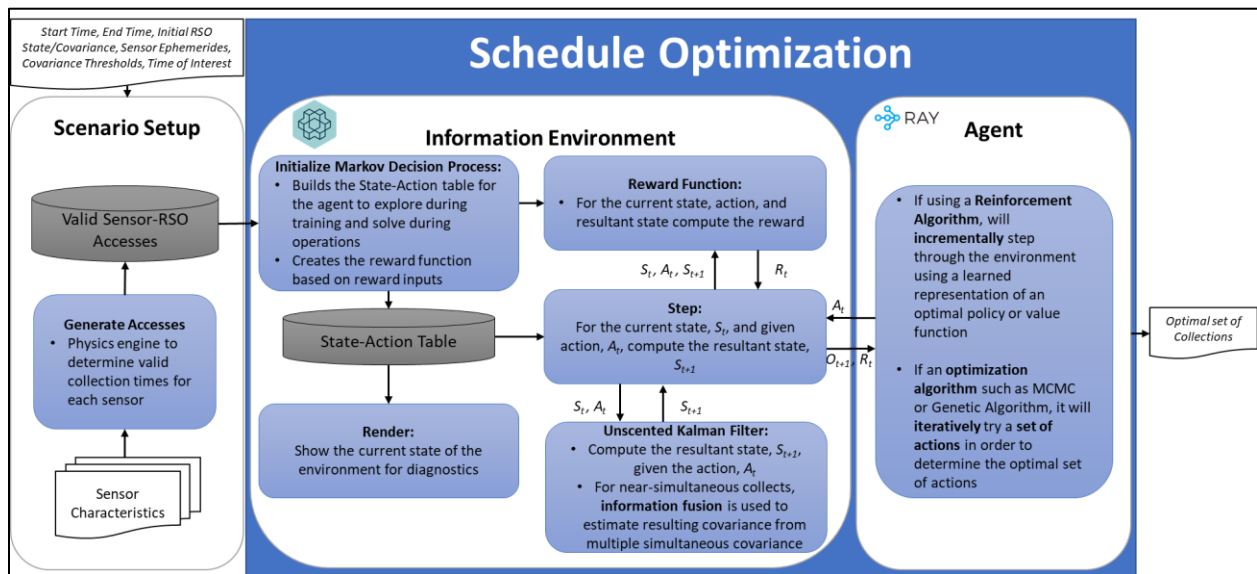


Fig. 3. Collection Recommender Exploded View

Two examples of resulting figures from the Information Environment's Render function are shown below in Fig. 4. and Fig. 5. The top plot shows possible collects the agent and algorithm has choose from. As we progress through the MDP, a black line will indicate where we are in the timeline and Gold markers will appear to indicate the times at which we did a collect from a sensor of the RSO of interest. The middle plot shows the current position covariance at time t in the desired R/I/C frame. In this plot we project our position covariance into three 2-D projections of R/I/C to the background of the 3-D plot. We also project the R/I/C position covariance thresholds. If the covariance thresholds box is green, that means both thresholds in that 2-D slice are met. Otherwise, a red box indicates at least one threshold is exceeded. The same plot is shown for the time of interest, T_F . The example in Fig. 4. is the initial environment before any steps are taken. Here we can see our initial covariance (middle plot) does not meet our Radial thresholds, and at the time of interest T_F , our R/I/C covariance greatly exceeds the R/I/C thresholds.

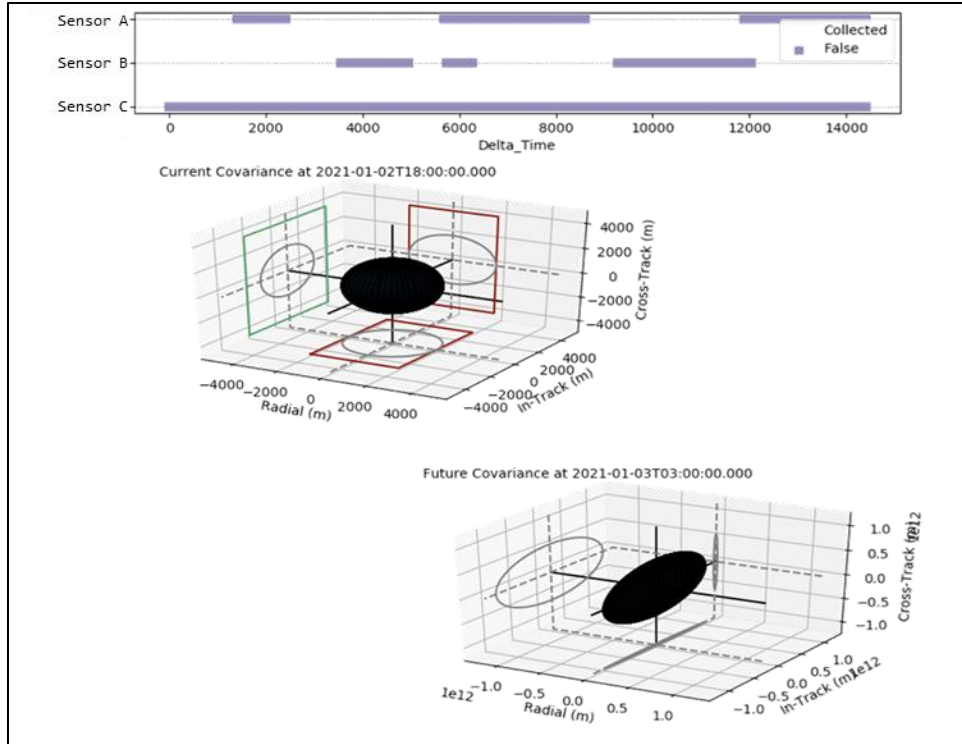


Fig. 4. Initial Rendering of the Information Environment

The example shown in Fig. 5. shows the end result after we reach the end of the MDP. Here we can see we meet the R/I/C thresholds at the end of the MDP and at the time of interest, T_F .

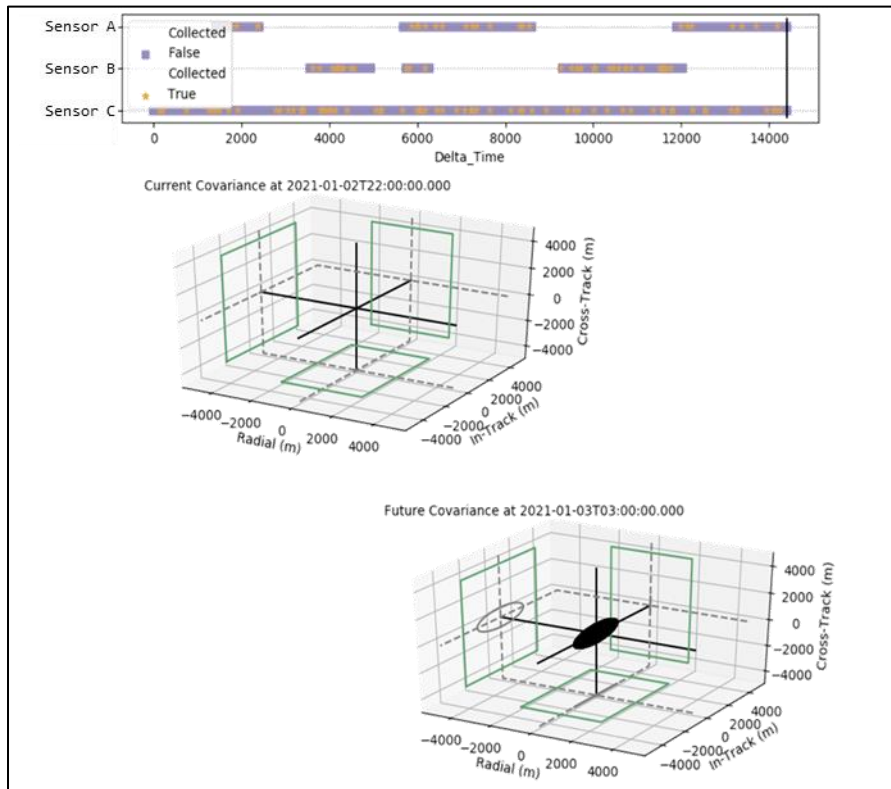


Fig. 5. Rendering of the End Result of the Information Environment

2.4 Simulated Scenario Overview

Below shows the parameters for initializing our MDP as defined above:

- **Start Time:** July 10th, 2020 15:30 GMT
- **End Time:** July 10th, 2020 18:00 GMT
- **Time of Interest:** July 10th, 2020 18:00 GMT
- **Delta Time:** 60 Seconds

To baseline our results, we implemented two optimization algorithms, Markov Chain Monte Carlo (MCMC) and a Genetic Algorithm (GA). These algorithms were then used to task collects of a low Earth orbit (LEO) RSO, the International Space Station (ISS). Our sensor constellation consisted of five fictitious space-based SDA sensors, Sensors A-E. We aimed to reduce the R/I/C position covariance in Sensor A's R/I/C frame at the time of interest to 2000 meters, 3000 meters, and 4000 meters, respectively. We will use an initial R/I/C position covariance of 2000 meters for each axis. Our initial R/I/C velocity covariance will be 200 meters / second for each axis. For position process noise used in our UKF, we will use 10,000 meters / day for the R/I/C position states and 500 (meters / second) / day, 500 (meters / second) / day, and 1,000 (meters / second) / day for our R/I/C velocity states.

