DEVELOPING PROBABILISTIC SATELLITE EXPLOSION MODELLING METHODS FOR SPACE DEBRIS ENVIRONMENT PREDICTION

S. Hawkins[1], A. Horstmann[2], B. Bastida Virgili[1], and F. Letizia[3]

[1]*European Space Agency, ESOC, Darmstadt, Germany, Email: {saskia.hawkins, benjamin.bastida.virgili}@esa.int*
[2]*IMS Space Consultancy at Space Debris Office, ESA/ESOC, Darmstadt, Germany Email: andre.horstmann@ext.esa.int*
[3]*European Space Agency, ESTEC, Noordwijk, The Netherlands, Email: francesca.letitzia@esa.int*

ABSTRACT

In-orbit explosions are currently the primary source of fragmentation events in space and significantly contribute to collision risk. Accurately modelling the frequency of these events is therefore critical for reliable long-term space environment predictions. This study presents a novel approach for modelling explosion events in ESA's long-term simulation tool, DELTA, using survival analysis of historical data. We distinguish between payloads and rocket bodies, demonstrating that each group follows a distinct survival distribution. A five-knot cubic spline is fitted to the cumulative hazard functions, providing a framework for generating synthetic explosion times. This method is implemented into a new DELTA module named EEM, which we demonstrate produces explosions that are successfully coupled to launch traffic and scale accordingly. Our results also show that the new model more accurately predicts explosion events between 1980 and 2020 compared to the previous implementation, though some under prediction remains.

Keywords: Modelling, Explosions, Fragmentations, DELTA.

1. INTRODUCTION

Over the past decade, significant efforts have focused on passivation measures and end-of-mission disposal techniques to reduce in-orbit fragmentation. However, accidental explosions currently remain the largest contributor to debris-generating fragmentation events. In the last decade, explosions due to leftover propellant alone account for a third of all breakup events [1]. The third largest recorded fragmentation event was due to over pressurisation of the HAPS vehicle propellant tank, producing a record 754 pieces of tracked debris, 64 of which remain in orbit.

In the latest long-term predictions [2], the number of collision fragments overtake the number of explosion fragments within a decade. However, explosion fragments are a key driver for increased collision risk, thus, accurately modelling the frequency and characteristics of these explosions is essential for predicting the evolution of the debris environment over the coming centuries.

In ESA's long term environment simulation tool, DELTA, explosions are currently modelled with the Breakup Event Module (BEM). The explosion events are predicted per object class, creating the Object Class Data file, which is derived from historical breakup analysis. This file contains event and object parameters for explosions, along with annual event rates for different object types, and is processed into an array of explosion events. To account for variations in explosion rates over time, an annual factor may be applied to scale the explosion number.

In each Monte Carlo (MC) run, the number of explosions at each timestep for a given object class is determined by randomly sampling from a Poisson distribution. This distribution is defined by the object class's explosion rate, the corresponding year's scaling factor, and the timestep size. The explosion event properties are determined from the historical explosion event array, then appended to an event array for the simulation. Once all the object classes and timesteps have been processed for the MC run, the event array is ordered in chronological order. This event file is then executed during the simulation.

There are several significant issues that have been identified with this approach.

a) Firstly, explosion events are not correlated with objects. Therefore, explosion fragments add mass into the simulation that were not accounted for in the launch traffic. While this additional mass is not expected to be greatly statistically significant in comparison to the launch traffic, the second issue poses a greater problem:

b) The explosion rate is not proportional to the launch rate. When the explosion rate is constant, the explosion probability decreases with increasing launch traffic.

c) Thirdly, the events occur in the exact same orbital

Proc. 9th European Conference on Space Debris, Bonn, Germany, 1–4 April 2025, published by the ESA Space Debris Office
Editors: S. Lemmens, T. Flohrer & F. Schmitz, (http://conference.sdo.esoc.esa.int, April 2025)

regions as historical explosion events, which may lead to build up and an overly conservative collision probability in certain orbital regions.

A probabilistic model based on a time-dependent Poisson distribution was recently developed for NASA's LEGEND simulator [3]. It showed significant improvement in modelling the explosion rate with respect to historical data, compared to the previously implemented constant rate of explosion. The aim of this paper is to provide an improved method to probabilistically model explosions, that are object and launch dependent in DELTA.

2. DATA

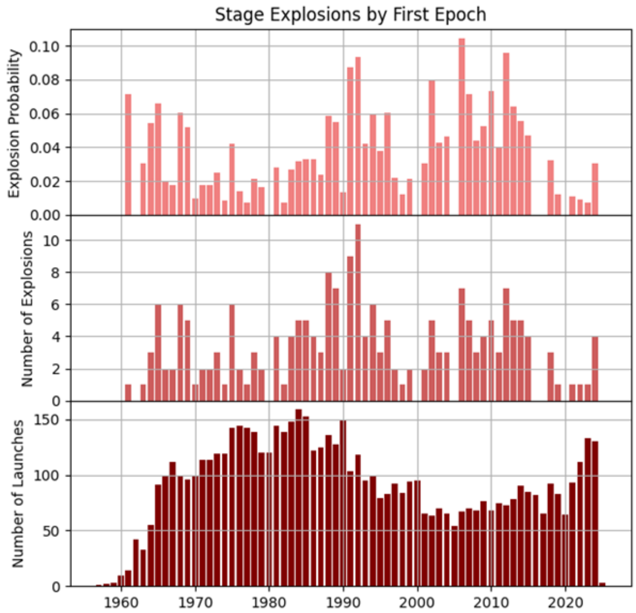
ESA's DISCOS (Database and Information System Characterising Objects in Space) database contains the breakup information of all recorded fragmentation events in orbit. Events are discovered and confirmed when additional objects are observed near the parent object using ground based optical or radar observations. The recorded number of fragments associated with a breakup event is highly dependent on the orbital regime due to observability, and therefore we have not included fragmentation number in this analysis. In DELTA, explosion fragments are modelled according to NASA's standard breakup model [4].

