This is the accepted version of:

INTERPOLATION AND INTEGRATION OF PHASE SPACE DENSITY FOR ESTIMATION OF FRAGMENTATION CLOUD DISTRIBUTION

Stefan Frey[*], Camilla Colombo[†], and Stijn Lemmens[‡]

To calculate the effects of on-orbit fragmentations on current or future space missions, accurate estimates of the fragment density and its time evolution are required. Current operational tools estimate the risks involved through representative objects. Such tools, however, cannot accurately estimate the fragment density at any point in space and time. Rather, they directly calculate the number of close approaches from the representative objects. As such, they require a large number of Monte Carlo (MC) simulations to accurately find the collision risk over a large domain. Instead, the continuity equation can be applied to model the fragment density as a continuum, and propagate it forward in time. To model the evolution in any orbital region, the continuum can be propagated semi-analytically along its characteristics. The difficulty arises in estimating the density in between the cloud of samples.

Here, the underlying density distribution is estimated by fitting a Gaussian Mixture Model (GMM) to the characteristics. An example of a break-up in three dimensions is given. It is shown that the model can accurately be fitted at different snapshots after the fragmentation, even with a low number of sample points. Given an analytical expression of the density enables the subsequent integration of the collision risk at any point in the phase space.

INTRODUCTION

Hundreds of on-orbit fragmentations have occurred in the last six decades, resulting in thousands of trackable objects.[1] Such fragmentations can occur in any orbital region and potentially release hundreds of thousands of space debris, the majority of which is too small to be tracked from Earth. The number of fragments orbiting Earth with a diameter larger than 1 mm is assumed to be in excess of 100 million.[2] Even small fragments can carry enough kinetic energy to corrupt on-board instruments or lead to a loss of mission in case of an impact.[3] For a proper analysis of the ramifications of an on-orbit break-up, hypothetical or real, on current and future missions, the size spectrum down to 1 mm needs to be considered. However, the integration of all fragment trajectories becomes infeasible for such a vast number of particles.

One way of handling this constraint is through sampling of Representative Objects (ROs), implemented in long-term evolutionary tools such as DELTA.[4] To extract statistical information, the sampling is repeatedly applied during many Monte Carlo (MC) runs. Estimating the underlying,

---

[*]PhD Student, Dept. of Aerospace Science and Technology, Politecnico di Milano, Via La Masa, 34, 20156 Milano, Italy.

[†]Associate Professor, Dept. of Aerospace Science and Technology, Politecnico di Milano, Via La Masa, 34, 20156 Milano, Italy.

[‡]Space Debris Analyst, Space Debris Office, ESA/ESOC, Robert-Bosch-Strasse 5, 64293 Darmstadt, Germany.

non-parametric density distribution requires a large number of ROs or MC runs, especially as the dimensionality of the problem grows.

Another way of propagating the density of a cloud is by integrating the characteristics of the continuum equation which directly yield the density along each trajectory. McInnes used an analytical formulation to derive the debris population as a function of altitude.[5] Letizia et al. applied the approach to model the evolution of a fragment cloud and extended it to increase its applicability to low eccentric orbits.[6] These analytic formulations reduce the dynamics to a point where they are not applicable to any generic force model or orbital region.

Instead, the distribution can also be propagated using numerically or semi-analytically integrated characteristics.[7,8] The underlying distribution is estimated using interpolation between the scattered characteristics. Using Delaunay triangulation and linear interpolation, the initial phase space density can be estimated using two orders of magnitude less samples than the estimation through RO.[9] However, the interpolation, as well as the integration, over the full domain, $\mathcal{D}$, from a population of scattered samples is challenging for several reasons; e.g. non-convex distributions need careful treatment when estimating the shape from the point cloud; limited available memory does not allow small grid bin sizes for interpolation, especially in higher dimensions; distributions that are highly localised – i.e. have a small spread – in at least one dimension could not be accounted for if a rigid grid is used; integration of piece-wise linear function can result in non-smooth behaviour; etc.

The estimated fragment density is the input for subsequent collision risk calculations requiring an integration. Preferably, the density is available as a continuous function, rather than a piece-wise linear function that is expensive to be evaluated. Herein, a method is presented that fits a parametric, continuous function – a Gaussian Mixture Model (GMM) – to the characteristics, to estimate the phase space density evolution.

The paper is organised as follows: The following Section describes how to fit the GMM to the weighted samples, obtained from propagating the characteristics of the dynamics. Subsequently, the application of the proposed method to an evolving fragment cloud is shown and discussed. Finally, concluding remarks are given and future work is outlined.

METHOD

The idea of this work is to fit a model to a population of samples carrying information about the fragment density, $n \in \mathbb{R}$, as a function of the phase space variable, $\boldsymbol{x} \in \mathbb{R}^d$.* The samples and their weights, i.e. the densities, are obtained through propagation of the characteristics of the continuum equation.[5,7] The fitted model allows to evaluate n at any point in $\mathcal{D}$, and find the marginals of the distribution.