While baselining our solutions for this specific two-and-a-half-hour timespan, we saw the best results when using the following objective function weights:

- ω_E : 23
- ω_t : (0, 0, 0)
- ω_T : (15, 15, 15)

Of note, we determined as we increase the duration of the scenario, we would need to increase ω_T in order to solve the MDP. This is because as we increase the time span, the more reward the agent gets for performing no collects and receiving the full ω_E . However, we can combat this by increasing ω_t and ω_T to add more incentive for the solution to task collections to reduce the RSO covariance at the time of interest.

2.4.1 Monte Carlo Markov Chain Results

When tasking our MCMC algorithm to schedule collections for this period of time, it took the agent 4.5 minutes to generate a solution. The derived optimal collection schedule consisted of 10 collections of the RSO, and this collection plan achieved a value of 1789 from the objective function. The rendering of the Information Environment at the time of interest is shown in Fig. 6. Here we can see the 10 scheduled collects in the timeline shown at the top. We can see for the first 2000 seconds (~33 minutes) no collects were scheduled. Afterwards, our MCMC solution began to schedule collects it found to be of value. The resulting R/I/C position covariance at the time of interest is within all thresholds as shown by the by the bottom figure in Fig. 6.

2.4.2 Genetic Algorithm Results

When tasking our GA to schedule collections for this time period, it took the agent ~3 minutes to generate a solution. The derived optimal collection schedule consisted of 27 collections of the RSO, and this collection plan achieved a value of 1464 from the objective function. The rendering of the Information Environment at the time of interest is shown in Fig. 7. Here we can see the 27 scheduled collects in the timeline shown at the top. These collects are scattered across the timeline unlike the MCMC algorithm. This is consistent with how the GA works by randomly selecting initial collection plans and evolving from there. As we go through iterations of evolution, the collection plans with the most impactful collects should then pass their collects onto the next iteration of collection plans and we should see less collects with each iteration. The resulting R/I/C position covariance at the time of interest is within all thresholds as shown by the by the bottom figure in Fig. 7.

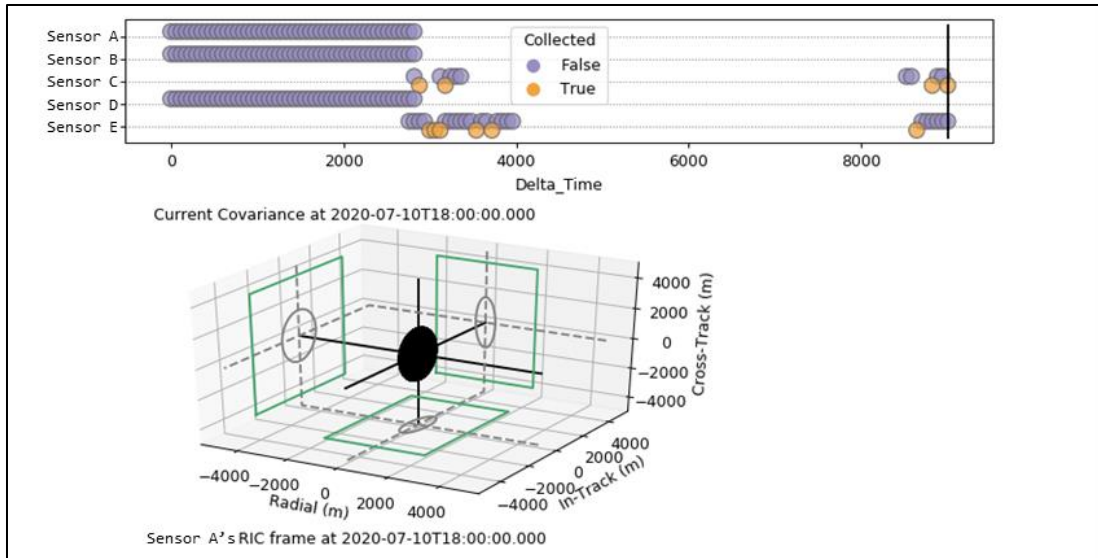


Fig. 6. Information Environment Rendering of the MCMC Results

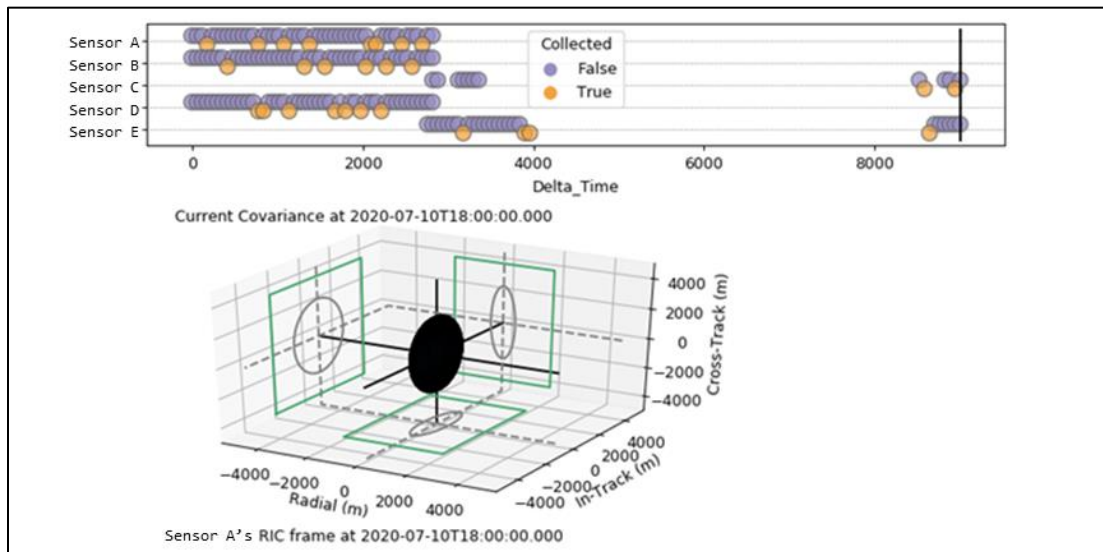


Fig. 7. Information Environment Rendering of GA Results

3. Anomaly Detector

Now, let's consider when these collects are taken, and we estimate an RSO's state from a track generated by sensors from our constellation of space based EO SDA sensors. Each of these sensors may have their own set of performance and mission requirements to satisfy. However, the observations from each system are ultimately combined and used to maintain accurate orbital knowledge of observed RSOs. Consider the case, where a single sensor now exhibits anomalous behavior, such as a sensor bias. Each sensor will generate an estimated state of RSOs they observed, and if consistent will result in a bias in the resultant RSO state. This is depicted in the Fig 8. Here we can see Sensor C and E both estimate RSO 1's state, but each have differing biases in their estimated RSO 1 state. Whereas Sensor D and F each observe RSO 2. In which Sensor F is performing nominally and does not present a biased RSO 2 state, but Sensor D does have biased data. A lot of effort by analysts is spent manually combing through this data, finding anomalous data, and resolving these discrepancies.

