

Data Mining and Knowledge Discovery from Store Separation Trajectories

S. S. Vaddi,¹ J. Nguyen,² and P. K. Menon³
Optimal Synthesis Inc, Los-Altos, CA, 94022

The trajectory of a store released from an aircraft is subject to the uncertainty in parameters such as inertia properties, aerodynamic coefficients and external factors. Extensive Monte-Carlo simulations are conducted for certifying the safety of store separation. An enormous amount of data is generated in this process demanding specific tools for further analysis. The objective of this work is to develop tools to address questions such as (i) what parameters cause un-safe store trajectories (ii) what are the worst-case combinations of parameters (iii) how can un-safe trajectories be avoided and (iv) what level of parameter uncertainty is acceptable for store certification. Techniques from data mining and machine learning tools such as recursive least squares, principal component analysis, K-means clustering, and probability binning are employed in this work to address these questions.

I. Introduction

Fighter and bomber aircraft carry stores such as missiles, bombs and drop-tanks to execute their missions. Safe and acceptable separation of these stores is essential for meeting the mission objectives. The jettisoning systems may incorporate mechanical springs, pneumatic actuators and pyrotechnics to ensure that sufficient forces are generated to eject the store away from the aircraft. Safe separation requires the store to have a constant or increasing vertical velocity with respect to the aircraft, while staying within the attitude and rate limits. Acceptable performance requires that the release transients do not cause the store to fail in its mission. Shown in Figure 1 is an illustration of acceptable and unacceptable store trajectories. Actual figures are not included in this version of the paper as Air Force approval for public release is still pending. The final version of the paper is expected to include these figures.

Due to the variations in the store parameters such as mass, moment of inertia, center of mass location, aerodynamic coefficients, and ejection system parameters, no two store trajectories are exactly alike. The objective of the store separation trajectory analysis is to establish safe and acceptable operating envelopes for the store based on preflight data, and to assure high likelihood of successful flight tests. One at a time variations, Monte-Carlo sampling and genetic algorithm searches are conducted to explore the parameter space of the store.

Initial investigations examine the sensitivity of a few key parameters on the store trajectories to help focus attention on the critical regions of the flight envelope. Monte-Carlo simulations are conducted to determine the likelihood of safe and acceptable separation in the presence of these parameter variations. Genetic algorithm searches are used to iteratively adjust the parameters within the range of expected variability to determine the worst-case combination of inputs causing unacceptable trajectories. If safe and acceptable performance can be assured from these simulation studies, positive flight test outcomes are highly probable. The post-flight trajectory results can be back correlated with Monte-Carlo simulations to reconcile with the preflight predictions.

This analysis process generates enormous number of trajectories, which may contain important relationships, not easily discernable by the analyst. Data mining and knowledge discovery (DMKD) techniques [1 - 7] can be employed to discover these hidden relationships. These can then be explored using a variety of data visualization tools such as 2-D/3-D line plots, bar graphs, contour, mesh and surface plots. Knowledge gained in this manner can be used by the analyst as decision aids for generating specifications for store separation systems and for assuring high likelihood of success in subsequent flight tests and operations.

¹Research Scientist, 95 First Street, Suite 240, AIAA Member.

²Research Engineer, 95 First Street, Suite 240.

³ Chief Scientist, 95 First Street, Suite 240, Associate Fellow, AIAA.

Subject to Air-Force Approval for Public Release

Figure 1. Store Separation Schematic

The focus of the present work is analysis of the Monte-Carlo simulation data. Shown in Figure 2 is a schematic of the Monte-Carlo simulation. Parameters are sampled from pre-modeled distributions such as normal, uniform and empirical. Trajectory corresponding to these parameters is obtained from a six degree of freedom simulation. Trajectories thus obtained are analyzed for safe and acceptable behavior.

- A linear dynamic model approximation based on recursive least squares formulation [8, 9] is presented in Section II. The model is further utilized to estimate relative sensitivity of store trajectory to its parameters, compute the worst case combination of input parameters and predict the covariance history of the store trajectory.
- Automatic clustering of trajectories based on principal component analysis [10] (PCA) and K-means [10] clustering technique is discussed in Section III. Trajectories are clustered in an unsupervised manner into qualitatively similar groups.
- Probability binning methodology [11] is used in Section IV to compute the sensitivity of parameters to a given failure criteria.
- A numerical technique called fail-safe clustering is developed in Section V to compute a hypercube in input parameter space that is safe with respect to a given failure criteria.