Model

The model used for the description of the fragment density is a GMM, defined as[10]

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta}_k) \tag{1}$$

---

*A section on mathematical notation is provided in the sequel.

with the model weights, $\boldsymbol{\pi}$, the multivariate normal distribution, $\mathcal{N}$, as

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} \tag{2}$$

and the parameters, $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are the mean, $\boldsymbol{\mu}$, the covariance, $\boldsymbol{\Sigma}$, and number of dimensions, D. The expectation-maximisation iterative method can estimate the latent parameters from given samples for finding the maximum likelihood solution for unweighted samples.[11] This method can be extended to take into account weighted samples, either by artificially adding identical data points where the weights are high, or similarly, by noting that raising the likelihood function of each sample point, $\boldsymbol{x}_i$, by its weight, w_i, corresponds to[12]

$$\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})^{w_i} \propto \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}/w_i) \tag{3}$$

However, for a good fit, the data needs to be independently and identically distributed. This cannot be guaranteed for a set of samples subject to a transformation.

Instead, the GMM is fit using a Gradient Descent (GD) optimization method to minimize a given cost function dependent on the weights, i.e. the densities, n_i, of each sample.

Gradient Descent

The GD uses the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno.[13] At every step, the numerically evaluated Hessian is updated rather than re-estimated from scratch. The implementation of the method in SCIPY[14] is used to derive the results presented herein. The chosen method is not necessarily the best nor fastest option, but it has proven to be reliable and accurate. Future work will include increasing the speed of convergence as well as the stability of the algorithm by calculating the Hessian analytically.

The initial condition for the optimization of the initial distribution is obtained through weighted k-means.[10] Subsequent distributions are optimized using the solution of the previous optimization procedure.

Cost function

First, a cost function leading to a constrained optimization problem is derived. The problem is bounded in $\boldsymbol{\pi}$, as the model weights are required to be positive. It is also constrained through the covariance matrix, $\boldsymbol{\Sigma}$, that is required to be symmetric and positive, semi-definite, and as such the elements of $\boldsymbol{\Sigma}$ are dependent. Then, through transformation of parameters, an unconstrained optimization problem is derived.

Constrained Problem The cost function to be minimized is defined as a least squares in the $\log$ space, $\mathcal{C}_N$, and a regularization term, $\mathcal{C}_R$, as

$$\mathcal{C} = \mathcal{C}_N + \mathcal{C}_R \tag{4a}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\ln f(\boldsymbol{x}_i) - \ln n_i\right)^2 + R\sum_{k=1}^{K}\pi_k^2 \tag{4b}$$

with the regularization term, R. The latter is needed to penalize kernels with large weights, with generally large covariances matrices. Such kernels add density to void spaces, which needs to be

3

avoided. In order to increase the speed of convergence during the GD optimization, an analytical expression of the derivative of $\mathcal{C}$ with respect to the model parameters is given. The derivative of $\mathcal{C}_N$ with respect to any model parameter, p, is

$$\frac{\partial \mathcal{C}_N}{\partial p} = \frac{2}{N} \sum_{i=1}^{N} \frac{(\ln f(\boldsymbol{x}_i) - \ln n_i)}{f(\boldsymbol{x}_i)} \frac{\partial f(\boldsymbol{x}_i)}{\partial p} \tag{5}$$

The derivative of f with respect to $\boldsymbol{\pi}$, using $\mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k} = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$, can be found as

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial \pi_k} = \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k} \tag{6}$$

The derivations of f in any $\boldsymbol{p}_\theta \in (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial \boldsymbol{p}_\theta} = \pi_k \frac{\partial \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k}}{\partial \boldsymbol{p}_\theta} = \pi_k \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k} \frac{\partial \ln \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k}}{\partial \boldsymbol{p}_\theta} \tag{7}$$

The derivation of $\ln \mathcal{N}$ in $\boldsymbol{\mu}$ can be found as

$$\frac{\partial \ln \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k}}{\partial \boldsymbol{\mu_k}} = \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) \tag{8}$$

and, using the identities[15]

$$\frac{\partial \ln(\det(\boldsymbol{X}))}{\partial \boldsymbol{X}} = \boldsymbol{X}^{-T} \tag{9a}$$

$$\frac{\partial \boldsymbol{a}^T \boldsymbol{X}^{-1} \boldsymbol{b}}{\partial \boldsymbol{X}} = -\boldsymbol{X}^{-T} \boldsymbol{a} \boldsymbol{b}^T \boldsymbol{X}^{-T} \tag{9b}$$

the derivation in Σ as

$$\frac{\partial \ln \mathcal{N}_{\boldsymbol{x}_i|\boldsymbol{\theta}_k}}{\partial \boldsymbol{\Sigma}_k} = -\frac{1}{2} \left(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1} \boldsymbol{S}_{\boldsymbol{x}_i|\boldsymbol{\mu}_k} \boldsymbol{\Sigma}_k^{-1} \right) \tag{10}$$

where we used the fact that the inverse of a symmetric matrix is also symmetric and

$$\boldsymbol{S}_{\boldsymbol{x}_i|\boldsymbol{\mu}_k} = (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \tag{11}$$