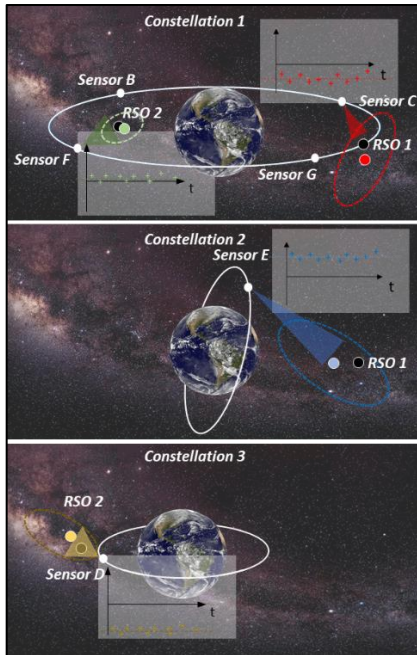


Fig. 8. Individual Sensor Collections

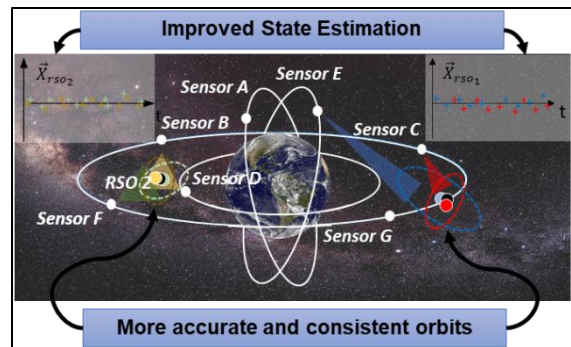


Fig. 9. Lumos Anomaly Detector Service

However, when comparing observations of RSOs from multiple sensors, we can detect anomalous observations due to sensor errors (e.g., sensor timing bias errors, out-of-date data used to compute sensor ephemerides). Additionally, we can detect unmodeled orbit maneuvers of RSOs which may result in “perceived” data anomalies.

Ultimately by consolidating the information from these observations, it helps to improve the reliability of data anomaly detection and utility to the end users. The Lumos Anomaly Detector takes a step towards automating a consolidated validation process. It does this by fusing observations of well-known and well-behaved RSOs from various space-based EO SDA constellations to monitor for sensor anomalies. This is depicted in the Fig. 9. We can see the observations from each of these constellations being fed to the multi-state Lumos Unscented Schmidt Kalman Filter which is designed to leverage information from the multiple tracking sources. This multi-state filter is estimating the RSO states and sensor timing biases.

The Lumos Anomaly detector will help result in reduced labor by analysts, and improved performance across the systems in state estimation and more accurate and consistent orbits.

3.1 Problem Framework

The Lumos Anomaly Detector Service is framed as an ITL (In-the-loop) Quality Assessment (QA) tool that processes data from multiple sensor systems on multiple satellites in a multi-state Unscented Schmidt Kalman Filter (USKF). The Lumos Anomaly Detector Service takes in data from different sensor systems and fuses the data to leverage information for the purpose of detecting anomalies in the data. Fig. 2. shows data being collected by separate sensor systems and being processed separately, as it is currently done in operations.

With data from each sensor system being processed separately, inconsistent processes result in inconsistent data quality assessments which ultimately leads to inconsistent tracking metrics. When the data is combined into a single multi-state filter information between sensors can be shared to estimate the states of the RSO’s better and estimate a sensor timing bias. An important feature of the USKF is the ability to estimate a sensor timing bias for calibration without the use of Calibration or Reference satellites. This feature allows for near real-time data calibration of the sensors.

The Lumos Anomaly Detector Service fuses the data of the different sensor systems into a single filter leading to improved orbit state and sensor bias estimates. The fusion of the different system data allows for anomaly detection in the data and the Lumos Anomaly Detector service to be a “Bell-Ringer” of anomalous data to an end user.

3.2 Anomaly Detection Definition and Metrics

3.2.1 Anomaly Detection Background

One of the challenges in determining anomalies from tracking data from multiple systems is distinguishing between true data anomalies and anomalies that are due to dynamic mis-modeling. For example, when a RSO that is being tracked maneuvers and the maneuver is not anticipated (i.e., not be modeled in the dynamics) the mis-modeling will be manifested as residuals with a systematic and growing bias (e.g., in the in-track direction). However, similar residual behavior might also be exhibited if there were, for example, a sensor timing anomaly. The key to robust and accurate anomaly detection and assessment is the effective utilization of key performance metrics from our USKF which enable detection and discrimination of anomalies. But the USKF metrics alone are not sufficient to unambiguously discriminate anomalies. When data from multiple sensors tracking multiple RSOs are combined in the estimation filter one is able to leverage dynamic constraints (“the laws of physics never lie...”) to improve the quality and expedite anomaly detection. This, in turn, also leads to a more robust and accurate assessment of data quality for the multiple sensor systems as any inconsistency between the sensors will manifest itself in performance metric anomalies. Of note, to see

Below we list these performance metrics and how they are used for anomaly detection and data quality assessment:

- **Data Quality Assessment Tests:** Uses multiple tests in a robust method to determine if there is a data quality anomaly present. Each individual test used provides a unique insight into the data, but all of the tests combined give a more comprehensive and accurate determination of the overall quality of the data.
- **Outlier Test:** An outlier is a sensor measurement that is outside of 3σ of the expected measurement values. It is typically due to a “bad” data point and any consistent outliers are indicators of either a systematic sensor error or dynamic mis-modeling (e.g., a maneuver). If the measurement is determined to be an outlier, then the measurement is not processed in the filter. This removal of outliers prevents the outlier measurement from negatively affecting the filter states while still being recorded for the determination of a data quality anomaly. The overall percentage of data points labeled as outliers is tracked as the filter is processing. If the percent of outliers is above a threshold then the Outlier Test is marked as failed.
- **Pre-/Post-Fit Residual Test:** Looks at the expected and actual measurement difference and compares it to the innovation covariance. The Pre-fit Residual Test looks at the expected and actual measurement before the update in the filter and the Post-fit Residual Test looks at the expected and actual measurement after the update in the filter. The residual should fall within the innovation covariance from the filter, if the residuals is outside of the covariance the test fails. If there is a percent of the Pre- and Post-fit residuals that exceed a threshold then the test is marked as failed for that sensor and RSO pair.
- **Sensor Bias Estimate Test:** Allows for operators to determine if there is a timing bias in the sensor and monitor if the sensor bias behaves in an anomalous way. The filter is able to estimate a timing bias associated with a specific sensor and can use the estimated bias to correct the timestamp of measurements that are processed in the filter to correct for the bias. Note, only the timestamps of the observations in the filter are adjusted to account for the timing bias. If the timing bias is validated, then it would need to be applied to the operational data for that sensor prior to dissemination. By estimating a “generic” bias in the sensors, we account for that source of potential error in the measurements and can be sensitive to detecting other data quality anomalies for the sensor, like errors in the sensor state. The Sensor Bias Estimate Test looks for anomalies in the estimation of the sensor bias. If the bias estimate falls outside of the uncertainty in the estimate, then the test fails. If the percent of failed sensor timing bias estimates is above a threshold, then the sensor has a failed test.
- **McReynold’s Consistency Test:** Shown in Fig. 10., this test uses the forward filtering and backwards smoothed results to determine if any of the updates of the filter are unexpected. The Consistency Test looks at all of the states in the filter (Position, Velocity, Area to Mass Ratio, and Bias Estimates) to determine if the update performed on the filter state is appropriate for the uncertainty in the state. The metric from the McReynold’s Consistency Test is a positive number that should be less than three for all components of

the state of the filter. If the metric is greater than three then the McReynold's Consistency test fails for that component of the state. If the percent of failed tests is greater than a threshold, then the test for the RSO and sensor pair fails. [10]

McReynolds' consistency check

Define the State and Covariance:

X_k^f = filtered state estimate at time t_k

X_k^s = smoother state estimate at time t_k

P_k^f = filtered covariance estimate at time t_k

P_k^s = smoother covariance estimate at time t_k

Form the State and Covariance Differences:

$X_{\Delta k} = X_k^f - X_k^s$

$P_{\Delta k} = P_k^f - P_k^s$

Define the Consistency Ratio:

$R_k^i = \frac{X_{\Delta k}^i}{\sigma_{\Delta k}^i}$

- If $\text{abs}(R_k^i) \leq 3$ for all i and k , then the test is satisfied globally for each estimate
- If $\text{abs}(R_k^i) > 3$ for all i and k , then the filter-smoother test fails globally indicating the possibility of modeling inconsistencies
- Thus, position, velocity and A/m estimation performance can be assessed in terms of the ratio of the estimates to the predicted/assumed modeling uncertainties.

