Dynamic Tasking of Networked Sensors Using Covariance Information

Keric Hill, Paul Sydney, Randy Cortez, Kris Hamada, and Daron Nishimoto
Pacific Defense Solutions, LLC, 1300 N. Holopono St., Suite 116, Kihei, HI 96753.

Kim Luu and Paul W. Schumacher, Jr.
Air Force Research Laboratory, 535 Lipoa Parkway, Suite 200, Kihei, Hawaii 96753.

ABSTRACT

A comprehensive high-fidelity simulation environment of networked optical space surveillance sensors has been created under an effort called TASMAN (Tasking Autonomous Sensors in a Multiple Application Network). One of the first studies utilizing this environment was focused on a novel resource management approach, namely covariance-based tasking. Under this scheme, the state error covariance of resident space objects (RSO), sensor characteristics, and sensor-target geometry were used to determine the effectiveness of future observations in reducing the uncertainty in orbit estimates. The different observation effectiveness metrics evaluated in this study predicted the amount of error reduction in the position, velocity, or semimajor axis estimate for the RSO. These observation effectiveness metrics were used to schedule the most effective times for the sensors to observe the RSOs in different sensor tasking scenarios. The tasking scenarios included fully distributed sensor schedulers, a fully centralized network scheduler, and a baseline case that did not use observation effectiveness in order to roughly mirror the current Space Surveillance Network tasking. The different tasking and scheduling techniques were compared by evaluating their impact on the accuracy of the orbit estimates in the RSO catalog. The observation effectiveness metrics measuring reduction in position or velocity error all substantially reduced the errors in the catalog estimates when compared to the baseline tasking scenario and would enable a more efficient use of the sensor network for catalog maintenance.

1. INTRODUCTION

A primary goal of the Space Surveillance Network (SSN) is the maintenance of a catalog with accurate orbit information for Resident Space Objects (RSOs). Efficiently tasking a centrally managed network of sensors is complex and potentially difficult.[1] Typical tasking schemes used by the SSN involve a central tasker which selects RSOs to observe and prioritizes those observations for each geographically distributed sensor based on orbit regime, constraints on the elevation angle, the age of the element sets, and historical sensor performance.[2, 3] Creating an observation timeline, or scheduling, is performed at the local sensor level with schemes that may vary from sensor to sensor. These scheduling schemes for optical sensors typically emphasize low phase angle observations for higher acquisition rates or minimum mount movement to maximize throughput. Within this current method of operations, the opportunity to optimally schedule observations for the express purpose of improving orbit accuracy tends to be lost in the gap between the central tasker and the distributed schedulers. For example, covariance information, which is the primary measure of orbit accuracy, is not known to the localized schedulers which decide on observational timing. However, as will be shown later, the use of the RSO orbit error covariance for scheduling can lead to a more intelligent use of sensor resources and result in better RSO orbit error covariance.

1.1 Optimizing Sensor Network Utilization

Following the same pattern as previous research in this area,[4, 5] three different sensor tasking schemes, or scenarios, were evaluated in this study in order to compare and contrast their effectiveness. Sections 1.2 through 1.4 describe these schemes.

1.2 Centralized Tasker with Distributed Schedulers (Scenario 1)

This scenario is similar to the current SSN and uses a centralized tasker which has knowledge of covariance information but only rudimentary knowledge of individual sensor capability. Sensor schedulers are distributed and have no knowledge of covariance information but do have detailed knowledge of sensor capability. A tasking list of visible targets is created daily by the central tasker for each sensor with the targets separated into three categories of tasking priority based on calculations involving the catalog orbit error covariance. The local sensor schedulers use probability of detection, the number of tracks previously scheduled, and the number of opportunities remaining to prioritize the RSOs and create an observation timeline.

1.3 Distributed Mission Planning (Scenario 2)

In this scenario there is no real distinction between the tasker and the scheduler. In effect, each RSO is assigned the same tasking priority and the distributed sensor schedulers have full knowledge of the orbit error covariance for each object. The ability to predict the relative effectiveness of observations using covariance is central to this scenario. Schedulers in this scenario assign observation times of RSOs based upon observation effectiveness from among the list of potential observation times during the tasking period. Once an observation is planned, the sensor predicts the effect that data will have on the RSO orbit error covariance and observation effectiveness when planning other observations. However, the sensor does not have information on how the schedules of the other sensors will affect the catalog.

1.4 Centralized Mission Planning (Scenario 3)

In this scenario, there is a single, centralized mission planner consisting of a co-located tasker and scheduler with knowledge of the orbit error covariance for each RSO and detailed knowledge of the parameters describing each sensor. As in Scenario 2, the scheduler uses observation effectiveness calculations to optimize observation timing and improve RSO catalog accuracy. Contrary to Scenario 2, the centralized nature of the scheduler potentially allows better coordination and utilization of the entire sensor network and thus better optimization of the space object catalog as a whole. Once a sensor is scheduled to observe a certain object at a certain time, the network scheduler predicts the effect that will have on the RSO orbit error covariance and observation effectiveness at others times in the tasking period for all the network sensors.