Finally, the derivation of $\mathcal{C}_R$ with respect to $\boldsymbol{\pi}$ is straightforward and can be found as

$$\frac{\partial \mathcal{C}_R}{\partial \pi_k} = 2R\pi_k \tag{12}$$

Unconstrained Problem Unconstrained problems are generally easier to solve as no restrictions constrain the parameters and only critical points with vanishing derivatives are potential minima of the problem. Here, the constrained problem is transformed into an unconstrained one simply by substitution of the constrained model parameters.

The model weights, $\boldsymbol{\pi}$, are required to be positive, as the Probability Density Function (PDF) is strictly non-negative in all of the domain, $\mathcal{D}$. Rather than optimizing in $\boldsymbol{\pi}$, the optimization is performed in the logarithmic model weights, $\boldsymbol{\nu}$

$$\boldsymbol{\nu} = \log \boldsymbol{\pi} \tag{13}$$

The Jacobian derived for the constrained problem requires small changes; The derivation of $\mathcal{C}_N$ with respect to $\boldsymbol{\nu}$ (replacing Equation 6) is

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial \nu_k} = \frac{\partial f(\boldsymbol{x}_i)}{\partial \pi_k} \frac{\partial \pi_k}{\partial \nu_k} = \pi_k \mathcal{N}_{\boldsymbol{x}_i | \boldsymbol{\theta}_k} \tag{14}$$

and the derivation of $\mathcal{C}_R$ with respect to $\boldsymbol{\nu}$ (replacing Equation 12) is

$$\frac{\partial \mathcal{C}_R}{\partial \nu_k} = \frac{\partial \mathcal{C}_R}{\partial \pi_k} \frac{\partial \pi_k}{\partial \nu_k} = 2R\pi_k^2 \tag{15}$$

Instead of optimizing the dependent components of $\boldsymbol{\Sigma}$, each covariance matrix is split into the lower, left triangle matrix, $\boldsymbol{L}$, such that

$$\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T \tag{16}$$

Starting from $\boldsymbol{\Sigma}$, $\boldsymbol{L}$ can be found efficiently using the Cholesky factorisation.[16] The derivation of $\boldsymbol{\Sigma}$ with respect to $\boldsymbol{L}$ is

$$\frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{L}} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{L}}\boldsymbol{L}^T + \boldsymbol{L}\frac{\partial \boldsymbol{L}^T}{\partial \boldsymbol{L}} \tag{17a}$$

$$= \boldsymbol{L} \otimes \boldsymbol{I} + (\boldsymbol{I} \otimes \boldsymbol{L})\boldsymbol{K} \tag{17b}$$

with the identity matrix, $\boldsymbol{I} \in \mathbb{R}^{D \times D}$, the Kronecker product, $\otimes$, and the square commutation matrix, $\boldsymbol{K} \in \mathbb{R}^{D^2 \times D^2}$.[17] Thus, the derivation of $\mathcal{C}_N$ with respect to the components of $\boldsymbol{L}$ (using Equation 10) is

$$\frac{\partial \ln \mathcal{N}_{\boldsymbol{x}_i | \boldsymbol{\theta}_k}}{\partial L_k^{(j,l)}} = \text{vec}\left(\frac{\partial \ln \mathcal{N}_{\boldsymbol{x}_i | \boldsymbol{\theta}_k}}{\partial \boldsymbol{\Sigma}_k}\right)^T \frac{\partial \boldsymbol{\Sigma}_k}{\partial \boldsymbol{L}_k}\bigg|_q \tag{18}$$

with

$$j = 1, \ldots, D$$
$$l = 1, \ldots, j$$
$$q = j + D(l - 1)$$

and $\boldsymbol{A}|_q$ the q^{st} column of $\boldsymbol{A}$. Note that the columns corresponding to the values above the diagonal ($l > j$) of $\boldsymbol{L}$ need not be computed as they are fixed to be zeros. The vectorization, vec, stacks the columns vertically such that

$$\text{vec}(\boldsymbol{A}) = \text{vec}\left(\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}\right) \tag{20a}$$

$$= (a_{11}, \ldots, a_{n1}, \ldots, a_{1m}, \ldots, a_{nm})^T$$

For a symmetric matrix, $\boldsymbol{A} = \boldsymbol{A}^T$, therefore $\text{vec}(\boldsymbol{A}) = \text{vec}(\boldsymbol{A}^T)$.

