Dragster: An Ensemble Assimilative Model for Satellite Drag

Ryan Blay, Connor Johnstone, Jeremy Highley, Junk Wilson, John Noto, Geoff Crowley
Orion Space Solutions, Louisville, CO 80027

ABSTRACT SUMMARY

As the population of resident space objects continues to grow steadily, the need for greater knowledge of the space environment and orbiting elements will increase as well. To truly achieve actionable and timely collision avoidance processes, improved atmospheric density and drag models are required. Orion Space Solutions has developed a drag specification and forecast tool called "Dragster" based on well-validated, full-physics atmospheric models and ensemble data assimilation techniques. Timely access to Dragster model output will allow analysts to utilize the enhanced satellite orbit specification and predictions provided by Dragster, reducing the number of false positives. In the comparisons between Dragster and the High-Accuracy Satellite Drag Model (HASDM) performed to date, Dragster generally outperforms HASDM. The primary innovation of the presented work is to explore the prediction of uncertainties associated with the density and drag predictions from the Dragster ensemble assimilative model. Presented here is a detailed look at Dragster's methodology and validation results, showing the improvement upon the state-of-the-art HASDM. The SSA/SDA technical community will be interested in the reduced error and importantly the quantification of this lower drag uncertainty.

1. INTRODUCTION

A significant challenge of global Space Domain Awareness is the sheer number of objects that are now in orbit around Earth. As the demand for satellites increases, and the density of satellites in space increases, there is a need for more robust space domain awareness capabilities, specifically the prediction of satellite orbits. Perturbing forces which change the orbital trajectory of space objects can make this process difficult, resulting in greater uncertainty around collision avoidance and maneuvering, causing optical sensors to spend additional time searching for objects they are tasked to track. When these forces are modeled or represented incorrectly, errors in orbital position accumulate, potentially causing ambiguity between objects for optical sensors. Atmospheric drag is especially difficult to specify for the Low Earth Orbit (LEO) regime (below 2000 km altitude) [8], yet this is where most satellites reside.

The accuracy of current modeling of space object density is unsophisticated enough that it does not provide estimates of the uncertainty regarding drag or object position. More accurate modeling can increase accuracy, decrease search volumes, and thus increase the capacity of individual sensors to make additional observations. For satellite operators, conjunction assessment and collision avoidance are daily issues which depend upon having high accuracy orbit estimates with well-characterized uncertainties and recent observations. Any loss in orbit estimate accuracy significantly increases the chances of a catastrophic collision. Existing atmospheric drag models are not robust enough to provide accurate predictions and cannot take advantage of additional available data, or improved forecasting techniques.

2. SPACE ENVIRONMENT OVERVIEW

The ionosphere and upper atmosphere play a major role in LEO operations, including communications, navigation, surveillance, and satellite drag. Satellite drag is the leading cause of orbit prediction error in low earth orbit. As the population of satellites in Earth orbit grows with time, higher orbital prediction accuracy is required to enable accurate catalog maintenance, collision avoidance for manned and unmanned space flight, reentry prediction, satellite lifetime prediction, defining on-board fuel requirements, and satellite attitude control. Satellite drag varies strongly as a function of the neutral thermospheric density and ballistic coefficient. Aerodynamic drag acceleration (a_{drag}) is expressed by the equation below in terms of atmospheric density (ρ), drag coefficient (C_D), cross-sectional area (A), spacecraft mass (m), and the spacecraft velocity relative to the atmosphere (V_r).

The drag acceleration is the aerodynamic acceleration projected in the direction of satellite velocity. Many satellites also experience non-negligible lift forces which can cause long-term changes in the orbital inclination as well as aerodynamic torques which can alter the attitude state of the satellite. The inverse ballistic coefficient is often used to describe the non-atmospheric contributions to satellite drag as shown below.

Equation 1: Aerodynamic Drag

$$a_{drag} = \frac{1}{2}\rho \frac{C_D A}{m} V_r^2$$

The drag acceleration is the aerodynamic acceleration projected in the direction of satellite velocity. Additionally, objects may experience non-negligible lift forces which can cause long-term changes in the orbital inclination as well as aerodynamic torques, which can alter the attitude state of the satellite. The inverse ballistic coefficient is often used to describe the non-atmospheric contributions to satellite drag, as shown below.

Equation 2: Ballistic Coefficient

$$B = \frac{C_D A}{m}$$

Thermospheric density is the most variable parameter, at a constant altitude, changing by as much as 200-800% [1] due to changes in geomagnetic activity levels. Here we define variability as the total change of a parameter divided by the initial value of the parameter. The product of C_D and A is the second contribution to drag variability with variations for elongated satellites flying above 180 km as large as approximately 100% [2, 3]. Another 25%-50% change can be expected in the product of C_D and A below 180 km due to transition effects [4, 5]. Changes in atmospheric winds can lead to changes in V_r on the order of 3% 1-σ, with maximum effects on the order of 15% during large geomagnetic storms. Some orbiting objects experience propellant leaks and breakup, in which case the changes in mass can cause drag to be altered by significant amounts (sometimes in excess of 100%) [4]. Uncertainty in changes from one solar cycle to the next adds additional urgency to the problem of modeling thermospheric density. Therefore, the density prediction must respond appropriately to solar and geomagnetic forcing terms to enhance the Space Traffic Management effort.

3. DRAGSTER OVERVIEW

Dragster is a modeling framework developed by Orion and based on state-of-the-art full-physics models of the coupled thermosphere-ionosphere running in real-time with inputs comprising near-real time and predicted space weather data and indices. It uses assimilative techniques to produce both a nowcast and 72-hour predictions of the global thermosphere-ionosphere system. Dragster's system architecture is shown below, in Fig. 1.