Subject to Air-Force Approval for Public Release

Figure 2. Schematic of the Monte-Carlo Simulation

II. Linear Model Approximation

The following will discuss the construction of the linear model from Monte-Carlo simulation data, and then show its use in conducting various analyses. The nonlinear 6-DOF equation of motion of the store in discrete time can be represented as:

$$X(i+1) = f(X(i), u) \quad (1)$$

The nonlinear functions $f(\cdot)$ are the right hand sides of the 6-DOF equations of motion. These functions consist of the forces and moments acting on the store, and kinematic nonlinearities arising from the coordinate transformations. In Equation (1), $X(i)$ is the (12×1) store state vector:

$$X \equiv [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \psi \ \theta \ \phi \ \dot{\psi} \ \dot{\theta} \ \dot{\phi}]^T \quad (2)$$

This vector consists of the store relative position with respect to the aircraft, velocity, attitude and attitude rates. The input vector u represents the parameters of the store and is a constant vector.

$$u = [u_1 \ u_2 \ u_3 \ \dots \ u_m] \quad (3)$$

Given the 12 initial conditions and the inputs u , the nonlinear model is used to simulate the store trajectories. Although useful for simulations, it is difficult to use this model for any formal analysis due to the nonlinearities represented by the functions $f(\cdot)$. However, following the well-recognized tradition in aeronautics [21], these equations can be approximated by a linear model about a *nominal* trajectory, and then used to derive useful results.

The linear model describes the perturbations of the states from the nominal trajectory. For the store separation problem, the nominal trajectory can be computed by setting all the inputs to their nominal values, and carrying out the trajectory simulation with nominal inputs and initial conditions. Next, individual input components are perturbed one by one in both positive and negative directions to generate the variation trajectories. Trajectory perturbations are then defined as the difference between a perturbed trajectory and the nominal trajectory. For the present development, define the trajectory perturbations as:

$$\delta(i) = X(i) - \bar{X}(i) \quad (4)$$

$\bar{X}(i)$ is the nominal trajectory generated using the nominal initial conditions and nominal inputs \bar{u} , and $X(i)$ is the trajectory with perturbed inputs u and initial conditions. Define the perturbations in the inputs as: $U = u - \bar{u}$. Perturbed states are given by the vector $\delta(i)$.

The linear dynamic model defines linear relationships between the input perturbations U and the trajectory perturbations $\delta(i)$. This approximate linear relationship can be derived using a Taylor series expansion [21] of the right hand sides of the 6-DOF equations of motion about the nominal trajectory, as:

$$\delta(i+1) = \left[\frac{\partial f}{\partial X} \right]_{X=\bar{X}, u=\bar{u}} \delta(i) + \left[\frac{\partial f}{\partial u} \right]_{X=\bar{X}, u=\bar{u}} U \quad (5)$$

The matrices multiplying the state perturbations and the input perturbations are the Jacobian matrices evaluated at the nominal values of the states and the inputs. If the system dynamic equations are not explicitly available, the given trajectories can be used to numerically construct the linear model, as will be illustrated in the following section.

The linear dynamic system can be expressed in a more compact form as:

$$\delta(i+1) = A \delta(i) + BU \quad (6)$$

where A (12×12 matrix) and B ($12 \times m$ matrix) are the Jacobian matrices given by:

$$A = \left[\frac{\partial f}{\partial X} \right]_{X=\bar{X}, u=\bar{u}} \quad B = \left[\frac{\partial f}{\partial u} \right]_{X=\bar{X}, u=\bar{u}} \quad (7)$$

At any sample instant k , the state perturbations can be computed in terms of initial condition, and the input as [22]:

$$\delta(k) = A^k \delta(0) + \sum_{j=0}^{k-1} A^{k-j-1} BU, \quad k = 1, 2, 3, \dots \quad (8)$$

In a more compact form:

$$\delta(k) = Q_1(k) \delta(0) + Q_2(k) U \quad (9)$$

The 12×12 matrix $Q_1(k)$ relates the perturbations in the initial conditions to the states, while the $12 \times m$ matrix $Q_2(k)$ relates the perturbations in the inputs to the perturbations in the states.

$$Q_1(k) = A^k \quad Q_2(k) = \sum_{j=0}^{k-1} A^{k-j-1} B \quad (10)$$

If the perturbations in the initial conditions and the inputs are Gaussian, this linear model is sometimes known as the *Gauss-Markov* model [15]. The following sections will first illustrate how the coefficients of the linear model can be derived from the given trajectories.

A. Recursive Least Squares Formulation

Recursive Least Squares (RLS) algorithm is used to estimate the A and B matrices in the linear model. The advantage of the RLS algorithm when compared to the more familiar batch least squares algorithm is that it can handle a large number of trajectories without encountering computer memory problems. Moreover, it offers the convenience of being able to refine the estimates of the A and B matrices as new trajectories become available.

As discussed in the previous section, the linear dynamic model is of the form:

$$\delta[i+1] = A\delta[i] + BU \quad (11)$$

Consequently, a complete trajectory can be represented by linear equations of the form:

$$Y = Q*S \quad (12)$$

Here, the vector Y and the matrix Q are formed by the state components and the vector S is formed by the elements of the A and B matrices.

$$S = [A(1,:) \quad A(2,:) \quad . \quad . \quad B(1,:) \quad B(2,:) \quad . \quad .]^T \quad (13)$$

Such systems of linear equations can be defined for every trajectory used in the variation analysis. Thus:

$$Y_i = Q_i*S \quad (14)$$

where $i = 1 \dots n$.