1.5 Covariance-based Scheduling

A method has been presented in which the orbit error covariance, observation geometry, and sensor noise characteristics were used to predict the effectiveness of future observations in reducing the position error variance of the RSOs.[4] This was done using the measurement update equations of the conventional Kalman filter.

$$K = \bar{P}\tilde{H}^T\left(\tilde{H}\bar{P}\tilde{H}^T + W^{-1}\right)^{-1} \tag{1}$$

$$\hat{P} = (I - K\tilde{H})\bar{P} \tag{2}$$

K is the Kalman gain and I is an identity matrix. $\tilde{H}$ is a matrix which consists of the partial derivatives of the observations with respect to the state. $\bar{P}$ is the covariance based on previous measurements propagated forward to the time of the current measurement. W is a matrix of weights and is usually the inverse of the observation error covariance matrix. The quantity $\left(\tilde{H}\bar{P}\tilde{H}^T + W^{-1}\right)$ is another form for the innovations covariance. The updated covariance, $\hat{P}$, is equal to $\bar{P}$ minus a matrix $K\tilde{H}\bar{P}$:

$$\hat{P} = \bar{P} - K\tilde{H}\bar{P}. \tag{3}$$

The matrix $K\tilde{H}\bar{P}$ is the reduction in the covariance. In previous studies, the observation effectiveness metric β_{pos} was selected to be the trace of the 3×3 position portion of that covariance reduction, $[K\tilde{H}\bar{P}]_{pos}$.

$$\beta_{pos} = tr\left([K\tilde{H}\bar{P}]_{pos}\right) \tag{4}$$

The values of β_{pos} are equal to the reduction in the 3D position error variance from one observation.

Three new observation effectiveness metrics were evaluated for this study to assess whether they offer advantages over β_{pos}. The first new metric, β_{vel}, is the trace of the 3×3 velocity portion of the covariance reduction, $[K\tilde{H}\bar{P}]_{vel}$.

$$\beta_{vel} = tr\left([K\tilde{H}\bar{P}]_{vel}\right) \tag{5}$$

Similar to β_{pos}, the β_{vel} values indicate the reduction in the 3D velocity error variance from one observation. Since velocity may have more of an effect on the error growth of an orbit estimate, a metric that more effectively reduces velocity error could result in less error when propagating the orbit estimate.

The second new metric, β_{semi}, measures the reduction in the semimajor axis error variance from one observation. It is made by transforming the 6×6 Cartesian orbit error covariance to a scalar semimajor axis variance using a matrix of partial derivatives, γ, which is

$$\begin{aligned} \gamma &= \begin{bmatrix} \frac{\partial a}{\partial x} & \frac{\partial a}{\partial y} & \frac{\partial a}{\partial z} & \frac{\partial a}{\partial \dot{x}} & \frac{\partial a}{\partial \dot{y}} & \frac{\partial a}{\partial \dot{z}} \end{bmatrix} && (6) \\ &= \begin{bmatrix} \frac{2a^2 x}{r^3} & \frac{2a^2 y}{r^3} & \frac{2a^2 z}{r^3} & \frac{2a^2 \dot{x}}{\mu} & \frac{2a^2 \dot{y}}{\mu} & \frac{2a^2 \dot{z}}{\mu} \end{bmatrix}. && (7) \end{aligned}$$

In these equations, a is the semimajor axis, r is the magnitude of the RSO radius vector, μ is the gravitational parameter of the Earth, and x, y, z, $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the position and velocity of the RSO. Using γ, β_{semi} is computed in this way

$$\beta_{semi} = \gamma[K\tilde{H}\bar{P}]\gamma^T. \tag{8}$$

Since the error in semimajor axis is the primary source of error growth in an orbit estimate, reducing error in the semimajor axis estimate should also reduce error growth.

The third new metric is called β_{frob} is similar to β_{pos}, but uses the Frobenius norm, $||\cdot||_F$, of the position portion of the covariance reduction matrix

$$\beta_{frob} = \left|\left|[K\tilde{H}\bar{P}]_{pos}\right|\right|_F \tag{9}$$

Since the Frobenius norm also accounts for the off-diagonal terms of the position covariance, it is possible that it captures more essential information for scheduling than the trace of the position covariance.

Using these estimates of the predicted effectiveness of observations, a method was devised to plan the timing and selection of RSOs for observations in order to take advantage of the most effective sensor-target geometry. To compute the observation effectiveness, or β-metrics, two passes through a Kalman filter are required. On the first pass, the latest orbit update for the target is used for the *a priori* state and covariance. The filter performs time updates to move forward over the time span to be scheduled. In addition, measurement updates are performed on the covariance for each observation that has already been committed to the mission plan. Note that the measurement update is not performed on the state since there are not any observations to process yet. At the end of the first pass through the Kalman filter, the covariance must be propagated backward in time to the beginning of the time span to be scheduled. Then the second pass through the Kalman filter uses the latest orbit update for the target state and the back-propagated covariance from the first filter pass for the *a priori* state and covariance. On the second pass, the selected β is computed and stored. Note that since β is computed in the second filter pass, it is in context of all observations already committed to the mission plan, both in the past and the future.

