NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZATION OF AIMPOINTS FOR COORDINATE-SEEKING WEAPONS

by

Wei Sheng Eugene Lee

September 2015

Thesis Advisor: Morris Driels
Second Reader: Christopher Adams

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2015	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE OPTIMIZATION OF AIMPOINTS FOR COORDINATE-SEEKING WEAPONS			**5. FUNDING NUMBERS**
6. AUTHOR(S) Lee, Wei Sheng Eugene			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The objective of this thesis is to develop a program that makes use of three types of damage functions to optimize the weapon aimpoints of multiple coordinate-seeking weapons against a unitary target in order to achieve the highest probability of damage (PD). A MATLAB program is used as the coding tool for the development of this algorithm and the optimization process. The program works by first taking in the number of weapons used and arranging them in a fixed uniform spacing on a circle centered on the assumed target location. Then, the weapon characteristics such as the radius of the circle containing the weapon aimpoint, impact angle, dependent (aiming) and independent (ballistic) errors are taken into account, before utilizing each of the three damage functions representing the weapon.

A Monte-Carlo simulation method is used to calculate the PDs at incremental radii of weapon placements from the target. Since the damage functions differ in terms of fidelity (accuracy), a comparison in terms of optimal aimpoint radius for the highest PD is made for the results generated for all three damage functions. The simulated results demonstrated that the optimal aimpoint radii for the maximum PD are slightly different for each damage function. In addition, the maximum PD at the optimal aimpoint radius generated for each damage function is lowest for a damage function that has the greatest fidelity (accuracy), which is consistent with the calculated results for single weapons against unitary targets. Also as expected, generating a PD using a higher fidelity damage function takes a longer time than that of a lower fidelity damage function. As such, the user of this program has to take into account the accuracy requirements and time limitations before selecting the damage function to be used to generate the PD.

14. SUBJECT TERMS Carlton damage function, rectangular damage function, lethal area matrix, Monte-Carlo simulation, probability of damage, coordinate-seeking weapon, dependent error, independent error, aimpoint radius			**15. NUMBER OF PAGES** 113
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

OPTIMIZATION OF AIMPOINTS FOR COORDINATE-SEEKING WEAPONS

Wei Sheng Eugene Lee
Civilian, Singapore Technologies Engineering (Kinetics)
B.Eng. (Mechanical), Nanyang Technological University, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Approved by: Morris Driels
Thesis Advisor

Christopher Adams
Second Reader

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The objective of this thesis is to develop a program that makes use of three types of damage functions to optimize the weapon aimpoints of multiple coordinate-seeking weapons against a unitary target in order to achieve the highest probability of damage (PD). A MATLAB program is used as the coding tool for the development of this algorithm and the optimization process. The program works by first taking in the number of weapons used and arranging them in a fixed uniform spacing on a circle centered on the assumed target location. Then, the weapon characteristics such as the radius of the circle containing the weapon aimpoint, impact angle, dependent (aiming) and independent (ballistic) errors are taken into account, before utilizing each of the three damage functions representing the weapon.

A Monte-Carlo simulation method is used to calculate the PDs at incremental radii of weapon placements from the target. Since the damage functions differ in terms of fidelity (accuracy), a comparison in terms of optimal aimpoint radius for the highest PD is made for the results generated for all three damage functions. The simulated results demonstrated that the optimal aimpoint radii for the maximum PD are slightly different for each damage function. In addition, the maximum PD at the optimal aimpoint radius generated for each damage function is lowest for a damage function that has the greatest fidelity (accuracy), which is consistent with the calculated results for single weapons against unitary targets. Also as expected, generating a PD using a higher fidelity damage function takes a longer time than that of a lower fidelity damage function. As such, the user of this program has to take into account the accuracy requirements and time limitations before selecting the damage function to be used to generate the PD.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A_{ET}	Effective Target Area
A_L	Weapon Lethal Area
ASW	Air-to-Surface Weapon
BOC	Bombing-on-Coordinates
CDF	Carlton Damage Function
CEP	Circular Error Probable
DEP	Deflection Error Probable
DMPI	Desired Mean Point of Impact
DPI	Desired Point of Impact
FD	Fractional Damage
GDF	Gaussian Damage Function
GPS	Global Positioning System
JDAM	Joint Direct Attack Munition
JMEM	Joint Munitions Effectiveness Manual
JWS	JMEM Weaponeering System
LAM	Lethal Area Matrix
L_{ET}	Effective Target Length (Rectangular)
L'_{ET}	Effective Target Length
MAE	Mean Area of Effectiveness
MAE_F	Mean Area of Effectiveness (Fragmentation)
MC	Monte Carlo
MPI	Mean Point of Impact
NATO	North Atlantic Treaty Organization
PD1	Calculated Probability of Damage for Single Weapon
PD	Probability of Damage
PGM	Precision-Guided Munition
RCC	Rectangular Cookie Cutter
RDF	Rectangular Damage Function
REP	Range Error Probable

SSPD	Single Sortie Probability of Damage
TLE	Target Location Error
W_{ET}	Effective Target Width (Rectangular)
W'_{ET}	Effective Target Width
WR_d	Weapon Radius (Deflection)
WR_r	Weapon Radius (Range)

ACKNOWLEDGMENTS

First and foremost, I would like to thank my thesis advisor Professor Morris Driels for his time, effort and utmost patience in guiding me through this research phase, without which I would not have been able to complete this thesis.

I would also like extend my gratitude to my second reader Professor Christopher Adams for his advice during my thesis selection phase and for his encouragement during the research phase.

And to the girl who's always with me through all the nights while I worked on this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. RESEARCH OBJECTIVE

The main objective of this thesis is to develop a program for optimizing the aimpoints for a given target for multiple coordinate-seeking weapons in order to achieve the highest probability of damage to the target. The MATLAB program tool is used in this thesis as the main coding tool to achieve the above-mentioned objective.

Comparisons shall be made with the probability of damage (PD) generated from three different damage functions and the optimal radius with which the highest PDs are obtained for each damage function.

B. APPROACH

In this thesis, given a set of specified aimpoints, the PD can be calculated using damage functions in an iterative procedure known as Monte Carlo simulations, which is explained in a later chapter. With the generated PD across a range of aimpoints, the optimal aimpoints (with highest PD) can be determined.

C. SCOPE OF RESEARCH

The scope of research covers coordinate-seeking weapons only. Coordinate-seeking weapons (CSW) are warheads or bombs that are configured or designed to be maneuvered onto specified coordinates via control surfaces (fins) attached to the weapon. Examples of these weapons are the air-delivered Mk80s series of munitions (Mk81, Mk82, Mk83, and Mk84) shown in Figure 1.

Figure 1. Mk80 series general purpose bombs, from [1]

To provide control surfaces for maneuvers in an air-delivery scenario, the Joint Direct Attack Munition (JDAM) guidance kit is affixed to the Mk80 series of weapons as demonstrated in Figure 2

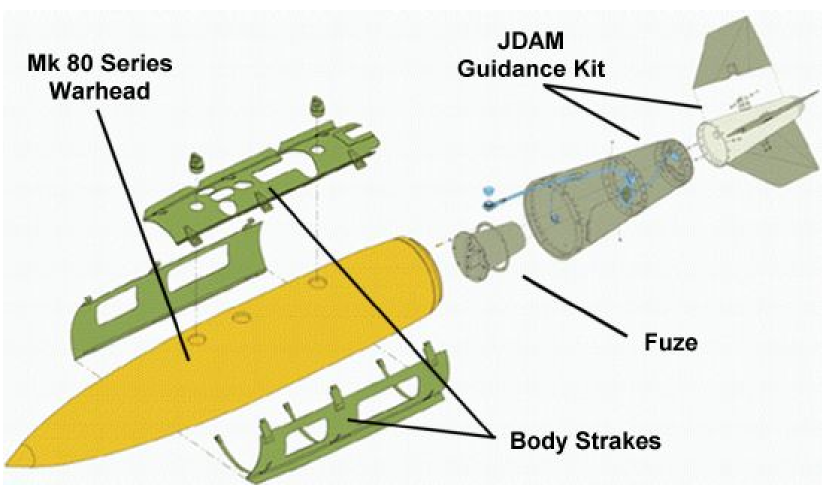

Figure 2. JDAM guidance kit for Mk80s series warheads, from [2]

Figure 3 illustrates the forward fin control surfaces on the M982 Excalibur round, which is another example of a coordinate-seeking weapon that is ground-launched (artillery).

Figure 3. M982 *Excalibur* artillery round, from [3]

THIS PAGE INTENTIONALLY LEFT BLANK

II. OVERVIEW OF WEAPONEERING CONCEPTS

This chapter discusses the relevant weaponeering concepts with which to understand the data analysis and discussions in the chapters that follow.

A. WEAPON TERMINAL CONDITION

1. Desired Point of Impact (DPI)

In coordinate-seeking weapons, a coordinate frame is represented two-dimensionally in the range and deflection directions. The desired point of impact (DPI) is the weapon aimpoint for a known target location, marked on the two-dimensional coordinate ground frame. Figure 4 illustrates the DPI (in red) on an enemy bridge position in the coordinate plane.

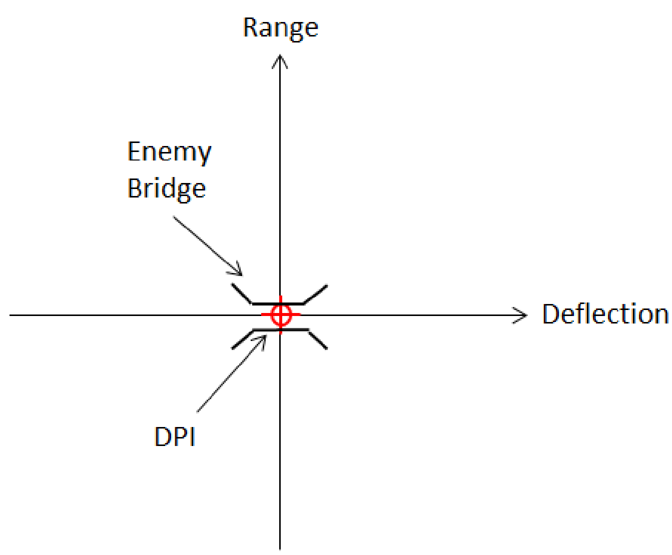

Figure 4. Definition of desired point of impact

The DPI is also known as the desired mean point of impact (DMPI) in cases where multiple weapons are released in a single salvo. Figure 5 illustrates the distribution of impact points around the DMPI.

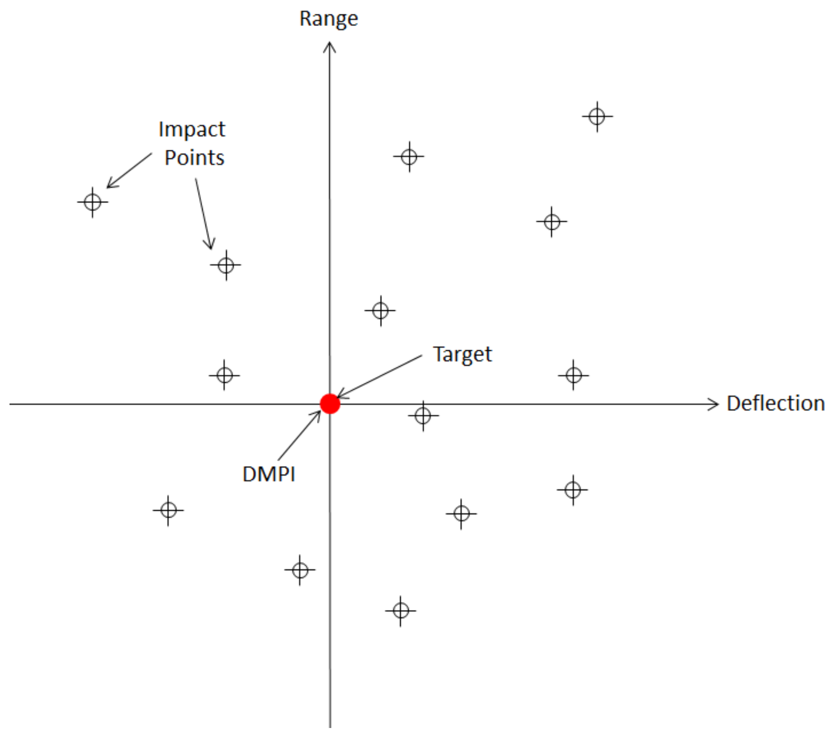

Figure 5. Distribution of multiple weapon impact points around desired mean point of impact

2. Impact Angle, *I*

The impact angle, *I*, refers to the angle from the ground at which the coordinate-seeking weapon impacts the target. The definition of the impact angle is illustrated in Figure 6.

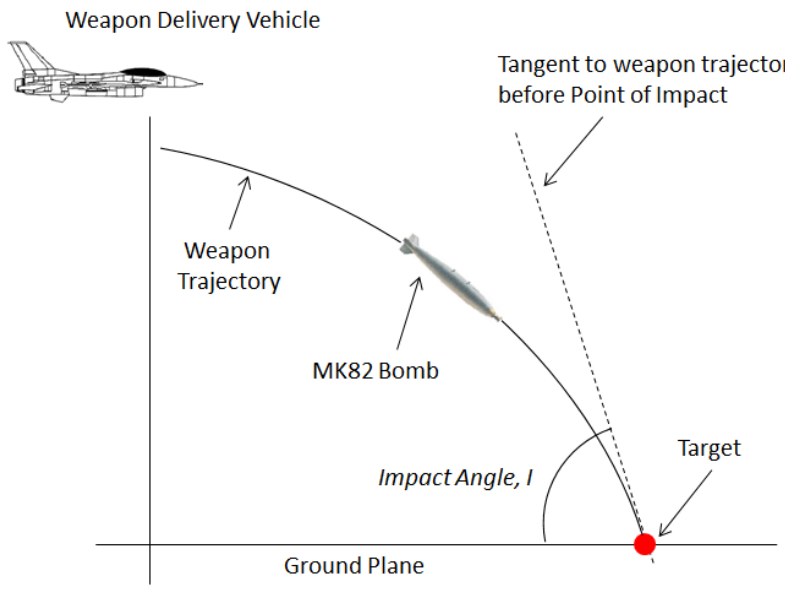

Figure 6. Definition of impact angle, *I*

Mission profiles vary in terms of the type of target to be bombed, collateral prevention, type of weapon used, and so forth. Hence, so do the impact angles for each mission profile. For comparison and discussion in this paper, the impact angle, *I*, is 65°.

B. TYPES OF ERRORS

Errors associated with coordinate-seeking weapons can be grouped into two main categories, dependent and independent.

1. Single Weapon Dependent (Aiming) Error

Dependent errors can be described as aiming errors, where the actual weapon impact point is at an offset location from the original weapon aimpoint at the target coordinates. Figure 7 illustrates an example of a dependent error for a single weapon aimed (single-shot) at the target, with the assumption that there is no independent error yet (to be discussed in next section).

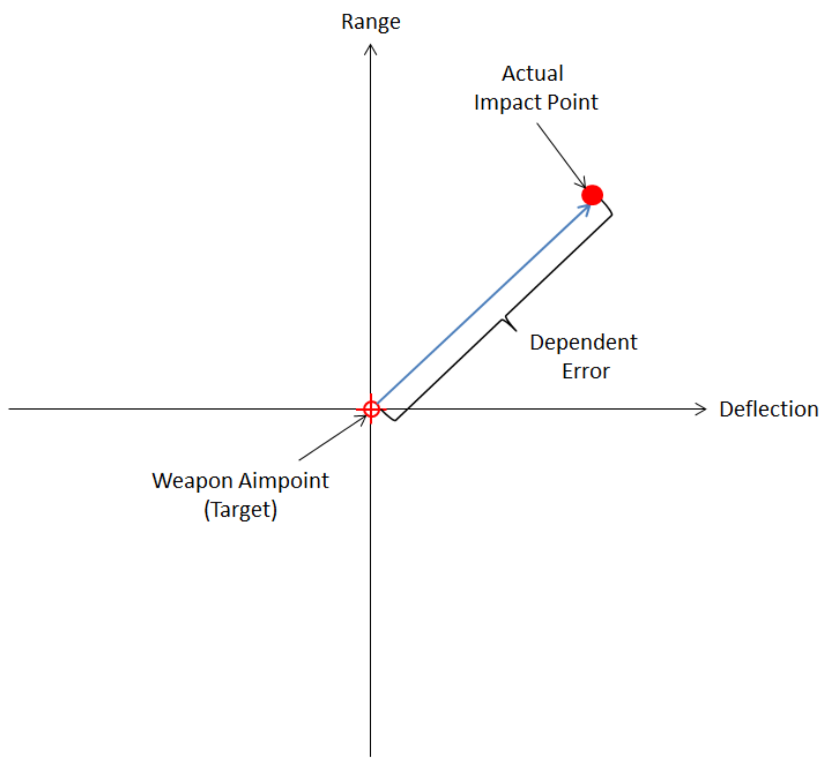

Figure 7. Definition of single-weapon dependent error

2. Single Weapon Independent (Ballistic) Error

Independent errors are errors where the impact points of the subsequent weapons are independent of the prior weapon impact point. Figure 8 illustrates the independent error for a single weapon used against a unitary target; the dependent error is also included in the figure.