Fig. 10. Filter-smoother Consistency Test [10]

3.2.2 Data Quality Anomaly Determination

For each sensor and RSO measurement pair, Lumos runs all of these data quality assessment tests to determine if there is a data quality anomaly. Fig. 11.a-e shows the various Sensor and RSO Data Quality Scenarios. If a majority of the tests fail, then the status for that sensor and RSO pair is changed to a warning state, shown in yellow in Fig. 11.b-c. If the data being processed continues to fail over time, the status of the sensor and RSO pair becomes an alert, shown in red in Fig. 11.d-e. A nominal status, shown in green in Fig. 11.a, represents the sensor and RSO pair data being processed does not have any data quality anomalies being detected.

		Sensor		
		A	B	C
RSO	1			
	2			
	3			
a.) Nominal Processing				
RSO	1			
	2			
	3			
b.) Sensor Data Quality Anomaly (Warning)				
RSO	1			
	2			
	3			
c.) RSO Data Quality Anomaly (Warning)				
RSO	1			
	2			
	3			
d.) Sensor Data Quality Anomaly (Alert)				
RSO	1			
	2			
	3			
e.) RSO Data Quality Anomaly (Alert)				

Fig. 11. Sensor and RSO Data Quality Scenarios

Lumos then determines if the failed tests are due to the result of either a sensor or RSO data quality anomaly. If there is a Sensor Data Quality Anomaly Alert, Fig. 11.d, this information gets sent to the Lumos UI for the operator to Acknowledge or Analyze. The RSO Data Quality Anomaly Alerts, Fig. 11.e, are not sent to the UI for the operator to acknowledge but can result in the satellite being removed from consideration in the filter.

3.3 Lumos Anomaly Detector CONOPS

We will now outline a couple concept of operations (CONOPS) for handling sensor and RSO data quality anomalies. The CONOPS assumes that the starting data being processed is nominal, then a data quality alert is correctly identified, the operator decides how to acknowledge and analyze the alert, then address the cause behind the alert and re-introduce the sensor/RSO for consideration in our filter.

3.3.1 Sensor Data Quality Anomaly Alert CONOPS

In the Sensor Data Quality Anomaly Alert CONOPS shown in Fig. 12., Sensor A has a data quality anomaly that is detected by the Lumos process and the operator is alerted of the anomaly on the UI. The operator has the opportunity to remove the sensor from processing or continue to monitor the data quality of the sensor. If the sensor is removed from processing, the data is re-processed without the sensor data included to remove any negative effects of the data quality anomaly. After the sensor data quality anomaly has been resolved, it can be turned back on for consideration in the filter to continue monitoring for data quality anomalies.

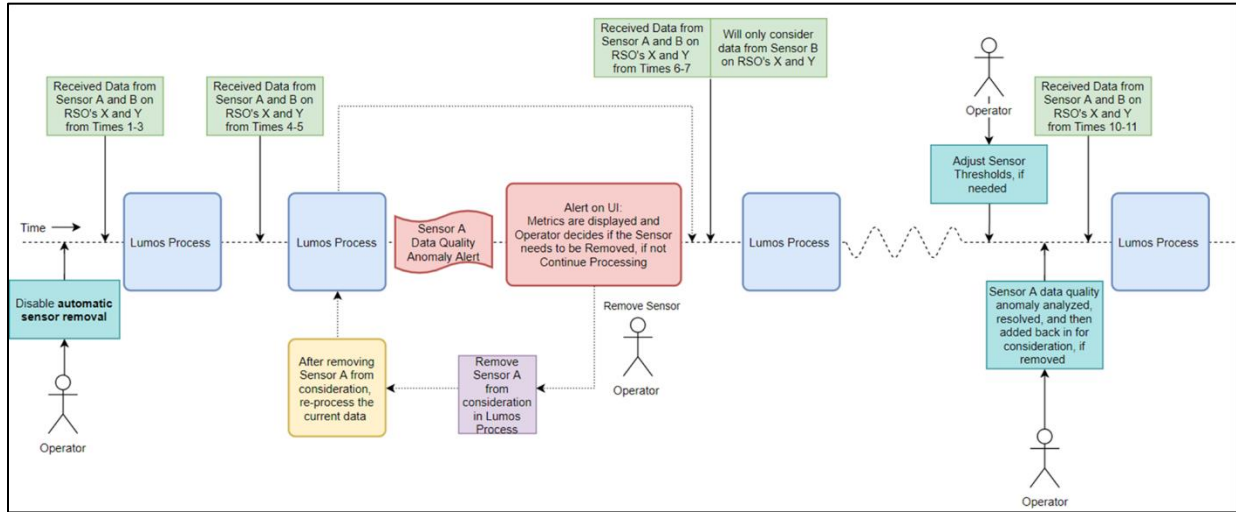


Fig. 12. Sensor Data Quality Anomaly Alert CONOPS

3.3.2 RSO Data Quality Anomaly Alert CONOPS

The RSO Data Quality Anomaly Alert CONOPS, shown in Fig. 13., follows a similar pattern as a Sensor Data Quality Anomaly Alert. Instead of removing the sensor from consideration in the filter, a RSO with a data quality anomaly is alerted on and removed from processing until it is evaluated and added back in for consideration.

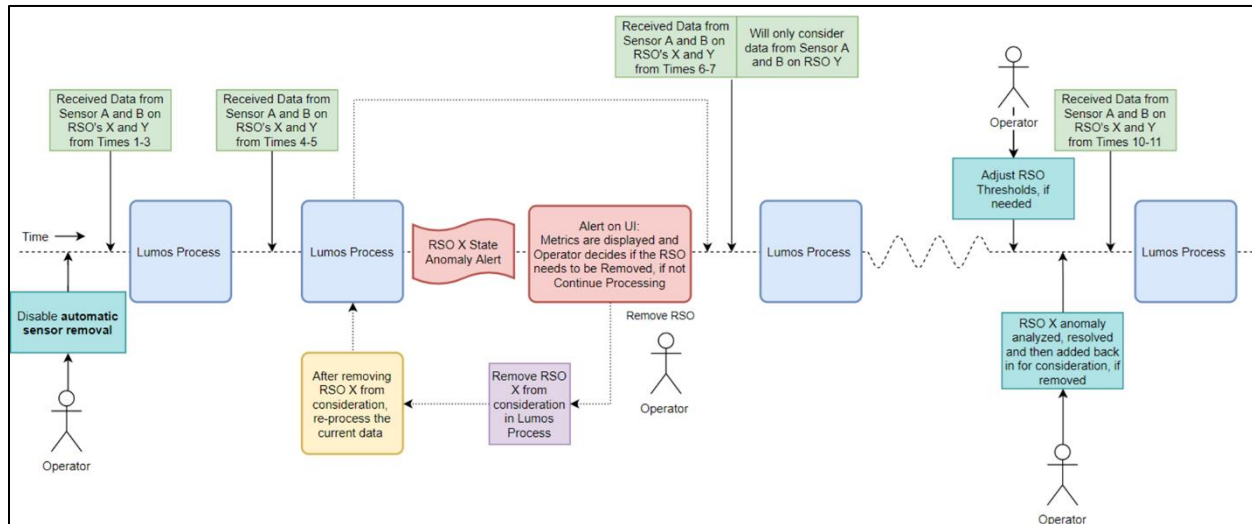


Fig. 13. RSO Data Quality Anomaly Alert CONOPS

3.4 Test Case Results

To demonstrate the ability of the Lumos Anomaly Detector to detect a Sensor anomaly, we built a simulated scenario using two notional GEO SDA sensors that are tracking multiple GEO RSOs. We use a TDRS-11 (39070) TLE to model Sensor 901, and a ANIK F1-R (28868) TLE to model Sensor 902. These sensors are tracking RSOs TDRS-10 (27566), ANIK F1-R (28868), TDRS-11 (39070), EUTELSAT 177W-B (41589), and SES-15 (42709).