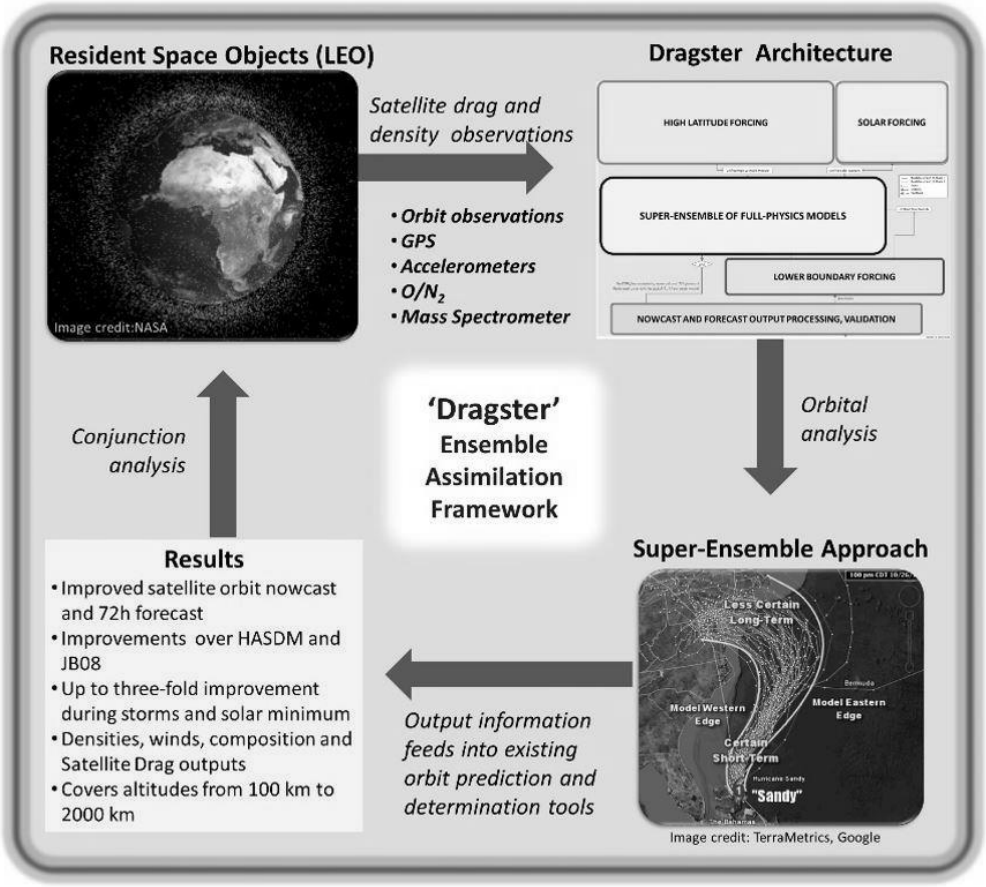

Fig. 1: Dragster Operation View 1 (OV-1)

Dragster utilizes the 3-D global Thermosphere-Ionosphere-Electrodynamics General Circulation Model (TIE-GCM) to solve for the time dependent non-linear momentum, energy, and composition equations to provide neutral dynamics, temperature, and the distribution of neutral species. The 3-D distribution of neutral density is obtained from the temperature and composition, which, together with the neutral winds, provide the necessary parameters for satellite drag prediction. The self-consistent ionosphere is important and necessary to ensure accurate conductivities and for characterizing high latitude Joule heating, ion drag, and realistic wind determination. The model is driven by three main inputs: solar radiation, high-latitude forcing, and tides propagating from the lower atmosphere. TIE-GCM runs include variable high-latitude forcing specification based on Assimilative Mapping of Ionospheric Electrodynamics (AMIE) model files. The lower boundary includes the specification of the upward propagating tides. The specification of seasonal eddy diffusion near 90 km can also be included via input flags of the TIE-GCM interface. The eddy diffusion affects the altitude distribution of various neutral species (e.g., N2, O2, O) and therefore the overall density profile [6]. The physical underpinnings of this variation are still being investigated and Orion recommends this feature should only be used for research purposes until further validation work is done. The upper boundary of Dragster has been extended into the Helium-rich exosphere and data assimilation has been demonstrated at altitudes up to and beyond 800 km. Helium is the dominant neutral constituent above ~ 500 km at solar minimum. Both the TIE-GCM and Naval Research Laboratory Mass Spectrometer and Incoherent Scatter (NRLMSIS-00, henceforth MSIS) models include Helium components, and the Assimilation Engine and TIE-GCM software is designed to accommodate objects well above the 500 km oxygen-helium transition. The current operational Space Force model, called the High Accuracy Satellite Drag Model (HASDM), does not account for this transition to a helium-rich atmosphere.

Dragster utilizes MSIS as an alternative background model. The software architecture was designed to be modular so that as atmospheric models evolve, such as MSIS v2, they can be included into Dragster for evaluation/testbed purposes. This forward-looking design means that the Dragster tool can be optimized to leverage an ensemble of models most appropriate for the task at hand, and future improvements to any such models will flow directly into

Dragster performance. For example, several "whole atmosphere" models currently being developed, including WAM-IPE, could be introduced into the Dragster framework to broaden the 'ensemble' or, if unsuitable for operational use, can be compared for operations. Dragster propagates the model ensemble members forward to predict the most probable trajectory of the thermosphere state and its uncertainty based on inter-model differences, similar to forecasting the path of a hurricane (See Fig. 1). Unlike tropospheric weather, the thermosphere is strongly driven by extraterrestrial inputs and depends less on the prior states.