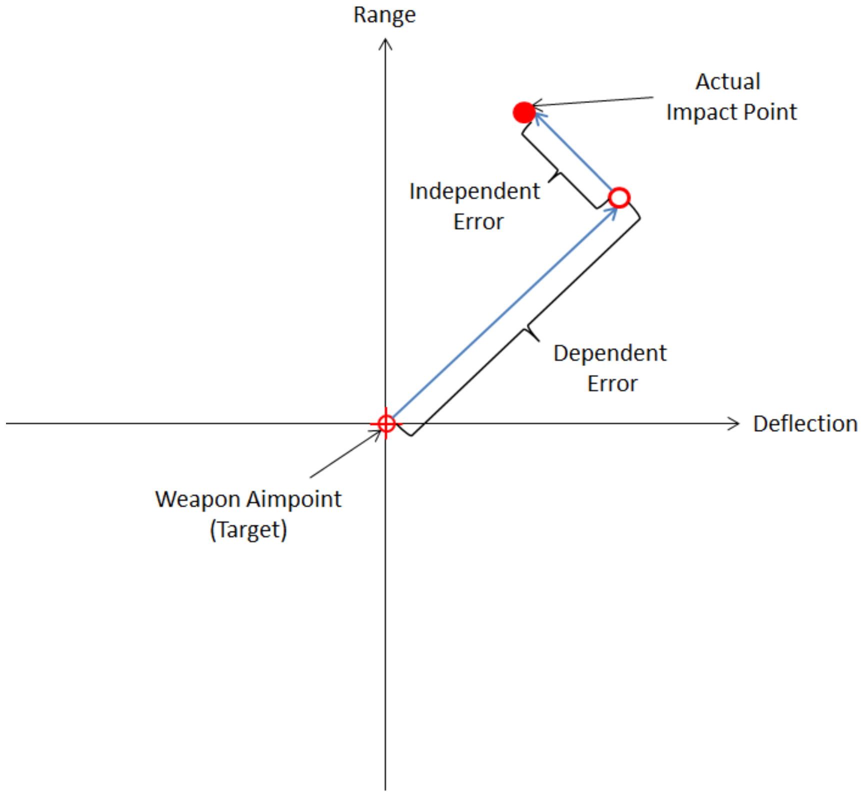

Figure 8. Definition of single-weapon independent error

3. Multiple Weapon Dependent (Aiming) Error

In the absence of independent errors, a multiple weapon salvo would theoretically impact the ground at fixed standoff distances from one another based on the intended salvo aimpoint placements around the target.

Figure 9 illustrates an example of the impact points from a four-weapon salvo around the weapon aimpoint (on target) without any errors, placed at the corners of a 10ft x 20ft rectangle.

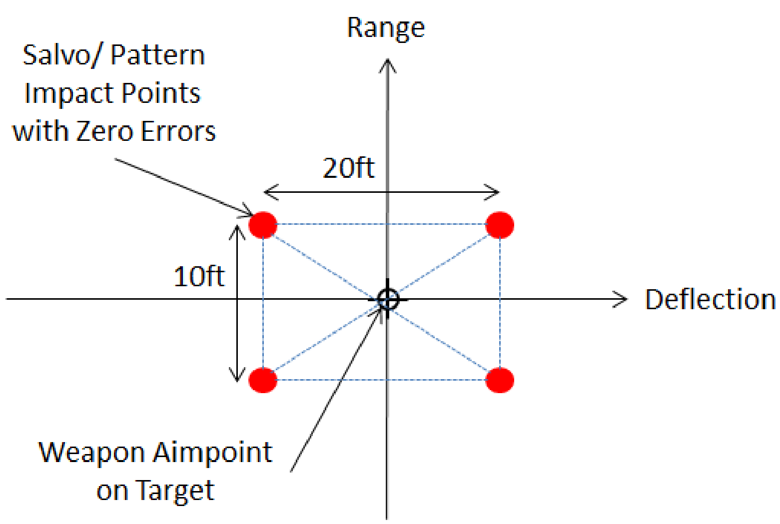

Figure 9. Impact points of four-weapon salvo around target without errors

Figure 10 illustrates an example of a dependent error for multiple weapons aimed at a target, with the assumption that there are no independent errors. The actual weapon aimpoint for the salvo of four weapons that impacted the ground is at an offset (dependent error) from the intended weapon aimpoint over the target.

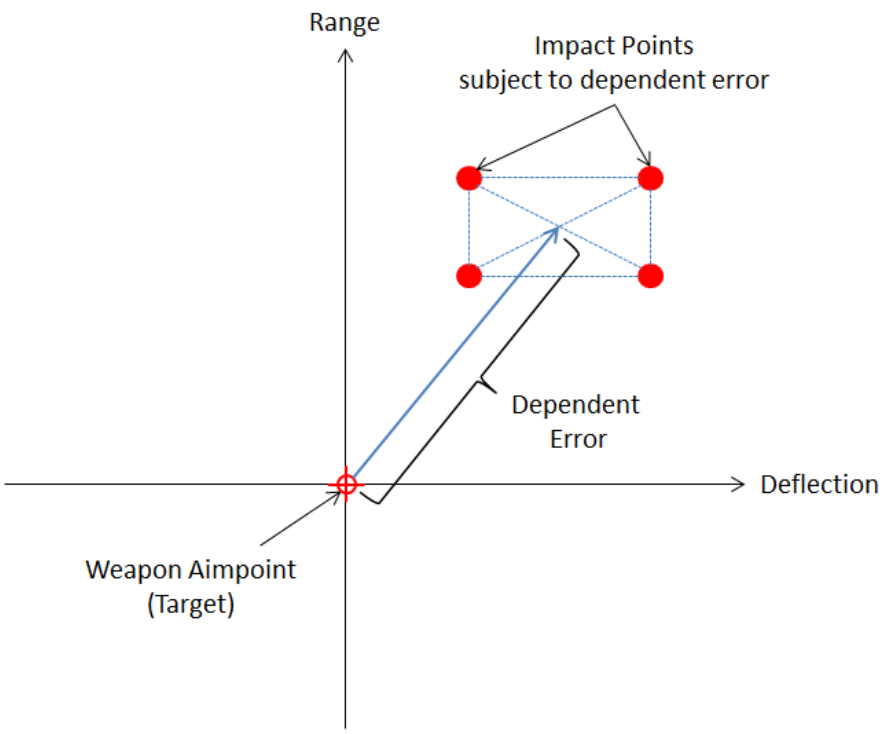

Figure 10. Definition of multiple-weapon dependent error

4. Multiple Weapon Independent (Ballistic) Error

Figure 11 illustrates the independent errors for multiple weapons used against a unitary target. The dependent error mentioned in the previous section is also included in the figure.

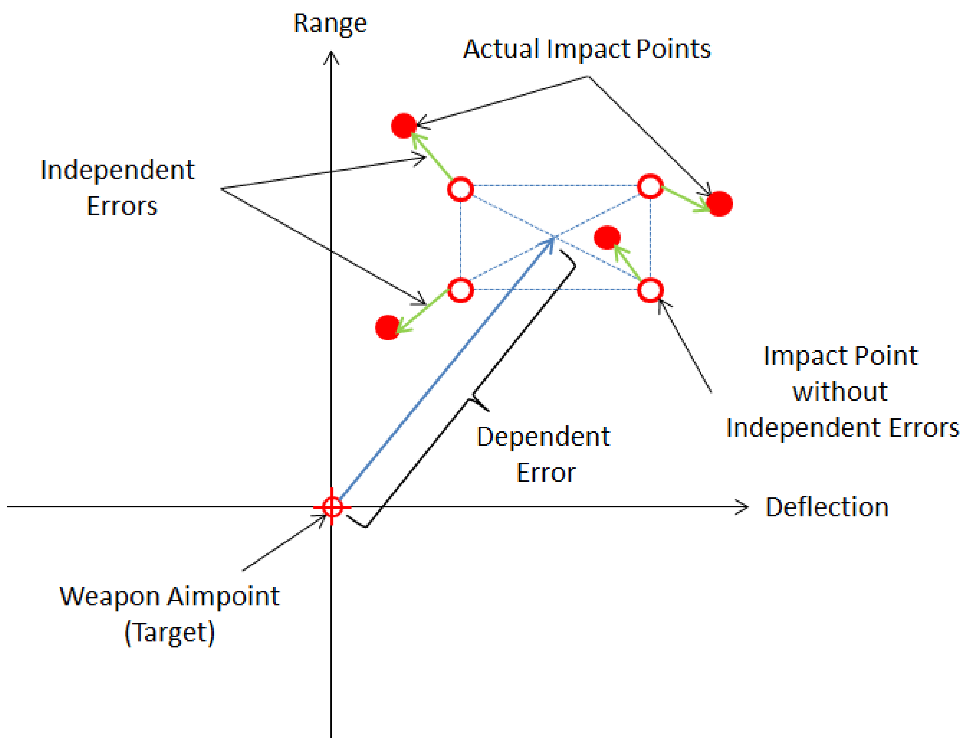

Figure 11. Definition of multiple-weapon independent errors

In addition to the dependent error that is already present, the actual impact points land at an offset distance away from the intended impact points when no independent errors are present.

C. DELIVERY ACCURACY MEASURES

1. Circular Error Probable

The circular error probable (CEP) is defined as the radius of a circle from the DPI where 50% of the impact points lie within. Figure 12 illustrates the definition of the CEP for single weapon independent impact points.

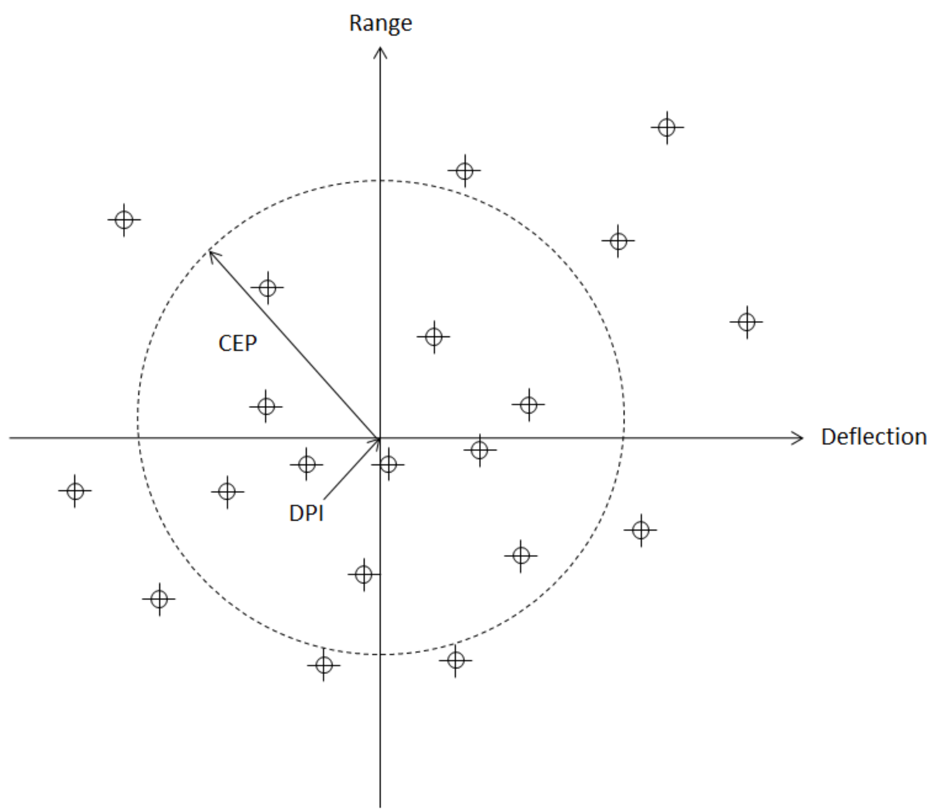

Figure 12. Definition of circular error probable

2. Range Error Probable

The range error probable (REP) is defined as the distance from the DPI that contains 50% of the impact points along both directions in the range axes. Figure 13 illustrates the definition of the REP with single weapon independent impact points.

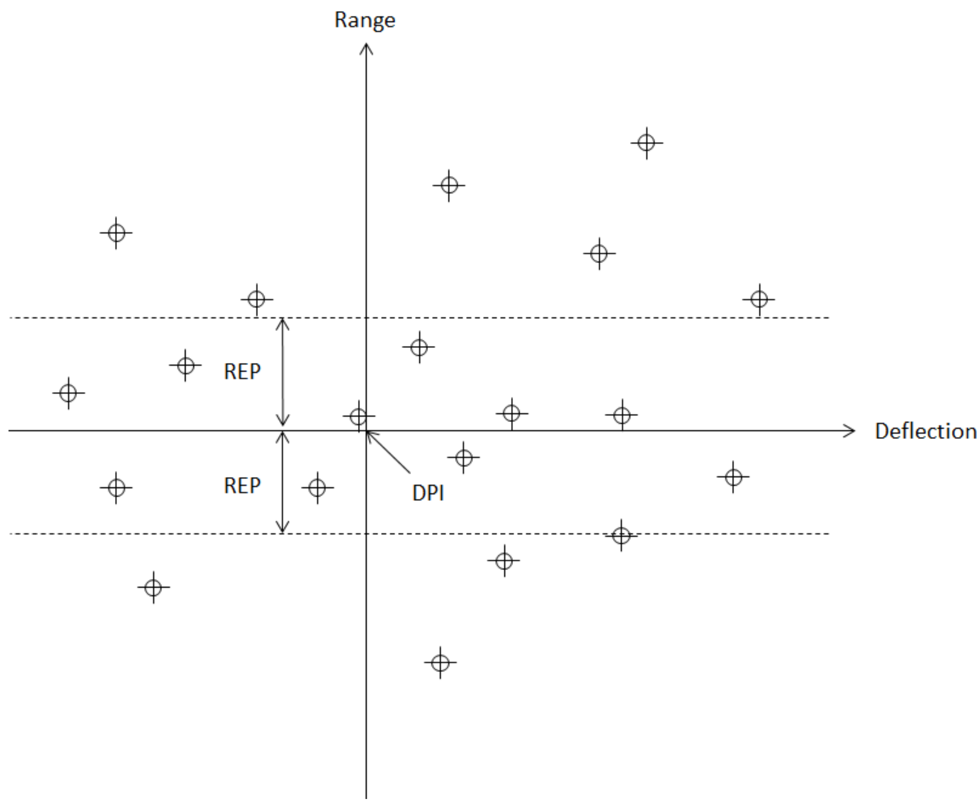

Figure 13. Definition of range error probable

3. Deflection Error Probable

The deflection error probable (DEP) is the distance from the DPI that contains 50% of the impact points along both directions in the deflection axes. Figure 14 illustrates the definition of the DEP with single weapon independent impact points.