From DISCOS, we may extract all fragmentation events related to an object type and exclude those associated with the following tags: ASAT, Deliberate, Collision and Small Impactor. The cause of events is not analysed in this paper, or required for time-to-event analysis, and therefore we include unknown and anomalous breakup events in the data pool. The data is split into two categories, payloads and rocket bodies (or 'stages'). We make a key distinction between these objects, as they are structurally different resulting in explosion events that are likely driven by distinct mechanisms. As of March 2025, there have been no reported explosions of constellations. These would likely be treated differently in long term propagation, as a design flaw that leads to an explosion significantly increases the explosion risk of objects with the same design within that constellation. For rocket bodies, this leaves 209 events from 1961 to 2024. For payloads, we observe 338 events.

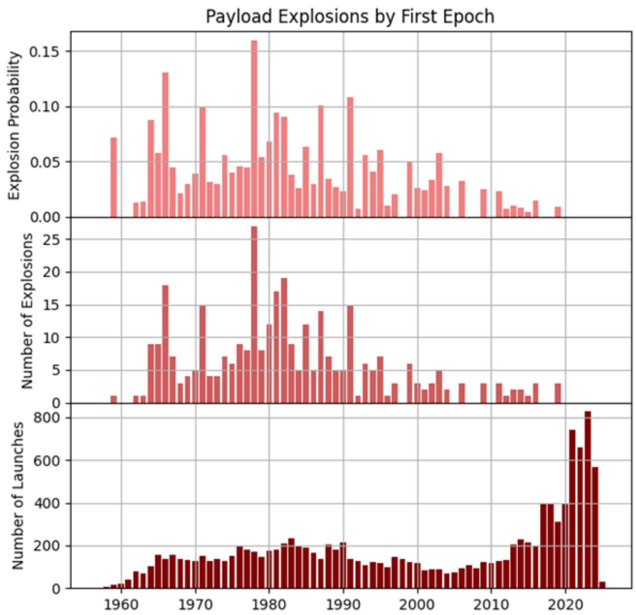
There are two key timescales to consider when analysing historical explosion rates: generational change and spacecraft age.

2.1. Generational Change

Generational change refers to the evolution of explosion behaviour as a function of launch year, capturing the influence of technological advancements and hardware trends at the time of launch. Predicting explosion probability based on launch epoch is similar to forecasting



(a) Rocket body explosions



(b) Payload explosions

Figure 1: Explosions by objects' launch epoch, demonstrating generational evolution of explosion probability.

future launch traffic trends. Currently, DELTA relies on repeated historical launch data. While efforts have been made to predict future launch trends [5, 6], the rapid surge in launches over the past decade demonstrates the difficulty of accurately forecasting such trends using historical data alone.

From Figure 1, we see the evolution of explosion probability as a function of an object's launch epoch. The explosion probability for launch year i is defined as:

Table 1: Mean average explosion probability per object type by objects' launch epoch

Object type	Mean probability	Std.
Rocket Body	0.0326	0.0274
Payload	0.0337	0.0349

$$p_i = \frac{n_{explosions}}{n_{launches,i}} \tag{1}$$

Where $n_{explosions}$ are the number of objects that exploded at any time, that were launched in year i. We exclude launches of constellations for the payloads.

While there does appear to be a reduced explosion probability in recent years, the recent population has a higher explosion probability than older generations (see subsection 2.2), and will likely still experience explosion events. New technology will bring new challenges, and we do not see a clear linear improvement with time. For this model we therefore have chosen to not assume any generational change with time.

We define the overall probability of explosion, p, as the mean average explosion probability by launch epoch. These averages are displayed in Table 1. Reference [3] finds that the total explosion probability for a payload is 0.044, and 0.019 for rocket bodies. The discrepancy in the latter value is due to object filtering, where SOZ spacecraft were analysed separately, and found to have a 0.57 explosion probability.

2.2. Days on Orbit

The second timescale we may consider is the object's time on orbit before explosion. This may address the various causes of explosions, for example, early failures due to crucial design flaws versus failures due to thermal degradation over long time periods. We may naively expect a bathtub curve commonly associated with reliability engineering [7], however, Figure 2 demonstrates that the vast majority of explosions occur within the first year of launch.

As of 2025, the longest time elapsed between an object launch and explosion of known cause was 49 years. The object responsible was a Titan Transtage 3C-17 rocket body, and this was the second event for this object in 2018. The first occurred in 2014, which was believed to be an outgassing event. This object has produced 100 reported fragments, 53 of which are catalogued. While it is clear from Figure 2 that events occurring greater than 30 years after their launch are rare, we are limited by a maximum observation period of 68 years. It is entirely feasible that explosions may occur after longer timescales.

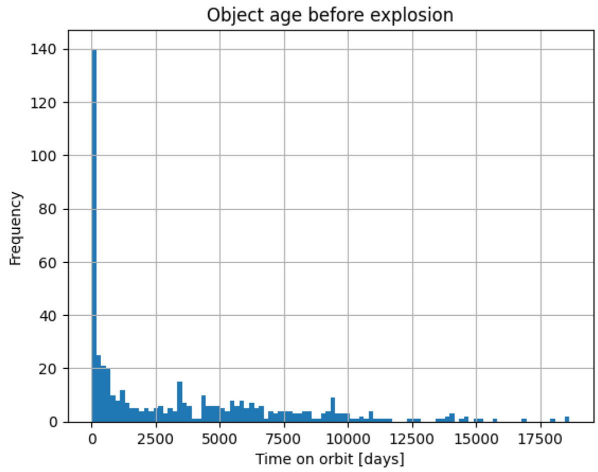


Figure 2: Object age before explosion for all object types

3. METHODOLOGY

3.1. Survival Analysis

Survival analysis is a statistical framework used to analyse, model, and predict the time until a specific event occurs. In this context, the event of interest is the explosion of a rocket body, with time measured from launch.

