Fragment Capture Simulation for Missile Blast Test Optimization

Michael J. Garee, Raymond R. Hill, and Brent Russell
Department of Operational Sciences
Air Force Institute of Technology, WPAFB, OH 45433, USA

Abstract

The assessment of aircraft survivability against explosive munitions is an expensive undertaking. Test articles for both aircraft and weapon are scarce due to their high costs, leading to a limited supply of test data. Unfortunately, the short supply of test data does not prevent the development of newer, hopefully more effective weaponry or protection measures. Therefore, test organizations need to explore methods for increasing the quality of test results while looking for ways to decrease the associated costs. This research focuses on Man-Portable Air-Defense Systems (MANPADS) as the weapon of choice and live-fire arena testing as the experimental data source. A simulation infrastructure is built and used to examine how to optimize the arena configuration to maximize the test information obtained.

Keywords
Survivability, simulation, fragment pattern, test and evaluation.

1. Introduction

The assessment of aircraft survivability against explosive munitions is an expensive but necessary undertaking within the Department of Defense. Test articles for both aircraft and weapon are scarce due to their high costs, leading to a limited supply of test data. Test data are needed for weapons assessments and survivability analyses. The short supply of test data does not prevent the development of newer, hopefully more effective weaponry or protection measures. Therefore, test organizations must explore methods for increasing the quality of test results while simultaneously looking for ways to decrease the associated costs of conducting those tests. This research focuses on the MANPADS as the weapon of choice and live-fire arena testing as the experimental data source.

MANPADS are a class of shoulder-fired anti-aircraft missiles measuring one to two meters in length and weighing 13 to 25 kilograms [2]. Examples of MANPADS include the SA-7b and the SA-14. A MANPADS missile is guided, unlike the superficially similar but unguided rocket-propelled grenade (RPG), and is best-suited for use against ground targets or low-flying aircraft. The most common guidance system for MANPADS is infrared seekers, though laser-guided and command line-of-sight systems also exist [7]. MANPADS are designed to destroy helicopters and small aircraft and can be operated by an individual or a small team [8]. The ease with which these weapons are transported and hidden facilitates their proliferation and makes them attractive weapons for criminal and terrorist activities.

The Department of Defense and the Department of Homeland Security outline four methods of protection for U.S. aircraft against MANPADS [5]:

1. “Denial of weapons to potential threat organizations and individuals
2. “Denial of opportunity to fire the weapon at an aircraft
3. “Prevention of impact of the missile on the aircraft
4. “Withstanding MANPADS impacts and landing the aircraft without system loss or casualties.”

This study supports the fourth point by exploring data collection and analysis techniques useful for aircraft survivability assessments.

“What is the potential for an aircraft kill given a hit?” is the key question in aircraft survivability analysis and is supported by the answers to other more fundamental questions [6]. These questions cover survivability aspects such as the amount of blast damage sustained by an aircraft and the penetration depth of missile fragments. These questions
are answered using Test & Evaluation (T&E) methods. Live-fire testing using actual aircraft and weapons produces the most realistic results, but comes at the highest resource cost, even when representative test articles, such as helicopter tail boom skeletons instead of production-quality helicopters, are used [3].

Simulation software can offer cost savings to the live-fire test establishment, particularly the MANPADS testing establishment. We design, prototype, and investigate such a simulation for the assessment of MANPADS testing. MANPADS fragment mass and velocity estimate data can be captured in a physical fragment test arena. A fragment capture arena is a full or partial enclosure around an explosive weapon, with walls made of catch bundles used for capturing fragments and make-screens used for calculating estimated fragment velocities. The capture arena configuration and fragment capture patterns can be modeled within a simulation from which quality assessments of selected arena configurations can be developed and ideally optimized with respect to test efficiency and effectiveness.

2. Problem statement
Survivability simulation authors have made specific requests for data improvements from physical MANPADS testing. In particular, they want missile fragment masses mapped to their velocities with a higher degree of fidelity than in the past. Ideally, the fragment capture arena would completely surround the weapon, capturing every fragment and allowing a complete three-dimensional fragment distribution model. Physical and financial constraints make such a complete arena infeasible. Smaller arenas are more affordable in both material and labor but yield lower data quality. There is currently no way to assess objectively and quantitatively the feasible "middle" configurations.