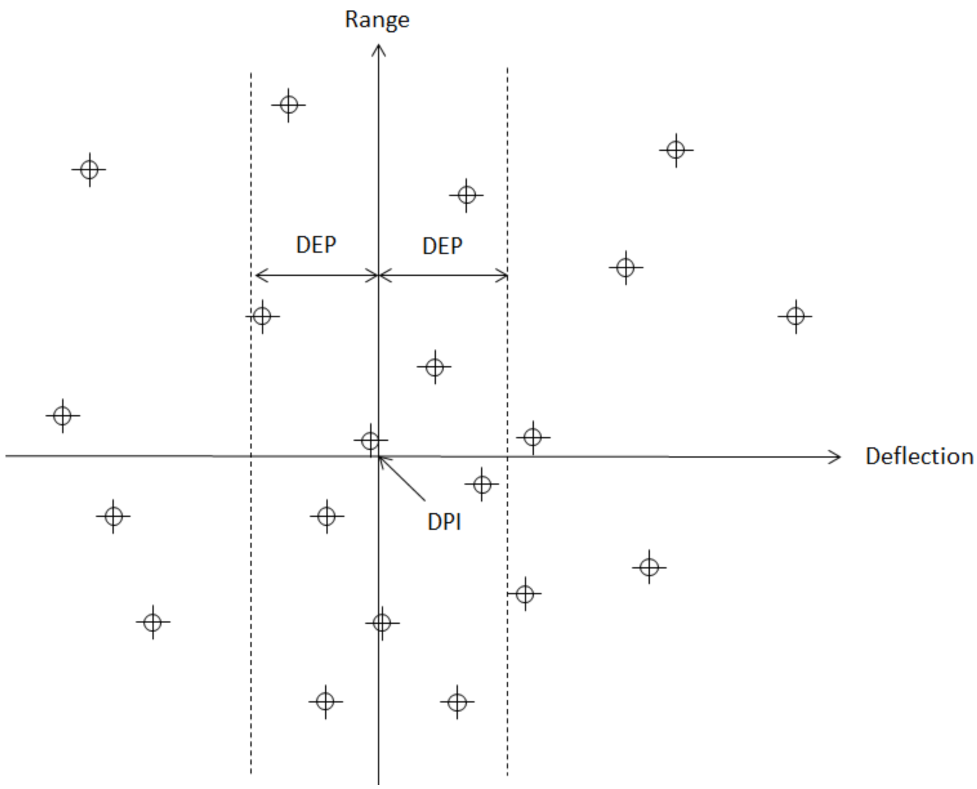

Figure 14. Definition of deflection error probable

4. Relationship of CEP with Respect to REP & DEP

To relate the CEP with the REP and DEP, assumptions are made in the following relations in that the distribution of impact points in the range and deflection directions are normally (Gaussian) distributed. As such, we may deduce the following relations as follows:

$$REP = 0.6745\sigma_x \tag{2.1}$$

$$DEP = 0.6745\sigma_y \tag{2.2}$$

The symbol σ denotes the standard deviation of a normal Gaussian statistical table. When the data distribution is further assumed to be circular with a zero mean where:

$$\sigma_x = \sigma_y = \sigma \tag{2.3}$$

Then the relationship between REP, DEP, CEP and σ can be represented as follows:

$$CEP = 1.1774\sigma \tag{2.4}$$

$$CEP = 1.7456REP = 1.7456DEP \tag{2.5}$$

D. DAMAGE FUNCTIONS

This section introduces the three damage functions that form the cornerstone of this paper and the basis of the developed MATLAB program for optimizing the probability of damage (PD) for multiple weapons on unitary targets. These damage functions have different complexities and hence produce varying levels of accuracies for use within the framework of the JMEM[1] Weaponeering System (JWS) methodologies. For example, depending on the requirements such as time constraint or fidelity (accuracy), a simple or more complex damage function may be selected for use within a particular JWS methodology respectively.

1. Lethal Area Matrix

a. Damage Matrix

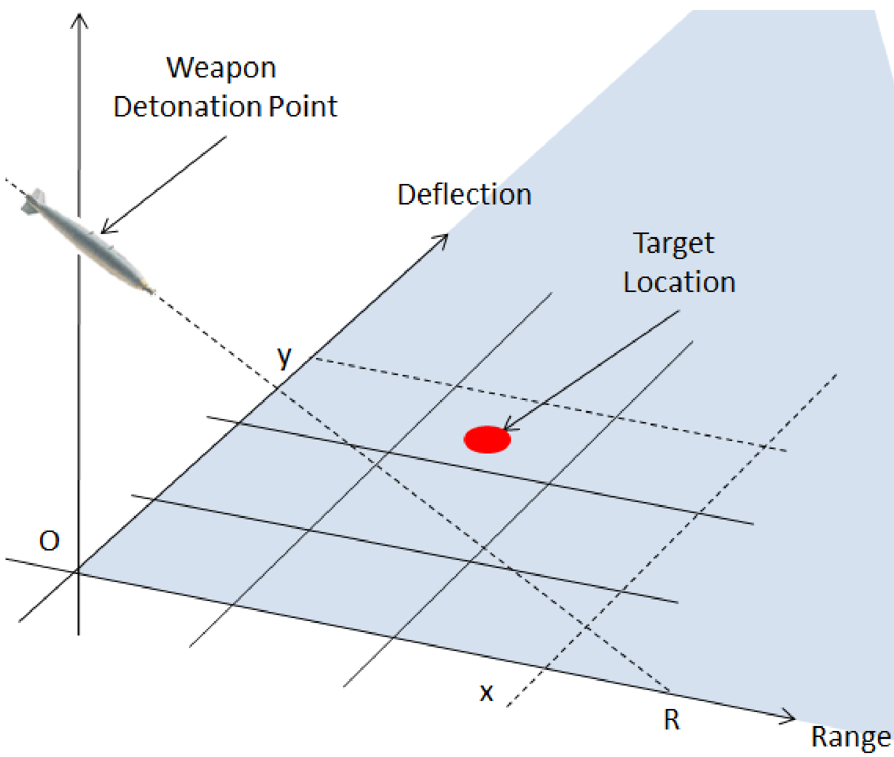

Figure 15. Weapon-target interaction geometry

[1] *Joint Munitions Effectiveness Manual* (JMEM).

Figure 15 provides an overview of the geometry of the weapon-target interaction. The weapon is placed on the ground plane location shown using methods referenced from [4], resulting in a calculated PD. The process is repeated for other locations within a grid of points on the ground plane. The resulting grid of cells with PD values inside each cell is known as the Lethal Area Matrix (LAM), a sample of which is shown in Figure 16.

In the LAM, the overall damage function is compartmentalized into separate but identical-sized cells, each with a PD value. Each individual identical cell area is assumed to be small enough such that the value of the PD is the same at any point within this individual cell. Figure 16 illustrates an example of a damage matrix of the LAM. The impact point is always taken to be at the center of the damage matrix, marked by the circle in the figure. The outlying areas in the range and deflection directions take the value of zero (not shown for deflection direction). This is due to the fact that as the kinetic fragments travel away from the impact point, they encounter air resistance and start to slow down to a point where the fragments do not possess enough kinetic energy to penetrate the target. As such, the values in the damage matrix taper out to zero values at the boundaries. Naturally, the onsets of the zero-value cells become the limits of the lethal region of the damage matrix.

	Deflection																			
Range	-379.0	-341.1	-303.2	-265.3	-227.4	-189.5	-151.6	-113.7	-75.8	-37.9	37.9	75.8	113.7	151.6	189.5	227.4	265.3	303.2	341.1	379.0
-128.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-114.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-100.1	0.0001	0.0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0001	0.0001
-85.8	0.0001	0.0001	0.0002	0.0001	0	0	0	0	0	0.0001	0.0001	0	0	0	0	0	0.0001	0.0002	0.0001	0.0001
-71.5	0.0001	0.0001	0.0002	0.0004	0.0003	0	0	0	0	0.0011	0.0011	0	0	0	0	0.0003	0.0004	0.0002	0.0001	0.0001
-57.2	0.0001	0.0001	0.0002	0.0004	0.0008	0.0009	0	0	0	0.0028	0.0028	0	0	0	0.0009	0.0008	0.0004	0.0002	0.0001	0.0001
-42.9	0.0001	0.0001	0.0002	0.0005	0.0009	0.0017	0.0029	0.0006	0.0001	0.0064	0.0064	0.0001	0.0006	0.0029	0.0017	0.0009	0.0005	0.0002	0.0001	0.0001
-28.6	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0042	0.0099	0.0059	0.1402	0.1402	0.0059	0.0099	0.0042	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
-14.3	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0045	0.0127	0.0459	0.5571	0.5571	0.0459	0.0127	0.0045	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
14.3	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0045	0.0156	0.0891	0.6794	0.6794	0.0891	0.0156	0.0045	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
28.6	0.0001	0.0001	0.0002	0.0005	0.0012	0.0041	0.0116	0.0325	0.0927	0.1741	0.1741	0.0927	0.0325	0.0116	0.0041	0.0012	0.0005	0.0002	0.0001	0.0001
42.9	0.0001	0.0002	0.0006	0.0016	0.0034	0.0063	0.0128	0.0258	0.0186	0.0060	0.0060	0.0186	0.0258	0.0128	0.0063	0.0034	0.0016	0.0006	0.0002	0.0001
57.2	0.0003	0.0006	0.0010	0.0017	0.0032	0.0061	0.0118	0.0105	0.0050	0.0007	0.0007	0.0050	0.0105	0.0118	0.0061	0.0032	0.0017	0.0010	0.0006	0.0003
71.5	0.0004	0.0006	0.0010	0.0017	0.0031	0.0056	0.0072	0.0015	0.0024	0	0	0.0024	0.0015	0.0072	0.0056	0.0031	0.0017	0.0010	0.0006	0.0004
85.8	0.0003	0.0005	0.0009	0.0017	0.0028	0.0045	0.0011	0.0012	0.0010	0	0	0.0010	0.0012	0.0011	0.0045	0.0028	0.0017	0.0009	0.0005	0.0003
100.1	0.0003	0.0005	0.0009	0.0015	0.0025	0.0012	0.0005	0.0009	0.0003	0	0	0.0003	0.0009	0.0005	0.0012	0.0025	0.0015	0.0009	0.0005	0.0003
114.4	0.0003	0.0005	0.0009	0.0014	0.0011	0.0002	0.0004	0.0006	0	0	0	0	0.0006	0.0004	0.0002	0.0011	0.0014	0.0009	0.0005	0.0003
128.7	0.0003	0.0004	0.0007	0.0009	0.0001	0.0002	0.0003	0.0003	0	0	0	0	0.0003	0.0003	0.0002	0.0001	0.0009	0.0007	0.0004	0.0003
143.0	0.0003	0.0004	0.0006	0.0001	0.0001	0.0001	0.0003	0.0001	0	0	0	0	0.0001	0.0003	0.0001	0.0001	0.0001	0.0006	0.0004	0.0003
157.3	0.0002	0.0004	0.0002	0	0.0001	0.0001	0.0002	0	0	0	0	0	0	0.0002	0.0001	0.0001	0	0.0002	0.0004	0.0002
171.6	0.0002	0.0002	0	0	0.0001	0.0001	0.0001	0	0	0	0	0	0	0.0001	0.0001	0.0001	0	0	0.0002	0.0002
185.9	0.0001	0	0	0	0.0001	0.0001	0.0001	0	0	0	0	0	0	0.0001	0.0001	0.0001	0	0	0	0.0001
200.2	0	0	0	0	0	0.0001	0	0	0	0	0	0	0	0	0.0001	0	0	0	0	0
214.5	0	0	0	0	0	0.0001	0	0	0	0	0	0	0	0	0.0001	0	0	0	0	0
228.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
243.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 16. Example of damage matrix of LAM, from [4]

In Figure 16, passing through the weapon impact point are two dashed lines in the vertical and horizontal directions; these lines represent the zero or reference axes in range and deflection.

The extreme left-most column of the damage matrix represents the range axes and contains the cell heights in unit increments of 14.3ft in either direction (up/down) from the range centerline (horizontal dashed line). Similarly, the top-most row represents the deflection axes and contains the cell widths in unit increments of 37.9ft in either direction (left/right) from the deflection centerline (vertical dashed line). A portion of the damage matrix bound by the rectangle in Figure 16 is shown in Figure 17 as an example, and highlights the range and deflection increments of the cells in the vicinity of the weapon impact point.

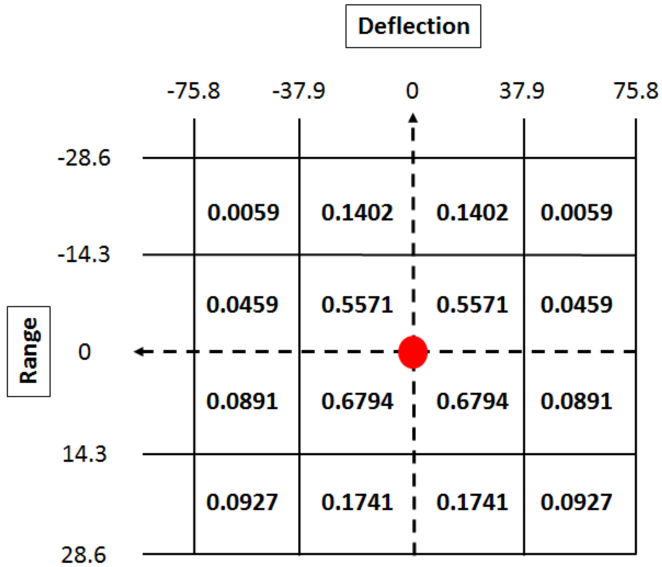

Figure 17. Cell incremental length and width of damage matrix

b. Lethal Area, A_L

The lethal area, A_L, of the LAM can be calculated by multiplying the individual cell areas with its corresponding PD value and then summing up all of the cells that have a non-zero PD. Mathematically, the weapon lethal area can be represented by the following equation:

$$A_L = \sum_{x=x_{\min}}^{x=x_{\max}} \sum_{y=y_{\min}}^{y=y_{\max}} P_D \triangle y \triangle x \tag{2.6}$$

Based on the given damage matrix in Figure 16, the previous equation gives rise to a lethal area of approximately 2270ft^2. This lethal area is also known as the Mean Area of Effectiveness (Fragmentation), MAE_F, where:

$$MAE_F = A_L \tag{2.7}$$

c. Probability of Damage (LAM)

Taking the weapon impact point to be the center of the damage matrix, the probability of damage (PD) for a unitary weapon can be determined by offsetting the placement of the target in the damage matrix with the same magnitude and direction as where the target would be from the weapon impact point. Figure 18 and Figure 19 illustrate how the PD can be read off the damage matrix given the location of the target.