The survival function, S(t), represents the probability that an object remains intact at least until time t. It is a smooth, monotonically decreasing function; as time progresses, the likelihood of survival decreases. A commonly used method for estimating S(t) from observational data is the Kaplan-Meier estimator, a non-parametric approach that provides a discrete, stepwise approximation of the survival function. It is defined as:

$$\hat{S}(t) = \prod_{i:,t_i \le t} \left(1 - \frac{d_i}{n_i}\right), \tag{2}$$

where d_i is the number of deaths at time t_i, and n_i is the number of subjects known to have survived until t_i.

The hazard function describes the instantaneous risk of an event occurring at a given timestep, given that it has survived up until that time. It is related to the survival function by:

$$h(t) = -\frac{\partial \log S(t)}{\partial t} \tag{3}$$

The cumulative hazard function is the accumulated hazard over time, and is related to the survival function by:

$$H(t) = \int_0^t h(u)du \ , \quad t > 0 \tag{4}$$

A key challenge in survival analysis is dealing with censored data. Datasets often include subjects which are not observed for the same duration or do not complete the full observation period. This can occur when subjects leave the study early (re-enters, collides or passivates) or are observed for different lengths of time (due to different launch dates). Right censoring is a common approach to handle this, where subjects that are no longer observed are assumed to have survived until a time greater than their last observation time. However, this assumption incorrectly assumes that given infinite time, these objects will eventually explode. This does not reflect the reality of a dynamic space environment, where objects may perform full passivation, re-enter, collide or are not capable of exploding.

Another challenge is extrapolation of the survival function. DELTA simulations typically run for 200 years, therefore capturing long term behaviour of exploding objects is required. As mentioned, if the survival function is projected beyond the observation period without convergence, it unrealistically assumes that all objects will eventually explode. However, avoiding extrapolation may limit the maximum lifetime to the observed data, potentially distorting the true lifetime by not accounting for the possibility that objects could survive beyond our current observation period.

To address these two challenges, a mixture, or 'cure' model can be used to model situations where a subset of the population never experiences the event of interest. A mixture model assumes that the population consists of two groups: one group that is susceptible and another that is immune to the event, where the probability of 'explodability' is p. The survival function is then defined as:

$$S_{total}(t) = (1 - p) + pS_0(t) \tag{5}$$

The function $S_0(t)$ is defined using only the subset of data that experienced the event. Right censoring is no longer applicable for this dataset, and extrapolation of $S_0(t)$ cannot result in a total explosion probability greater than p. For our model, we define p as the mean average probability of explosion by launch epoch per object type, as described in subsection 2.1.

3.2. Model Selection

The aim of this paper is to generate synthetic survival times for future launch traffic. By fitting a model to the survival distribution, we may produce data that mimic the patterns observed in the historical data. These functions can be estimated using nonparametric methods such as the Kaplan-Meier estimator, as shown in Figure 3, semi-parametric approaches like the Cox proportional hazards model, or parametric models such as an exponential or Weibull distribution.

Unlike parametric models, which impose rigid assumptions about the underlying data distribution, splines provide a non-parametric approach that can more accurately capture complex, non-linear patterns in survival data. It offers greater flexibility for an evolving dataset that is not expected to follow a specific distribution. They are defined as piecewise polynomials, with a degree n, and are joined together at locations called knots, k. Choosing the location and number of knots is a key step in performing spline regression. The degrees of freedom of the spline is $df = k + (n + 1)$ [8], where the greater the degree of freedom, the more flexibility and curvature the spline has. With no prior knowledge of the knot locations, one simple approach is to place the knots at equally spaced intervals along the dataset. However, this may miss information if data is not uniformly distributed. Another approach which addresses this is to define knots based on quantiles. This ensures that where there is more information, the knots are placed closer together, capturing curvatures on smaller timescales.

The Akaike information criterion (AIC) is a method commonly used to compare how well different models fit a dataset [9]. It is calculated using the maximum likelihood estimate of the model $\hat{L}$, and the number of independent variables used v. The minimum AIC describes a model that maximises the log likelihood using the least number of parameters to avoid overfitting.

$$AIC(model) = -2ln(\hat{L}) + 2v \tag{6}$$

The lifelines python package provides an inbuilt AIC comparison method to compare univariate parametric models [10]. This was used to determine that a 5-knot, cubic (n=3) spline provided the best fit to both the payload and rocket body cumulative hazard curve. The cumulative hazard function is fitted to the following spline distribution.

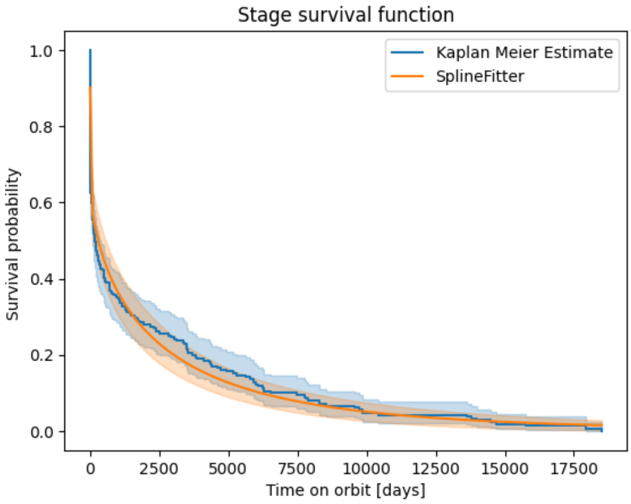
$$H(t) = exp\left(\phi_0 + \phi_1 log(t) + \sum_{j=2}^{N} \phi_j v_j log(t)\right) \tag{7}$$

Where v_j are cubic basis functions at N given knots. For a detailed definition on the spline function, see [11].