The simulation is designed for a single sensor to collect three images on an RSO and then slew to track the next tasked RSO. After all visible objects have been collected on at least once, the sensor starts over and begins collecting on the first object and slews through all objects. This process is repeated for the duration of the simulation. The filter has the capability to process EO measurements and also reference state observations. The use of reference state observations improves the overall accuracy and efficiency of the filter, they were not used for this specific test case.

The goal of this scenario is to detect an applied anomaly in the sensor state being sent to the filter. An error or bias in the sensor state can happen for a variety of reasons, such as a timing bias in the sensor state, issues with the calculation/propagation of the state, or other reasons. In this simulated scenario, the observations of the sensors were generated with the true sensor states, but the Sensor 901 reported state had a slight perturbation applied and propagated.

The simulated observations were broken into three different datasets to sequentially process through the Lumos Anomaly Detector in three batches. This was done to demonstrate (1) the ability to alert on an anomaly detected within the data (2) turn of processing for the sensor/satellite with the data anomaly and continue processing and monitor for additional anomalies and (3) add back in the sensor/satellite into the filter process after the anomaly has been analyzed. Fig. 14. shows how the data was generated for the different runs for this test case.

		Sensor	
		901	902
RSO	42709	x	x
	27566	x	-
	41589	-	x
	28868	-	x

a) **Dataset 1:** Sensor 901 with Data Anomaly

		Sensor	
		901	902
RSO	42709	x	x
	27566	x	-
	41589	-	x
	28868	-	x

b) **Dataset 2:** Sensor 901 with Data Anomaly Still Present

		Sensor	
		901	902
RSO	42709	x	x
	27566	x	-
	41589	-	x
	28868	-	x

c) **Dataset 3:** Sensor 901 without Data Anomaly

Fig. 14. RSO-Sensor Pairs and Whether an Anomaly is Present (red) or Not Present (green)

The results from processing the first dataset through the Lumos Anomaly Detector show there was an anomaly detected for Sensor 901. Fig. 15. shows the Pre-Fit and Post-Fit Measurement Residuals Test results. As we can see the measurement residuals are outside of their normal bounds and have abnormal values. The Lumos Anomaly Detector correctly identifies this as a failed test for Sensor 901. In comparison, the measurement residuals for Sensor 902 pass their test. This helps isolate the anomaly to Sensor 901.

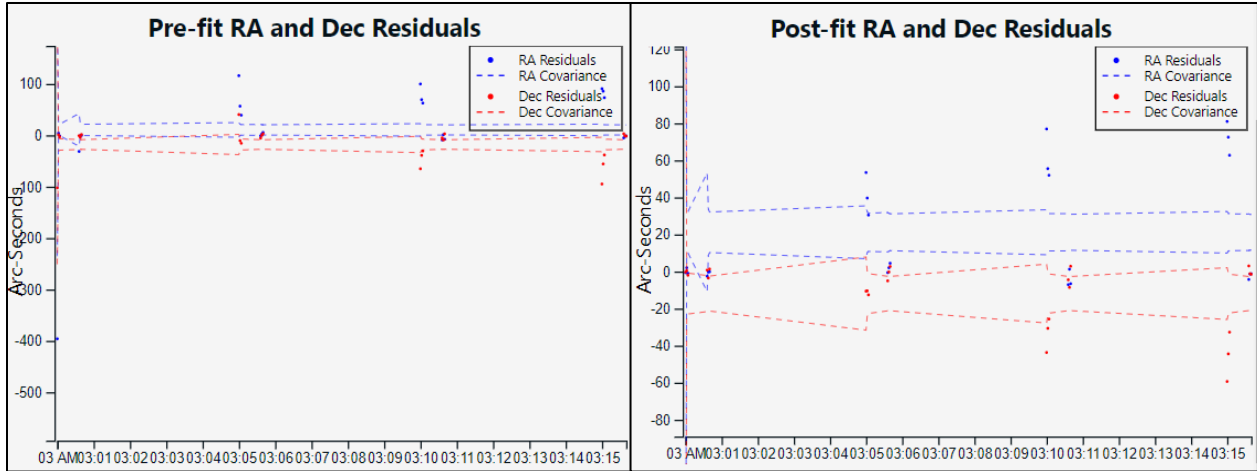


Fig. 15. Sensor 901 Measurement Residuals After Processing Dataset 1

Additionally, the Consistency Test for Sensor 901, shown in Fig. 16., fails as well. While the Consistency Test for Sensor 902 passes. For the Sensor Timing Bias Estimate, the results for Sensor 901 can be seen in Fig. 17. The magnitude of the timing bias estimate for Sensor 901 is much larger than any expected timing bias and the estimate goes outside of the large covariance bounds, which fails the timing bias estimate test. Due to our filter design, Sensor 902 is able to have its own timing bias estimated without issues from Sensor 901.

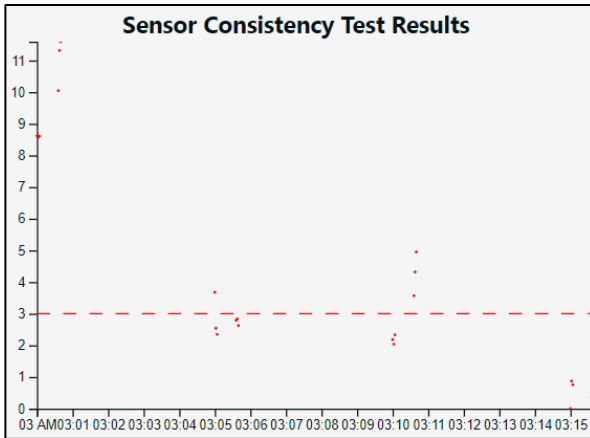


Fig. 16. Consistency Test Results After Processing Dataset 1 for Sensor 901

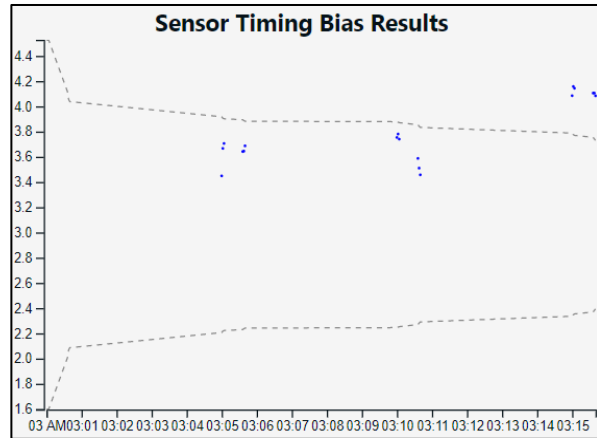


Fig. 17. Timing Bias Estimate for Sensor 901 After Processing Dataset 1 Showing Potential Data Anomaly

The combination of failed tests for Sensor 901 across multiple RSO-sensor pairs as shown in Fig. 18. allows for the Lumos Anomaly Detector to conclude there is an anomaly for Sensor 901.

		Sensor	
		901	902
RSO	42709	FAIL	PASS
	27566	FAIL	
	41589		PASS
	28868		PASS

Fig. 18. Test Summary After Processing Dataset 1

It is important to note that the tests were failed for multiple RSOs (42709 and 27566), but the failed tests were exclusive to Sensor 901 observations of these RSOs. This information is used to distinguish between a Sensor Data Quality

Anomaly and a Satellite Data Quality Anomaly. The combinations of failed tests for the sensor causes an alert to be sent to the UI so an operator can take a closer look at the data, as shown in Fig. 19.

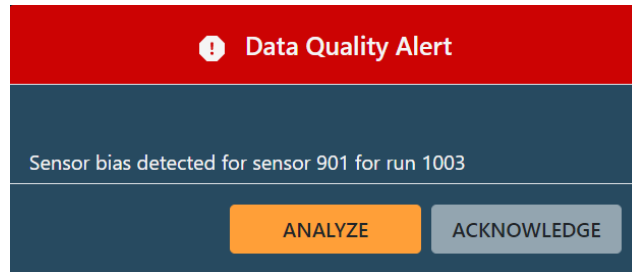


Fig. 19. Lumos Data Quality Alert After Processing Dataset 1

The Analysis tab of the UI will also keep a log of the status of different runs and allows the opportunity to go back and look at previous run results. Fig. 20. shows the alert on the Analysis page of the UI.