A key requirement for improving thermosphere density models, and to provide the best current densities, is the ability to assimilate data. Dragster incorporates the ability to assimilate data into an ensemble of model instantiations using "full-physics" first-principles general circulation models (e.g., TIE-GCM) as well as empirical models (e.g., MSIS) of the atmosphere. It also has the ability to assimilate multiple sources and types of satellite drag-related data. This includes tabulated Energy Dissipation Rate (EDR) values that can be output by a number of Precise Orbit Determination (POD) techniques as well as from accelerometer data. It also includes operational datasets such as the DC summary files resulting from Special Perturbation (SP) orbit fits. Orion developed Dragster to assimilate density data by using a version of the Ensemble Kalman Filter (EnKF) to provide nowcasts of various atmospheric and drag parameters. Fig. 2 shows the Dragster EnKF algorithm top-level flow diagram. The algorithm begins in the upper left with the definition of the initial atmospheric state (X_0) for every ensemble member.

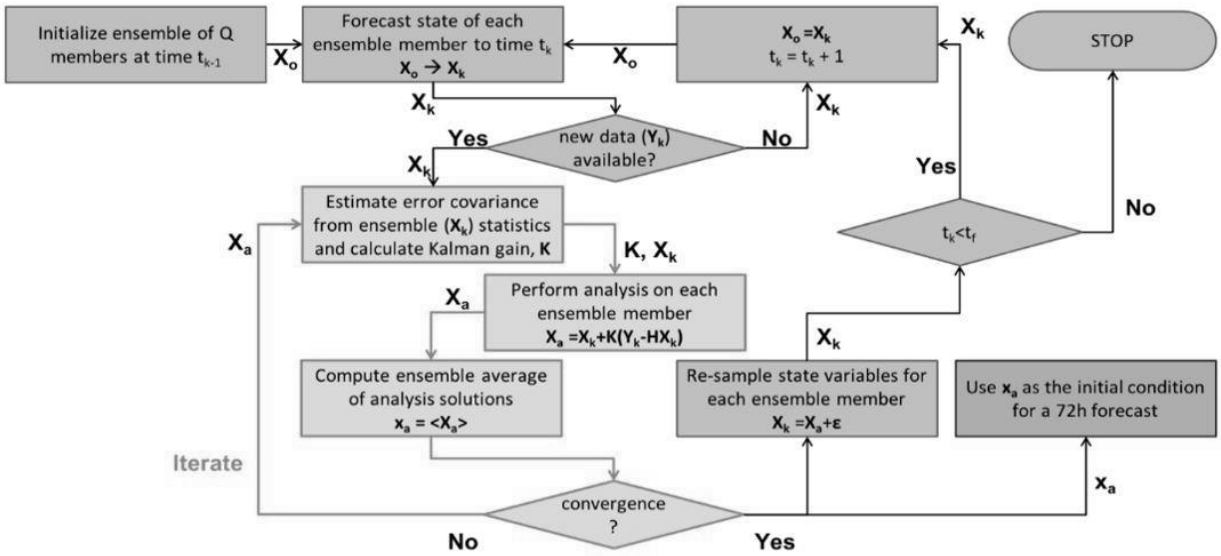

Fig. 2: Process flow diagram for Dragster

The states include a selectable span of model times to accommodate multi-bandwidth datasets. The software propagates all the atmospheric states to the current time and ingests new satellite drag data, if available. At this point, the ensemble computes the covariance matrices and Kalman gain. Unlike a classical EnKF, Dragster estimates covariance and forms state estimates iteratively at each assimilation time window, giving it similarities to a batch processing estimator. As the filter moves through time, the first step is to forecast the models forward. Then, as new data becomes available, the iterative part (orange boxes) of the EnKF is entered. The main objective of this loop is to calculate the error covariances from the ensemble members and the data measurements. These covariances then allow for the calculation of a Kalman gain (K). With the Kalman gain and the model densities, the analysis states can be computed for each ensemble member. An array of K values is stored reflecting the magnitude of change from time step T_0 to T_1. A solution X_a is obtained for each ensemble member and the average of these solutions (x_a) is used to initiate a forecast of satellite drag parameters at time T_1 in the future. This part of the process is performed iteratively until a pre-specified convergence criterion is met. The iteration and wide time-range incorporated into each state cause this part of the algorithm to resemble a batch processor within an EnKF architecture. Conjunction analysis and orbit prediction algorithms for Space Traffic Management can then be executed using the new drag and density values at time T_1.

Dragster delivers time-dependent density fields and includes the drag-force along the orbit for any given object specified by the operator. Object area, mass, and drag coefficient are specified by the user. The drag coefficient can also be specified by a choice of empirical model, a fitted-ballistic approach, or hybrid of the two. The user can also interpret the output density fields along with a catalogue of space objects.

4. DRAGSTER SOFTWARE ARCHITECTURE

To operationalize Dragster from its current TRL 5 status to a TRL 8 scalable platform that can be used simultaneously by multiple users, the existing application architecture needs to be transitioned from a monolithic to a cloud-based infrastructure. Monolithic applications are typically large, complex software systems that perform a wide range of tasks, often with a specific focus on scientific or engineering research. These applications are typically built as a single, integrated system that includes all the necessary functionality and data storage. As new features and technologies are added, and technologies evolve and change, the codebase becomes increasingly complex and difficult to maintain. This technical debt can be especially challenging for institutions that have limited resources and budget for maintaining and updating these applications, which can lead to a lack of innovation and a decline in competitiveness.