	-379.0	-341.1	-303.2	-265.3	-227.4	-189.5	-151.6	-113.7	-75.8	-37.9	37.9	75.8	113.7	151.6	189.5	227.4	265.3	303.2	341.1	379.0
-128.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-114.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-100.1	0.0001	0.0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0001	0.0001
-85.8	0.0001	0.0001	0.0002	0.0001	0	0	0	0	0	0.0001	0.0001	0	0	0	0	0	0.0001	0.0002	0.0001	0.0001
-71.5	0.0001	0.0001	0.0002	0.0004	0.0003	0	0	0	0	0.0011	0.0011	0	0	0	0	0.0003	0.0004	0.0002	0.0001	0.0001
-57.2	0.0001	0.0001	0.0002	0.0004	0.0008	0.0009	0	0	0	0.0028	0.0028	0	0	0	0.0009	0.0008	0.0004	0.0002	0.0001	0.0001
-42.9	0.0001	0.0001	0.0002	0.0005	0.0009	0.0017	0.0029	0.0006	0.0001	0.0064	0.0064	0.0001	0.0006	0.0029	0.0017	0.0009	0.0005	[illegible]	[illegible]	0.0001
-28.6	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0042	0.0099	0.0059	0.1402	0.1402	0.0059	0.0099	0.0042	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
-14.3	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0045	0.0127	0.0459	0.5571	0.5571	0.0459	0.0127	0.0045	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
14.3	0.0001	0.0001	0.0002	0.0005	0.0009	0.0019	0.0045	0.0156	0.0891	0.6794	0.6794	0.0891	0.0156	0.0045	0.0019	0.0009	0.0005	0.0002	0.0001	0.0001
28.6	0.0001	0.0001	0.0002	0.0005	0.0012	0.0041	0.0116	0.0325	0.0927	0.1741	0.1741	0.0927	0.0325	0.0116	0.0041	0.0012	0.0005	0.0002	0.0001	0.0001
42.9	0.0001	0.0002	0.0006	0.0016	0.0034	0.0063	0.0128	0.0258	0.0186	0.0060	0.0060	0.0186	[illegible]	[illegible]	0.0063	0.0034	0.0016	0.0006	0.0002	0.0001
57.2	0.0003	0.0006	0.0010	0.0017	0.0032	0.0061	0.0118	0.0105	0.0050	0.0007	0.0007	0.0050	0.0105	0.0118	0.0061	0.0032	0.0017	0.0010	0.0006	0.0003
71.5	0.0004	0.0006	0.0010	0.0017	0.0031	0.0056	0.0072	0.0015	0.0024	0	0	0.0024	0.0015	0.0072	0.0056	0.0031	0.0017	0.0010	0.0006	0.0004
85.8	0.0003	0.0005	0.0009	0.0017	0.0028	0.0045	0.0011	0.0012	0.0010	0	0	0.0010	0.0012	0.0011	0.0045	0.0028	0.0017	0.0009	0.0005	0.0003
100.1	0.0003	0.0005	0.0009	0.0015	0.0025	0.0012	0.0005	0.0009	0.0003	0	0	0.0003	0.0009	0.0005	0.0012	0.0025	0.0015	0.0009	0.0005	0.0003
114.4	0.0003	0.0005	0.0009	0.0014	0.0011	0.0002	0.0004	0.0006	0	0	0	0	0.0006	0.0004	0.0002	0.0011	0.0014	0.0009	0.0005	0.0003
128.7	0.0003	0.0004	0.0007	0.0009	0.0001	0.0002	0.0003	0.0003	0	0	0	0	0.0003	0.0003	0.0002	0.0001	0.0009	0.0007	0.0004	0.0003
143.0	0.0003	0.0004	0.0006	0.0001	0.0001	0.0001	0.0003	0.0001	0	0	0	0	0.0001	0.0003	0.0001	0.0001	0.0001	0.0006	0.0004	0.0003
157.3	0.0002	0.0004	0.0002	0	0.0001	0.0001	0.0002	0	0	0	0	0	0	0.0002	0.0001	0.0001	0	0.0002	0.0004	0.0002
171.6	0.0002	0.0002	0	0	0.0001	0.0001	0.0001	0	0	0	0	0	0	0.0001	0.0001	0.0001	0	0	0.0002	0.0002
185.9	0.0001	0	0	0	0.0001	0.0001	0.0001	0	0	0	0	0	0	0.0001	0.0001	0.0001	0	0	0	0.0001
200.2	0	0	0	0	0	0.0001	0	0	0	0	0	0	0	0	0.0001	0	0	0	0	0
214.5	0	0	0	0	0	0.0001	0	0	0	0	0	0	0	0	0.0001	0	0	0	0	0
228.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
243.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

-68ft

200ft

Figure 18. Target location from weapon impact point

For example, a target that has coordinates of -68ft in the range and 200ft in the deflection directions relative to the weapon impact point, falls within the cell as illustrated in Figure 19, where the PD of the target can simply be read off as 0.0003.

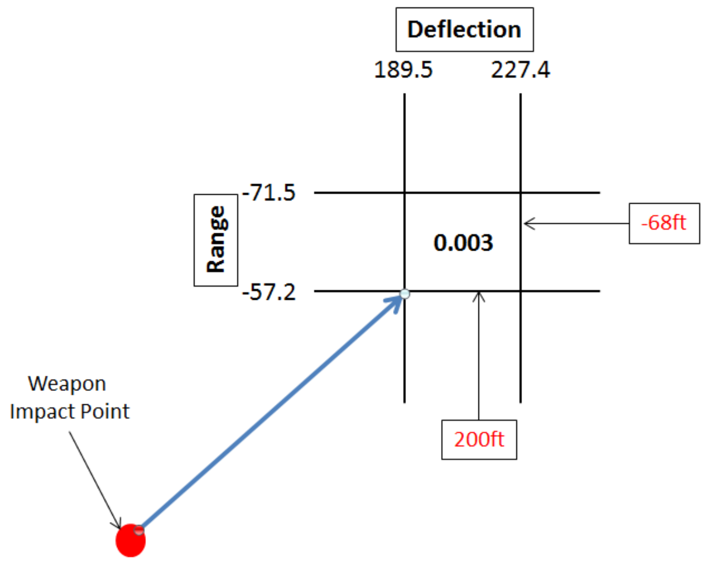

Figure 19. Selecting the correct cell and PD value from damage matrix

2. Carlton Damage Function

a. Simplification of Damage Matrix

The Carlton Damage Function (CDF) is a simplification of the LAM damage function, and as a consequence, has a lower fidelity than that of the LAM. By plotting contour lines along cells with the same PD values in the damage matrix, a plot of the form as shown in Figure 20 is generated.

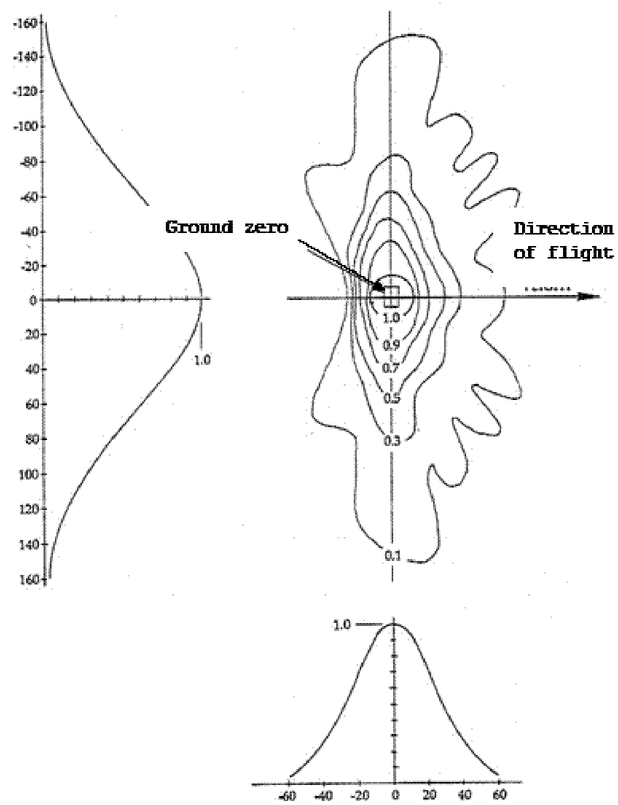

Figure 20. PD contour lines of LAM damage matrix, from [4]

And by utilizing Gaussian-like approximations on the contour lines, Figure 20 can be approximated into a smooth three-dimensional version of the damage matrix as illustrated in Figure 21.

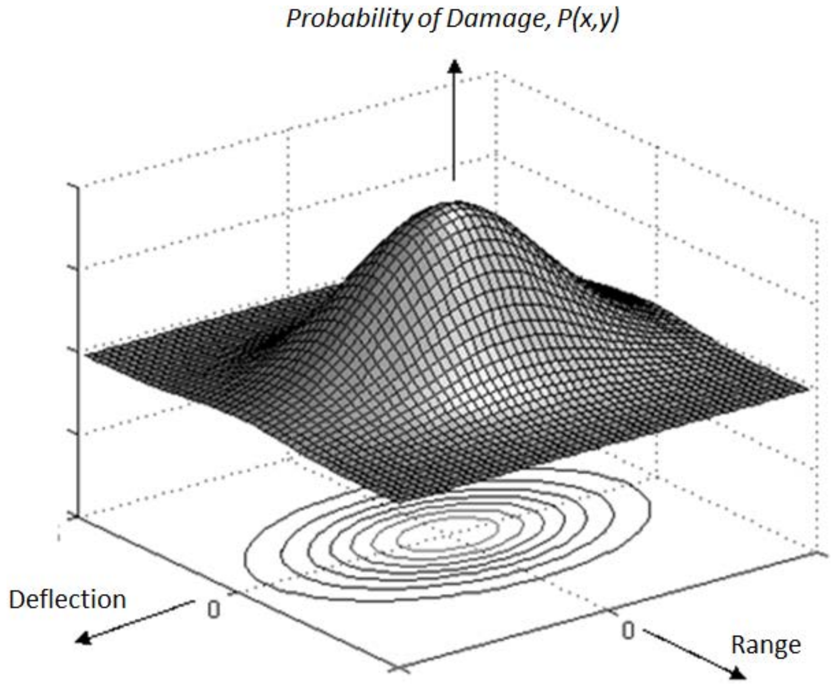

Figure 21. Definition of Carlton Damage Function, from [4]

b. Damage Function

The CDF can now be described mathematically by the following equation:

$$PD = P(x, y) = \exp\left\{-\left[\frac{x^2}{WR_r^2} + \frac{y^2}{WR_d^2}\right]\right\} \tag{2.8}$$

The notations WR_r and WR_d from the CDF equation (2.8) represent the weapon radii in the range and deflection directions respectively, which are discussed in the next section.

c. Weapon Radii and Aspect Ratio

As was previously mentioned in the LAM damage function, the effectiveness of the weapon decreases with increasing distance away from the center. However, there is no fixed limit to the weapon effectiveness; it tends to zero as the distance from the target tends to infinity.

(1) Weapon Radius (Range), WR_r

The Weapon Radius (Range), WR_r, describes the spread of weapon effectiveness in the range axes in both directions.

(2) Weapon Radius (Deflection), WR_d

The Weapon Radius (Deflection), WR_d, describes the spread of weapon effectiveness in the deflection axes in both directions.

(3) Aspect Ratio, a

Aspect ratio, a, is the ratio of the weapon radii of the CDF; their relationship can be represented by the following equation:

$$a = \frac{WR_r}{WR_d} \tag{2.9}$$

It may be shown that the aspect ratio is a function of the impact angle and has been found to be represented by the following empirical equation:

$$a = MAX\left(1 - 0.8\cos I, 0.3\right) \tag{2.10}$$

d. Weapon Lethal Area, A_L

The weapon lethal area, A_L, of the CDF is calculated as:

$$\begin{aligned} A_L &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P\left(x, y\right) dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\left[\frac{x^2}{WR_r^2} + \frac{y^2}{WR_d^2}\right]\right\} dxdy \\ &= \pi \times WR_r \times WR_d \end{aligned} \tag{2.11}$$

3. Rectangular Damage Function

A further simplification of the CDF is to represent the damage function in the form of a Rectangular Cookie-Cutter (RCC) as shown in Figure 22. The RCC demarcates the weapon lethal area, A_L, of the Rectangular Damage Function (RDF).

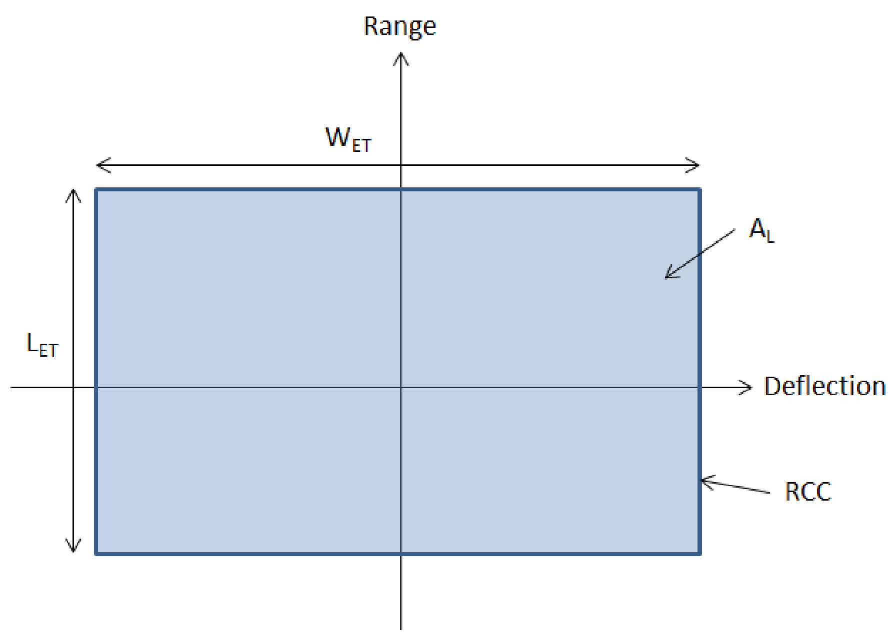

Figure 22. Definition of RCC or rectangular weapon lethal area, A_L

a. *Effective Target Length, Width and Aspect Ratio*

As illustrated in Figure 22, the effective target length, L_{ET}, and width, W_{ET}, denotes the length and width of the RCC respectively.

(1) Length, L_{ET}

The effective target length, L_{ET}, represents the weapon effectiveness limit in the range direction as shown in Figure 22.

(2) Width, W_{ET}

The effective target width, W_{ET}, represents the weapon effectiveness limit in the deflection direction as shown in Figure 22.

(3) Aspect Ratio, *a*

The Aspect Ratio, *a*, is the ratio of the Effective Target Length to Width; and their relationship can be represented by the following equation:

$$a = \frac{W_{ET}}{L_{ET}} \tag{2.12}$$

It may be shown that the aspect ratio is a function of the impact angle and has been found to be represented by the same equation (2.10) shown previously.

b. Weapon Lethal Area, A_L

The weapon lethal area, A_L, for the RDF is simply the multiplication of the effective target length, L_{ET}, and width, W_{ET}, as per the following equation:

$$MAE_F = L_{ET} \times W_{ET} \tag{2.13}$$

c. Probability of Damage (RDF)

The PD for the RDF can be defined using the RCC concept by taking unity (PD = 1) for any target that falls on or within the limits of the RCC and a PD of zero (PD = 0) for targets falling anywhere outside of the RCC. Figure 23 illustrates the above-mentioned PD = 1 and PD = 0 scenarios.

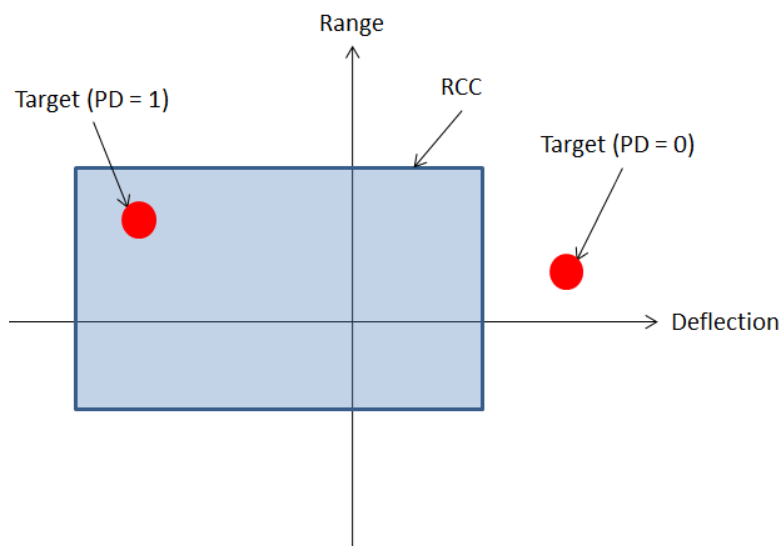

Figure 23. Scenarios for PD = 1 and PD = 0

4. Conserving Lethality for Different Damage Functions

For the three different damage functions, the lethality of the weapon is always conserved from the most detailed (LAM) through to the simplest (RDF) damage function in terms of fidelity. Using an example, the mean area of effectiveness (fragmentation), MAE_F, of 2270ft^2 calculated from the LAM damage function in equation (2.6) as a reference, the following steps demonstrate that the MAE_F for the CDF and RDF is conserved. Using the impact angle, I, of 65°, the aspect ratio can be calculated using equation (2.10):

$$a = MAX\left(1-0.8\cos(65), 0.3\right) = \frac{WR_r}{WR_d} = 0.662 \tag{2.14}$$

For the CDF, the previously stated integral equation (2.11) can be evaluated to the following form:

$$A_L = MAE_F = \pi \times WR_r \times WR_d = 2270\, ft^2 \tag{2.15}$$

The Weapon Radii for the CDF can now be computed from the MAE_F:

$$MAE_F = 2270 = \pi \times WR_r \times WR_d = \pi \times WR_r \times {}^{WR_r}\!/_{a} \tag{2.16}$$

$$WR_r^{\ 2} = MAE_F \times {}^{a}\!/_{\pi} \tag{2.17}$$

$$WR_r = \sqrt{MAE_F \times {}^{a}\!/_{\pi}} = \sqrt{2270 \times {}^{0.662}\!/_{\pi}} = 21.87\, ft \tag{2.18}$$

$$WR_d = {}^{WR_r}\!/_{a} = {}^{21.87}\!/_{0.662} = 33.04\, ft \tag{2.19}$$

translating to a Carlton Damage Function (CDF) of:

$$\begin{aligned} c(x, \mathrm{y}) &= \exp\left(-\left[\frac{x^2}{WR_r^2} + \frac{y^2}{WR_d^2}\right]\right) \\ &= \exp\left(-\left[\frac{x^2}{21.87^2} + \frac{y^2}{33.04^2}\right]\right) \end{aligned} \tag{2.20}$$

When equation (2.20) is integrated over the ground plane and with the limits of the elliptical weapon lethal area as the limits of the integral, the MAE_F of 2270ft^2 will be obtained.