1003	1013	2021-02-22T16:35:33.319Z	2020-01-01T10:01:11.999Z	902	41589 - EUTELSAT 117W...	NOMINAL
1003	1012	2021-02-22T16:35:33.286Z	2020-01-01T10:00:35.999Z	902	39070 - TDRS 11	NOMINAL
1003	1011	2021-02-22T16:35:33.254Z	2020-01-01T09:59:59.999Z	902	42709 - SES-15	NOMINAL
1003	1010	2021-02-22T16:35:33.222Z	2020-01-01T10:00:35.999Z	901	42709 - SES-15	BIAS ▲
1003	1009	2021-02-22T16:35:33.140Z	2020-01-01T09:59:59.999Z	901	27566 - TDRS 10	BIAS ▲

Fig. 20. Analysis Alert Log After Running Dataset 1 Showing the Detected Bias for Sensor 901 RSO Observations

After an operator evaluates the data that has caused the alert, the operator can “turn off” Sensor 901 from consideration in the filter as shown in Fig. 21. Sensor 901 may still be collecting data, but it will no longer be processed by the filter.



Fig. 21. Switch to Toggle Sensor 901 Observations “Off” and No Longer Considered in the Filter

After the sensor with anomalous data is removed from consideration, the second dataset can be processed in the filter, as shown in Fig. 22. The second data set has anomalous data from the Sensor 901, but it will not be considered in the filter. Without anomalous data present in the filter, the filter has nominal processing for the available satellites and sensors. The nominal results can be seen in the Analysis tab of the UI.

1004	1016	2021-02-22T20:31:34.017Z	2020-01-01T10:20:12.000Z	902	41589 - EUTELSAT 117W...	NOMINAL
1004	1015	2021-02-22T20:31:33.974Z	2020-01-01T10:19:36.000Z	902	39070 - TDRS 11	NOMINAL
1004	1014	2021-02-22T20:31:33.935Z	2020-01-01T10:18:59.999Z	902	42709 - SES-15	NOMINAL

Fig. 22. Analysis Alert Log After Running Dataset 2 Showing Only Sensor 902 Observations Being Processing

With all of the observations being processed nominally, there is no alert on the Lumos UI. After some time, the anomaly in the Sensor 901 can be assessed and corrected. After the correction has been evaluated and implemented, the sensor can be turned back on for consideration in the filter. This allows for the Sensor 901 data to be processed in the filter. In our scenario, the anomaly was corrected for Dataset 3. So, we turn Sensor 901 back “on” for consideration in the filter for Lumos to process the last batch of observations, as shown in Fig. 23.



Fig. 23. Switch to Toggle Sensor 901 Observations “On” and Now Being Considered in the Filter

In Fig. 24. we can now see the pre- and post-fit measurement residuals are within their covariance thresholds for Sensor 901. Additionally, we can see the consistency test results, Fig. 25., and filter estimated sensor timing bias for

Sensor 901, Fig. 26., are nominal. With nominal processing the Lumos Anomaly Detector Service is able to continue processing data from satellites and sensors to detect anomalies within the data.

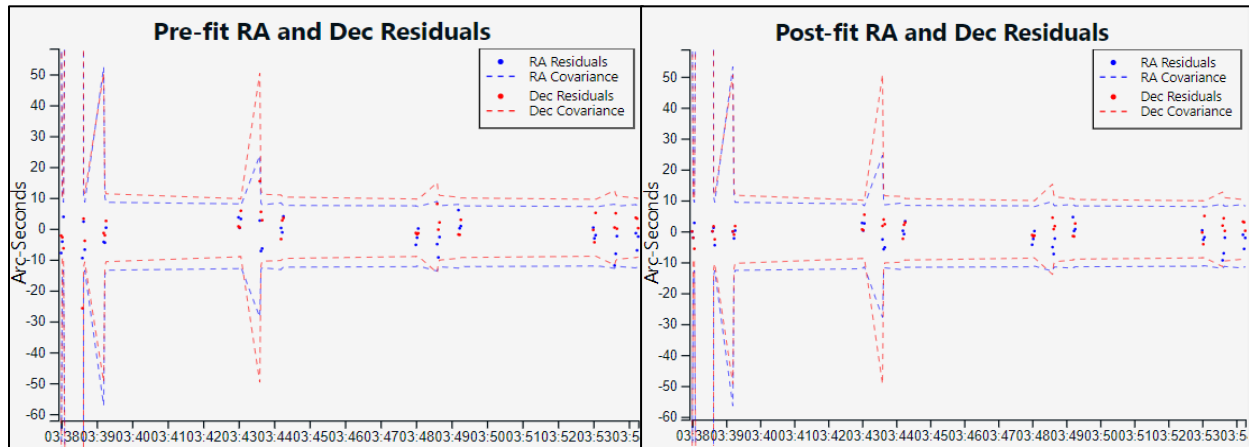


Fig. 24. Sensor 901 Measurement Residuals After Processing Dataset 3

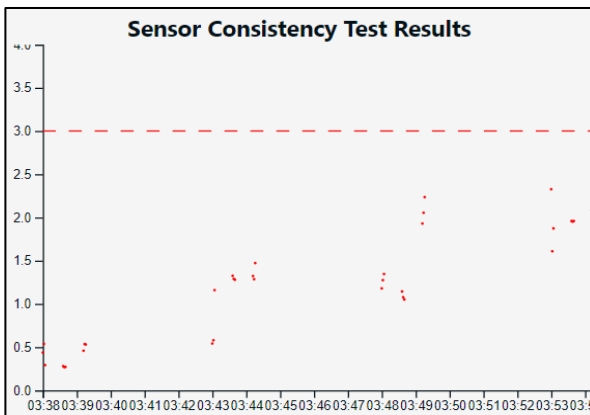


Fig. 25. Consistency Test Results After Processing Dataset 3 for Sensor 901

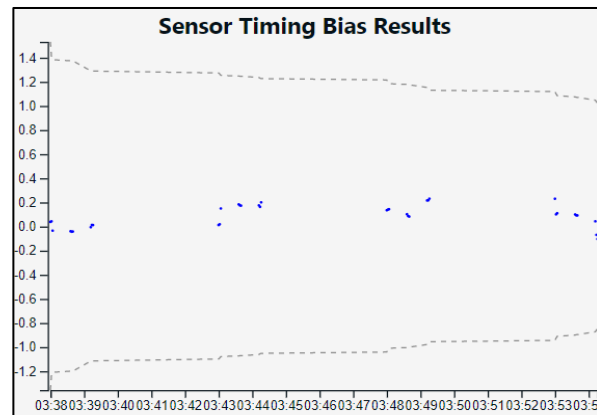


Fig. 26. Timing Bias Estimate for Sensor 901 After Processing Dataset 3 Showing Nominal Sensor Bias

4. Conclusion

In this paper, we presented two automated capabilities within Lumos, the Collection Recommender and Anomaly Detector. Both capabilities will reduce the labor involved in tasking coordinated collects of RSOs of interest across multiple sensor constellations and detect data anomalies. The Lumos Collection Recommender has been designed to provide the capability to find the best set of collections across a constellation to reduce the R/I/C covariance of an RSO of interest within user specified R/I/C thresholds in a specified R/I/C frame at time of interest. The results from the simulated scenario showed the OpenAI Gym environment within the Lumos Collection Recommender providing the ability to easily interface with both a MCMC and GA search algorithm to derive an optimal solution to the underlying MDP. The Lumos Anomaly Detector process has been designed to capture both data and dynamic anomalies and be the “Bell Ringer” to and end user. Results from the example use case presented in this report demonstrate these capabilities. Fictitious data was simulated and processed for a specific use case intended to show a data anomaly most likely to happen being flagged in near real-time. The proposed CONOPS for utilization of the Anomaly Detector was demonstrated for how this process can be used by an operator and illustrates specific actions required under operational circumstances. As illustrated in the use case, a data anomaly was reliably detected and distinguished using the USKF metrics and analyzation techniques.