To mitigate these issues, the modern technique is to transition to a more modular, microservices-based approach, allowing for more flexibility and scalability. This involves breaking down the monolithic application into smaller, independently deployable services that can be updated and scaled more easily, as shown in Fig. 3. By adopting this approach, institutions can reduce their technical debt, maintain their agility, and increase their ability to innovate. Further, the associated containerization gives the ability to host the containers in cloud platforms, which can be beneficial. Existing Cloud platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), handle the hosting, maintenance, backup, scaling, and networking of the hosting infrastructure for users. These services have become standard and can handle both secure and public deployments. Cloud platforms have the added benefit of moving expenses from capital to operational accounts since both the hardware and maintenance are ‘pay-as-needed’ services in cloud environments.

Containerization is the process of packaging software and all its dependencies into a single unit, which can then be deployed and run independently of the compute environment. This allows for consistent and reproducible environments, as all the necessary components are included within the container, rather than relying on the host system's configurations. Containerization allows services to be deployed in a variety of environments, to allow services to be duplicated to meet demand and testing requirements, and to allow the same functions to be run in parallel with duplicate container instances. By breaking down the monolithic application into smaller, containerized components and services, it becomes easier to test and deploy each part separately. This allows for more efficient testing and easier scaling of the application.

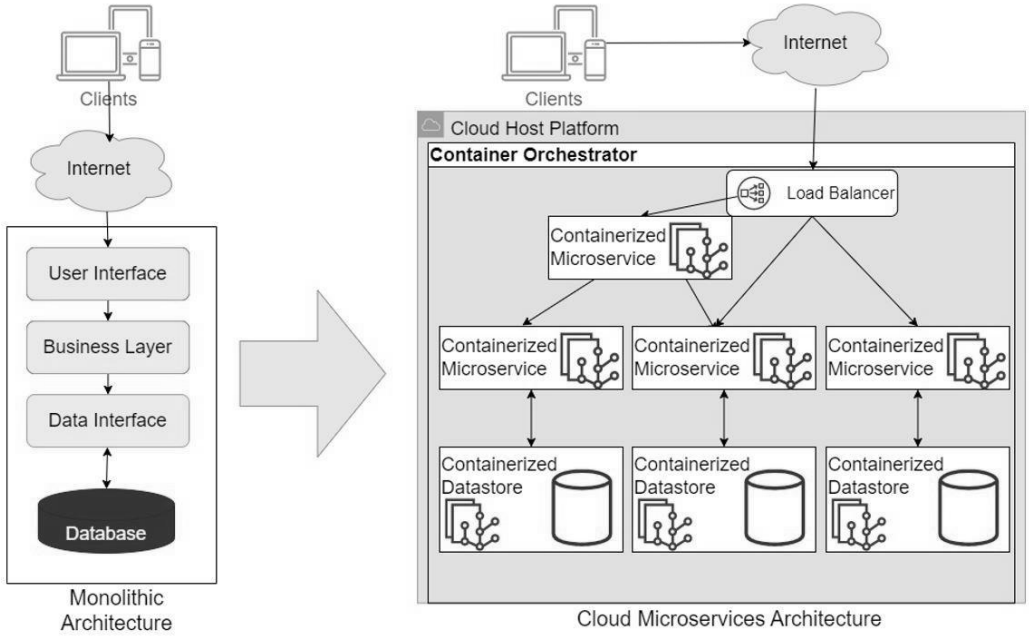

Fig. 3: A comparative view of a monolithic and cloud microservices architecture. On the left all components are built and deployed together, requiring the whole stack to be installed/configured. In cloud microservice design, each service can be built, tested, and deployed independently.

If most of a monolithic application can be separated into small reproducible containerized services, the next task becomes one of managing and orchestrating the different components of the system. Orchestration involves managing the order and scheduling of services, monitoring their performance and health, and ensuring the system remains stable and secure. The most common way of achieving this is to use a container orchestration platform, such as Kubernetes. Kubernetes is a secure, open-source platform, used for a variety of applications at Orion, that allows for the deployment, scaling, and management of containerized applications. It provides features such as resource allocation, automatic scaling, service discovery, and health monitoring. It also provides a unified interface for managing multiple services, allowing for easier deployment and maintenance of the system. Orchestration can handle the distribution of requests to available services and scale the pool of services based on the needs of the clients to provide high availability. In addition, orchestration can be used to cluster services to allow for side-by-side deployment of development, test, and production code. Further, the deployment of modified code is handled through the graceful spool down and startup of different versions of the same services for increased reliability and availability, even with updates in the code.

As an added benefit, containerized cloud architecture can apply the principles of test-driven development and Continuous Integration/Continuous Deployment (CI/CD) at each phase of the process. CI/CD is a method to frequently transition applications to operation by automating the process of build, test, and deployment. In doing so, each containerized microservice will be automatically deployed to one of the deployment environments and the entire test suite will be performed to provide a record of the validity of that version of the code. Every requirement and technical objective will inform a corresponding test (either unit, regression, or end-to-end). The requirement or objective is not considered complete until the corresponding test passes its validation. In this way, the team can guarantee traceability and performance to the original requirements. This process also ensures that any future upgrades or improvements to the application can be instantly validated on-demand.

Containerization also permits for fast environment setup and deployment so that each service can be deployed to multiple environments for robust validation and isolation. First deploying to local desktop for development, then to testing environments for internal validation, then finally deploying to production environments for their active use. One is guaranteed that each step will use the same tested and stable version of the code without having to deal with system specific configurations that can destabilize deployed applications.