Curse of Dimensionality

Given the number of dimensions, D, and the number of kernels, K, the number of parameters to be fitted can be calculated. Each kernel is defined in D dimensions and as such is fully described with D means, $\frac{D^2+D}{2}$ covariance parameters (lower left triangle), and one weight. Thus, the total number of parameters to be fitted is $K(\frac{D^2+3D}{2}+1) \propto KD^2$. This is a modest growth comparing to techniques that require to evaluate the density at fixed bins that grow with number of bins to the power of D. To further reduce the growth in D – with an accompanied increase in K – only the diagonal of Σ could be fit.

Hyperparameter Selection

The number of kernels, K, as well as the regularisation term, R, are hyperparameters to be set before the optimization. To find appropriate values, Cross-Validation (CV) is performed;[10] First, the data is randomly split into B bins. Then, the model in Equation 1 is fit to the combined training data of $B - 1$ bins, and $\mathcal{C}_N$ is evaluated using the validation data in the remaining bin. This process is repeated for all combinations of bins. Finally, the average cost is calculated. Performed for different K and R, CV is a powerful tool of discerning between under- and overfitting the data. Herein, the number of bins is set to $B = 10$, as a trade-off between speed and accuracy of the CV.

Marginalisation

The marginalisation of the multivariate normal distribution over one or more distributions is beautifully simple, as the result is another multivariate normal distribution. The new mean and covariance are simply the partitioned mean and covariance of the marginalised distribution.[10] The extension to the marginalisation of a GMM is trivial. The number of fragments, P_f, being represented by the GMM at each snapshot at time t is

$$P_f(t) = \sum_{k=1}^{K} \pi_k(t) \tag{21}$$

The result is compared to the total number of fragments found through the method of representative objects, P_r

$$P_r(t) = \frac{N_c(t)}{N_c(t_0)} \tag{22}$$

with the number of characteristics, N_c, present at each time step. I.e. the representative objects are the same as the characteristics, however all with a fixed weight of $N_c^{-1}(t_0)$. Since P_r is a 0-dimensional quantity, it is accurately estimated even for a low number of samples. Note that P_r could also be used during the optimization procedure. However, as it only gives an idea about the total number and not the underlying distribution, it is only used as a measure of quality of fit herein.

Pruning and Normalization

The data to be fitted is pruned and normalized at each snapshot. The lowest weighted samples with a combined weight of 0.1% of the total weight are removed. After pruning, the data is normalized such that each dimension has a unity spread, i.e.

$$\boldsymbol{X}|_d = \frac{\boldsymbol{Y}|_d}{(\max \boldsymbol{Y}|_d - \min \boldsymbol{Y}|_d)} \quad \forall\, d \tag{23}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the normized and non-normalized sample set, and d denotes the dth column.

The initial distribution from which the samples are drawn is normalized, such that it integrates to unity. However, the total number of fragments, N_f, can always be recovered from the estimated normalized number of fragments, P_f, by multiplying it with the initial number of fragments, $N_f(t_0)$, given by the break-up model, as

$$N_f(t) = P_f(t)N_f(t_0) \tag{24}$$

RESULTS AND DISCUSSION

Characteristics

The proposed method is put to the test using an evolving cloud in Low Earth Orbit (LEO) with a phase space consisting of $\boldsymbol{x} = (a, e, A/m) \in \mathbb{R}^3$, with semi-major axis, a, eccentricity, e, and the logarithmic area-to-mass ratio, χ. The latter is defined as

$$\chi = \log_{10}\left(\frac{A}{m}\right) \tag{25}$$

with the area-to-mass ratio, A/m. The initial distribution in the three variables is converted from the National Aeronautics and Space Administration (NASA) break-up model,[18] simulating a rocket body explosion.

The characteristics are selected by sampling the initial distribution using the Metropolis-Hastings algorithm.[19] The samples, in the extended phase space containing the density, n, are integrated as in[9]

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{F} \tag{26a}$$

$$\frac{\mathrm{d}n}{\mathrm{d}t} = -n\operatorname{tr}\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} \tag{26b}$$

with the dynamics, $\boldsymbol{F}$, and the trace, tr. Only atmospheric drag is considered as perturbing force here. Figure 1 shows the initial and propagated characteristics for the snapshots at $t = 0$, 20 and 40 years after fragmentation. Three figures per snapshot show the distribution, each with one dimension collapsed. The colour scales are the same for all characteristics plots and represent the fragment density at each point. From an initial $N_c = 1000$ there are 542 and 262 remaining characteristics in orbit after 20 and 40 years respectively. The others are removed through atmospheric drag acting as a sink. After 80 years, N_c drops to below 100. The density in each characteristics decreases due to the exponential increase of the drag forces with lowering altitude, which has a diverging effect on the density. I.e. the radial distance between fragments with the same A/m increases over time and thus the fragment density decreases.