For the RDF with reference to equations (2.12) and (2.13), the following equations will be obtained:

$$L_{ET} = \frac{MAE_F}{W_{ET}} = \frac{MAE_F}{L_{ET} \times a} \tag{2.21}$$

$$L_{ET}{}^2 = MAE_F \times a \tag{2.22}$$

$$L_{ET} = \sqrt{MAE_F \times a} = \sqrt{2270 \times 0.662} = 38.76\,ft \tag{2.23}$$

$$W_{ET} = \frac{L_{ET}}{a} = \frac{38.76}{0.662} = 58.56\,ft \tag{2.24}$$

The weapon lethal area, A_L, would be:

$$\begin{aligned} A_L &= L_{ET} \times W_{ET} \\ &= 38.76 \times 58.56 \\ &= 2270\,ft^2 \end{aligned} \tag{2.25}$$

which is equivalent to the initial MAE_F value of 2270ft^2 and thus, lethality is shown to be conserved through all three damage functions.

E. MONTE-CARLO SIMULATION FOR WEAPON IMPACT POINTS

1. Example of Monte-Carlo Simulator

The Monte-Carlo (MC) simulation method is an iterative mathematical procedure that involves simulating the randomness of a particular process, with the conduct of a large enough number of trials to return a sufficiently consistent value. The MATLAB program uses a random generator function (‘*randn*’) to produce randomized numbers with a normal distribution as inputs for each iterative process of the simulator. Suppose a weapon has accuracy (σ_x, σ_y), the weapon impact points can be simulated using the MC approach.

Figure 24 illustrates the MC simulation algorithm in the form of a generalized flow diagram with 100 iterations as an example. An example of the Monte-Carlo simulation program can be found in Appendix A.1.

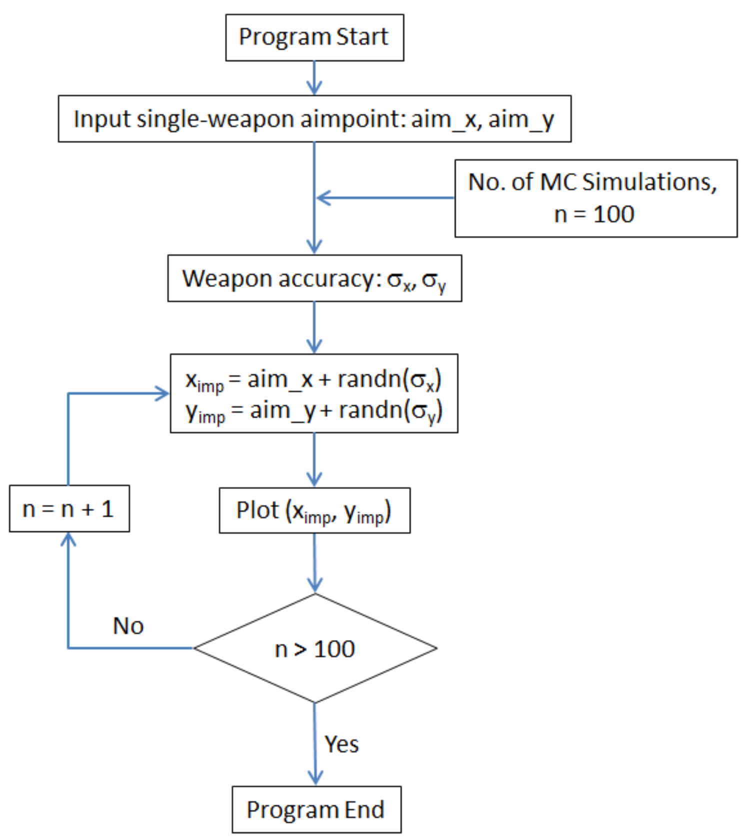

Figure 24. Flow diagram for MC-simulation of 100 iterations

Figure 25 presents the MC-simulated plot of 100 impact points with weapon accuracies of $\sigma_x = \sigma_y = 15$ around the target (*aim_x=aim_y=0*) based on the algorithm illustrated in the flow diagram in Figure 24.

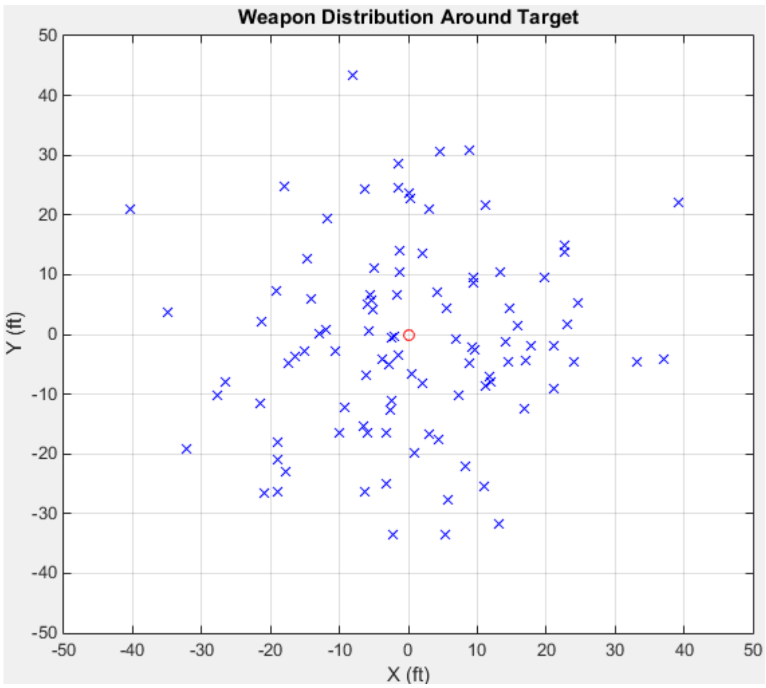

Figure 25. MC-simulated plot of 100 impact points around target (circle)

2. Single Weapon Impact Point

In order to simulate the probability of damage (PD) for a single weapon impact point using the Monte-Carlo (MC) simulation method, an iterative algorithm based on the MC approach was developed to produce single impact points similar to that previously illustrated in Figure 8. Dependent and independent errors are included in this algorithm.

Figure 26 presents the algorithm in the form of a flow diagram required to achieve the PD for single weapon impact point. An example of the single-weapon impact point generator program in MATLAB can be found in Appendix A.2.

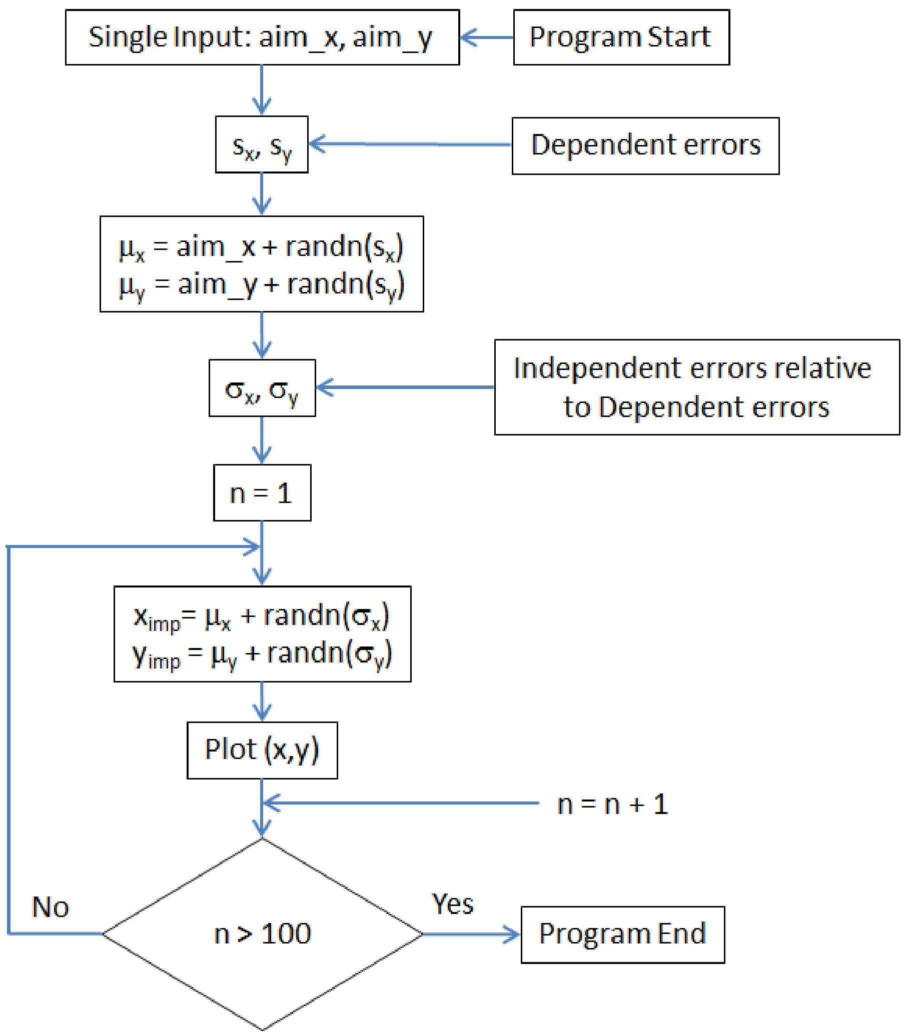

Figure 26. Single-weapon MC-simulation algorithm

Figure 27 presents the actual plot from running the MC-simulated MATLAB program for single weapon impact points with dependent and independent errors of 15 and 5 respectively, centered on the target (circle).

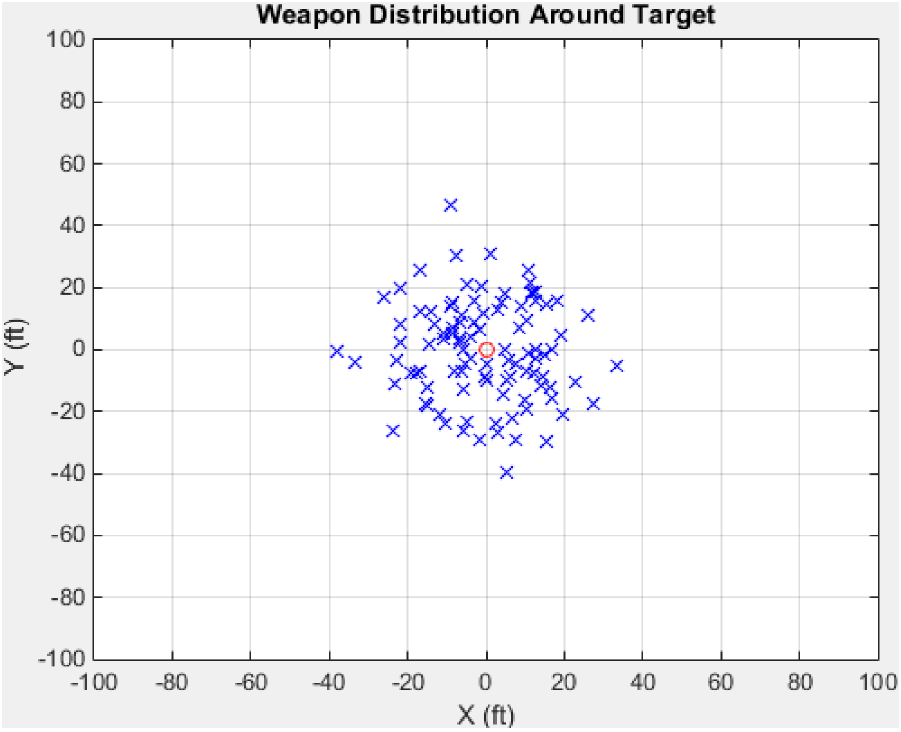

Figure 27. Single weapon impact points using MC-simulation (100 iterations)

3. Effectiveness for Single Weapon

The single weapon effectiveness is obtained through any one of the three damage functions mentioned previously, which are explained in the following three sub-sections.

a. LAM Effectiveness (Single Weapon)

The effectiveness or PD for a single weapon utilizing the LAM is reflected by the position of the target within the damage matrix of the LAM (Figure 16), relative to the weapon impact point. The position of the target is set at the origin (*0,0*) of the range-deflection plane, and assuming that the single weapon impact point includes both dependent and independent errors, the PD obtained for a single weapon through the LAM is explained through Figure 28 and Figure 29.

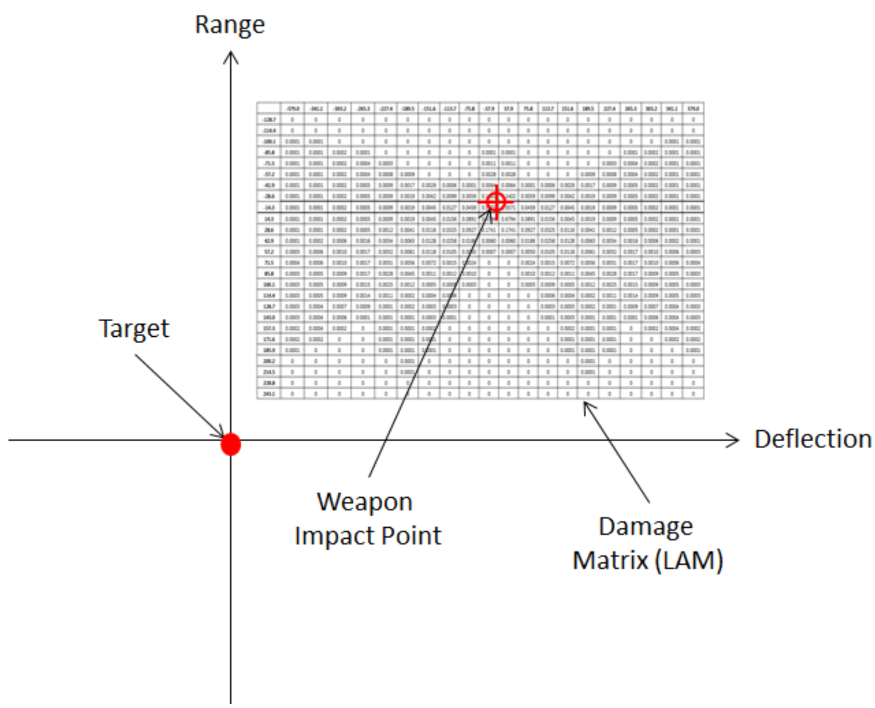

Figure 28. Example of PD that is zero using the LAM

With reference to Figure 28, the weapon impact point is at an offset location from the target. Since the target does not fall within the damage matrix of the LAM, the effectiveness or PD is zero in this case.