5. References

- [1] L. Mao and S. Liu, "Multi-sensor Distributed Information Fusion UKF Filter for Passive Location," *Chinese Control and Decision Conference*, pp. 672-675, 2010.
- [2] A. Juliani, "Simple Reinforcement Learning with Tensorflow Part 8: Asynchronous Actor-Critic Agents (A3C)," 6 June 2018. [Online]. Available: <https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2>.
- [3] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," *The IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153-158, 2000.
- [4] R. Linares and R. Furfaro, "An Autonomous Sensor Tasking Approach for Large Scale Space," *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2017.
- [5] J. Stauch and M. Jah, On the Unscented Schmidt-Kalman Filter Algorithm. *Journal of Guidance, Control, and Dynamics* 38(1): 117-123, 2014.
- [6] Kececy, T., E. Lambert, B. Sunderland, J. Stauch, T. Kubancik, V. Mallik, M. Jah, J. Paffett, N. Sanches-Ortiz and Jaime Nomen Torres, "Automated Near Real-time Validation and Exploitation of Optical Sensor Data for Improved Orbital Safety," Proceedings of the 69th International Astronautical Congress, Bremen, Germany, October 2018.
- [7] Kececy, T., E. Lambert, B. Sunderland and J. Stauch, "Automated Near Real-time Validation and Data Integrity Assessment Using an Unscented Schmidt Kalman Filter (USKF)," Proceedings of the Space Flight Mechanics Meeting, Ka'anapali, HI, AAS 19-521, January 13-17, 2019.
- [8] Kececy, T. and M. Jah, "Analysis of Orbit Prediction Sensitivity to Thermal Emissions Acceleration Modeling for High Area-to-mass Ratio (HAMR) Objects," AMOS Technical Conference, Maui, HI, Maui Economic Development Board, 2009.
- [9] Vallado, D., Kececy, T., and M. Jah, "Data Integrity in Orbital Data Fusion," 63rd International Astronautical Congress. Naples, Italy: International Astronautical Federation, 2012.
- [10] Wright, J. R., "McReynolds' Filter-Smoother Consistency Test," Internal Analytical Graphics Inc. Internal white paper, May 15, 2009.
- [11] Raley, J. et al, "The OrbitOutlook: Autonomous Verification and Validation of Non-Traditional Data for Improved Space Situational Awareness," 17th Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference. Maui, HI: Maui Economic Development Board, 2016.

6. Appendix A: Derivation of Lumos Collection Recommender Framework

For this problem, we will define our state space, measurement space, and information space for our RSO of interest. Consider a single RSO, o , with a state vector as a function of time, $x(t)$,

$$x(t) = \begin{bmatrix} pos_x(t) \\ pos_y(t) \\ pos_z(t) \\ vel_x(t) \\ vel_y(t) \\ vel_z(t) \end{bmatrix}, \quad (6-1)$$

where $[pos_x(t) \ pos_y(t) \ pos_z(t)]^T$ and $[vel_x(t) \ vel_y(t) \ vel_z(t)]^T$ represent a position and velocity in the J2000 reference frame. Furthermore, the dynamics of the object can be assumed to be governed by the following general stochastic differential equation,

$$\dot{x}(t) = f(x(t), u(t)) + \omega(t), \quad (6-2)$$

where $f(x(t), u(t))$ represents the underlying orbital motion model with control inputs, $u(t)$, and $\omega(t)$ is a Gaussian random variable representing the process noise of the model. For this work we are assuming that we have knowledge of the underlying motion model and control inputs (including low thrust maneuvers). However, they are not known exactly, thus x is a multivariate random variable.

We consider a set of space-based sensors S with cardinality M , $S = \{s_1, s_2, \dots, s_M\}$. This set consists of a variety of sensor types including fixed stare optical, and steerable Electro-Optical (EO). Observations from these sensors at a time t are defined in our measurement space as follows,

$$z_k(t) = H_j(x(t)) + v_j(t), \quad (6-3)$$

for $k \in K(t) = \mathbb{Z}_{\bar{k}(t)} \setminus \{0\}$ where $\bar{k}(t)$ is the number of sensors that can make an observation of the RSO at time t . Please note, if no sensors can make an observation at time t , then $K(t) = \emptyset$. We will define a function

$$g_t : K(t) \rightarrow \{1, \dots, M\} \quad (6-4)$$

such that $g_t(k) = j$ maps the observation index, k to which sensor, j collected the observation, $z_k(t)$ at time t . Furthermore, at time t , if the RSO can be observed by the j^{th} sensor, we will only consider one observation that represents the maximum likely observation from the sensor. Thus, $|K(t)| \leq M$. Additionally, H_j represents the j^{th} sensor model with corresponding measurement noise, $v_j(t)$ which is assumed to be Gaussian distributed with zero mean.

We will now define the set of measurements at time t as

$$Z(t) = \{z_k(t) \mid k \in K(t)\}, \quad (6-5)$$

and the collection of all measurements from a given initial time t_0 to the current time t as

$$\Psi(t) = \{Z(s) \mid t_0 \leq s \leq t\}. \quad (6-6)$$

The posterior probability distribution function (PDF) for the RSO, o , at time t in the J2000 reference frame can then be given by the conditional density $P(x(t) \mid \Psi(t))$. Note that to consider multiple sensor measurements at any simultaneous time, t , the posterior density must be computed using an appropriate tracking algorithm capable of optimal sensor information fusion. To accomplish this, we will define the posterior PDF in the J2000 reference frame to be the information state of the system $\chi_k(t)$ resulting from the k -th measurement at time t ,

$$\chi_k(t) = P(x(t)|\Psi_k(t)), \quad (6-7)$$

Where the prior measurements are defined as

$$\Psi_k(t) = \{Z(s)|t_0 \leq s < t\} \cup \{z_k(t)\}, \quad (6-8)$$

for $k \in K(t)$. Note that the combined observation space can be denoted

$$\Psi(t) = \{Z(s)|t_0 \leq s \leq t\} = \{Z(s)|t_0 \leq s < t\} \cup \{z_k(t) | k \in K(t)\}, \quad (6-9)$$

$$= \bigcup_{k=1}^{K(t)} \{Z(s)|t_0 \leq s < t\} \cup \{z_k(t)\}, \quad (6-10)$$

$$\Psi(t) = \bigcup_{k=1}^{K(t)} \Psi_k(t). \quad (6-11)$$

When computing the resultant posterior PDF of all $K(t)$ collects at time t in the J2000 reference frame, $\chi(t) = P(x(t)|\Psi(t))$, we will use a linear combination of $\{\Psi_k(t) | k \in K(t)\}$ as shown below: [1]

$$\rho = \sum_{k=1}^{K(t)} \left(\frac{1}{tr \chi_k(t)} \right)^{-1}, \quad (6-12)$$

$$a_k = \frac{\rho}{tr \chi_k(t)}, \quad (6-13)$$

$$\chi(t) = P(x(t)|\Psi(t)) = \sum_{k=1}^{K(t)} a_k \chi_k(t), \quad (6-14)$$

where $\chi(t)$ will represent the information state of the system after performing information fusion of all $K(t)$ collects at time t .

We can see the information state is clearly a function of the measurement sequence $Z(t)$. Furthermore, we can define sensor configurations $a_j(t)$ and $A(t)$ from $z_k(t)$,

$$a_j(t) = \{z_k(s) | z_k(s) \in \Psi(t), g_s(k) = j, t_0 \leq s \leq t\}, \quad (6-15)$$

$$A(t) = \bigcup_{j \in \{1, \dots, M\}} a_j(t), \quad (6-16)$$

where the $z_k(s) \in A(t) \subseteq \Psi(t)$ represent the measurements the collection plan $A(t)$ has selected for all times in the interval $[t_0, t]$ which achieves the optimization goals.

7. Appendix B: Lumos Anomaly Detector Unscented Schmidt Kalman Filter

7.1 USKF Background

The proposed USKF-based data anomaly detection processes, to be run in near-real time, require an estimation implementation that enables key parameters to be estimated. Other parameters that are not estimated but may contribute errors to estimation uncertainty will be “considered” during the estimation process (e.g., observation timing biases). This section describes the analytical implementation of this approach.