The recursive least squares algorithm is given the following equations [8]:

$$S_{k+1} = S_k + K_k [Y_{k+1} - Q_{k+1}S_k] \quad (15)$$

$$K_k = P_k Q_{k+1}^T [Q_{k+1} P_k Q_{k+1}^T + I]^{-1} \quad (16)$$

$$P_{k+1} = P_k - P_k Q_{k+1}^T [Q_{k+1} P_k Q_{k+1}^T + I]^{-1} Q_{k+1} P_k \quad (17)$$

$P(1)$, the initial value of P is assumed to be a large value. Initial values of A is assumed to be an identity matrix, while all the initial entries of the B matrix are all assumed to be zero. These values are used to define the initial value of S .

The parameter estimation process can be further simplified by recognizing that the store trajectory dynamics consists of dynamic and kinematic equations. Since the relationships between these are given by pure integrations, only the parameters associated with the dynamic equations need be estimated. With this fact in view, the A and B matrices partitioned into the form:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \quad (18)$$

with,

$$A_{11} = 0 \quad A_{12} = I\Delta t \quad B_1 = 0$$

Only the elements of the sub-matrices A_{21} , A_{22} and B_2 need be estimated by the RLS algorithm. Note that the partitioning of the system dynamics into kinematic and dynamic equations reduces the number of parameters to be estimated in half.

This linear model is used to derive a variety of useful results as will be illustrated in the following sections.

B. Sensitivity to Parameters

An important step in store separation analysis is that of estimating the relative importance of the input components on the trajectory behavior. Although these relationships can be established for any of the trajectory parameters, the following development will illustrate the relative importance of the input components on the store states.

The relationship between the input vector and the state vector at any sample instant k was shown in Equation (9) to be:

$$\delta(k) = Q_1(k)\delta(0) + Q_2(k)U \quad (19)$$

The matrices Q_1 and Q_2 are time varying. The notation $Q_1(i, j, k)$, $Q_2(i, j, k)$ will be used to refer to the i^{th} row j^{th} column elements of these matrices at the k^{th} time instant. The expressions for individual state components are of the form:

$$\delta_i(k) = Q_1(i,1,k)\delta_1(0) + \dots + Q_1(i,12,k)\delta_{12}(0) + Q_2(i,1,k)U_1 + \dots + Q_2(i,35,k)U_m \quad (20)$$

The quantity $Q_2(i, j, k)$ is a measure of the influence of the j^{th} input on the chosen state variable at time k . However, straight forward comparison of two $Q_2(i, j, k)$'s corresponding to two different inputs determined by the choice of two j 's can be misleading. This is due to the fact that the dynamic range of different inputs may be different.

In order to enable the comparison between any input components, it is necessary to normalize the input variables so that all of them have the same dynamic range. For the present research, normalizing constants for the inputs are chosen as their individual standard deviations, thus:

$$\tilde{U}_j = \frac{U_j}{u_{std_j}} = \frac{u_j - \bar{u}_j}{u_{std_j}} \quad (21)$$

As the mean input has already been subtracted from individual inputs during linearization, normalization with individual standard deviations u_{std_j} maps all inputs to the range of ± 3 . Rewriting the influence equation in terms of the normalized inputs,

$$\delta_i(k) = \text{Initial Condition Terms} + Q_2(i,1,k)u_{std_1}\tilde{U}_1 + Q_2(i,2,k)u_{std_2}\tilde{U}_2 + \dots + Q_2(i,m,k)u_{std_m}\tilde{U}_m \quad (22)$$

Rewriting the above equation by grouping the $Q_2(i, j, k)$ and the u_{std_j} terms and defining

$$\tilde{Q}_2(i, j, k) = Q_2(i, j, k)u_{std_j},$$

$$\delta_i(k) = \text{I.C.Terms} + \tilde{Q}_2(i,1,k)\tilde{U}_1 + \tilde{Q}_2(i,2,k)\tilde{U}_2 + \dots + \tilde{Q}_2(i,m,k)\tilde{U}_m \quad (23)$$

At any sample k , the input normalized equation allows the direct comparison of the relative influence of the inputs on the state perturbations using the coefficients $\tilde{Q}_2(i, j, k)$.

In order to determine the influence of individual inputs over the entire time history of the trajectory, define the RMS influence coefficient as:

$$RMS(\tilde{Q}_2(i, j)) = \sum_{k=0}^N \sqrt{\frac{\tilde{Q}_2^2(i, j, k)}{N}} \quad (24)$$

C. Worst-Case Combination of Parameters

The linear dynamic model can be used to determine the worst-case combinations of inputs that will drive the outputs towards the direction of failures. Finding the worst-case combinations are particularly simple if the failure cases are defined as maximum or minimum values of the state variables at a specified sample instant. This will be illustrated in the following.