These middle configurations include numerous ways to place velocity-measuring make-screens on the arena walls. Smaller screens capture fewer fragments on average, giving better estimates of individual fragment velocities. Each make-screen has a material cost and requires a data channel; both money and data channels are limited resources, so an upper bound exists on the number of make-screens available for a test. Predicting fragment distributions that might be seen on the arena walls allows make-screens to be sized and placed more efficiently – smaller screens can be placed in areas of higher fragment density and larger screens in areas of lower fragment density.

The scenario under study involves a MANPADS detonated within a fragment capture arena used to measure fragment masses and velocities. Arena design constraints capture only a portion of the weapon’s fragments. Data extrapolation is required to construct a complete distribution. This research develops methods to perform simulated arena tests and to enhance data quality from physical tests by suggesting arena configuration improvements.

3. Literature Review
The Joint Aircraft Survivability Program Office (JASPO) and the 96th Test Group are responsible for much of the published information on MANPADS testing in support of aircraft survivability. Their work includes the design of tactics, techniques, and procedures (TTPs) for protecting against MANPADS attacks near airfields [4]; assessment of large aircraft control surface and engine vulnerability [5, 6]; and use of MANPADS miss distances to evaluate low-altitude aircraft survivability [1].

Figure 1 depicts a static fragment capture arena [6]. In the static configuration, the weapon is centered in an arena consisting of fragment capture bundles, make-screens for recording fragment velocities, and other data collection equipment. The weapon can also be launched into the arena at a controlled velocity and set to detonate at the exact center of the arena to capture dynamic test data. The shape of the arena is determined by the testers but is typically a square or half-square. Capture bundles are thick layers of fibrous material used for arresting weapon fragments. After the test, researchers record the position where each fragment entered the capture bundle and remove layers from the bundle until the fragment itself is located and its mass is recorded. Collecting fragments in this manner is a very slow process, causing significant delay between conducting the test and analyzing the results.

A make-screen is a device used to record the timing of a weapon fragment impact. In its simplest form, it consists of an insulating material sandwiched between two charged, conductive surfaces (e.g. a thin piece of cardboard placed between two sheets of aluminum foil) that are attached to a data channel. These can be made to fit any dimension as required by the test; Czarnecki et al. [6] report using screens one to four square-feet in surface area. When a fragment penetrates the make-screen, it creates an electrical connection between the front and back surfaces. An attached data channel registers the connection and records the time of impact from which a time-of-flight is calculated and a fragment velocity is estimated.

There are several drawbacks with this design. Important for this project is that when a make-screen is struck more than once, multiple impact times are recorded, but there is no way to determine which impact time is associated with a
particular fragment. To mitigate this, screen sizes can be selected to keep the expected number of fragments per screen low, and the average velocity of all fragments incident on a screen can be used as a measurement. The challenge is how to determine the best mix of screen sizes and locations. To date, this mix is based on expert opinion.

4. Arena Configuration Model
A computational model of a fragment capture arena was designed and realized in MATLAB. The model mimics the MANPADS device detonation with subsequent fragment impact pattern. The fragments strike the walls of the capture arena that contain representations of various sized make-screens.

The key elements of the model are implemented as class objects. The primary classes used in the model are:

**Arena** Container for all aspects of the simulation configuration. The Arena object specifies a test weapon, a collection of make-screens, and methods for constructing the arena from external input. Methods for transforming data between Cartesian and spherical coordinate systems are also defined here.

**Screen** Representation of an individual make-screen, containing the screen’s position within the arena and a list of fragments that impact the screen after the simulated weapon is detonated.

**Weapon** Model for the MANPADS missile being tested. This class specifies the fragmentary mass of the weapon, the distributions from which fragment masses and velocities are sampled, methods for generating fragments at the time of detonation, and a list of the fragments themselves.

**Frag** Model for an individual weapon fragment that specifies its weight, the impact position, and its impact velocity.