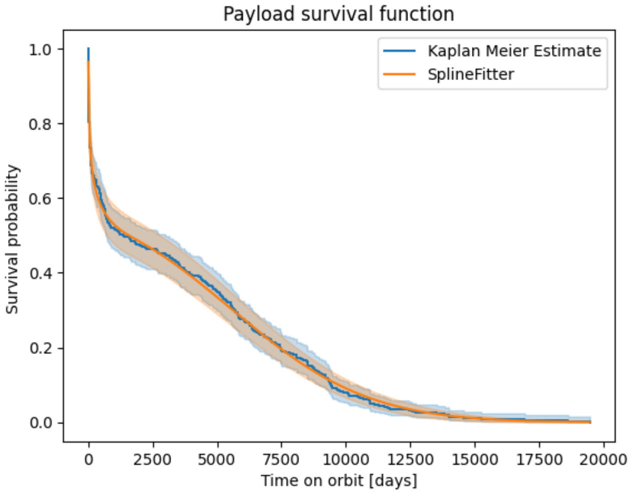
Figure 3 demonstrates the fit of the 5 knot cubic splines to both the stage and payload survival functions, where the 5 knots are placed at equally spaced quantiles along the data. This successfully captures the two step decrease of the payload survival probability, and the smoother continuous decrease of the rocket body survival probability.

3.3. Synthetic Time Generation

To avoid being constrained by the maximum observed explosion time, we extrapolate the cumulative hazard distribution beyond the observation period. In alignment with



(a) Rocket body survival function



(b) Payload survival function

Figure 3: Survival function of rocket bodies and payloads, constructed using the subset of data where objects experienced an explosion. A 5 knot cubic spline is regression fitted to the data, and the Kaplan Meier discrete estimate is plotted for comparison.

the cure mixture model, we assume that only a fraction, p, of objects is capable of exploding. Consequently, after the maximum observation time, we assume that the hazard accrued at each time step is constant and equal to the hazard value at the last time step. This results in a linearly increasing cumulative hazard function and an exponentially decreasing survival function $S_0(t)$. The survival probability for the exploding population asymptotically approaches zero, and the total survival function $S_{total}(t)$ converges to $(1-p)$.

Once the form of the cumulative hazard curve has been determined, we can sample survival times directly from this distribution. Since the cumulative hazard function, $H(t)$, is a monotonically increasing function, there is a one-to-one mapping between the values of H and t. The linearity of $H(t)$ at high t values ensures greater numerical stability in both extrapolation and interpolation than sampling directly from the survival function, which decreases exponentially with t.

For a given object, we determine whether it will explode using Bernoulli sampling based on the object type. This probability may be scaled according to a user input annual explosion scale factor α, which is nominally equal to 1. We scale the odds, which is the ratio of the probability of success to failure, equal to $p/(1-p)$. This is transformed back to probability, so that the probability remains constrained between 1 and 0. At small probabilities, the function scales the new probability almost linearly.

$$p_{new} = \frac{\alpha p}{\alpha p + 1 - p} \tag{8}$$

Then, if the object is deemed to explode, we sample a uniform distribution $U(0,1)$, which represents a random value of $S(t)$. The time can then be found by inverting the cumulative hazard distribution and solving for time.

$$t = H^{-1}(ln(U)) \tag{9}$$

We add this value to the object's beginning of life to find its explosion epoch.

4. IMPLEMENTATION

The implementation of this model is separated into three parts, as illustrated in Figure 4

4.1. DISCOS4DELTA process

The DISCOS4DELTA program produces datapackages for DELTA runs, including initial population files, launch information of the last N years, and the Object Class Data file for the previous explosion module.

We extend this program to include the survival analysis and spline regression for this model. The probability of explosion and cumulative hazard are derived per object class (rocket body or payload) and saved to data files. These files are added to the datapackage for the DELTA run.

4.2. DELTA Preprocesses

The new pre-processesor is named the Explosion Event Module (EEM). It parses the cumulative hazard CSVs, probability datafiles, and input scale factor and allocates