Hyperparameter Selection

To select an appropriate number of kernels, K, and regularisation term, R, CV is performed over the range of $K = 1, \ldots, 9$ and $R = 10^{-3}, \ldots, 10^1$, for the three different snapshots. From this grid, the turning point of under- (decreasing cost) to overfitting (increasing cost) can be discerned. Figure 2 shows the averaged least squares cost – evaluated on the validation data – per scenario. For $K \leq 5$, and $R \geq 10^{-1}$, the cost decreases with increasing K and decreasing R, i.e. the model fits the underlying data better. For $K \geq 7$, and $R \leq 10^{-2}$ overfitting can be observed for increasing K

7
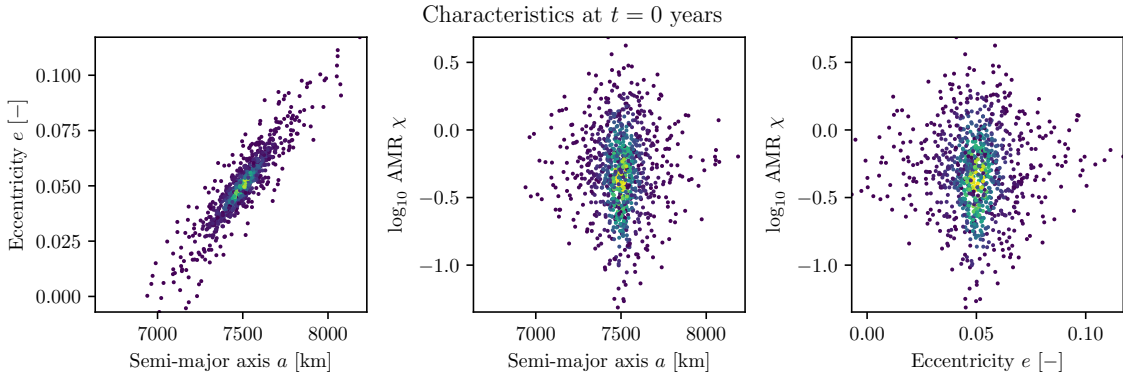
(a) Initial 1000 characteristics.
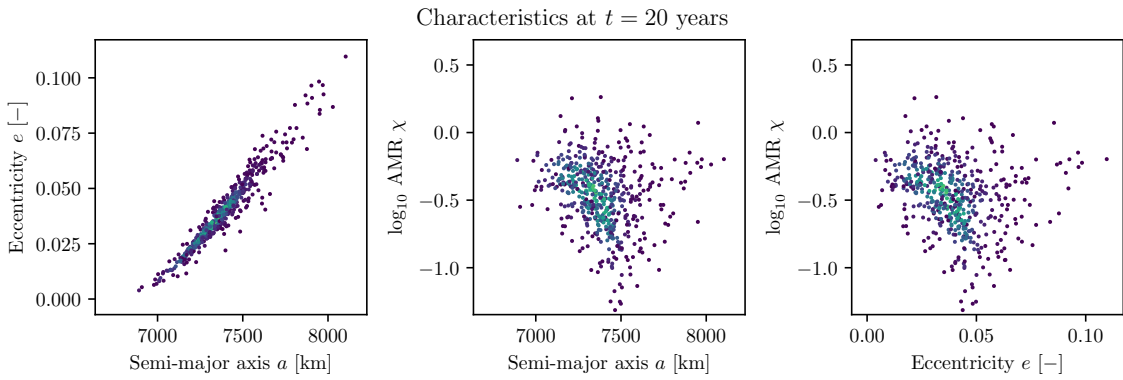
(b) Remaining 542 characteristics after $t = 20$ years.
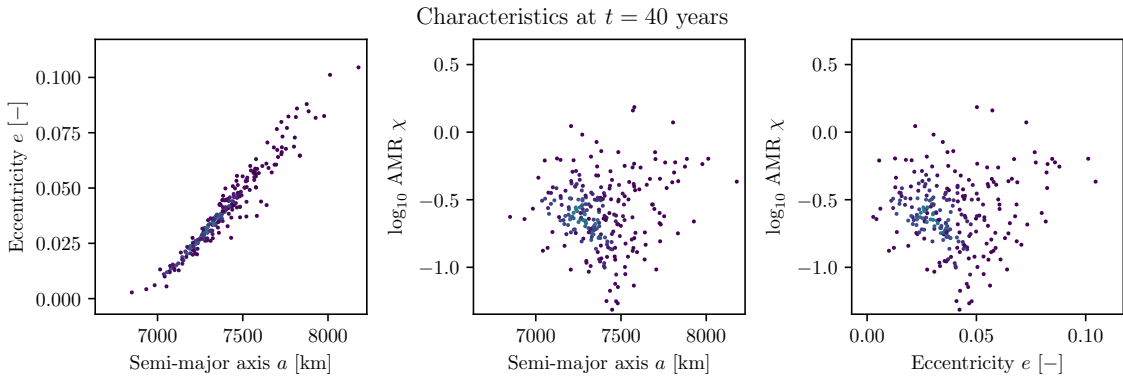
(c) Remaining 262 characteristics after $t = 40$ years.