Positions within the arena are principally specified in Cartesian coordinates using the arena coordinate system. The weapon centroid is the origin of this system; the y-axis follows the long axis of the weapon (nose to tail); the x-axis is perpendicular to the y-axis and parallel to the arena floor; and the z-axis is the arena’s vertical axis. Positions in Cartesian coordinates \((x, y, z)\) are converted to spherical coordinates \((r, \theta, \phi)\), where radius \(r\) is the straight-line distance from the origin, \(\theta\) is the azimuth angle measured counter-clockwise from the positive x-axis in the x-y plane (i.e. from the right side of the missile) as in standard polar coordinates, and \(\phi\) is the elevation angle measured from the x-y plane. The MATLAB functions \texttt{cart2sph} and \texttt{sph2cart} are used for converting between Cartesian and spherical coordinates, internally using the following equations:

\[
\begin{align*}
  r &= \sqrt{x^2 + y^2 + z^2} \\
  \theta &= \arctan\left(\frac{y}{x}\right) \\
  \phi &= \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right)
\end{align*}
\]
Weapon fragments in the MANPADS model have three key attributes: mass, position on the arena walls, and velocity at time of impact. These are set as properties within each Frag object.

The Weapon object defines the properties numFrags for the number of fragments produced by the weapon and fragMass as the total mass of those fragments. \( N \), the number of fragments of mass \( m \) or greater, is a uniformly-distributed random number for producing random variates in the range \([0, \text{numFrags}]\). As each Frag object is generated, a uniform random draw for \( N \in [0, \text{numFrags}] \) is used to assign the fragment mass \( m \). If the user has fragment mass data from physical testing, an empirical distribution can be used.

Fragment trajectories are sampled from user-specified distributions, with all fragments originating from the weapon’s center of mass (the arena origin) and provided some initial velocity. Trajectories are designated using spherical coordinates and stored in the Frag properties theta and phi. The properties \( x \), \( y \), and \( z \) are used to record the fragment’s impact position on a make-screen, calculated later in the simulation.

The Arena object contains a set of Screen objects. Each make-screen in the arena is represented with a Screen class instance, which has properties for the screen’s ID number, the positions of its upper-left and lower-right corners (in the arena coordinate system), and a list of fragments that impact the make-screen (populated after the Weapon object is detonated). Arena configuration data can either be provided by the user as an input into the model or created by the simulation as an output.

Impact data for each fragment has three components:

- impact position in arena coordinates,
- identification of the make-screen struck by the fragment, and
- velocity at time of impact (initial velocity if no drag modeled).

A fragment strikes a make-screen at a point given by the intersection of the fragment’s trajectory vector (defined by \( \theta \) and \( \phi \)) and the plane containing the make-screen. The plane of the make-screen is defined by any three points on the screen. The model uses the upper-left, upper-right, and lower-right corners of the screen, denoting them as \( \vec{x}_1 \), \( \vec{x}_2 \), and \( \vec{x}_3 \), respectively. If the fragment impacts the plane within the boundaries of the make-screen, then the impact position and screen of the fragment are updated and the fragment is added to that screen’s list captured fragments.

The baseline model has two modes of operation. Mode A scatters weapon fragments on defined walls of an arena, with no concern given to make-screens. The user provides the model with characteristics of the weapon under test, positions of the arena walls (not make-screens), and environmental factors. The model generates fragment positions on the arena walls, the corresponding impact velocities, and graphical representations of the arena and fragments. This mode is useful for generating fragment distribution data for external analysis.

Mode B is used for performing fragment impact analysis on a given make-screen configuration. With the additional make-screen information, the model places fragments on the make-screens and reports the velocity and position of each fragment impact with respect to the affected make-screen. Once all fragment impacts are computed, the make-screen velocities (the mean velocity of all fragments on a particular make-screen) are available. This mode is well-suited for postmortem evaluation of physical test data.

Figure 2 shows a graphical output from the model. The figure depicts the outline of the modeled make-screens along with the fragment distribution. The algorithms employed map the fragment location to the corresponding screen object membership.