To effectively develop containerized services, all services are written to be stateless allowing for easier parallelization and composability of service container instances. A stateless service is one that does not depend on previous calls to perform its function. By adhering to stateless design, it allows for pools of services to satisfy requests from any client in the environment regardless of previous activity or requiring costly context initialization.

For successful service-based Application Programming Interfaces (API), Representational State Transfer (REST) API design allows for maximum reuse and flexibility. RESTful API design involves making the services stateless and following the standard internet HTTP protocol for their interaction. This creates APIs independent of calling language but prescribes enough rigidity in specification to make them consistent in structure, function, and callability. By leveraging HTTP as its communication protocol, RESTful design allows for a broad range of deployment scenarios of both local and cloud deployed services, while simultaneously taking advantage of caching and scalability inherent to modern HTTP systems. Furthermore, a large set of libraries and tools exists to interact with RESTful APIs, so less work will be required of clients to use API service endpoints.

The modular service architecture of Dragster is shown in Fig. 4.

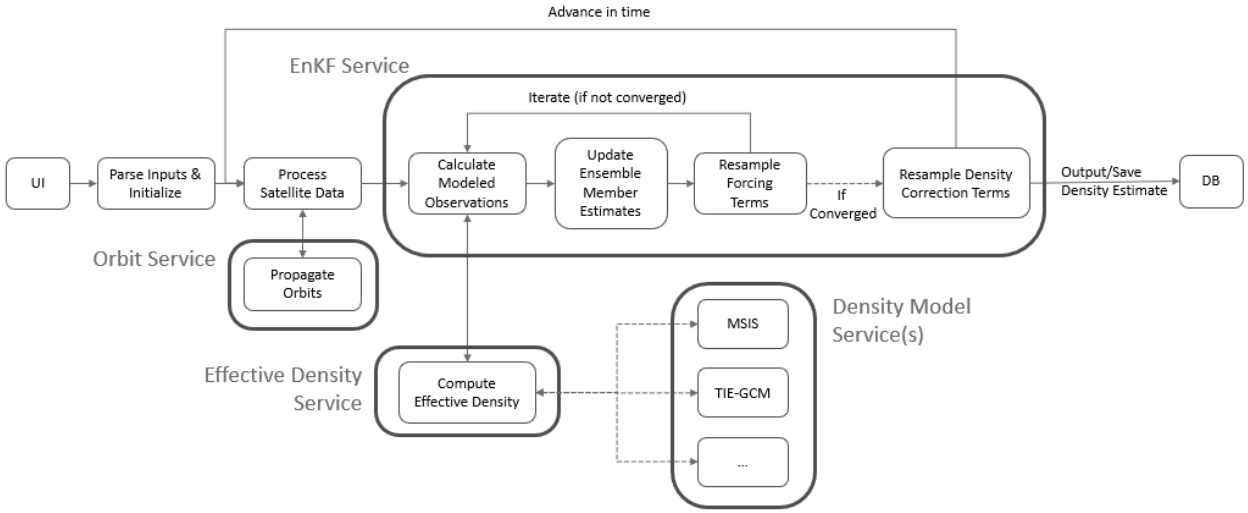

Fig. 4: Dragster Service Architecture

The Dragster software is split into several services. The first of which is the "main" service which will act as a broker between the initial input data sources (post-ingestion, tagging, and enrichment) as well as orchestrate the processing of the data into a form as will be required by the Ensemble Kalman Filter service. This service will handle the processes that aren't captured by a separate service (bold outline). The next step in the processing of the data is the propagation of the input satellite telemetry. The main functionality of the Dragster project is in the running of the ensemble extended Kalman Filter which produces the atmospheric drag correction factor "state" estimate. This service represents the largest part of the Dragster code. The Dragster filter functions by first sampling the forcing terms for a variety of different ensemble member sets. From here, an iterative process begins in which the modeled observations are calculated (by first calculating the underlying model effective density and then propagating the system), and then updating the correction factor state estimates for each ensemble member. Upon each iteration, should this state estimate not converge, the process begins again with a new resampling of the forcing terms, based on the covariance of the current state estimate. If the set does converge, then the density correction terms are resampled from the ensemble member states and the output is the state estimate for that time. The process is then repeated for the next time window as defined by the user configuration until each time point is complete. The largest piece of calculation that occurs during this filter process, however, is the computation of the effective density needed to calculate the modeled observations. In the current codebase, there are two options for underlying background models to use in this calculation: MSIS and TIE-GCM. In the future models JB-08 and WAM will be included as background options.

5. RESULTS/VALIDATION

Orion has developed verification and validation procedures for use with Dragster. The Dragster model outputs have enabled comparison with a number of satellite data sets. Side-by side comparisons have been made between the drag predictions of Dragster versus JB-08, HASDM and the empirical MSIS model. Below, we compare the Dragster neutral density and drag against values from the JB-08, HASDM and the MSIS model, and demonstrate that Dragster offers statistically significant improvement. Dragster often outperforms HASDM in the 200-500 km regime during both solstice and equinox conditions. Dragster fundamentally outperforms HASDM above 500 km as Dragster accounts for helium being the major gas above this altitude, whereas HASDM assumes oxygen is the dominant constituent.