As shown in Equation (9), the perturbations in the states are related to the input perturbations through the linear relationship:

$$\delta_i(k) = \text{I.C.Terms} + Q_2(i,1,k)U_1 + \dots + Q_2(i,m,k)U_m \quad (25)$$

Since the input perturbations appear linearly in Equation (25), the values of the input components U_j that maximize $\delta_i(k)$ are given by:

$$U_j = \begin{cases} U_{j_{\max}}, & Q_2(i, j, k) > 0 \\ U_{j_{\min}}, & Q_2(i, j, k) < 0 \end{cases} \quad (26)$$

The values of the inputs that minimize $\delta_i(k)$ can be determined by simply by reversing the inequalities in the above expression.

D. Uncertainty Prediction

Monte-Carlo simulation using the nonlinear model can be considered as a methodology for computing the statistics of the states, given the statistics of the inputs and the initial conditions. It is known that the statistics of the output in terms of expected values and the covariance matrix can be computed algebraically [9] if the system dynamics is linear, and if the inputs have Gaussian distributions.

Since the linear model developed in this section approximates the trajectory dynamics of the store, it can be used to obtain approximate results of the Monte-Carlo simulation, without explicitly conducting large number of numerical simulations. This semi-analytical computation of the input-output statistical characteristics can be used as an initial prediction of the statistics of the outputs, and as the basis for verifying the quality of the Monte-Carlo simulations.

Equation (9) showed that the inputs and the initial conditions are related to the states through a linear relationship:

$$\delta(k) = Q_1(k)\delta(0) + Q_2(k)U \quad (27)$$

In order to find the covariance of the states in terms of the covariance of the inputs and initial conditions, first compute the quadratic matrix

$$\delta(k)\delta^T(k) = \{Q_1(k)\delta(0) + Q_2(k)U\}\{Q_1(k)\delta(0) + Q_2(k)U\}^T \quad (28)$$

Expanding,

$$\begin{aligned} \delta(k)\delta^T(k) &= Q_1(k)\delta(0)\delta^T(0)Q_1^T(k) + Q_2(k)U\delta(0)^T(0)Q_1^T(k) \\ &\quad + Q_1(k)\delta(0)U^T(0)Q_2^T(k) + Q_2(k)UU^TQ_2^T(k) \end{aligned} \quad (29)$$

Taking expectations on both sides, and noting that the perturbation in the inputs U and the perturbation in the initial conditions $\delta(0)$ are uncorrelated,

$$E(\delta(k)\delta^T(k)) = Q_1(k)E(\delta(0)\delta^T(0))Q_1^T(k) + Q_2(k)E(UU^T)Q_2^T(k) \quad (30)$$

Thus, if the covariance matrix of the input perturbations $E(UU^T)$ and the covariance matrix of the initial condition perturbations $E(\delta(0)\delta^T(0))$ are known, the covariance matrix of the states $E(\delta(k)\delta^T(k))$ at a sample instant k can be determined using the above relationship. Note that these computations can be carried out for any set of output variables, as long as the linearized equations are available. Shown in Figure 3 is the covariance prediction of the vertical velocity.

Subject to Air-Force Approval for Public Release

Figure 3. Covariance Prediction of \dot{z}

III. Automatic Clustering of Trajectories

Automatic clustering of trajectories attempts to group the Monte-Carlos simulation trajectories into groups of similar looking trajectories. Shown in Figure 4 is a flow diagram of the automatic clustering implementation. Principal Component Analysis (PCA) followed by K-Means clustering is used in this composite formulation.

PCA is conducted on the trajectory time histories to reduce their high dimensionality. Length of time histories could vary from 50 – 500 depending on the simulation step size and the final time. Those principal components that account for 95% of the “energy” associated with all the singular values are included in this work. Actual trajectories can be compared with PCA estimated trajectories to estimate the loss of accuracy due to the truncation of the principal components. The user can increase the number of principal components if the data loss is not acceptable.

Lower dimensional data resulting from PCA is clustered using the *k-means* algorithm. Each trajectory is assigned to a cluster based on its proximity to the centroid of the cluster. The iterative process involves assignment of trajectories based on proximity to centroid, and re-computation of centroids based on the assignment. It terminates when no further change in centroid location and trajectory assignment are observed. Clustered trajectories can be visually evaluated to determine the merit of cluster assignment. Choice of too few numbers of clusters results in trajectory assignment not consistent with the majority in the cluster. On the other hand choice of too many clusters results in clusters that are almost identical to each other.

Subject to Air-Force Approval for Public Release

Figure 4. Flow Diagram of Automatic Clustering

E. PCA

Shown in Figure 5 is the scatter plot of a two dimensional data sample. Principal axes computed using singular value decomposition [6] are shown in red color. Principal axes for a full rank square matrix can be the eigen vectors. It can be inferred from the figure that variation of data along one of the principal axis is much larger than the other. The data points can be projected on to this axis with little loss of accuracy and 50% reduction in the size of the data. The reduction is even more significant for higher dimensional systems.