5. Results

High-quality test data describes reality with a high degree of accuracy. Data from fragment capture tests includes fragment impact positions and make-screen velocity measurements. There is little error in the positions of fragment impacts, apart from human measurement error. However, there is great potential for error in measuring fragment velocities. The most notable source of error is in the way make-screens function – make-screens collect fragment impact times, compute times of flight, average those times, and use the screen’s distance from the weapon to provide a
Make screens and fragments in arena coordinate system

Figure 2: Sample arena populated with fragments.
mean fragment velocity, which is then attributed to each fragment on the make-screen. The mean velocity for a large group of fragments is less representative of individual fragment behavior than the mean velocity for a smaller group. Velocity data quality is therefore maximized by using enough make-screens so that each screen captures a single fragment at most. This gives the most accurate estimate of fragment velocity possible but is completely impractical. Conversely, the lowest quality data is obtained by using one make-screen to cover an entire wall. Note that the actual measure of interest is the error between the make-screen velocity (estimated) and the corresponding fragment velocities (often unknown), not simply the number of make-screens.

Predicting the quality of data that can be produced by an arena configuration allows decision makers to choose between competing designs. The high resource costs of each physical test encourages careful arena design to maximize the utility of test data. The development of a data quality metric supports these goals.

A measure of data quality should satisfy the following criteria:

1. encourages arena configurations with low mean numbers of fragments per make-screen, achievable by adding screens or repositioning screens based on expected fragment density;
2. encourages arena configurations with low error between a make-screen’s measured velocity and the corresponding individual fragment velocities;
3. does not penalize configurations with make-screens that are not hit by any fragments – while wasteful of resources, they do not affect data quality;
4. does not reward configurations for adding make-screens without decreasing measurement error (e.g. using two screens to capture two same-speed fragments instead of one); and
5. allows comparison with other configurations.

Criteria 1 and 2 can be combined, since fewer fragments per make-screen naturally leads to more representative make-screen velocities, and more representative make-screen velocities correspond to reduced error. They are separated here for clarity.

Consider the arena configuration problem as follows. A configuration consists of \( S = \{1, \ldots, n^s\} \) make-screens, each of which captures a set of fragments \( T^s, s \in S \). Each screen captures \( n^s = |T^s| \) fragments, and a total of \( N = \sum_{s \in S} n^s \) fragments are recorded with \( T = \bigcup_{s \in S} T^s \). We require that \( T^i \cap T^j = \emptyset \) \( \forall i, j \in S, i \neq j \). Every fragment has some true velocity \( v_f, f \in T \) and an estimated velocity \( \hat{v}_f = v_s, f \in T^s, s \in S \) where \( v_s \) is the average velocity for all fragments impacting screen \( s \). The mean absolute velocity error for screen \( s \) is

\[
MAVE^s = \frac{1}{n^s} \sum_{f=1}^{n^s} |v_f - v_s|.
\]  

(7)

\( MAVE \) achieves a maximum value when a screen captures only the fastest and slowest of all fragments captured by the arena. \( MAVE \) achieves a minimum value of zero when a screen captures a single fragment; this aligns \( MAVE \) with criteria 1 and 2.

The actual magnitude of \( MAVE \) depends on the number of fragments captured by \( s \). It is normalized by dividing by the worst-case value \( MAVE^* \) (the single-screen case); this allows comparisons among screens with different numbers of fragments, since the ratio is in the unit interval. The per-screen quality score is thus defined as

\[
Q^s = 1 - \frac{MAVE^s}{MAVE^*}.
\]  

(8)

The complement is taken so that better performance yields a higher score. It is possible for a make-screen to be hit by no fragments; such screens are deemed “inactive”, while all screens with fragments are “active.” \( Q^s \) is not computed for inactive screens, in support of criterion 3 above.