2. TEST-BED AND RESEARCH ENVIRONMENT

Tasking Autonomous Sensors in a Multiple Application Network (TASMAN) is a flexible optical sensor network simulation that can be used to combine advanced mission planning algorithms with a distributed set of sensor simulation nodes. TASMAN is capable of simulating a single point tasking interface or distributed taskers while incorporating different local sensor measurement error models. Previous studies with TASMAN evaluated covariance-based scheduling techniques and incorporated dynamic retasking due to sensor outages.[5] The TASMAN software suite consists of a network communication architecture, simulation clock, sensor model, detection model, orbit determination, and tasker/scheduler modules.

2.1 Network Communication Architecture

The sensor network in TASMAN consists of a central node and multiple sensor nodes that communicate via the Simulation Network Auto-configuration Protocol (SNAP). SNAP is a flexible publish-and-subscribe network communication protocol that allows different TASMAN software modules to communicate with one another by passing XML wrapped TASMAN Message objects between modules during a simulation.

2.2 Simulation Clock

The Simulation Clock is an asynchronous clock that controls the simulation time for all modules in the TASMAN architecture. The clock publishes a time window within which modules can operate. New clock updates are sent only when all modules are finished performing their respective functions for the previously published time window.

2.3 Truth Orbit Generation

On simulation startup, the truth orbits are generated for the entire simulation period and stored in a MySQL database on the Central Node. TASMAN includes various orbit propagators to generate truth orbits for the simulation using a development branch of a package called TurboProp.[6] The options for orbit propagators in TurboProp include a fixed-step fourth-order Runge-Kutta solver, a variable-step fourth/fifth-order Runge-Kutta solver, and a variable-step seventh/eighth-order Runge-Kutta solver. The surface station/vehicle propagator generates the coordinates of a stationary sensor or a simple vehicle on the Earth or the Moon in inertial space. A SGP4/SDP4 orbit propagator uses NORAD general perturbation theories and Two-Line Element (TLE) sets. The force models include the JPL planetary ephemerides DE403, DE405, and DE421; the lunar gravity models GLGM-2, LP100K, and LP150Q; the Earth gravity models GGM02C, JGM-3, and WGS-84; exponential atmosphere models, and the NRLMSISE-00 atmospheric drag model for Earth. Solar radiation pressure (SRP) is modeled with a simple umbra/penumbra representation and a constant reflectance. Both drag and SRP use an assumed constant spacecraft cross-sectional area.

2.4 Sensor Model

The sensor models on the Sensor Nodes generate simulated observations for RSOs to different levels of fidelity using the truth ephemerides. TASMAN development proceeded in two phases. Phase 1 used a low-fidelity sensor model in the initial phase of development. Under this phase, geometric angles measurements with Gaussian noise are generated by a government-developed sensor simulation package called jSim or by the sensor schedulers. The measurements are selected based on a probability of detection algorithm that uses a pre-generated table based on the estimated signal to noise ratio (SNR) performance of real detection algorithms on a large selection of estimated target brightness levels with Monte-Carlo generated noise levels. The resulting observations are sent directly to the Orbit Determination module. Figure 1 shows the configuration of the TASMAN simulation for Phase 1 and Phase 2.

Under Phase 2, high-fidelity sensor models were added for realistic scene generation. The Naval Observatory Merged Astrometric Dataset (NOMAD) is used to simulate the space environment for the sensors. The Sensor Model for Phase 2 has two parts, the jSim and the High Fidelity Scene Simulation (SceneSim) packages.

jSim is the physics engine that provides target geometry as seen by the focal plane (sensor). It has the capability to simulate sensor gimbal motion and pointing using a variety of mount models, at the specified geographic site and temporal sampling rate. It also selects stars and orients them within the sensor field of view. The generated target geometries are written to a file that another government-developed package called SceneSim reads in to create images. SceneSim was developed as a method of generating realistic imagery for wide field of view sensors. It uses both spatial and temporal oversampling to precisely position objects to an angular accuracy that is a fraction of the inscribed field of view of a single pixel. Realistic noise terms such as read noise, zodiacal light, Poisson noise, and dark current are included. There also exists the capability of inserting gain-maps and optical distortion terms to generate highly realistic imagery challenges for detection algorithms.