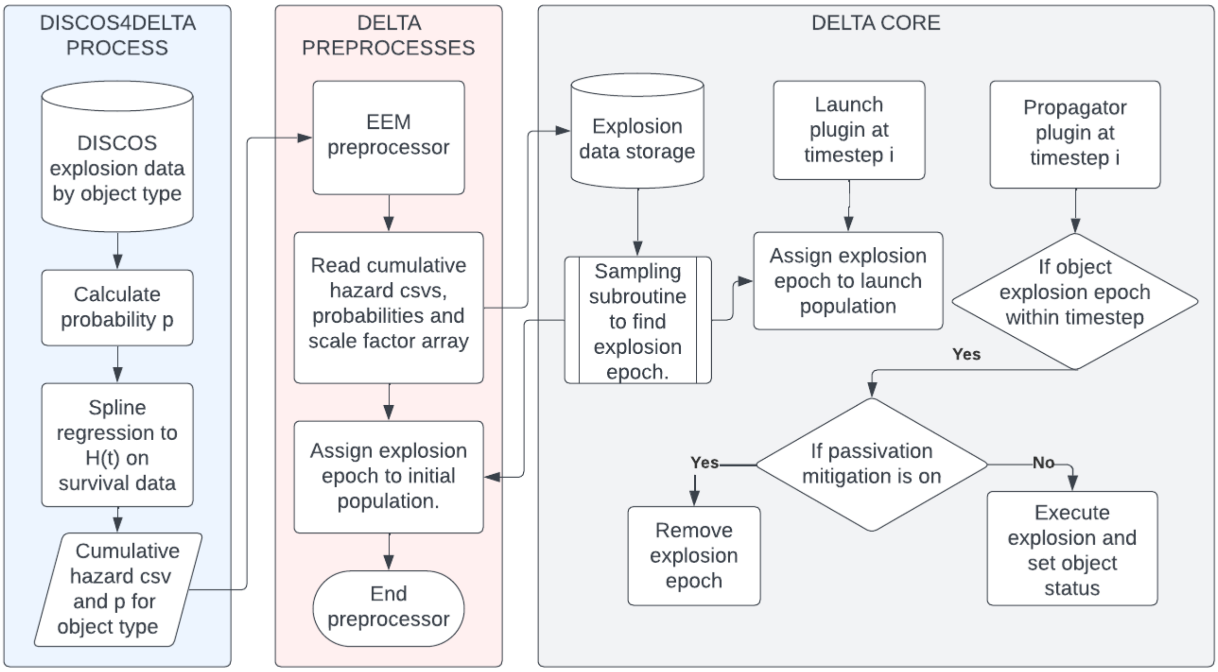


Figure 4: New explosion module implementation workflow overview

these data to a globally accessible explosion data storage module. This module contains subroutines that interpolate and extrapolate $H(t)$, and generate explosion epochs for an object, as described in subsection 3.3.

The EEM preprocessor then loops through the initial population and assigns explosion epochs to objects that are deemed to explode based on the Bernoulli sampling. If the explosion epoch is in the past, the explosion will not be executed.

4.3. DELTA Core Processes

At each timestep, the launch plugin is executed. For newly launched objects, explosion epochs are assigned in the same way to the initial population before they are added to the simulation.

The propagator plugin is also executed at each timestep: updating each objects orbital properties, removing them if they reenter. We adapt the propagator plugin to execute explosions when the epoch is within the given timestep, and the object has not passivated.

5. RESULTS

As the previous explosion module was not coupled to the DELTA population, the explosion output files are limited in their reported information. We intend to expand the explosion output datafiles to include launch epoch and object ID, but as of the writing of this report, this has not yet been implemented. The results presented here are therefore limited. With no launch epoch or object ID, we cannot examine the time to event of the simulated explosions, or split by object type. Only the overall environment effect may be analysed.

5.1. Historical Verification

To verify the new explosion module, a DELTA run was executed to simulate the environment evolution from 1980 to 2020.

To set up a DELTA run simulating the future environment, a datapackage is usually created based on launch traffic from year n to year m ($m > n$), where year m is the simulation start year. The input population is created from the environment in year m, and DELTA stochastically samples a launch traffic cycle of length $(m-n)$, and is repeated from year m onwards.

For historical verification, we instead use the input population from 1980 (year n), and create a launch traffic cycle with a length equal to the simulation length, from 1980 to 2020. The launch traffic is therefore still stochastically sampled and will vary between MC runs, but should reflect reality more closely than a predictive launch traffic based on data prior to 1980.

This simulation was executed using both the old (BEM) and new (EEM) modules, with 25 MC runs each. No post mitigation measures were used. The aim is to see how accurately the explosion modules predict the recorded number of explosions during this timeperiod. The results are displayed in Figure 5.

The first thing to be noted is that the launch traffic was not accurately predicted. To see the differences in observed launch traffic compared to DELTA launch traffic, see Figure A.1. Investigating this difference is not in the scope of this paper, but to account for it, we have plotted the number of explosions normalised by the number of launches in that year. The recorded number of explosions normalised by launch number from DISCOS is also plotted.

We can see that the old explosion module under predicts the explosion number by approximately a factor of 5, despite sampling from explosions that occurred during the 1980 to 2020 timeframe. The new explosion module makes a reasonable improvement on this, however, there is still an under prediction. The higher standard deviation compared to the old module indicates greater annual variation, which is perhaps a more accurate reflection of reality.

The lack of explosions in 1980 is a setup misalignment. The simulation starts at 1980-11-01, and the initial population is from 1980-01-01. Many of the explosions assigned to the initial population will not have been able to execute. This simulation will be rerun to correct this, and the discrepancy will be further investigated.

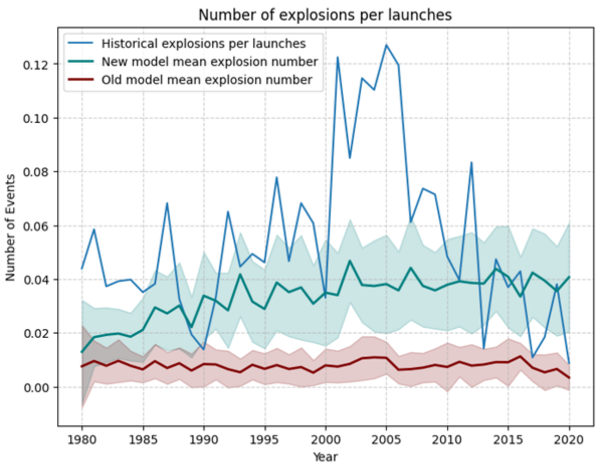


Figure 5: Predicting the explosion traffic from 1980 to 2020 using the old and new explosion module.

The under prediction of the new EEM model may be attributed to a number of factors. The explosion rate is highly dependent on the launch traffic. The under prediction of the launch traffic shown in Figure A.1 will explain some of this discrepancy. Furthermore, objects that were assigned to explode may naturally re-enter, de-orbit after 25 years at 0.1 success rate, or be involved in other fragmentation events. This could provide motivation to adjust the explosion probability p to ensure that the number of explosion events executed is in line with the expected rate.

5.2. 2024 Extrapolation

The second simulation ran was the environment extrapolation case for ESA's 2024 Environment Report [2], with both the old and new model. Figure 6 shows the overall environment effect of the new explosion model, including explosion fragment number, collision fragment number and overall object number.

The amplitude of the explosion fragments is higher, likely because they are coupled to the amplitude of the launch traffic.

There is a clear increase in explosion fragment number, approximately by a factor of 2. However Figure A.2 demonstrates that the total number of explosion events is much higher than this, producing approximately 6 times more explosions total over 200 years. This difference may be due to a number of reasons:

a) The explosions may occur at lower altitudes in the new model, and therefore deorbit faster.

b) The explosions may produce fewer fragments on average, due to more explosions of smaller objects.