Subject to Air-Force Approval for Public Release

Figure 5. Principal Component Analysis

As a first step time histories of the chosen variables are stacked in the form of columns of a matrix. Rows represent the time instants and columns represent the run numbers.

$$X = \begin{pmatrix} x_1(0) & x_2(0) & \dots & x_n(0) \\ x_1(\delta t) & x_2(\delta t) & \dots & x_n(\delta t) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ x_1(t_f) & x_2(t_f) & \dots & x_n(t_f) \end{pmatrix} \quad (31)$$

The second step involves the normalizing the data by subtraction of mean from individual rows of the data matrix.

$$X_n = \begin{pmatrix} \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & \sum_{k=1}^m x_{kj} & \cdot \\ \cdot & x_{ij} - \frac{\sum_{k=1}^m x_{kj}}{m} & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \end{pmatrix} \quad (32)$$

Next step involves computing the singular value decomposition of the normalized data matrix:

$$X_n = USV^T \quad (33)$$

Singular values arranged in descending order:

$$[\sigma_1 \ \sigma_2 \ \dots \ \sigma_n] \quad (34)$$

Selection criteria for number p of principal components based on 95% energy associated with singular values is given by:

$$1 - \frac{\sum_{i=1}^p \sigma_i}{\sum_{i=1}^n \sigma_i} \leq 1 - 0.95 = 0.05 \quad (35)$$

Transformation matrix to lower dimensional space is given by:

$$T = V(:,1:p)^T \quad (36)$$

Transformation to lower dimensional space is obtained as:

$$X_L = XT \quad (37)$$

PCA estimated trajectories are computed as:

$$X_{pca} = XTT^T \quad (38)$$

Accuracy loss due to truncation of principal components is given by:

$$e = |X - X_{pca}| \quad (39)$$

F. K-Means

Shown in Figure 6 is the flow diagram of the K-Means algorithm. The algorithm uses as input the lower dimensional data defined in Eq. (37). Euclidean norm between vectors is used as the distance metric between the runs and the centroids.

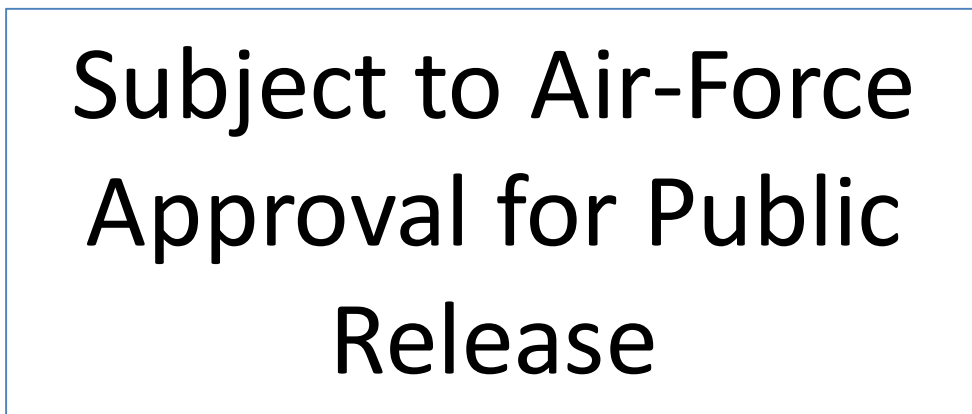


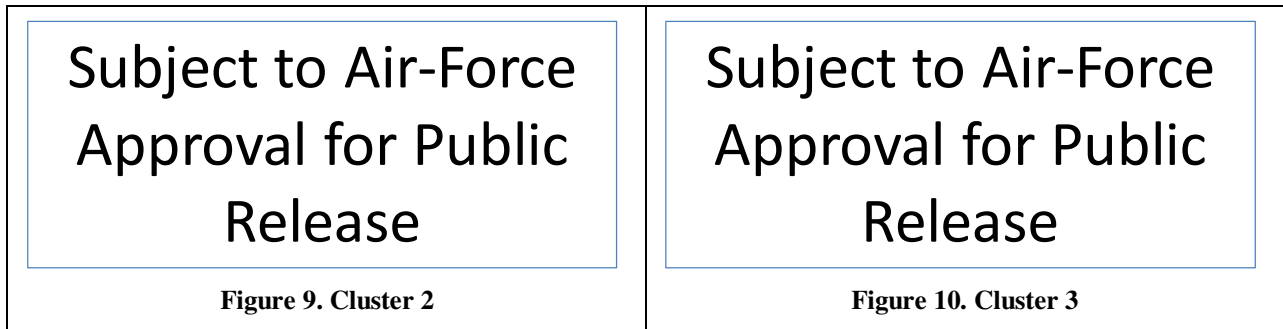
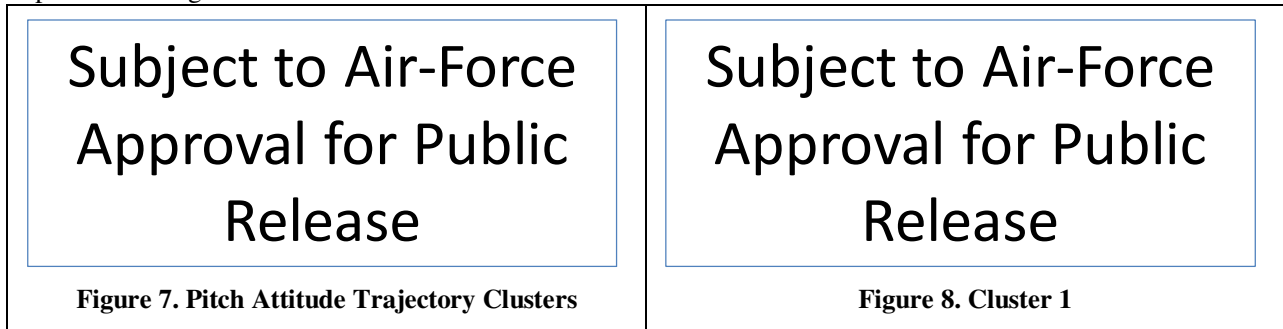
Figure 6. Flow Diagram of k-Means Algorithm