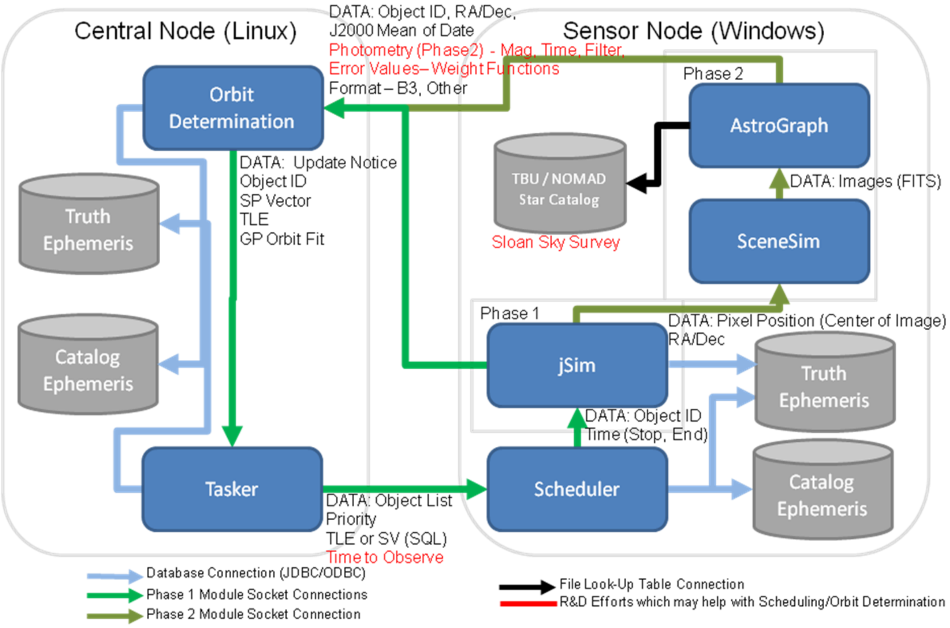

Figure 1: TASMAN Phase 1 and Phase 2 Illustration.

2.5 Detection Algorithm

The automated target detection and processing module used with the Phase 2 high-fidelity sensor models is called AstroGraph. AstroGraph is both a library and a tool that automatically performs astrometric and photometric measurements on simulated or real optical sensor imagery. It is used to detect objects in the scenes generated by SceneSim and associate the metric and photometric measurements to an object in the catalog. The resulting lists of angles measurements, or tracks, are sent to the Orbit Determination module.

2.6 Orbit Determination

The orbit determination module on the Central Node receives and processes tracks from the Sensor Nodes and updates the RSO catalog state vectors using an Unscented Kalman Filter (UKF). The UKF allows real time orbit updates and was written using descriptions by Julier and Uhlmann[7] and Crassidis and Markley.[8] The UKF models the dynamics of state deviations by propagating a group of states called sigma-points instead of linearizing about a single reference trajectory. When using the UKF, several sigma-point trajectories are propagated for each time update, but the increase in computation time is partially offset by eliminating the need to compute the partial derivatives used when integrating the state transition matrix, as well as the partials that define the observation-state relationship. The UKF therefore provides a robust method of estimation from inaccurate initial conditions that avoids inaccuracies caused by linearization while providing an acceptably rapid estimation of the orbit state.

2.7 Tasker and Scheduler

The orbit updates from the orbit determination module are used to create task lists and schedules in a variety of tasking architectures. These are passed to the Sensor Nodes, creating a closed simulation loop.

3. METHODS

3.1 Current Simulation Configuration

For this paper, high-fidelity scenes were not required so TASMAN was configured to run in a modified Phase 1 mode. Instead of using jSim, the sensor schedulers were modified to produce observations of the RSOs in