Both of these possibilities will be investigated.

The number of collision fragments does not appear to be significantly different between models in Figure 6. However, we can see in Figure A.4 that the additional explosion fragments does marginally increase the collision risk, as expected.

5.3. Scaling with launch traffic

To test how the explosion traffic scales with the launch traffic, we compare the 2024 extrapolation in subsection 5.2 to two alternative future scenarios. The first is no future launches, so the explosion traffic will be generated only from the initial population. The second is a traffic 'surge', where we assume the traffic scales up by a factor of 2 over the next 200 years. It should be noted, this is not the same scale factor as described in subsection 3.3, instead it is applied directly to the launch traffic number. This scale factor has only been applied to payloads and stages, not constellations. 25 MC runs of each case were run.

The results are shown in Figure 7. The high spikes are expected to smooth out with increased MC runs. As expected, the no future launch case has explosions that decrease with a shape and timescale that aligns with Figure 2. The traffic surge case shows an increase in explosion number that is consistent with the increase in launch

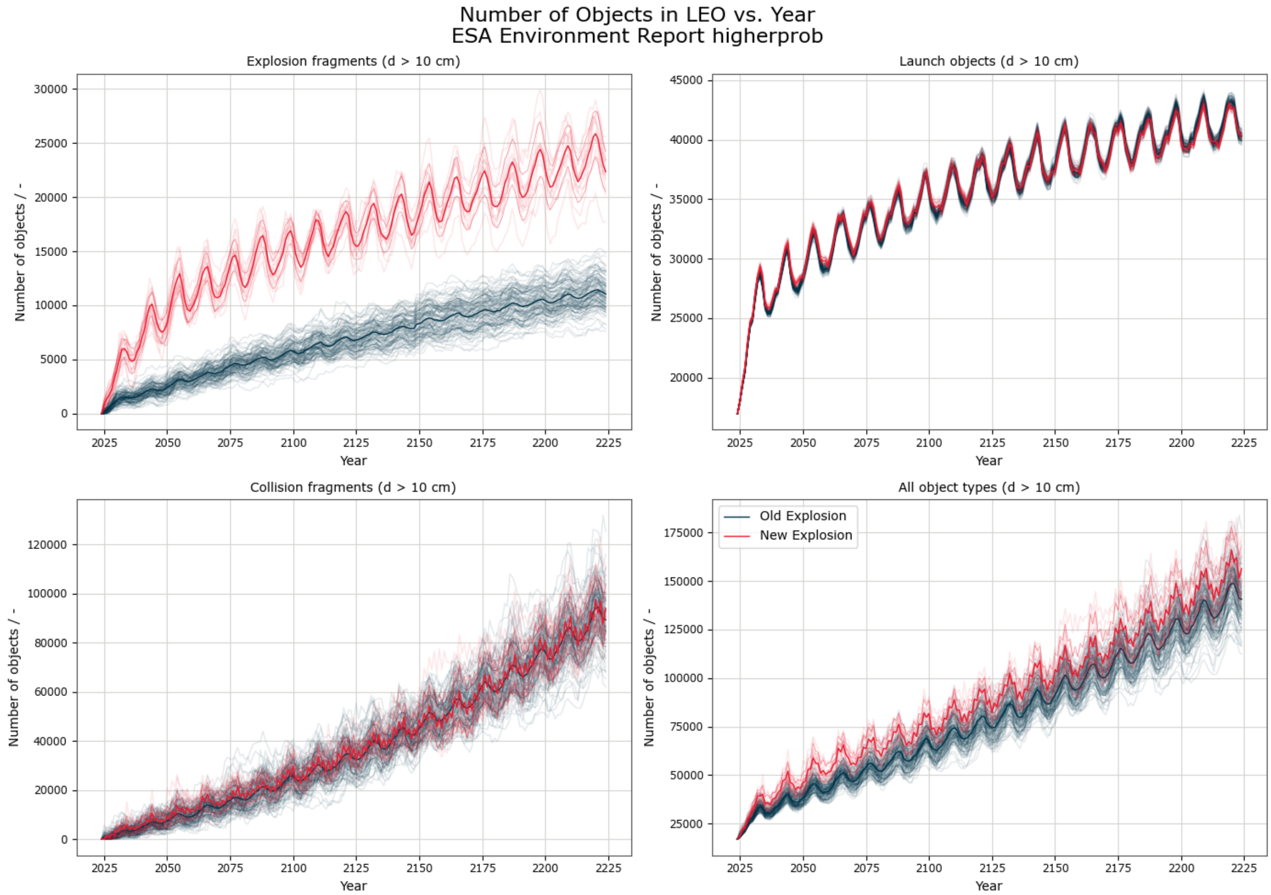


Figure 6: ESA 2024 environment report extrapolation case with old and new explosion model

traffic. By plotting the ratio of the explosion number between the surge case and extrapolation case in Figure 8, it is clear the explosion rate correctly scales with the launch rate increasing from a factor of 1 to 2 over 200 years. Figure A.3 shows the cumulative explosion number.