G. Representative Trajectories

Trajectory closest to the cluster centroid is selected as the representative trajectory. Euclidean norm is used as the distance metric between two trajectories. The representative trajectory unlike the centroid is a real trajectory with a known input. The trajectory and its input can be used to characterize the cluster and the inputs that lead to the cluster. Representative trajectories are very useful in characterizing large number of Monte-Carlo trajectories using a single representative trajectory. Input corresponding to the representative trajectory can be considered as the representative input for the cluster.

H. Results

Shown in Figure 7-Figure 10 are the clustering results of pitch attitude trajectories. Note the y-axis of these plots is non-dimensionalized. Cluster 1 shown in Figure 8 contains those trajectories where the store pitch attitude assumes sustained positive values. Cluster 2 shown in Figure 9 contains trajectories where the pitch attitude mostly remains negative. Slightly different from cluster 2 is cluster 3 where the trajectories exhibit a late tendency to pitch up as seen in Figure 10.



Representative trajectories for the above three clusters are shown in Figure 11. These three trajectories can be considered as qualitative representatives of the 1000 Monte-Carlo trajectories.

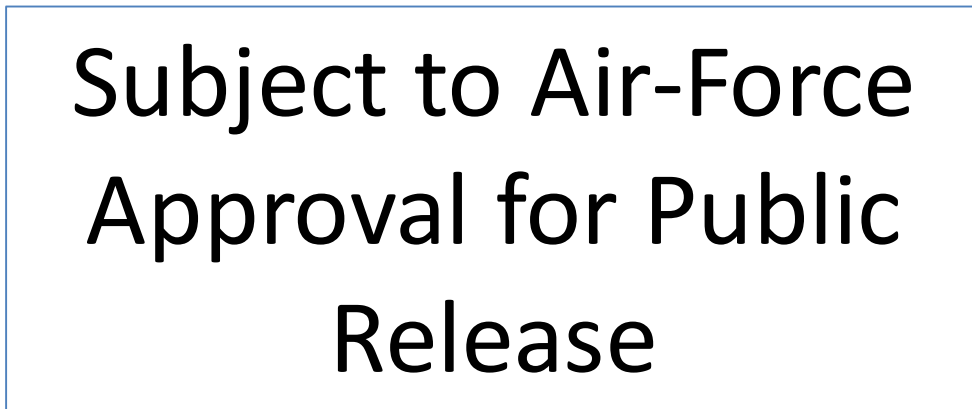


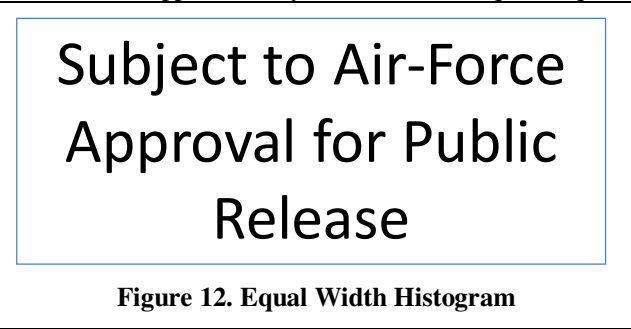
Figure 11. Plot of Representative Trajectories

IV. Probability Binning

Probability binning [11] methodology is presented in this section to evaluate the sensitivity of input parameters to a prescribed failure criterion. The approach fundamentally involves comparison of the population distribution of the parameter with the distribution of those samples that match the prescribed failure criterion. The population distribution function is first represented by an equal frequency histogram. This is done by using variable width bins that accommodate the same frequency in all the bins. Failed samples are then binned in the unequal width equal frequency histogram. A difference metric between the two distributions is then computed which serves as a measure of the sensitivity of the particular parameter to the prescribed failure criterion. Parameters whose failed sample distribution differs significantly from the population distribution are theorized to be more sensitive to the failure criterion. On the contrary parameters whose failed sample distribution is very similar to the population distribution are considered insignificant to the chosen mode of failure.