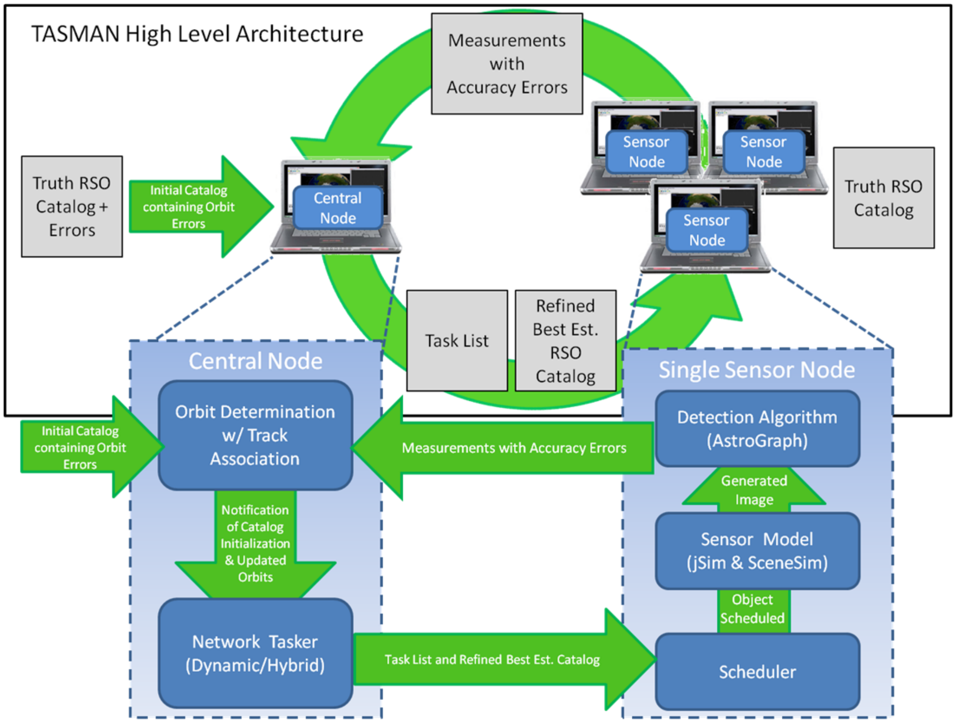
TASMAN High Level Architecture
Measurements with Accuracy Errors
Truth RSO Catalog + Errors
Initial Catalog containing Orbit Errors
Central Node
Sensor Node
Sensor Node
Sensor Node
Truth RSO Catalog
Task List
Refined Best Est. RSO Catalog
Central Node
Orbit Determination w/ Track Association
Initial Catalog containing Orbit Errors
Measurements with Accuracy Errors
Notification of Catalog Initialization & Updated Orbits
Network Tasker (Dynamic/Hybrid)
Task List and Refined Best Est. Catalog
Single Sensor Node
Detection Algorithm (AstroGraph)
Generated Image
Sensor Model (jSim & SceneSim)
Object Scheduled
Scheduler

Figure 2: TASMAN Architecture.

48-second tracks of five angle pairs each with added random Gaussian error having a standard deviation of 1 arc second. Objects were only observed above 20 degrees of elevation angle. These observations are directly sent to the orbit determination module to estimate a new orbit. The runs span an 8-day simulation period. Truth orbits were generated using TLEs accessed from the Space-track website (www.space-track.org) on December 15, 2009. Of all the TLEs available in the catalog, objects in GPS- or Glonass-type orbits were selected using the criteria in Table 1. The 213 resulting objects included more than just the operational satellites.

Table 1: Criteria for Selection of Objects in GPS- or GLONASS-type Orbits.

	Semimajor axis (km)	Eccentricity	Inclination (deg)	Radius of perigee (km)
Minimum	25,000	0	50	25,000
Maximum	28,000	0.05	70	28,000

For the purpose of simplifying the simulation initially, the TLE coordinate system was treated as True of Date instead of the NORAD coordinate frame and converted to Cartesian position and velocity. The TurboProp orbit integrator used to propagate truth orbits and estimate orbits in the orbit determination module was a fourth-order Runge-Kutta method using a 60-second step size. The force model includes Earth point mass gravity plus J_2 and J_3. Sensor sites were selected to give adequate global coverage. More sites can be added to the simulation as needed to assess various sensor deployment strategies. The selected sites for the initial configuration are shown in Table 2.

Table 2: Simulation Site Locations.

Site	East Longitude (deg)	North Latitude (deg)	Ellipsoid Height (m)
Kwajalein	192.2667	8.716667	50
Albuquerque	253.502717	34.96305	1725
Moron, Spain	354.41194	37.1511	101

The initial catalog estimates were obtained by generating observations in a four-day precursor period immediately before the simulation time span. For the precursor period, observations were simulated for all RSOs for all visible passes at intervals of eight minutes. The resulting covariance matrix for each RSO was scaled to make the square root of the trace of the 3×3 velocity portion equal to 1×10^{-5} km/s. Scaling the covariance gives control over the error magnitudes while preserving the proper relationships between parameters through the correlation terms. Then the catalog orbit states were computed by randomly sampling the scaled covariance matrices and adding the resulting error vectors to the true orbit states.

Three tasking scenarios were evaluated while assuring that each resulted in the same number of observations per day. The limit on observations was set to 200 tracklets per sensor per day. This was so that any changes in catalog orbit accuracy would only be due to the sensor schedule and not the total number of observations. All the scheduling scenarios were based on 24-hour tasking periods. At the beginning of each 24-hour tasking period, before generating and processing the observations, the scheduler first determines when each RSO will be observed. After the scheduler is done creating the observation plan, the plan is used to generate the observations using the truth orbits and random noise, bypassing the high-fidelity sensor model and detection algorithm in this initial study.