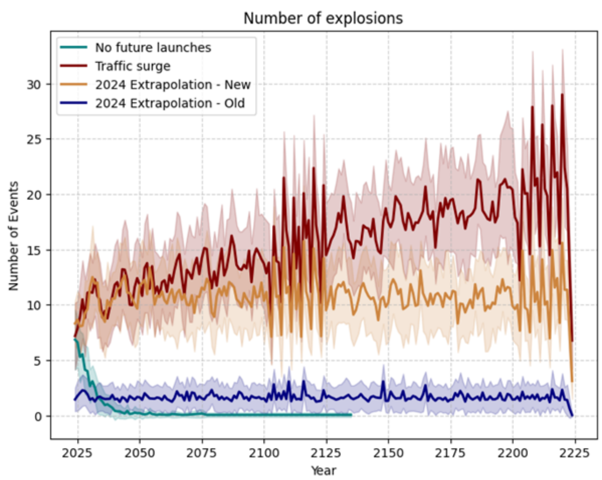


Figure 7: Explosion number with varying launch traffic

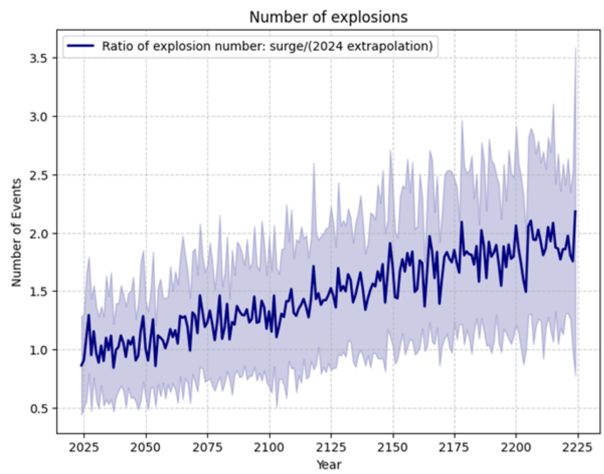


Figure 8: Ratio of explosion number of surge case to nominal 2024 extrapolation case with new explosion model.

6. FUTURE DEVELOPMENTS

There are a number of future developments that are expected following the preliminary results presented in this paper.

a) The explosion output files will be expanded to include information such as object ID and launch epoch, to allow for time-to-event analysis. The fragmentation event analyser will also be updated.

b) The scaling factor α for the explosion probability will be investigated to gauge its sensitivity

c) The historical verification case in subsection 5.1 will be rerun with correctly aligned dates.

d) Compare the differences in mass, location and fragment number of explosions between models

e) A scaling of the overall explosion probability p will be investigated to attempt to capture the correct proportion of explosion events before re-entry or collision.

7. CONCLUSION

This paper aimed to improve the explosion module (BEM) in ESA's long-term environment simulator DELTA. Previously, the explosions were modelled by sampling from historical explosion events. This added unaccounted mass into the simulation, decreased the explosion probability as the launch traffic increased, and restricted the orbital locations in which explosions could occur.

We used survival analysis to incorporate a cure mixture model, assuming that only a fraction, p, of the population are capable of exploding. This value was estimated by finding the mean average probability of explosion per launch epoch for each object type.

A five knot cubic spline was regression fitted to the cumulative hazard function of the survival data of rocket bodies and payloads separately. We extrapolated these functions, assuming the explosion hazard is constant beyond the last observed time. For the future launch traffic, a survival time may be generated by sampling directly from this function.

We produced a method in DELTA that successfully couples the explosion traffic to the launch traffic, and scales accordingly. We found that the new explosion model provides a more realistic prediction of the historical explosion traffic between 1980 and 2020 than the old model, which under predicted by approximately a factor of 5. The new model also under predicts explosions, as objects that were assigned explosions may re-enter, collide or passivate before exploding.

Further implementations and investigations are necessary to understand the differences between the models' fragment number production, orbital locations, and the model scaling factors.

ACKNOWLEDGMENTS

Thank you to Andre Horstmann for support on understanding the DELTA framework, and to Vitali Braun for lending me his famous FORTRAN 95 book.

REFERENCES

1. F Mclean, S Lemmens, Q Funke, and V Braun. Discos 3: An improved data model for esa's database and information system characterising objects in space, 2017.
2. European Space Agency. Esa's annual space environment report, Jul 2024.
3. Alyssa P Manis, Mark J Matney, Phillip D Anz-Meador, and Andrew B Vavrin. Time-dependent satellite explosion probabilities for long-term orbital debris environment modeling, Jan 2025.
4. N.L. Johnson, P.H. Krisko, J.-C. Liou, and P.D. Anz-Meador. Nasa's new breakup model of evolve 4.0. *Advances in Space Research*, 28(9):1377–1384, Jan 2001.
5. Callum J Wilson, Massimiliano Vasile, Jinglang Feng, Keiran McNally, Alfredo Antón, and Francesca Letizia. Modelling future launch traffic and its effect on the leo operational environment - strathprints. *Strath.ac.uk*, Jan 2024.
6. Mariel Borowitz, Brian C Gunter, Alaric C Gregoire, Clifford Stueck, Surya Venkatram, and Lauri Newman. Exploring the effectiveness of maneuvering guidelines for space traffic management, Oct 2024.
7. G. Klutke, P.C. Kiessler, and M.A. Wortman. A critical look at the bathtub curve. *IEEE Transactions on Reliability*, 52(1):125–129, Mar 2003.
8. Dr S. Jackson. Chapter 9 splines — machine learning, May 2024.
9. P. Stoica and Y. Selen. Model-order selection. *IEEE Signal Processing Magazine*, 21(4):36–47, Jul 2004.
10. Cameron Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019.
11. Patrick Royston and Mahesh K. B. Parmar. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine*, 21(15):2175–2197, Aug 2002.