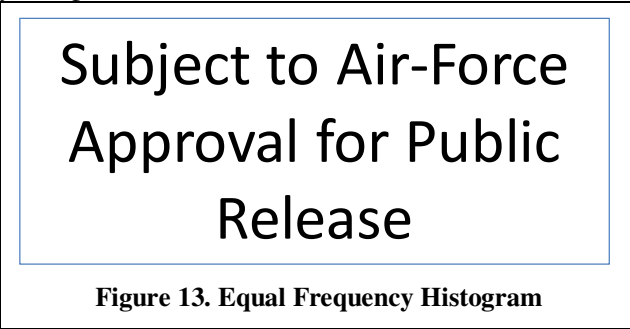
I. Equal Frequency Histogram

Standard binning algorithms use bins of equal width for ease of computation. However, it is not necessary for the bins to be of equal width. Chi-square statistic computed with equal width bins is weighed towards the distribution containing more events making it less sensitive to outlier samples. Choosing the bins to adapt to the location of the data is known as “adaptive-binning”. Probability binning is an “adaptive-binning” strategy suggested by Roederer et.al [11]. Instead of choosing bins of equal width the reference distribution is divided such that all bins have the same number of samples. Therefore, randomly selected sample has an equal probability of falling into any of the bins. The resulting bins are of unequal width. Shown in Figure 12 and Figure 13 are the equal width and equal frequency histograms of a particular store parameter. The total number of runs in the population is 1000 therefore each bin has approximately 100 runs in the equal frequency histogram.



Subject to Air-Force
Approval for Public
Release

Figure 12. Equal Width Histogram




Subject to Air-Force
Approval for Public
Release

Figure 13. Equal Frequency Histogram

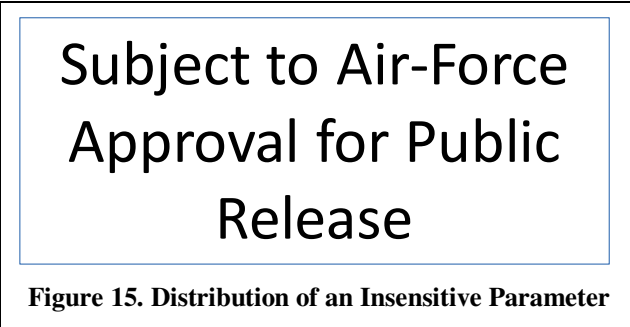
J. Sensitivity Metric Computation

To compute the sensitivity metric it is first necessary to obtain the samples of the parameters that correspond to a given failure criterion. Typically this is a small number. The failed samples are then binned in the equal frequency histogram of the entire population. Shown in Figure 14 and Figure 15 are the failed sample distributions (red) against the backdrop of the population distribution. The number of failed runs in this example is only 27 therefore the small frequencies corresponding to the red bins.



Subject to Air-Force
Approval for Public
Release

Figure 14. Distribution of a Sensitive Parameter



Subject to Air-Force
Approval for Public
Release

Figure 15. Distribution of an Insensitive Parameter

A scalar quantity similar to the chi-squared statistic can be computed to compare the two distributions.

$$\chi^2 = \sum_{i=1}^n \frac{\left(\frac{O_i}{n_s} - \frac{E_i}{n_e} \right)^2}{\left(\frac{O_i}{n_s} + \frac{E_i}{n_e} \right)} \quad (40)$$

Where,

- O_i is the frequency of the failed samples in the i^{th} bin
- E_i is the frequency of the population samples in the i^{th} bin
- n_s is the total number of failed samples
- n_e is the total number of population samples in the reference distribution.

This metric a single scalar quantifies the difference between the two distributions. A large value indicates that the sample distribution is significantly different from the population distribution. On the other hand a small value indicates that the sample distribution is very similar to the population distribution. The comparison can be done for each input parameter one at a time or a combination of inputs. Inputs or combinations of inputs with large probability binning metric are theorized to have more impact on the chosen failure criterion than those with small probability binning metrics. The probability binning metric can be computed for different inputs and combination of inputs. Shown in Figure 16 is a pareto plot of the input components arranged in a descending order of the sensitivity metric. The failure criteria used in generating this plot was number of vertical velocity sign flips $\Rightarrow 1$.

Subject to Air-Force Approval for Public Release

Figure 16. Pareto Plot of the Probability Binning Sensitivity Metric

V. Fail-Safe Clustering

The purpose of fail-safe clustering is to compute the acceptable ranges of the parameter components to avoid a given mode of failure. Particular approach adopted in this research involves construction of a normalized hypercube in the parameter space that includes as many non-failed parameter samples as possible while completely avoiding the failed parameter samples.