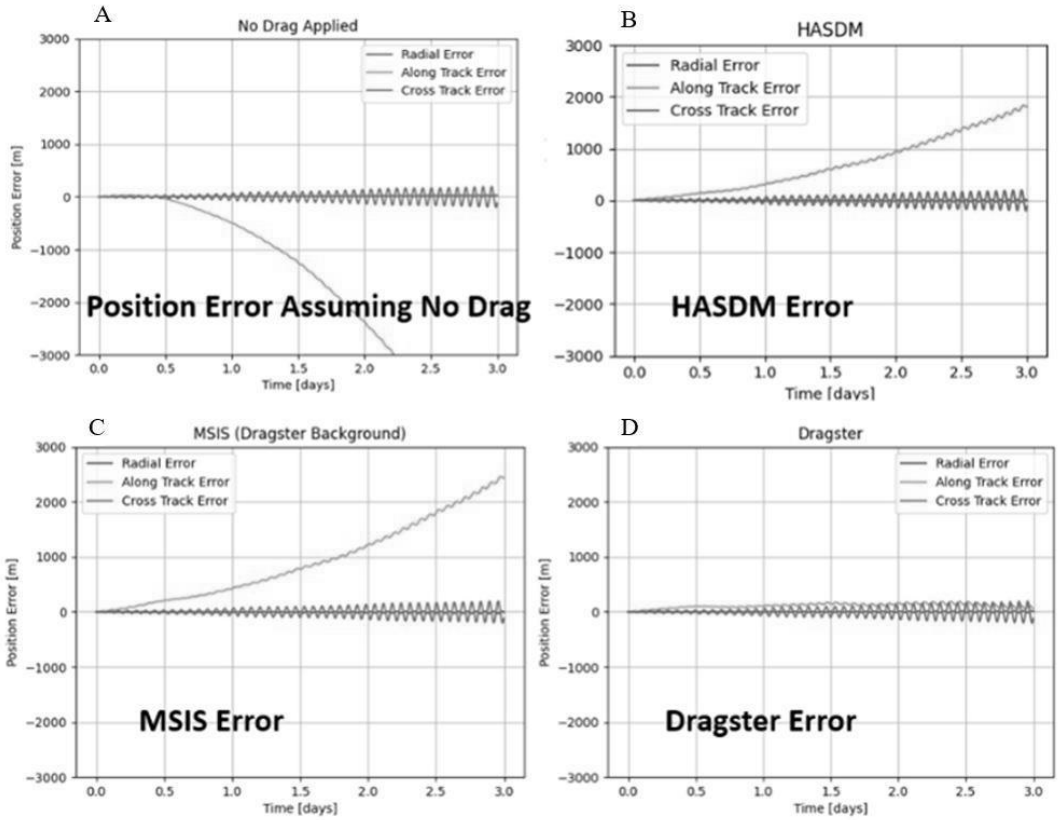

Fig. 5: SWARM satellite fly-through simulation comparing no drag, MSIS drag, HASDM drag, and Dragster drag. Dragster is shown to vastly reduce the along-track error and remains bounded throughout the time simulated.

This fly-through capability was used to validate the Dragster model nowcasting performance against two other models (MSIS and HASDM) as well as a 'no-drag' assumption, as shown in Fig. 5. To estimate the impact of satellite drag on the position error of a satellite, Fig. 5 (panels A-D), depicts the position error that accumulates over 3 days due to satellite drag when comparing the actual Swarm-A satellite location with position estimates based on different drag models. The time period used for this study was three days from the 08/16/17–09/20/17 equinox period described in the previous section. In the simplest case (panel A), we assumed no drag on the satellite, and in panels B-D we compare the actual satellite location with position predictions based on using the MSIS model HASDM, and Dragster, respectively, to specify the thermospheric density. Each panel shows the along-track (yellow), cross-track (green) and radial (blue) error. In the astrodynamics simulation, sun and moon gravity are included, and earth gravity is modeled up to order 100.

In panel A of Fig. 5, assuming no thermospheric drag, the predicted position diverges rapidly away from the actual position in the along-track direction, reaching 3000 m (3 km) in just over 2 days. Using MSIS (panel C), the error is just over 1000 m (1 km) in 2 days. The HASDM result in panel B reveals that HASDM has reduced the uncertainty

by about 750 m (0.75 km) relative to MSIS after 3 days. With Dragster (panel D), the along-track error is limited to about 100-200m even after 3 days, and this error does not appear to grow exponentially like the other three cases. In all four panels, the cross-track and radial error are both well-bounded and remain close to zero for the radial error, and within about 200 m for the cross-track error after two days. The small error derived from using the Dragster model would clearly reduce collision uncertainties and the number of false alarms. The HASDM model output for this validation was provided by Space Environment Technologies (SET).

This representative example reveals that the Dragster approach outperforms other data assimilation methods when assimilating data into strongly forced systems such as the Earth's upper atmosphere. Orion has been testing this model architecture with various types of satellite drag data, using two-line-elements, daily averaged densities, and accelerometer data as sources. Observation objects were selected according to a set of simple criteria, including a known shape exhibiting little variation in the observed ballistic coefficient and/or a stable fitted ballistic coefficient whose orientation is known. These criteria allow the ballistic coefficient to be estimated by both a priori means as well as by orbital observation. Orion currently maintains a catalog of 50+ calibration objects and 10+ validation objects. Validation objects are not used in assimilation, but instead serve as independent evaluations of assimilation performance. A wide range of orbital inclinations and perigee heights allows Dragster calibration to be sensitive to density changes at all latitudes and altitudes of interest. This reduction in the uncertainty would certainly have an impact on NASA and Space Force operations, including the higher orbital prediction accuracy required for accurate catalog maintenance, conjunction analysis and collision avoidance for manned and unmanned space flight, reentry prediction, satellite lifetime prediction, defining on-board fuel requirements, and satellite attitude dynamics.