Figure 1: Evolution of the characteristics in the phase space t years after the fragmentation. The colour ranges are fixed and the same for each plot (light for high and dark for low densities). Characteristics with low perigee and/or high A/m re-enter the fastest.
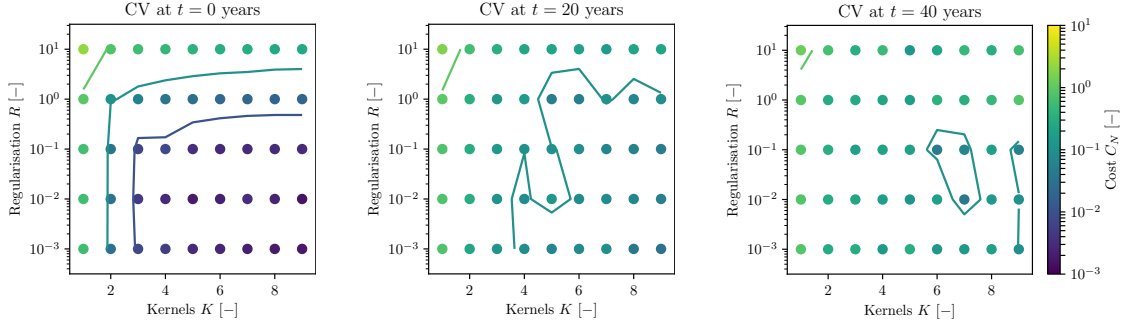
(a) Using the initial 1000 charac-
teristics.

(b) Using the remaining 542 char-
acteristics.

(c) Using the remaining 262 characteristics.

Figure 2: CV for the hyperparameter selection. Each dot shows the average least squares cost for the given selection of K and R. The lines show the contours of the cost function. The color scale is fixed across the three plots.

and decreasing R, especially if the number of samples are low. Note that the cost becomes higher with time, a combination between a more complex underlying distribution and fewer points to fit against. Subsequently, the hyperparameters are fixed at $K = 6$ and $R = 10^{-1}$, which seems a good selection even for the evolving density and a changing N_c.

The selection of hyperparameters is only valid for the given distribution. It needs to be shown whether or not K and R can be fixed for different types of fragmentations in any orbit.

Density Fitting

Figure 3 shows the resulting density fit for the three snapshots, using the proposed method. For illustration purposes, each figure is marginalised over one dimension and the resulting distribution plotted in the remaining two dimensions. The ranges in color scale are different for the different marginalisations (since the scale is different in each dimensions). However, they are fixed for the different snapshots, to give a feeling of the diminishing nature of the density. While initially, the distribution is rather simple, it quickly becomes more complex due to the dynamics. The distribution of the fragments starts to deviate towards re-entry most where the drag forces are highest. This is true for fragments with perigees in low altitudes, as well as fragments with high A/m.

The average processing time on a Intel® Xeon® Processor E5-4620 v4, 2.10 GHz processor is 8.8 seconds per snapshot. There is still a lot of improvement possible as the used optimization procedure is not optimized for speed.

To get a feel of the quality of the fit, the total number of fragments, P_f, is compared to the number estimated through the RO method. Figure 4a shows the comparison of the total marginal using the two different methods for the cloud evolution over 100 years. Initially, the fitted distribution slightly underestimates the number of fragments, due to high peaks in the initial distribution that cannot be covered with $K = 6$ kernels. Over time, the fit starts overestimating the number of fragments. Probable causes could be that the underlying distribution is too complex to be modelled by a GMM, or that it fills voids where there are no characteristics to act against. However, the least squares error remains stable after an initial period of worsening (see Figure 4b). After 50 years to the end of the