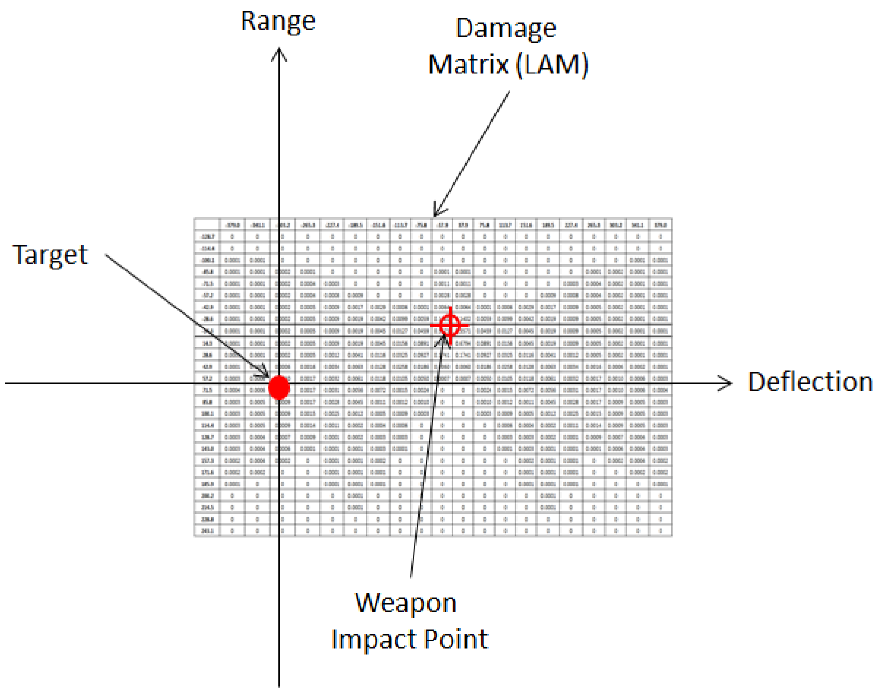

Figure 29. PD of greater than zero using the LAM

For a different impact point shown in Figure 29, the target falls within the influence of the damage matrix of the weapon impact. As such, the effectiveness or PD of the weapon on the target can be read off the particular cell of the LAM damage matrix within which the target falls.

b. CDF Effectiveness (Single Weapon)

The effectiveness or PD for a single weapon utilizing the CDF is explained using Figure 30.

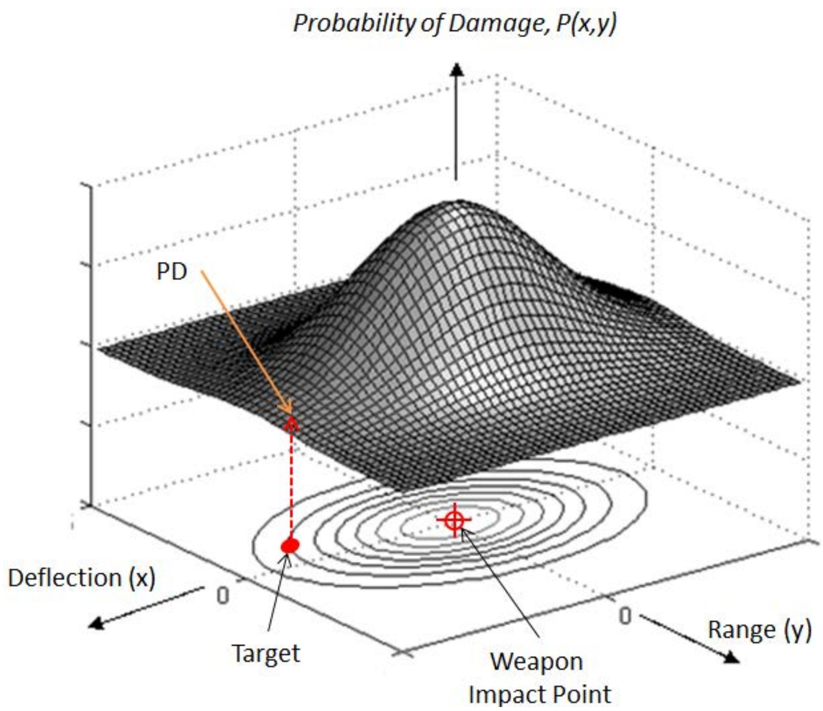

Figure 30. Effectiveness or PD on the target using the CDF, after [4]

With reference to Figure 30 and setting the weapon impact point as the origin of the range-deflection ground plane, the single-weapon PD of the target can be obtained by inputting the (x,y) coordinates of the target relative to the weapon impact point into equation (2.8).

c. RDF Effectiveness (Single Weapon)

The effectiveness or PD for a single weapon utilizing the RDF in the form of a RCC is represented by whether the target falls within the lethal area of the RCC. The position of the target is set at the origin (*0,0*) of the range-deflection plane, and assuming that the single weapon impact point includes both dependent and independent errors, the PD obtained for a single weapon through the RDF is explained in Figure 31.

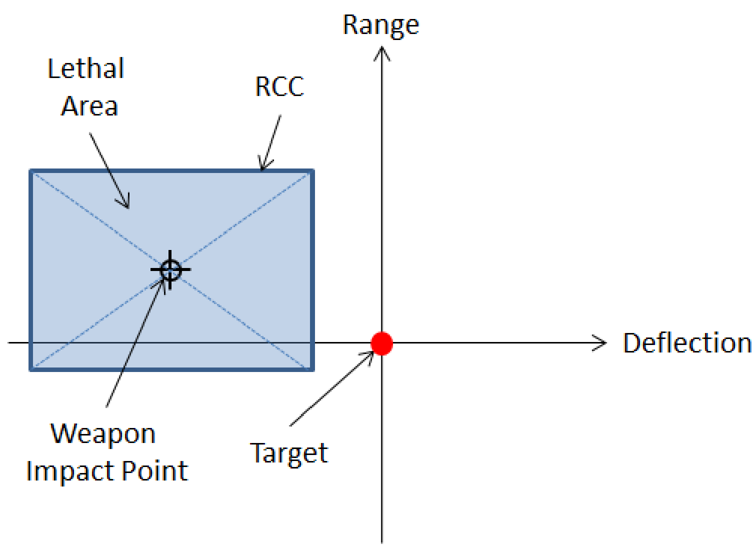

Figure 31. Zero PD using the RDF

With reference to Figure 31, from where the weapon impacts the range-deflection plane, the lethal area of the RCC does not enclose the target. As such, the PD equates to zero.

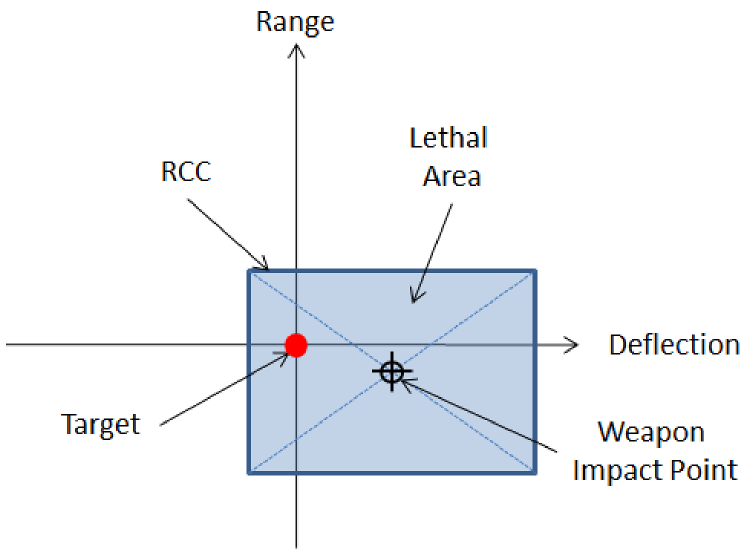

Figure 32. Unitary PD using the RDF

With reference to Figure 32, the lethal area of the RCC encloses the target based on the weapon impact point. As such, the PD is unity.

4. Multiple Weapons Impact Points

Similar to the single weapon impact point, an iterative algorithm based on the MC approach was also developed to produce multiple weapon impact points similar to that previously illustrated in Figure 8. Likewise, dependent and independent errors are also included in this algorithm.

Figure 33 presents the algorithm in the form of a flow diagram required to achieve the PD for multiple weapons impact points. An example of the multiple-weapon impact point generator program in MATLAB can be found in Appendix A.3.

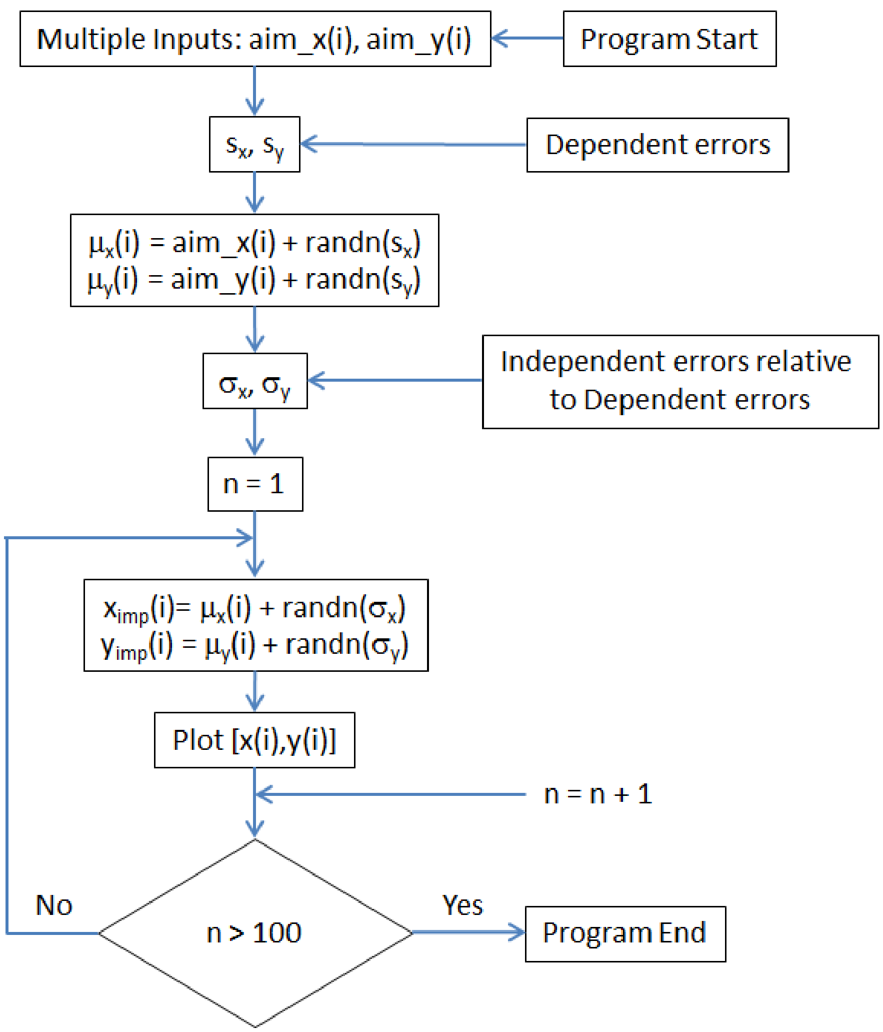

Figure 33. Multiple-weapon MC-simulation algorithm

Consider a pattern of four shots as shown in Figure 34.

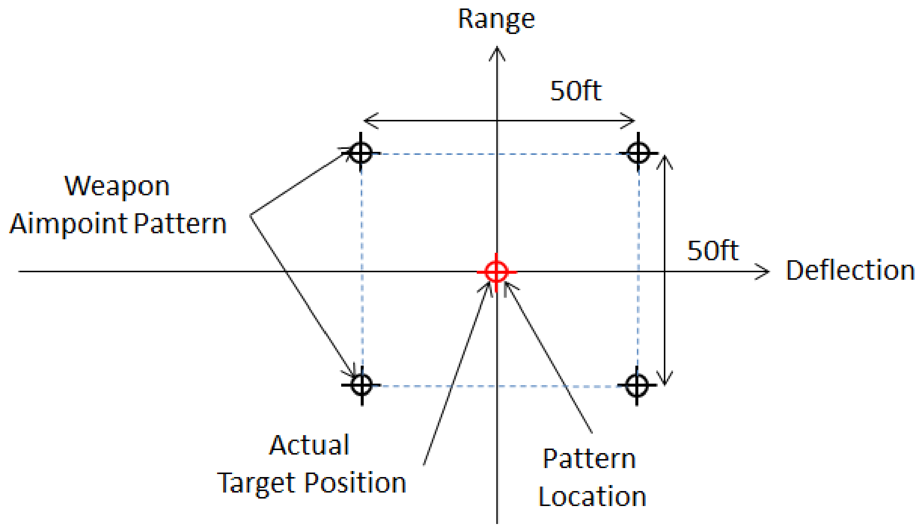

Figure 34. Pattern of four shots centered at target

Figure 35 presents the actual plot of multiple four-weapon impact points with dependent and independent errors, centered at the aimpoints of (25, 25), (-25, 25), (25, -25) and (-25, -25) for (aim_x, aim_y).

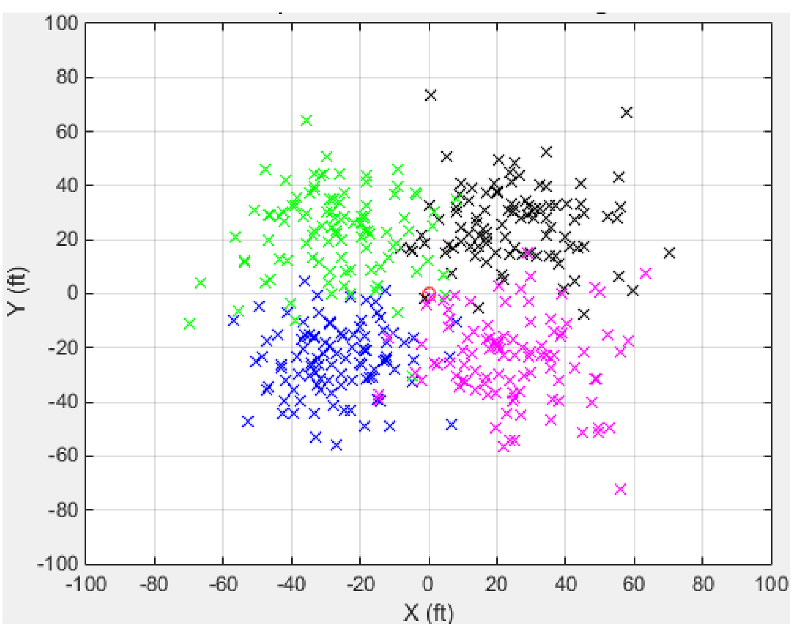

Figure 35. Multiple (4) weapon impact points using MC-simulation (100 iterations)

5. Effectiveness for Multiple Weapons

a. "Survivor Rule"

For multiple weapons, the effectiveness or PD of the target is a combination of the individual PDs for each weapon. This overall PD can be derived using the '*Survivor Rule*', which is explained as follows.

Logically, the probability of survival, PS_i, for each weapon in relation to the PD would be:

$$PS_i = 1 - PD_i \tag{2.26}$$

where '*i*' denotes the '*i*-th' number of weapon in multiple weapons, and by combining the PS for individual weapons, the following equation is used to obtain the overall PS:

$$PS = \prod_{i=1}^{n} \left(1 - PD_i\right) \tag{2.27}$$

where the notation '*n*' denotes the total number of weapons used. The overall PD for multiple weapons can now be represented by the following equation:

$$\begin{aligned} PD &= 1 - PS \\ &= 1 - \prod_{i=1}^{n} \left(1 - PD_i\right) \end{aligned} \tag{2.28}$$

Hence, by taking the individual single-weapon PD for multi-weapon salvos and putting them into equation (2.28), the overall PD for multiple weapons can be obtained using any of the three damage functions. For the purpose of explaining the effectiveness for multiple weapons, the number of weapons was assumed to be four (n = 4) for each of the three cases of damage functions in the following sections.

b. *LAM Effectiveness (Multiple Weapons)*

The range of effectiveness or probability of damage for one weapon (PD_i) in a multi-weapon salvo in a LAM is given by the inequality equation as:

$$0 < PD_i < 1 \tag{2.29}$$

Figure 36 illustrates the multi-weapon (4) impact points with their accompanying LAMs. In the figure, the lethal area of each weapon overlaps the target position at the center of the origin on the range-deflection ground plane.