The quality scores for each active make-screen contribute to the overall arena score. The arena data quality score for arena configuration \( a \) is a measure of how well the arena configuration records the true velocities of the captured fragments. It is defined as

\[
Q^a = \frac{1}{|S|} \sum_{s \in S_{active}} Q^s,
\]  

(9)

where \( S_{active} \subset S \) includes only active make-screens. Dividing the sum by the number of screens weights each screen’s contribution to \( Q^a \) according to the number of active screens. Adding screens to a configuration causes each screen to
Table 1: Summary of data quality scores.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quality score</th>
<th># of screens hit</th>
<th># of fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 screen</td>
<td>0</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>4 screens</td>
<td>0.0332</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>16 screens</td>
<td>0.2481</td>
<td>11</td>
<td>50</td>
</tr>
<tr>
<td>64 screens</td>
<td>0.5848</td>
<td>23</td>
<td>50</td>
</tr>
<tr>
<td>16 screens (custom)</td>
<td>0.2970</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>16 screens (perfect)</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Contribute less to the arena score (provided they are all active), so the additional screens must improve the actual data quality in order to improve the arena score. This aspect of $Q^a$ supports criterion 4. Since $Q^a \in [0, 1]$, the maximum sum of screen scores is $|S|$; therefore $Q^a$ is also in the unit interval. An arena data quality score of unity indicates all fragment velocities are measured correctly, and a score of zero means all fragment velocities are estimated as well as they would be in a single-screen arena. Bounding the quality score enables natural comparisons between arenas (criterion 5).

An experiment was designed to test the quality score. In the following runs, the same fragment pattern of fifty fragments is used. The arenas have a single wall; the number of make-screens vary.

Figure 3 shows arena configurations with one, four, sixteen, and sixty-four evenly-tiled make-screens. Qualitatively, the data quality improves with increasing screen count since the mean fragment count per screen decreases. $Q^a$ is calculated for each layout and annotated below the plots. The calculated values support the qualitative relationship. In the sixteen and sixty-four screen arenas, several make-screens capture only one fragment. These screens report the true velocity of the fragment (zero error) and make the greatest contribution to the overall arena scores.

To demonstrate that data quality is not a direct function of make-screen quantity, a new sixteen screen configuration is created such that each screen captures two to four fragments (Figure 4). Arena data quality increases about two percent. A final sixteen make-screen arena is constructed (Figure 5) to provide an example of an arena with the maximum data quality possible. All active screens capture exactly one fragment, so each make-screen velocity corresponds exactly to its fragment’s velocity. This arena only has eleven weapon fragments, so five screens are inactive. The perfect data quality indicates that the inactive screens do not detract from the quality score. The scores from the experiments in this section are summarized in Table 1.

These results indicate that the quality metric provides a means to compare arena configurations for a given fragment pattern. These results are focused purely on solution quality – which configuration yields the most accurate fragment velocity estimates. The quality metric does not do anything with respect to arena cost or complexity considerations.

6. Conclusions
Survivability analyses are critical for defense systems, and the MANPADS threat is quite real. Live testing is used to generate data necessary for survivability analytical codes to provide effective representations of the threat and the countermeasures in place for that threat. Pressures on testing are dictating the use of efficient and effective test programs. For the MANPADS environment, this means a better way to design and assess fragment capture arena configurations.

This work provides some background on the research problem, relates the development of the initial prototype analytical tool for assessing fragment capture arena configurations, and provides some initial work using that tool. Once refined and fielded, this tool can be used to assess configurations against known fragment patterns and to empirically search for configurations with robust performance over a range of foreseeable fragment patterns.

Disclaimer
The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.
Fifty fragments with a varying number of make-screens

One make-screen

\[ Q^a = 0.4111 \]

Sixteen make-screens

\[ Q^a = 0.5583 \]

Sixty-four make-screens

\[ Q^a = 0.7450 \]

Fifty fragments on a wall with a varying number of make-screens. Square, evenly-tiled make-screens are used for convenience. Several screens for the sixteen and sixty-four make-screen configuration capture only one fragment, so they report the true velocity for that fragment.
Fifty fragments on sixteen make−screens

Configuration 1 – Evenly tiled make−screens

Configuration 2 – Make−screens with custom placement

Figure 4: Fifty fragments with sixteen make−screens in two configurations. Configuration 1 has evenly tiled screens as seen in Figure 3. Configuration 2 has make−screens adjusted so each screen has between two and four fragments. The arena data quality scores show a two percent difference between the two configurations.

Example of perfect arena quality

Figure 5: Example of configuration with $Q_{arena} = 1$. Fragments are synthetically placed so that each make−screen has no more than one fragment. A number of screens have no fragments, which demonstrates that inactive screens do not count against the quality score.
References