9
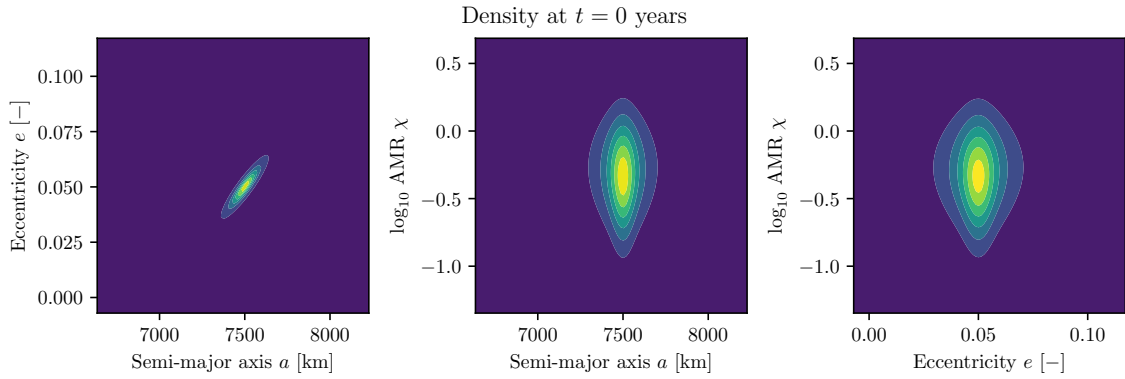
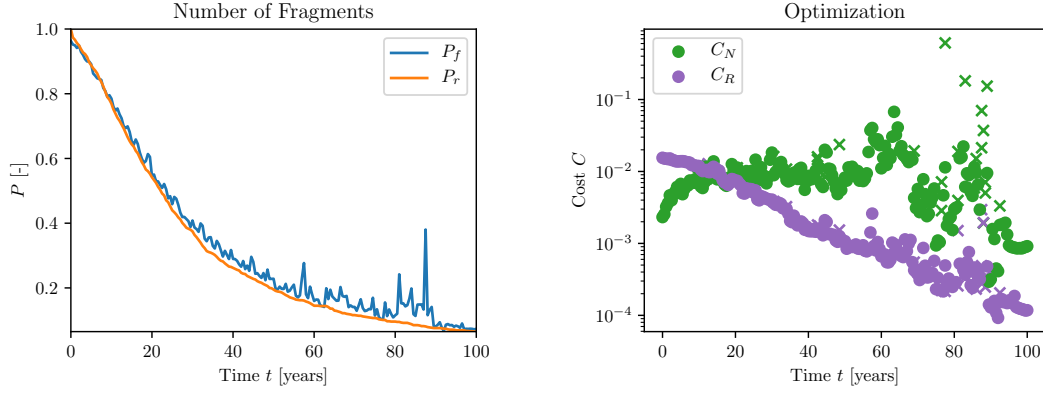(a) Density estimated from 1000 initial characteristics.
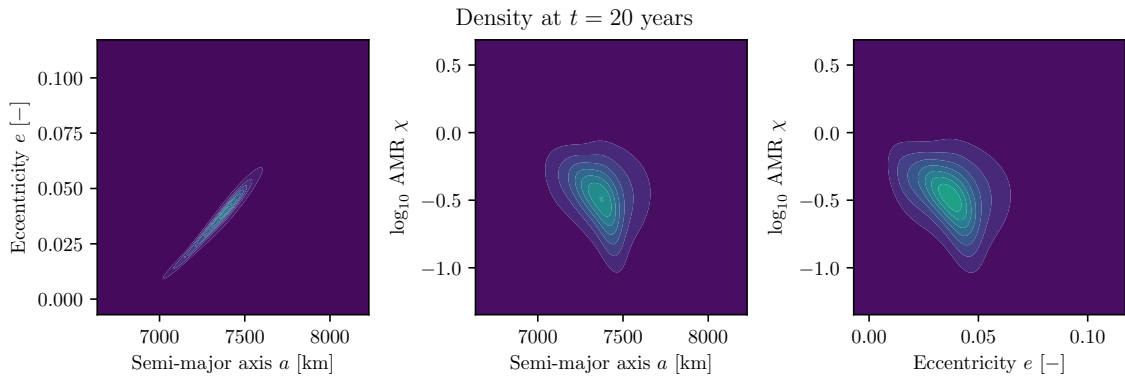
(b) Density estimated from 542 characteristics after $t = 20$ years.
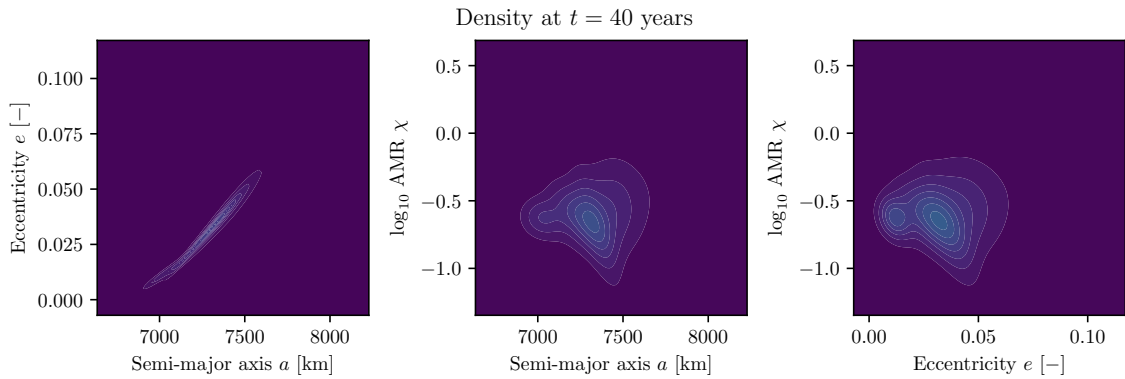
(c) Density estimated from 262 characteristics after $t = 40$ years.