In Scenario 1, the scheduler creates three priority bins for the catalog objects based on observation effectiveness. This is similar to how routinely tasked deep space objects are assigned as Category 3 through Category 5 for the SSN based on the element set age. Within each priority bin, the scheduler sorts the objects according to a merit function. The merit of an RSO is based on the combination of several factors. The probability of detection for the RSO is based on the normalized signal, $\widehat{S}_{rso}$ from a diffuse object with a solar-illuminaton phase angle, ϕ and a range, R from the sensor:

$$S_{rso} = \frac{sin(\phi) + (\pi - \phi)cos(\phi)}{R^2} \tag{10}$$

$$\widehat{S}_{rso} = \frac{S_{rso}}{max(S_{rso})} \tag{11}$$

The number of tracks already scheduled, $N_{scheduled}$ for the RSO is defined as the merit, $M_{scheduled}$:

$$M_{scheduled} = \begin{cases} 2 & \text{if } N_{scheduled} = 0 \\ \frac{1}{N_{scheduled}} & \text{otherwise} \end{cases} \tag{12}$$

and the number of available time slots for the RSO in the mission plan, $N_{available}$. The total merit of an RSO is then:

$$M_{RSO} = w_{S_{rso}} \cdot \widehat{S}_{rso} + M_{scheduled} + \frac{1}{N_{available}} \tag{13}$$

where $w_{S_{rso}}$ is a weight factor for probability of detection, which was set to 0.5. After the priority list is generated, the scheduler goes through the Category 3 list and first schedules the RSO with the highest merit. Then the merit is recomputed and the list is sorted again. Again, the RSO with the highest merit is scheduled. The process repeats until there are no more opportunities to observe Category 3 objects. Then the Category 4 list is then processed in order, according to merit, followed by the Category 5 list. The process stops once 200 tracklets are planned for that 24-hour period.

In Scenario 2, each sensor computes its own schedule independent from the others. The observation effectiveness is computed for one observation every two minutes during each visible pass, and the pass with the highest resulting β is scheduled first. Then the observation effectiveness is regenerated for each visible pass for the scheduled object, except during the intervals that the sensor has already been scheduled. Out of this new list, the track with the highest β is then scheduled. The process repeats until the sensor has scheduled 200 tracklets.

In Scenario 3, the observation effectiveness is computed for all the RSO-sensor combinations in the tasking period. The highest β from the combined list is scheduled, and the list is recomputed. In this way, sensors are collaborating more intelligently, and duplication of effort is reduced.

3.2 Catalog Accuracy

In order to assess the effect of different tasking and scheduling algorithms, the effect on catalog accuracy must be quantified. An accuracy metric for a single object is computed by taking the estimated catalog orbit for that object and differencing it with the truth orbit over a 24-hour prediction. The resulting accuracy metric for that object is called, “MaxErr,” and is the largest 3D position error over the 24-hour prediction. A single quantity representing the accuracy of the entire catalog, called “Catalog Median,” is computed by taking the median of the MaxErr values of all the objects. The single worst case in the catalog is defined as having the largest value of MaxErr and is called the “Catalog Max.” Comparisons of the three studied scenarios are interpreted within the context of these performance metrics.

4. RESULTS

Nine different configurations were simulated, one for Scenario 1, and four for each type of β for both Scenario 2 and Scenario 3. For each configuration, 10 eight-day simulations were conducted. Table 3 shows the mean and standard deviation of the Catalog Median and Catalog Max values at the end of the eight days for the 10 runs of each configuration, with the same information presented graphically in Figures 3 and 4. The covariance-based schedulers all did better than Scenario 1. The metrics β_{pos}, β_{vel}, and β_{frob} all performed similarly as far as catalog accuracy. It is curious to see that β_{semi} did not perform as well as the others. The Scenario 3 centralized scheduler usually performed slightly better than the distributed Scenario 2 schedulers.

Table 3: Catalog Accuracy Statistics for the 10 Runs of Each Tasker/Scheduler Configuration.

Configuration	mean of Catalog Median (m)	stdev of Catalog Median (m)	mean of Catalog Max (m)	stdev of Catalog Max (m)
Scenario 1	53.87	2.8	16099	9644
Scenario 2 β_{pos}	29.42	1.3	195	63
Scenario 2 β_{vel}	30.21	1.4	215	47
Scenario 2 β_{semi}	45.85	1.8	731	714
Scenario 2 β_{frob}	29.50	1.2	211	74
Scenario 3 β_{pos}	27.98	0.75	196	45
Scenario 3 β_{vel}	29.71	1.6	246	84
Scenario 3 β_{semi}	45.71	2.0	972	1955
Scenario 3 β_{frob}	28.07	1.1	186	34

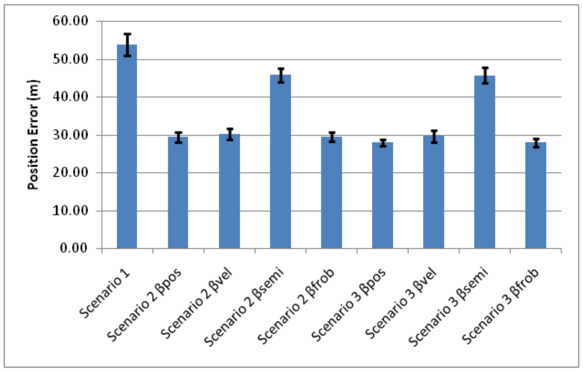

Figure 3: Ten-run mean for Catalog Median values at the end of simulation day 8, with black error bars showing the standard deviation.