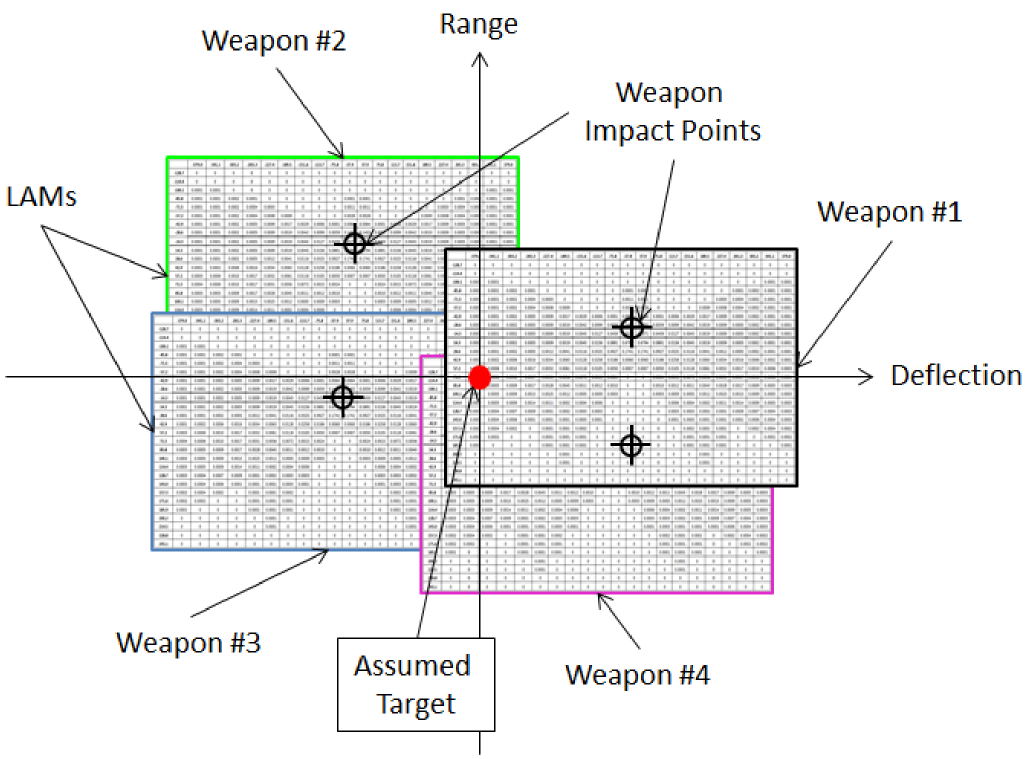

Figure 36. Representation of LAM matrix for multi-weapon (4) impact points

Based on the single weapon PD using the LAM for each corresponding weapon in the multi-weapon salvo within the LAM context, the overall PD is calculated using equation (2.28) of the 'Survivor Rule' where for four weapons,

$$PD = 1 - PS$$
$$= 1 - (1 - PD_1)(1 - PD_2)(1 - PD_3)(1 - PD_4) \qquad (2.30)$$

c. CDF Effectiveness (Multiple Weapons)

The range of effectiveness or probability of damage (PD_i) for one weapon in a multi-weapon salvo in a CDF is given by the following inequality equation:

$$0 < PD_i < 1 \qquad (2.31)$$

Figure 37 illustrates the four-weapon impact points around the target; the spread of the CDF is represented by the ellipses accompanying each weapon impact point. Note that the weapon effectiveness extends beyond the displayed elliptical boundaries for each weapon in the figure.

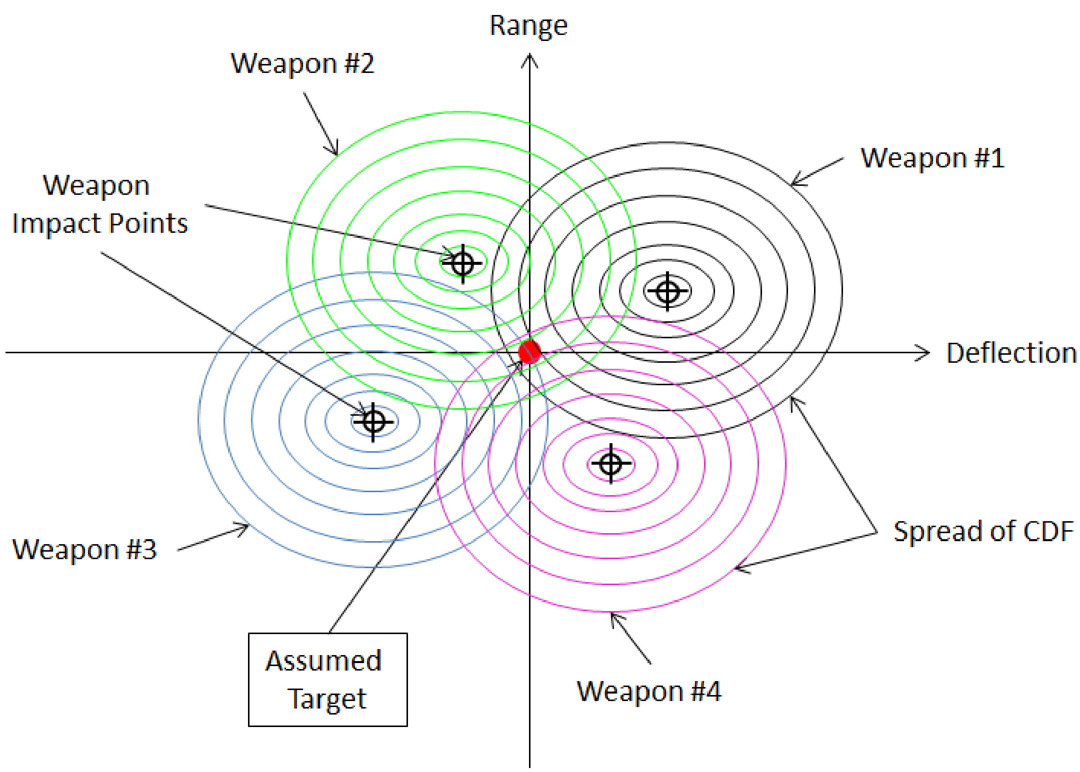

Figure 37. Representation of CDF damage matrix for multi-weapon (4) impact points

Based on the single weapon PD using the CDF for each corresponding weapon in the multi-weapon salvo, the overall PD is calculated using equation (2.28) of the 'Survivor Rule' where for four weapons,

$$\begin{aligned} PD &= 1 - PS \\ &= 1-(1-PD_1)(1-PD_2)(1-PD_3)(1-PD_4) \end{aligned} \tag{2.32}$$

d. RDF Effectiveness (Multiple Weapons)

Since the damage function is represented by the RCC, the effectiveness or PD_i for one weapon in a multi-weapon salvo obtained through the RDF is of the binary form:

$$PD_i = 1, 0 \tag{2.33}$$

Based on equation (2.28), the overall PD would be unity if at least one RCC lethal area falls over the target position on the range-deflection plane; otherwise, the overall PD would be zero. Figure 38 illustrates the scenario where at least one RCC for a four-weapon salvo encloses the target.

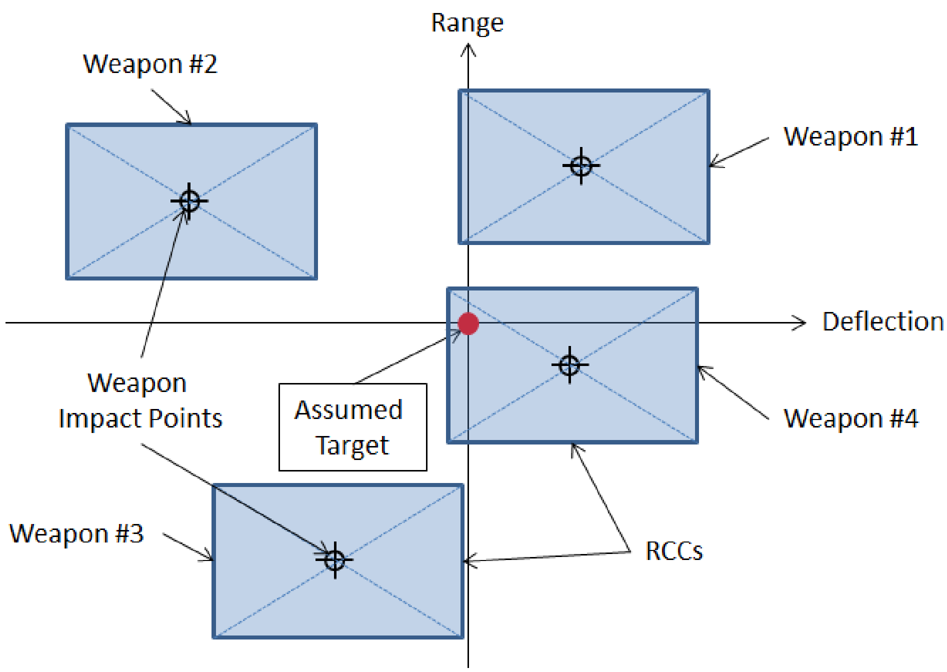

Figure 38. Example of at least one RCC enclosing target for RDF, overall PD=1

Based on the single weapon PD using the RDF for each corresponding weapon in the multi-weapon salvo, the overall PD is calculated using equation (2.28) of the 'Survivor Rule' where for four weapons,

$$\begin{aligned} PD &= 1 - PS \\ &= 1 - \left(1 - PD_1\right)\left(1 - PD_2\right)\left(1 - PD_3\right)\left(1 - PD_4\right) \end{aligned} \tag{2.34}$$

For the case shown above, $PD_1 = PD_2 = PD_3 = 0$, $PD_4 = 1$. Therefore, PD = 1.

THIS PAGE INTENTIONALLY LEFT BLANK

III. WEAPON EFFECTIVENESS SIMULATIONS FOR MULTIPLE WEAPONS

For comparison purpose, the Mean Area of Effectiveness, MAE_F, and impact angle, I, are kept constant at 2270ft^2 and 65° respectively through the three damage functions, wherever applicable.

A. AIMPOINT GENERATION

Traditionally, single weapons are aimed directly at the location of the target. It presents the highest probability of damage (PD) for a solitary weapon aimed at a unitary target without any errors. In reality, the impact point will more often than not, land at a point that is offset from the actual target location due to the influence of dependent and independent errors as illustrated in Figure 39.

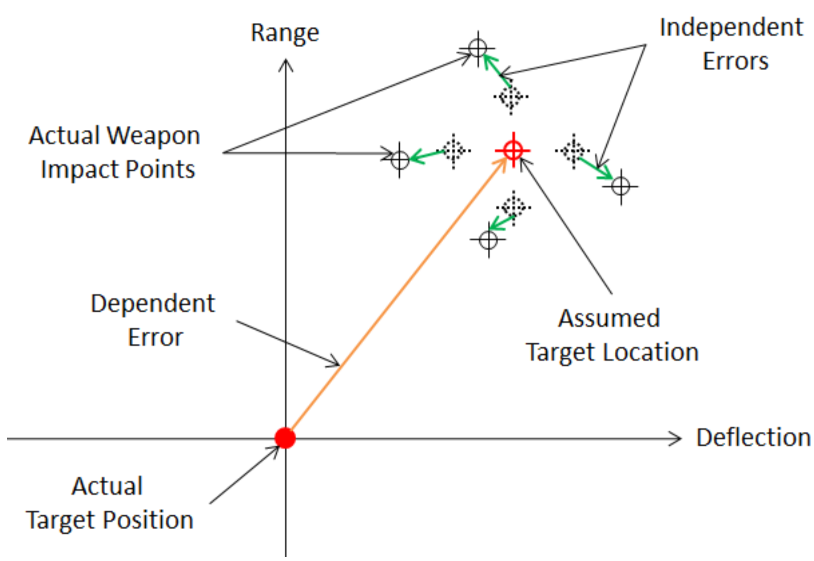

Figure 39. Multiple weapon impact points with dependent & independent errors

In other words, although coordinate-seeking weapons (CSW) are precise with small independent errors, the target coordinates may be poorly known which gives rise to a large dependent error. Since the target coordinates are not correct, all weapons will go to the wrong position precisely. Hence, there is a need to develop an aimpoint strategy to maximize the PD for a given *n* number of weapons.

1. Proposed Solution

By aiming all weapons at the assumed target location illustrated in Figure 40 and given a high enough dependent error, chances are that none of the weapon impact points would land at a position close enough to the actual target position to cause any significant damage on the target.

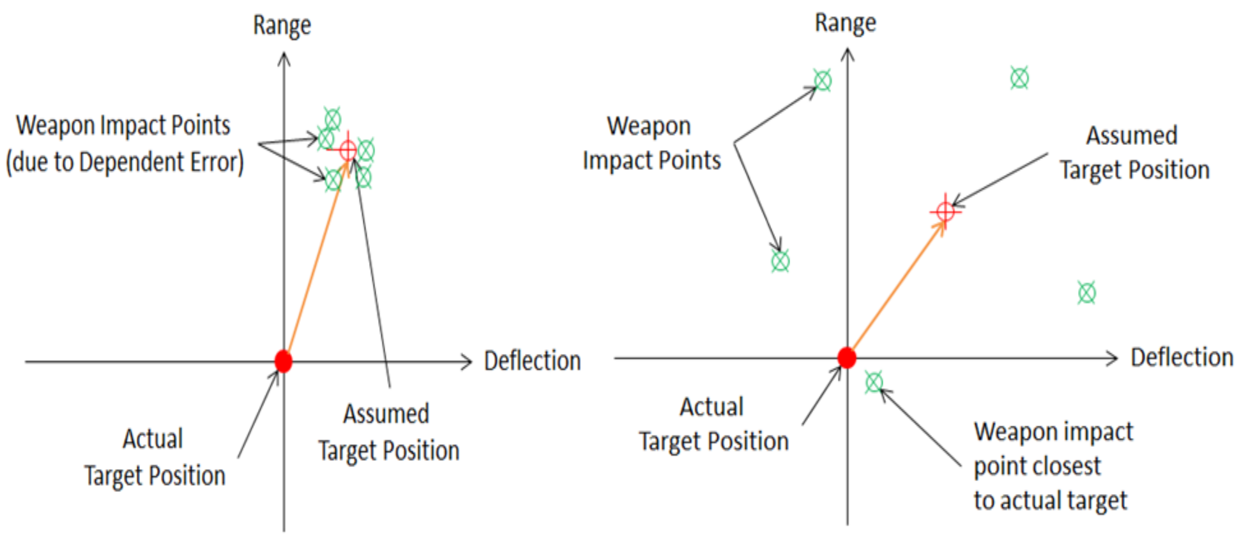

Figure 40. Weapon impact points aimed at and spread out around assumed target position

In the case of multiple weapons, weapon aimpoints that are spread around the assumed target are more likely to produce higher PDs on the actual target given the accumulation of the damage effects from multiple weapons as illustrated in Figure 40.

In addition, the spread of the weapon impact points around the assumed target position increases the probability that at least one weapon impact point would land close enough to the actual target location to cause substantial damage to the target. As such, the proposed solution is not to aim all weapons at the 'target,'[2] but to spread them out instead so that at least one of these weapons may be close enough to the actual target to cause more damage than all of the weapons precisely missing the target.

2. Arrangement of *n* Number of Weapons in a Circle ("All-Around")

Figure 41 and Figure 42 illustrate the arrangement of *n* number of weapons equally-spaced in a circle with aimpoint radius *r*, centered at the assumed target location. By virtue of the arrangement layout, the term '*all-around*' is used to denote the circular aimpoint patterns. One weapon is always on the positive deflection-axes as part of the aimpoint strategy.

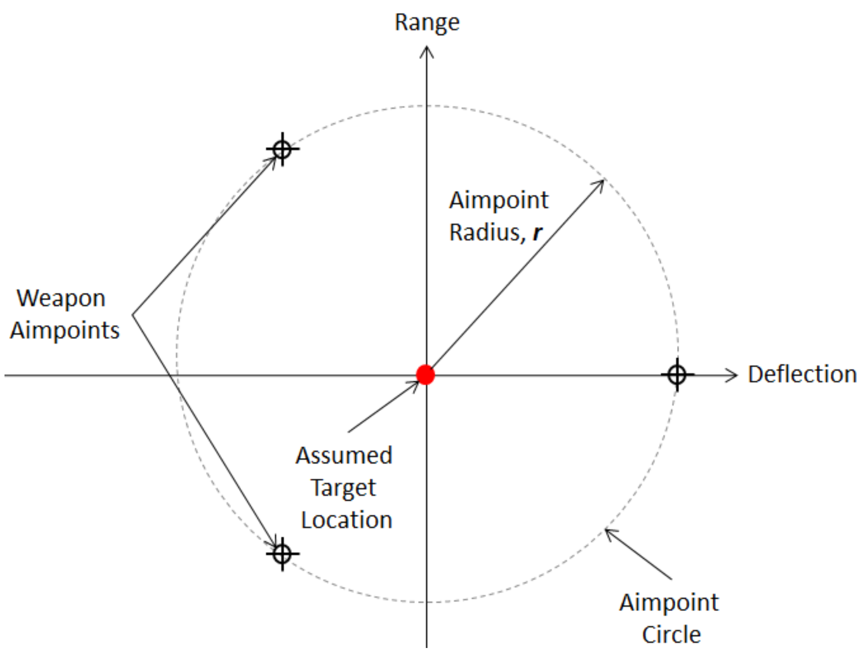

Figure 41. Example of three-weapon aimpoints arranged in a circle for $n = 3$

[2] Assumed target location.