The USKF is utilized in order to account for, or “consider”, the uncertainty associated with non-estimated parameters. It incorporates the “consider covariance analysis” concept whereby known errors in model and state parameters can be “considered” to make the estimation uncertainty more representative (realistic). This allows a user to account for so-called *known unknowns* which is ultimately reflected as more “realistic” estimation covariances. Using key metric by-products of the USKF algorithm, anomalies can be detected within the data provided to the filter.

Stauch and Jah [5] presented the USKF and noted that there are two general categories of consider techniques. One is consider covariance analysis, in which a typical state filter is executed and after the measurement update, the uncertainties of the consider parameters are mapped into the state space. The other is a consider filter, in which the state itself is augmented with the consider parameters while the consider parameter value and uncertainty are forced to be unchanged. Thus, the consider parameters are directly included in the filtering process. The USKF algorithm used in this work is provided in equation form in Fig. 27. where the predictive and update steps are highlighted.

Note that $X_{i,k}$ and P_k are the state and covariance of the estimated parameters only, $Z_{i,k}$ and $P_{zz,k}$ are the augmented state and covariance (i.e., both estimated and considered parameters) that include the consider parameters. Notice that the key difference between the filters is that the update to the consider state and covariance terms are forced to be zero, while the consider-estimated parameter cross-covariance term updates are maintained. In reality this makes the USKF a sub-optimal filter but one that is useful in preventing a falsely optimistic estimate. This is sometimes referred to as “covariance realism.” Parameters such as measurement related biases can be considered until reference satellite data are available.

Predictive

$$S_{zz,k-1} = \text{Cholesky}(P_{zz,k-1})$$

$$Z_{i,k-1} = \hat{z}_{k-1} \pm \sqrt{n_x + n_c} s_{i,k-1}$$

where $S_{zz} = [s_1, \dots, s_{n_x+n_c}]$

$$w = \frac{1}{2(n_x+n_c)}$$

$$Z_{i,k} \leftarrow \dot{Z}_i = f(Z_{i,k-1}, t)$$

$$\hat{z}_k = \sum_{i=1}^{2(n_x+n_c)} w_i Z_{i,k}$$

$$P_{zz,k} = \sum_{i=1}^{2(n_x+n_c)} w_i (Z_{i,k} - \hat{z}_k)(Z_{i,k} - \hat{z}_k)^T$$

Corrective

$$Y_i = h(Z_i, t)$$

$$\hat{y} = \sum_{i=1}^{2(n_x+n_c)} w_i Y_i$$

$$P_{yy} = \sum_{i=1}^{2(n_x+n_c)} w_i (Y_i - \hat{y})(Y_i - \hat{y})^T + R$$

$$P_{zy} = \sum_{i=1}^{2(n_x+n_c)} w_i (Z_i - \hat{z})(Y_i - \hat{y})^T$$

$$\begin{bmatrix} P_{xy} \\ P_{cy} \end{bmatrix} = P_{zy}$$

$$K_z = P_{zy} P_{yy}^{-1} = \begin{bmatrix} K_x \\ K_c \end{bmatrix} \quad \left(\begin{array}{l} \text{NOTE:} \\ K_c \neq 0!! \end{array} \right)$$

Force correction to consider terms to be 0:

$$\hat{z}^+ = \hat{z}^- + \begin{bmatrix} K_x \\ 0 \end{bmatrix} (y - \hat{y})$$

$$P_{zz}^+ = \begin{bmatrix} P_{xx}^- & P_{xc}^- \\ P_{cx}^- & P_{cc}^- \end{bmatrix} - \begin{bmatrix} K_x P_{yy} K_x^T & K_x P_{yy} K_c^T \\ K_c P_{yy} K_x^T & 0 \end{bmatrix}$$

Fig. 27. USKF Formulation as Derived In [5]

7.2 Consider Parameter Implementation

Parameters in an estimator can either be ignored, considered, or estimated (referred to as “ice”). In this application the state consists of satellite position, velocity, solar radiation pressure and sensor related biases (e.g., time-tag bias). In some instances, some of these filter parameters may not be “observable,” i.e., there is insufficient information in the observations to estimate them. In this case we might “consider” the parameter – that is, account for our knowledge of its uncertainty in the filter estimates and covariance without estimating it. In the data anomaly implementation, we might start with a “new” or uncalibrated sensor with a “consider” error to account for an as yet undetermined bias. This would allow the filter to continue processing until sufficient data and/or information are available to estimate the true bias.

7.3 USKF Time Bias Formulation

Though, in general, all forms of error (e.g., biases, systematic and periodic errors) are of interest, the subsequent use-cases model a sensor timing bias to illustrate the near real-time calibration using the USKF. Hence, in order to either estimate or consider the timing bias, it must be included in the USKF state along with any other estimated parameters (e.g., position, velocity, and solar radiation pressure). The timing bias finds its way into the USKF via the EO reference measurements derived from the reference satellite. The reference satellite is tracked by the EO sensor and the EO sensor inertial state (derived from the site coordinates) at the measurement time is corrected for the timing bias. At the time of each measurement update the state-vector sigma points are used to compute an equivalent measurement sigma point and these are adjusted for the current best estimate of the timing bias as follows

$$t_{corrected} = t_{observation} - t_{bias} \quad (7-1)$$

$$\vec{R}_{J2000} = [T_{ITRF \rightarrow J2000}(t_{corrected})] \vec{R}_{ITRF} \quad (7-2)$$

$$\vec{\rho} = \vec{r}_{J2000} - \vec{R}_{J2000} - \vec{v}_{J2000} \cdot (t_{bias} + \delta t_{LTC}) \quad (7-3)$$

$$\rho = |\vec{\rho}| = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2} \quad (7-4)$$

$$\alpha = \tan^{-1} \left(\frac{\rho_y}{\rho_x} \right) \quad (7-5)$$

$$\delta = \sin^{-1} \left(\frac{\rho_z}{\rho} \right) \quad (7-6)$$

where \vec{R}_{J2000} is the sensor inertial position; \vec{r}_{J2000} and \vec{v}_{J2000} are the satellite inertial position and velocity; $\vec{\rho}$ is the range vector between the sensor and satellite; α and δ are the “computed” optical measurements right ascension (RA) and declination (DEC); and δt_{LTC} is the light travel time correction that is applied to the optical measurements.

Within the USKF the state of the filter contains multiple RSOs and the timing bias of the different sensors being processed. For example, for two RSO states “1” and “2” and two sensor biases we would formulate our multi-state vector as

$$\vec{X} = \begin{bmatrix} \vec{X}_{rso_1} \\ \vec{X}_{rso_2} \\ \delta \vec{t} \end{bmatrix} \quad (7-7)$$

the RSO state is

$$\vec{X}_{rsoi} = \begin{bmatrix} \vec{r}_{rsoi} \\ \vec{v}_{rsoi} \\ \gamma_{rsoi} \end{bmatrix} \quad (7-8)$$

the optical sensor bias state is

$$\delta \vec{t} = \begin{bmatrix} \delta t_1 \\ \delta t_2 \end{bmatrix} \quad (7-9)$$

The two optical sensor biases assumed to be timing (other biases can be included in the state and estimated as appropriate) are δt_1 and δt_2 . The position and velocity cartesian vectors for the reference and RSO are

$$\vec{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (7-10)$$

$$\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (7-11)$$

and the relevant solar radiation pressure term for each is defined as

$$\gamma = C_r \frac{A}{m} \quad (7-12)$$

where C_r is the radiation pressure coefficient, A is the effective cross-sectional area and m is the mass.

The filter is also designed to process multiple measurement types, such as EO and state observations either in RA and DEC or \vec{r}_{rso} as shown below.

$$y = \begin{bmatrix} RA \\ Dec \end{bmatrix} \quad (7-13)$$

$$y = [\vec{r}_{rso}] \quad (7-14)$$

Range (e.g., RADAR or laser) measurement types have also been incorporated into the USKF to accommodate these ground-based sensor types. For an EO observation we get a RA and DEC measurement with a timestamp. For the state observations we get a position measurement of the state and a timestamp. Generally, the state observation is on a reference satellite and can be incorporated if available to help improve the results of the filter.