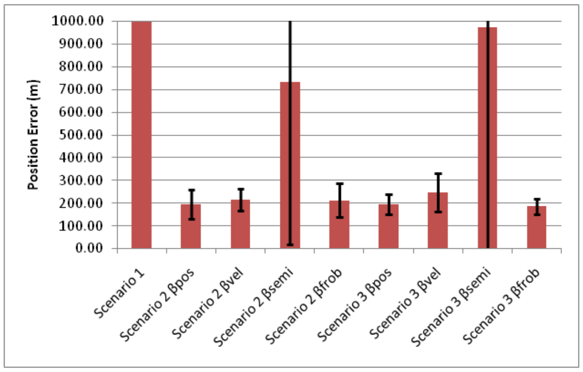

Figure 4: Ten-run mean for Catalog Max values at the end of simulation day 8, with black error bars showing the standard deviation.

Figures 5 and 6 show the the time histories of Catalog Max and Catalog Median for Scenarios 1, 2, and 3. For these figures, the catalog accuracy was plotted at the beginning of each 24-hour tasking cycle, as well as whenever an orbit update was performed for one of the RSOs. Notice Scenario 1 has a very large value for Catalog Max, and it does not get much better as the simulation progresses. That means that some objects have large errors in the orbit estimate, but they just are not tasked and scheduled enough to bring the error

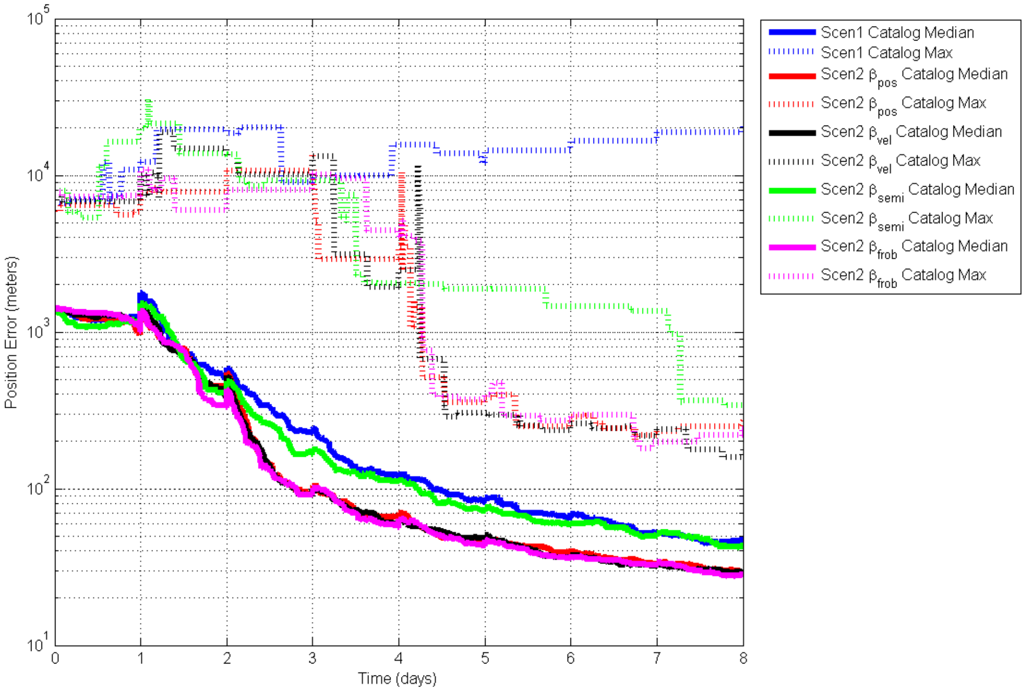

Figure 5: Sample time histories of catalog accuracy for Scenario 1 and Scenario 2 configurations.