A. ADDITIONAL PLOTS

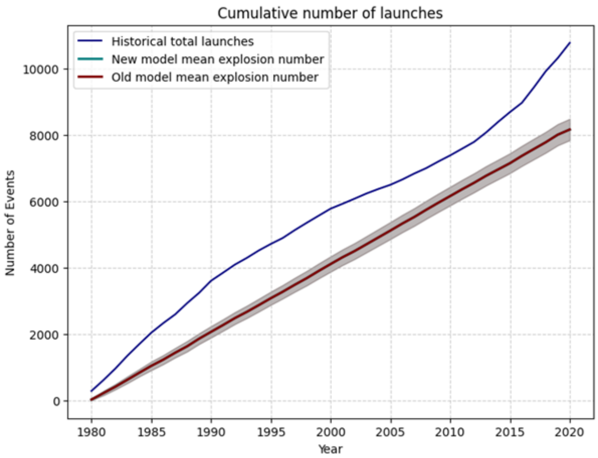


Figure A.1: Cumulative number of launches of historical verification case compared to reality

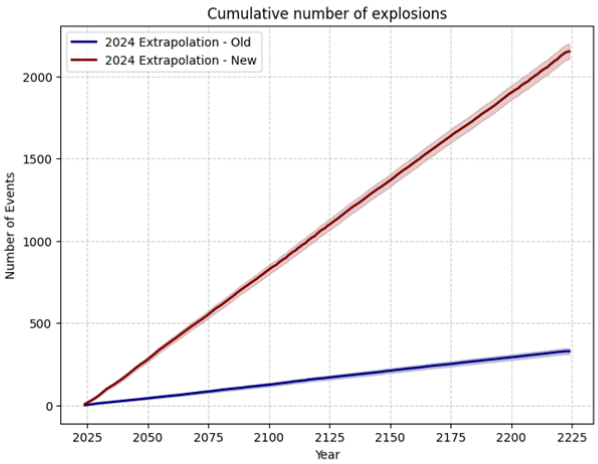


Figure A.2: Cumulative number of explosions in 2024 extrapolation case with new and old model

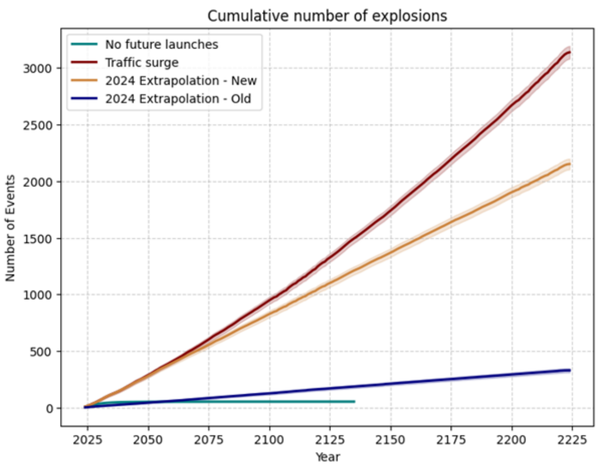


Figure A.3: Cumulative number of explosions in cases with varying launch traffic.

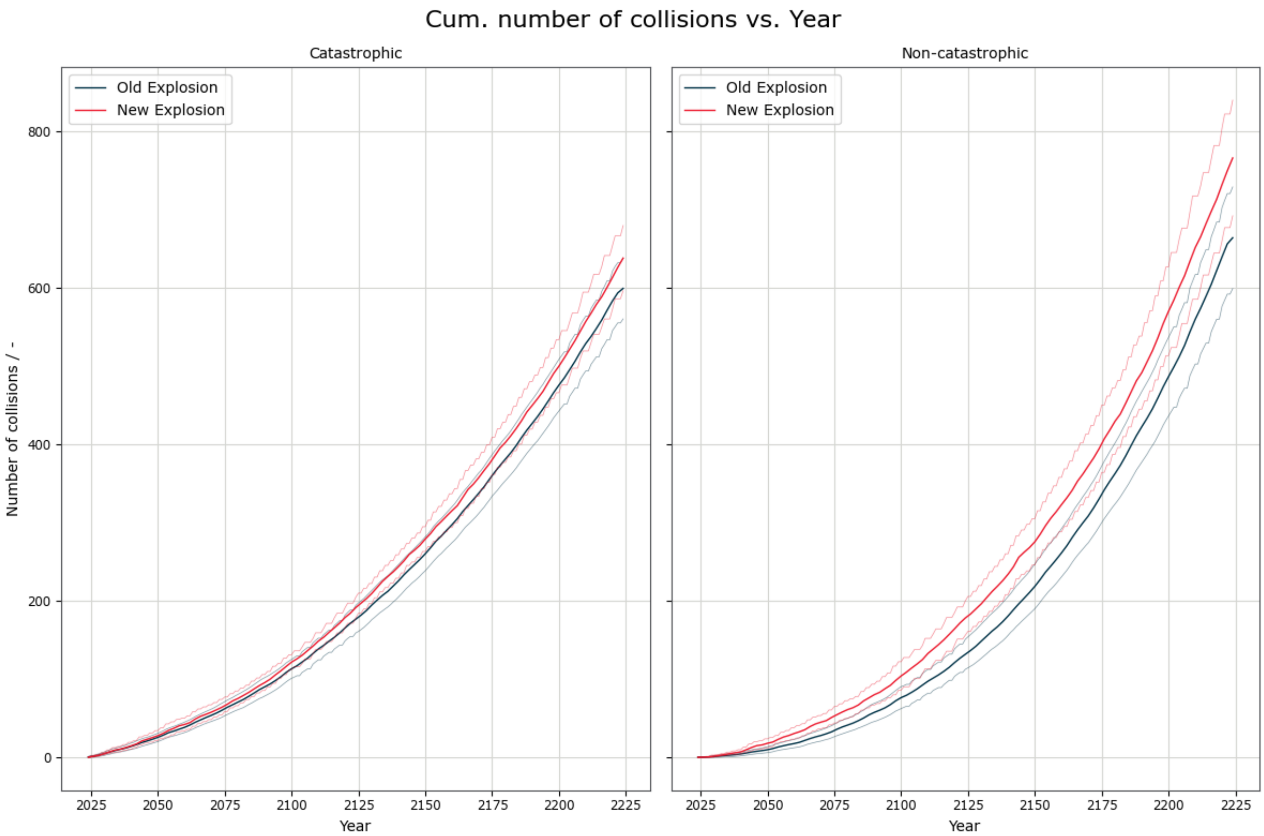


Figure A.4: Cumulative number of collisions in 2024 extrapolation case with new and old explosion module.