For a fuller validation study, the years 2015, 2016, and 2017 were studied with defined validation metrics. The validation metrics are computed in both logarithmic space and linear space. For the logarithmic metrics, Orion focuses on the mean of the observed (O) to computed (C) mass density ratios as well as the standard deviation of the ratios [7, 8]. An O/C ratio of one means that the model reproduces the observed thermospheric state perfectly. The standard deviation of a time series of these ratios represents the model's ability to capture temporal and, due to the spacecrafts motion and orbital evolution, spatial variations in the upper atmosphere. A part of the standard deviation is also a result of observation errors. Consequently, the standard deviation is one of the most comprehensive and useful metrics. A mean of the O/C ratios can be very consequential in orbit propagation due to the sometimes-large difference between aerodynamic forces experienced by the satellite and the forces computed using an atmospheric model. However, an O/C that is far from unity is sometimes caused by biases inherent in the observations and density estimation method (O) and cannot be easily distinguished from biases in the modeled density (C). Furthermore, a history of orbital observations can be used to de-bias the aerodynamic force model prior to propagating an orbit forecast. Computing O/C metrics in logarithmic space has the added benefit of reducing metric disparities when comparing model performance at different places and times in the atmosphere. This is because the mass density along the orbits of satellites flying at different altitudes can vary by several orders of magnitude. Mass density along the orbit of a single satellite can also vary significantly from solar maximum to solar minimum conditions. This variability makes it difficult to assess performance of a model by comparing linear metrics computed using a variety of satellite observations distributed in both space and time. The logarithmic mean is computed as follows:

Equation 3: Logarithmic Mean [9].

$$\text{Mean}\left(\frac{O}{C}\right) = \exp\left(\frac{1}{N}\sum_{n=1}^{N}\ln\frac{O_n}{C_n}\right)$$

Then, the logarithmic standard deviation (SD) is computed as:

Equation 4: Logarithmic Standard Deviation.

$$\text{SD}\left(\frac{O}{C}\right) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\ln\frac{O_n}{C_n} - \ln\text{Mean}\left(\frac{O}{C}\right)\right)^2}$$

Linear metrics can be useful when comparing the performance of models for a single satellite at similar locations and solar cycle conditions. Therefore, we include a linear standard deviation based on the model-data residual or (C - O)/O. In other words, SDlin = StdDev((C-O)/O). We do not restrict our calculations to the intrinsic sampling rate of

the dataset nor to the orbit-averaged densities that are often used for validation in the literature. Instead, the O/C ratio and residuals are smoothed using windows of varying length prior to computing any metrics. This is done to highlight changes in metrics at different scales ranging from sub-orbital to multi-day.

For independent density observations (O), we use Swarm satellite densities archived by ESA [10] as they are not assimilated into Dragster. Data is used from both Swarm-A (near 450 km altitude) and Swarm-B (near 515 km altitude). Any negative or near-zero densities are removed from these data products prior to computing error metrics. The computed densities (C) are based on Dragster output cubes sampled along the Swarm trajectories to produce a modeled density time series. We performed a similar process for the HASDM data cubes [11]. The JB-08 and MSIS models were interrogated only at the satellite positions so there was no interpolation necessary.

Table 1 lists the performance metrics for 2015, 2016, and 2017 respectively. The table lists validation metrics computed for four different models along orbits of Swarm-A and B. The 1/4 Orbit validations are shown and illustrate higher frequency temporal and spatial variability associated with sub-orbital sampling of latitudinal and local time structure by the Swarm observations. Each table entry corresponds to a logarithmic standard deviation with the linear standard deviation shown in parentheses. The best error metric for each case is highlighted in green.

Table 1: Comparison of Dragster and other leading neutral density models.

	2015 ¼ Orbit Average - Validation Results 2015, SD Logarithmic (Linear), 5 storms with Kp>5+			
	Dragster	HASDM	JB-08	NRLMSISE-00
Swarm-A (450km)	0.115 (0.101)	0.117 (0.133)	0.180 (0.202)	0.267 (0.318)
Swarm-B (515km)	0.198 (0.202)	0.219 (0.295)	0.258 (0.329)	0.340 (0.497)

	2016 ¼ Orbit Average - Validation Results 2016, SD Logarithmic (Linear), 1 storm with Kp>5+			
	Dragster	HASDM	JB-08	NRLMSISE-00
Swarm-A (450km)	0.166 (0.164)	0.157 (0.202)	0.236 (0.278)	0.274 (0.413)
Swarm-B (515km)	0.384 (0.871)	0.369 (1.065)	0.425 (0.930)	0.456 (1.563)

	2017 ¼ Orbit Average - Validation Results 2017, SD Logarithmic (Linear), 2 storms with Kp>5+			
	Dragster	HASDM	JB-08	NRLMSISE-00
Swarm-A (450km)	0.176 (0.160)	0.188 (0.220)	0.259 (0.303)	0.278 (0.442)
Swarm-B (515km)	0.377 (0.616)	0.389 (0.724)	0.440 (0.740)	0.437 (1.227)