K. Fail-Safe Cluster Synthesis

First step in the synthesis of fail-safe cluster is normalizing of input parameter with individual components that have different dynamic ranges. The normalization procedure adopted is as follows:

$$u_n = \frac{u - u_{\min}}{|u_{\text{nom}} - u_{\min}|} \text{ if } u < u_{\text{nom}} \quad (41)$$

$$u_n = \frac{u - u_{\max}}{|u_{\text{nom}} - u_{\max}|} \text{ if } u > u_{\text{nom}} \quad (42)$$

Once normalized all these components lie between -1 to +1. Shown in Figure 17 is a schematic of the fail-safe cluster synthesis procedure using only two input components. The approach involves constructing iterative hypercubes in the input space that do not enclose any failed inputs.

Subject to Air-Force Approval for Public Release

Figure 17. 2D Example of Fail-Safe Cluster Synthesis

A small region close to the nominal input parameter defined by

$H_1 : u_{j-l} < u_j < u_{j+u} \quad j = 1 \dots n$ is initially selected, where u_j is the j^{th} input parameter and n is dimension of the input space. The region is chosen such that it does not enclose any failed parameter samples. It is assumed that the nominal parameter vector is a non-failure. Furthermore, it is perfectly reasonable to assume that there exists a small region around this nominal vector where no failures are found.

The chosen region H_1 describes a hyper cube in the n dimensional input space and is the first cut estimate of the fail safe cluster.

The center of H_1 corresponds to the normalized nominal input which is the zero vector.

The parameters u_{j-l} and u_{j+u} determine the size of the hypercube along each input component.

The initial values of u_{j-l} and u_{j+u} are initialized to -0.001 and 0.001 respectively.

The two parameters are then incrementally updated as $u_{j-l} = u_{j-l} - 0.001$ and $u_{j+u} = u_{j+u} + 0.001$. The region $H_2 : u_{j-l} < u_j < u_{j+u}$ describes a larger hypercube that encloses its predecessor H_1 .

A search is then conducted over the database to determine if there are any failures corresponding to the inputs that lie in H_2 . If there are no failures then H_2 is set as the new estimate of the fail safe region.

The incrementing procedure and ensuing database search are continued till a failure is found within the hypercube.

Once a failure is found some dimensions of the fail safe region are frozen and not updated further. Assuming the first failure occurs after $(k+1)$ iteration; the failed input lies in H_{k+1} but not in H_k . Those input components whose value lie in the annular region of H_k and $H_{(k+1)}$ are frozen.

The hypercube is updated along the remaining directions until a new failure is encountered at which time the same process of freezing those input dimensions is followed.

The process terminates when no further incrementing is possible without encountering a failure or without violating the ± 1 limits of the normalized input components.

L. Utility and Limitations

The following are some of the useful features of the fail safe cluster:-

- Based on the data that has been used to construct this region it can be said that failure is not expected while inputs lie inside this region.
- The hypercube characterization helps in data understanding as each input can be isolated. It also provides insight such as which input components are critical based on the width of the cube along those dimensions. Large width implies that input component has smaller influence on the chosen failure criteria; in contrast a smaller width implies higher influence.
- The hypercube also indicates the direction on either side of the nominal in which the perturbations are more harmful.

Limitations of the fail safe cluster:-

- The fail safe region generated by the procedure described in this section is only an estimate of the exact region.
- It does not necessarily include all the non-failed samples.

Acknowledgments

The work presented in this paper was supported under Air Force contract number FA9200-05-C-0185.

References

- ¹Adriaans, P. and Zantinge, D., *Data Mining*, Addison-Wesley, Menlo Park, CA, 1996.
- ²Groth, R., *Data Mining*, Prentice-Hall, Upper Saddle River, NJ, 2000.
- ³Witten, I. H. and Frank, E., *Data Mining*, Morgan Kaufman, San Francisco, CA, 2000.
- ⁴Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, MA, 1996.
- ⁵Piatetsky-Shapiro, G. and Frawley, W., *Knowledge Discovery in Databases*, MIT Press, Cambridge, MA, 1991.
- ⁶Seidman, C., *Data Mining with Microsoft® SQL Server 2000*, Microsoft, Redmond, WA, 2001.
- ⁷Pyle, D., *Business Modeling and Data Mining*, Morgan Kaufman, San Francisco, CA, 2003.
- ⁸Brogan, W. L., *Modern Control Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
- ⁹Gelb, A. (Editor), *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1989.
- ¹⁰Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, New York, NY, 2001.
- ¹¹Roederer, M., Tresiter, A., Moore, W., and Herzenberg, L. A., "Probability Binning Comparison: A Metric for Quantifying Univariate Distribution Differences," *Cytometry*, Vol. 45, 2001, pp. 37-46.