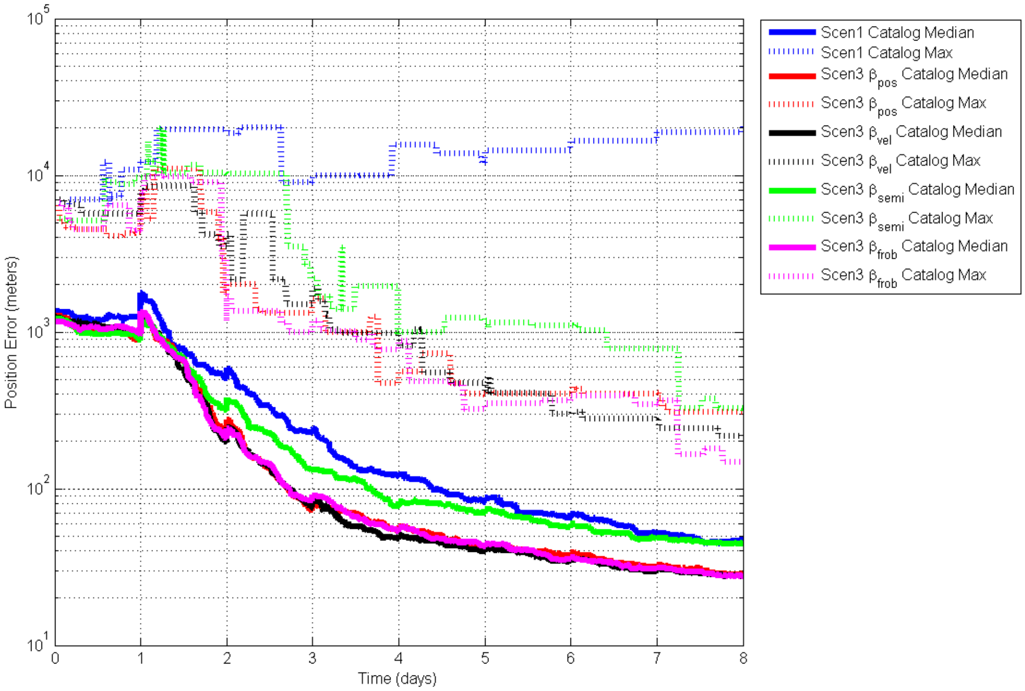

Figure 6: Sample time histories of catalog accuracy for Scenario 1 and Scenario 3 configurations.

down down, even though they are in the highest tasking category. However, the covariance-based schedulers (Scenario 2 and 3) are more effective at reducing errors for the worst objects in the catalog.

5. CONCLUSION

The most significant conclusion is that using covariance to schedule observations offers significant improvements in catalog accuracy and allows much more efficient use of a sensor network. In this initial study, three of the proposed observation effectiveness metrics, β_{pos}, β_{vel}, and β_{frob} resulted in similar catalog accuracies, which were much improved over the baseline tasking scenario. However, β_{semi} did not perform as well and was only slightly better than the baseline scenario. Although minimizing error in semimajor axis should minimize error growth, it is possible that periodic or constant position errors are dominating the maximum position error for the 24-hour predictions. Periodic errors would be due to uncertainty in eccentricity, right ascension of the ascending node, and inclination, and constant-offset position errors would be due to uncertainty in mean anomaly. This shows that it is important to target the correct state parameters for error reduction. Further research is required to understand the most effective way to apply RSO covariance information for sensor network tasking.

In reality, the catalog accuracy metric used in this work represents only one aspect of catalog maintenance. Consideration must also be given to other requirements such as timeliness, tracking priorities for RSOs, and accuracy requirements in different orbit regimes. For example, active payloads could have different orbit accuracy and timeliness requirements than debris objects. Future work will continue to investigate other means of using covariance to more efficiently task and schedule this type of sensor network that will be able to account for those aspects as well as the catalog accuracy.

References

[1] B. C. Weeden and T. S. Kelso, “Analysis of the Technical Feasibility of Building an International Civil Space Situational Awareness System,” Paper IAC-09.A6.5.2, International Astronautical Congress, Daejeong, South Korea, Oct. 12-16, 2009.

[2] J. G. Miller, “A New Sensor Resource Allocation Algorithm for the Space Surveillance Network in Support of the Special Perturbations Satellite Catalog,” AAS 03-669, AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, Aug. 3-7, 2003.

[3] J. G. Miller, “A New Sensor Allocation Algorithm for the Space Surveillance Network,” 74th Military Operations Research Society Symposium, Colorado Springs, CO, Aug. 28, 2006.

[4] K. Hill, P. Sydney, K. Hamada, R. Cortez, K. Luu, M. Jah, P. W. Schumacher, M. Coulman, J. Houchard, and D. Naho’olewa, “Covariance-based Network Tasking of Optical Sensors,” AAS 10-150, AAS/AIAA Space Flight Mechanics Conference, San Diego, CA, Feb. 14-17, 2010.

[5] K. Hill, P. Sydney, K. Hamada, R. Cortez, K. Luu, P. W. Schumacher, and M. Jah, “Covariance-based Scheduling of a Network of Optical Sensors,” AAS 10-325, AAS/AIAA Kyle T. Alfriend Astrodynamics Symposium, Monterey, CA, May 17-19, 2010.

[6] K. Hill, “TurboProp Version 3.0,” Colorado Center For Astrodynamics Research, http://ccar.colorado.edu/geryon/papers/TurboProp_3_0_Manual.pdf, February 12, 2007.

[7] S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182–193.

[8] J. L. Crassidis and F. L. Markley, “Unscented Filtering for Spacecraft Attitude Estimation,” 2003. AIAA 2003-5484, AIAA Guidance, Navigation, and Control Conference Austin, Texas, Aug. 11-14.