The assimilative models (Dragster and HASDM) always outperform the non-assimilative models (JB-08 and MSIS) for all three years regardless of the metric or timescale used. These differences can correspond to a three-fold reduction in error or greater. The JB-08 model also tends to outperform MSIS, especially at longer timescales. The differences between HASDM and Dragster are smaller than the differences between the other models. This is especially true in logarithmic metrics where the differences between the two are generally within a few percentage points. Note that the two assimilative schemes use similar datasets, which is a likely cause of the similarities. Beyond the similarities in performance, Dragster outperforms HASDM in terms of linear standard deviation for the entire time period. Dragster also outperforms HASDM in logarithmic standard deviation for two of the three years. Dragster achieves the best relative performance during the least active time period (2017). The relative strength of Dragster in 2017, during very low solar activity, is an indication that the assimilation scheme along with the background model are better suited to converge on very cold thermospheric states where compositional effects (e.g., helium) can dominate both the background distribution of density as well as its transient response. In contrast, the HASDM background model is almost devoid of composition effects. The Swarm-A and Swarm-B validation focuses on the 400km –500 km region and shows that Dragster often outperforms HASDM. Additional validation has shown that Dragster often outperforms HASDM throughout the 200-500 km regime. It is noteworthy that Dragster is expected to consistently outperform HASDM above 500 km because Dragster accounts for helium being the major gas above this altitude, whereas HASDM does not take into account this transition to a helium-rich atmosphere but assumes oxygen remains the dominant constituent, and therefore HASDM is expected to significantly underperform Dragster above about 500 km. Data assimilation has been demonstrated with Dragster at altitudes up to and beyond 800 km. Additional model

improvements and tuning are yet possible in Dragster that would give an additional advantage to Dragster in future validation exercises.

Since the analysis conducted and shown in Table 1, HASDM model outputs have become publicly available [12] spanning several years. As such, further validation could be performed over multiple years of data spanning solar minimum and solar maximum conditions. The objects included could span a range of inclinations, eccentricities, and cover perigee heights between ~200 km and ~900 km, or beyond.

6. CONCLUSIONS

Dragster has been shown to improve the specification of thermospheric density and drag significantly. This has a major impact on the orbit prediction capabilities of constellation providers and satellite owners. With more accurate density and drag estimates, conjunction alerts will be less often and will have more certainty. Another key benefit will be that optical systems will not have to track resident space objects as frequently, clearing up time for higher priority targets. Future work will continue to operationalize the modular service architecture and validating the forecast predictions.

7. ACKNOWLEDGEMENTS

This work has been supported by NASA Heliophysics SBIR contract #80NSSC23CA072. We greatly appreciate their support.

8. REFERENCES

[1] Guo, J., Wan, W., Forbes, J. M., Sutton, E., Nerem, R. S., Woods, T. N., Bruinsma, S., and Liu, L. 2007. Effects of solar variability on thermosphere density from CHAMP accelerometer data, J. Geophys. Res., 112, A10308, doi:10.1029/2007JA012409.

[2] Doornbos, Eelco. (2012). Thermospheric Density and Wind Determination from Satellite Dynamics. 10.1007/978-3-642-25129-0.

[3] Pilinski M. D., B. M. Argrow, S. E. Palo, B. R. Bowman (2013b), Semi-Empirical Satellite Accommodation Model for Spherical and Randomly Tumbling Objects, Journal of Spacecraft and Rockets, Vol. 50, pp. 556-571, doi: 10.2514/1.A32348.

[4] Bowman, B. R., & Hrncir, S. 2007. Drag coefficient variability at 100-300 km from the orbit decay analysis of rocket bodies. In AIAA astrodynamics specialist conference. Mackinaw Island, MI: AIAA

[5] Pilinski M.D., and B. M. Argrow (2013a), Aerodynamic Analysis Based on Challenging Minisatellite Payload Satellite Lift-to-Drag Measurements, Journal of Spacecraft and Rockets 2013 50:6, 1162-1170, DOI: 10.2514/1.A32394

[6] Pilinski, M. D., and G. Crowley (2015), Seasonal variability in global eddy diffusion and the effect on neutral density, J. Geophys. Res. Space Physics, 120, 3097, doi:10.1002/2015JA021084.

[7] Sutton, E., Thayer, J. Pilinski, M., Mutschler, S., Berger, T., and Nguyen, D. Masters. "Toward Accurate Physics-Based Specifications of Neutral Density using GNSS-Enabled Small Satellites," Space Weather, https://doi.org/10.1029/2021SW002736, 2021.

[8] Bruinsma, S. Boniface, C., Sutton, E.K. Fedrizzi, M. 2021. Thermosphere modeling capabilities assessment: geomagnetic storms. J. Space Weather Space Clim. 11 12 (2021), DOI: 10.1051/swsc/2021002.

[9] Mutschler, S. 2022. "Global Thermospheric Density Estimation using CubeSat GPS Data and a Physics-based Space Environment Model," Doctoral Thesis, University of Colorado Boulder,. https://spacewx.com/wp-content/uploads/2022/06/SET-TR-2022-001.pdf

[10] Lomidze, L., Knudsen, D. J., Burchill, J., Kouznetsov, A., & Buchert, S. C. 2018. Calibration and validation of Swarm plasma densities and electron temperatures using ground-based radars and satellite radio occultation measurements. Radio Science, 53, 15–36. https://doi.org/10.1002/2017RS006415.

[11] Mutschler, S. M., Axelrad, P., Sutton, E. K., & Masters, D. 2023. Physics-based Approach to Thermospheric Density Estimation using CubeSat GPS Data. Space Weather, 21, e2021SW002997. https://doi.org/10.1029/2021SW002997

[12] Tobiska, W. K., Bowman, B. R., Bouwer, S. D., Cruz, A., Wahl, K., Pilinski, M. D. 2021. The SET HASDM density database. Space Weather, 19, e2020SW002682. https://doi.org/10.1029/2020SW002682.