Figure 3: Evolution of the estimated density, marginalised over one dimension each. The colour ranges are fixed for each vertical row of plots.

(a) The estimated initially normalized number of fragments on orbit, for the two different methods.

(b) Cost per iteration. The x mark a non successful optimization result.

Figure 4: Optimization results.

propagation, the quality of the density fit becomes poor due to the low number of sample points present, which is in the same order of magnitude as the number of parameters to be fitted. The optimization routine starts to reach the maximum number of iterations and stops unsuccessfully. This is shown in Figure 4b where the crosses mark non successful optimization results.

To improve the quality of the fit, and suppress spiky results that will negatively affect the collision risk estimation accuracy, new characteristics could be sampled from the estimated distribution, then back-propagated to get the true density, n. Subsequent re-fitting using the increased sample population would increase the quality of the fit.

CONCLUSION

The proposed model and method accurately estimates the underlying distribution for an explosion in LEO, given that a sufficient number of characteristics is present. As such, it could be applied in assessing the ramifications of hypothetical break-ups in terms of collision risk for current or future missions. Current tools rely either on MC simulations, which are not sensitive towards addition of a single cloud of fragments, and are too slow to be used for many different initial conditions. The assessments could be one part of a rating scheme, classifying future missions in terms of risks towards the space environment, ultimately mitigating the risk of space debris. For such an index to be accepted on an international level, in needs to consist of accurate and understandable methods and tools, to which this work aims to contribute.

Future work include extending the method into more dimension, i.e. to cover all the slow moving orbital elements plus A/m, and make the resulting density fit more smooth, imperative for accurate collision risk estimates. The phase space density needs to be translated into a spatial density for subsequent collision risk analysis. Additionally, the speed of the fitting procedure needs to be increased for application of the method in operations.

ACKNOWLEDGEMENT

REFERENCES

[1] ESA Space Debris Office, "ESA's Annual Space Environment Report, Produced with the DISCOS Database," 2018.

[2] IADC, "Space Debris, IADC Assessment Report for 2011," 2013.

[3] G. Drolshagen, "Hypervelocity Impact Effects on Spacecraft," *Proceedings of the Meteoroids 2001 Conference*, 2001.

[4] B. Bastida-Virgili, "DELTA Debris Environment Long-Term Analysis," *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2016.

[5] C. R. McInnes, "An Analytical Model for the Catastrophic Production of Orbital Debris," *ESA Journal*, Vol. 17, 1993, pp. 293–305.

[6] F. Letizia, C. Colombo, and H. G. Lewis, "Multidimensional extension of the continuity equation method for debris clouds evolution," *Advances in Space Research*, Vol. 57, 2016, pp. 1624–1640.

[7] N. N. Gor'kavyi, L. M. Ozernoy, and J. C. Mather, "A new approach to dynamical evolution of interplanetary dust," *The Astrophysical Journal*, Vol. 474, 1997, pp. 496–502.

[8] A. Wittig, C. Colombo, and R. Armellin, "Long-term density evolution through semi-analytical and differential algebra techniques," *Celestial Mechanics and Dynamical Astronomy*, Vol. 128, 2017, pp. 435–452.

[9] S. Frey, C. Colombo, and S. Lemmens, "Evolution of Fragmentation Cloud in Highly Eccentric Orbit through Continuum Modelling," *Proceedings of the 69th International Astronautical Congress*, 2018.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] A. P. P. Dempster, N. M. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, 1977, pp. 1–38.

[12] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 12, 2016, pp. 2402–2415.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[14] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open source scientific tools for Python," 2001.

[15] K. B. Petersen and M. S. Pedersen, "The Matrix Cookbook," 2012.

[16] R. A. Horn and C. R. Johnson, *Matrix Analysis, Second Edition*. Cambridge University Press, 2013.

[17] J. R. Magnus and H. Neudecker, "The commutation matrix: some properties and applications," *The Annals of Statistics*, Vol. 7, No. 2, 1979, pp. 381–394.

[18] N. L. Johnson, P. H. Krisko, J.-C. Liou, and P. D. Anz-Maedor, "NASA's new breakup model of EVOLVE 4.0," *Advances in Space Research*, Vol. 28, No. 9, 2001, pp. 1377–1384.

[19] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, Vol. 49, No. 4, 1995, pp. 327–335.