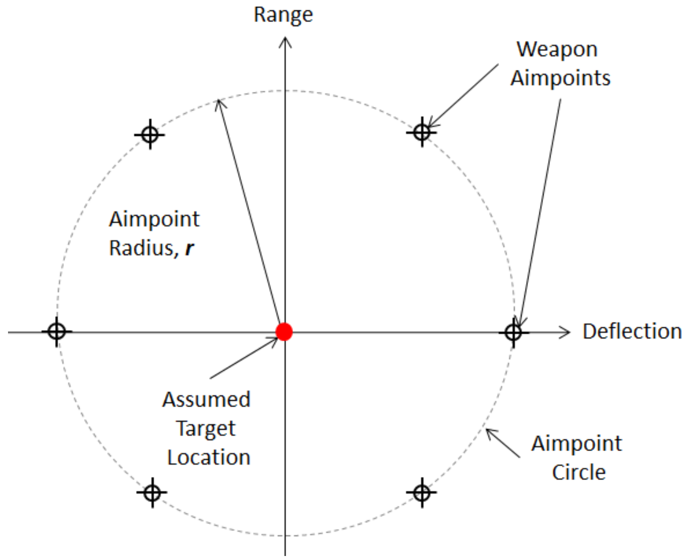

Figure 42. Example of six-weapon aimpoints arranged in a circle for *n* = 6

Subsequently, a MATLAB code was generated for the all-around aimpoint arrangement. Figure 43 presents this MATLAB-generated aimpoint plot of four weapons (cross) arranged in a circle of radius 25 centered at the assumed target location (circle). An example of the all-around weapon aimpoint generator program in MATLAB can be found in Appendix B.1.

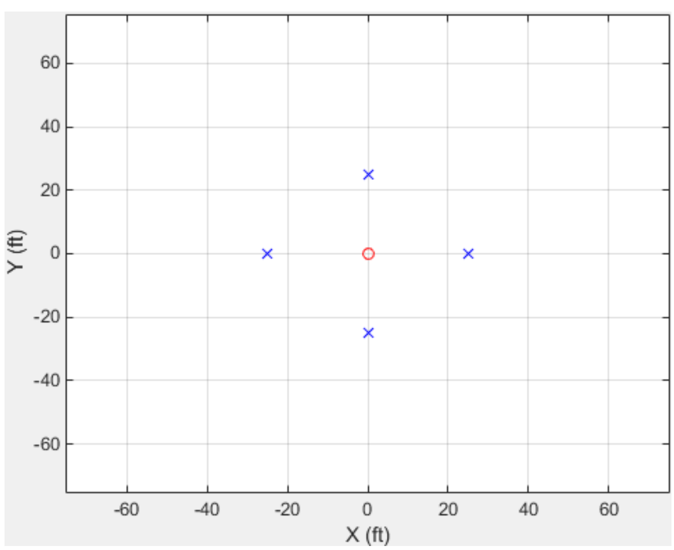

Figure 43. All-around weapon aimpoint arrangement for *n* = 4

3. Arrangement of One Weapon at Center for *n* Number of Weapons ("Centered")

However, it would also seem logical that at least one weapon out of the given *n* number of weapons is aimed at the assumed target location. As such, the arrangement in which one weapon is centered at the assumed target location while the rest (*n*-1) of the weapons are placed in a circle similar to the all-around arrangement is also considered. Figure 44 illustrates this '*centered*' aimpoint arrangement.

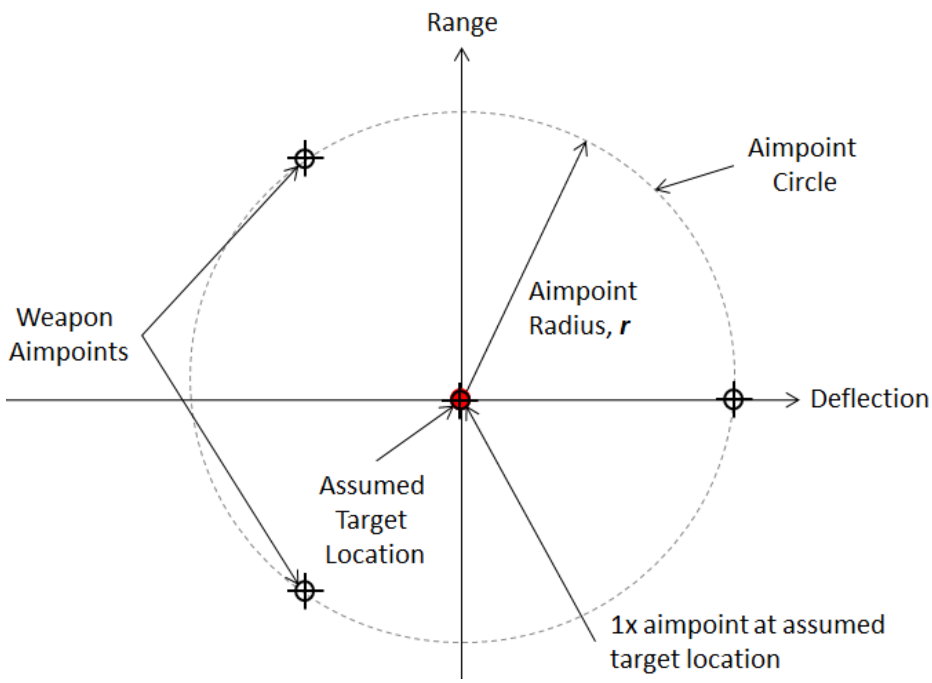

Figure 44. Example of one weapon aimpoint placed at and three weapons arranged in a circle centered at assumed target location for $n = 4$

Figure 45 illustrates the subsequent MATLAB-generated aimpoint plot of one weapon (cross) aimed at the assumed target and three weapons arranged in a circle centered at the assumed target location (circle). An example of the centered weapon aimpoint generator program in MATLAB can be found in Appendix B.2.

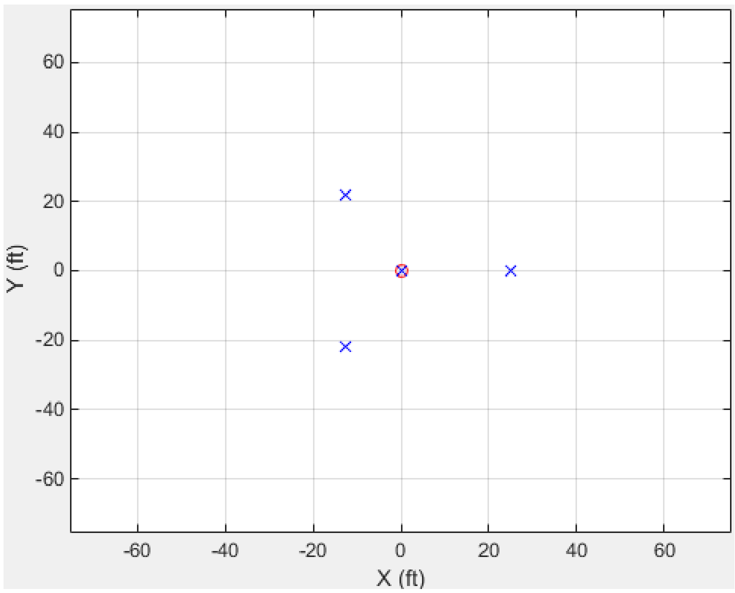

Figure 45. Centered weapon aimpoint arrangement for $n = 4$

B. TEST CASE: CARLTON DAMAGE FUNCTION (CDF) FOR N NUMBER OF WEAPONS

In the previous section, two weapon aimpoint arrangements were presented. By virtue of their different arrangements, it is quite certain that their PD would differ. As such, a test utilizing the CDF is used to differentiate the aimpoint arrangement that translates to a higher PD.

For a fair test, both cases utilized the same values of inputs. Table 1 summarizes the input values utilized for the test cases.

Table 1. Table of input values for test comparison

Table of Inputs	
Inputs	**Values**
Dependent Error (σ_{DEP})	30*ft*
Independent Error (σ_{INDEP})	5*ft*
Number of Weapons	4
Mean Area of Effectiveness, MAEF	2270ft^2
Impact Angle	65 degrees
Weapon Radii (Deflection)	*15.46ft*
Weapon Radii (Range)	23.36*ft*

1. Evaluate Centered vs. All-Around Weapon Aimpoint Arrangement

The aimpoint generation algorithm for both arrangements from the previous section was arranged into a Monte-Carlo simulation utilizing the CDF to generate the PD values at given a range of aimpoint radii. Figure 46 presents the test results for both weapon aimpoint arrangements and their simulated PD values against the aimpoint radii, *r*, ranging from 0 to 70ft.

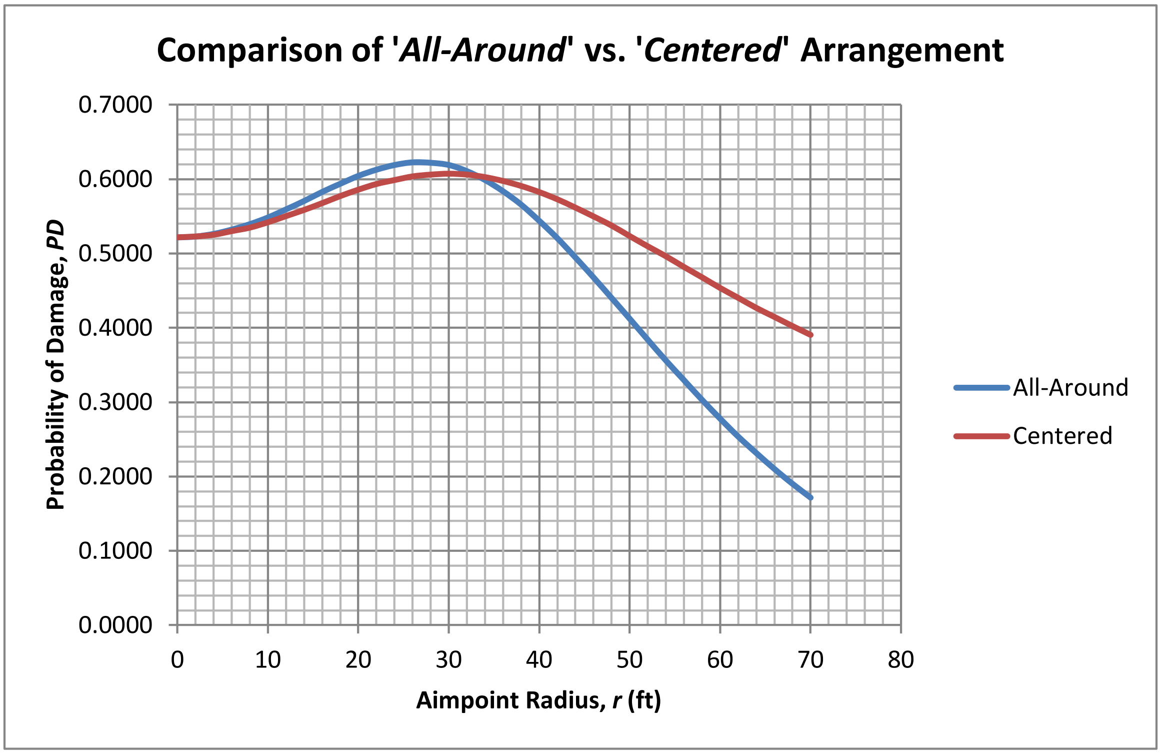

Figure 46. Comparison of all-around vs. centered aimpoint arrangement

2. Optimal Aimpoint Arrangement Selection

Based on the test data presented in Figure 46, the all-around weapon aimpoint arrangement has a marginally higher PD than the centered weapon aimpoint arrangement.

In addition, the smaller aimpoint radius taken to achieve a higher PD for the all-around arrangement than the centered arrangement aids in lowering any collateral damages.

As such, the all-around arrangement is taken to be the best aimpoint arrangement for multiple weapons. The all-around aimpoint arrangement is used as the default arrangement for the rest of this paper.

C. TEST CASE: OPTIMUM AIMPOINT RADIUS USING MC-SIMULATED CARLTON DAMAGE FUNCTION (CDF)

Based on the all-around aimpoint arrangement, the PD values are generated using the Monte-Carlo approach with input values listed in Table 1 for a multi-weapon scenario of four weapons using the CDF.

Figure 47 presents the PD values generated for the range of aimpoint radii from 0 to 70ft using the CDF. The MATLAB code used to generate this set of PD values using the CDF is found in Appendix C.1.

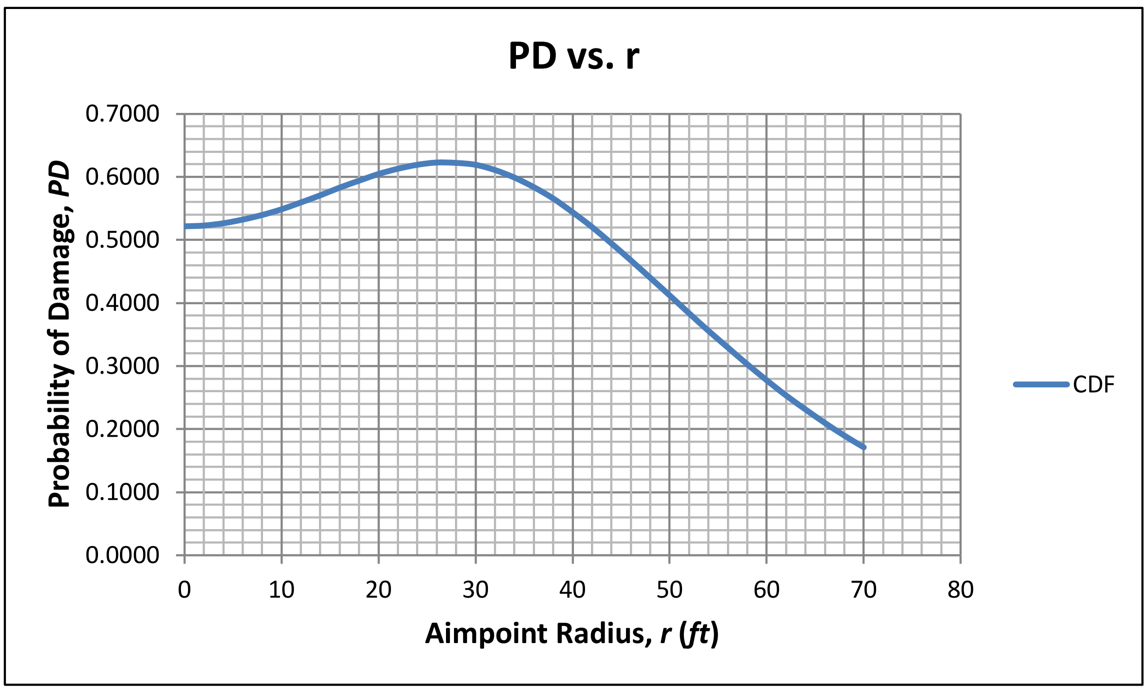

Figure 47. Effect of varying aimpoint radii on PD value using the CDF

From the graph in Figure 47, the PD value steadily increases from the aimpoint radius of 0ft until it hits a maximum of 0.6227 at 26ft. Thereafter, the PD value steadily decreases until it reaches a low of 0.17 at the aimpoint radius of 70ft. Note that when r = 0 for the case of all weapons aimed at the assumed target, the PD value obtained is 0.52. Since PD_{max} = 0.6227, the potential benefit of spreading out the weapons is an increase in the PD of 19.75%.

D. MC-SIMULATED RECTANGULAR DAMAGE FUNCTION (RDF)

For the same input values listed in Table 1 for a multi-weapon scenario of four weapons, Figure 48 presents the PD values generated for the range of aimpoint radii from 0 to 70ft using the RDF. The MATLAB code used to generate this set of PD values using the RDF is found in Appendix C.2.

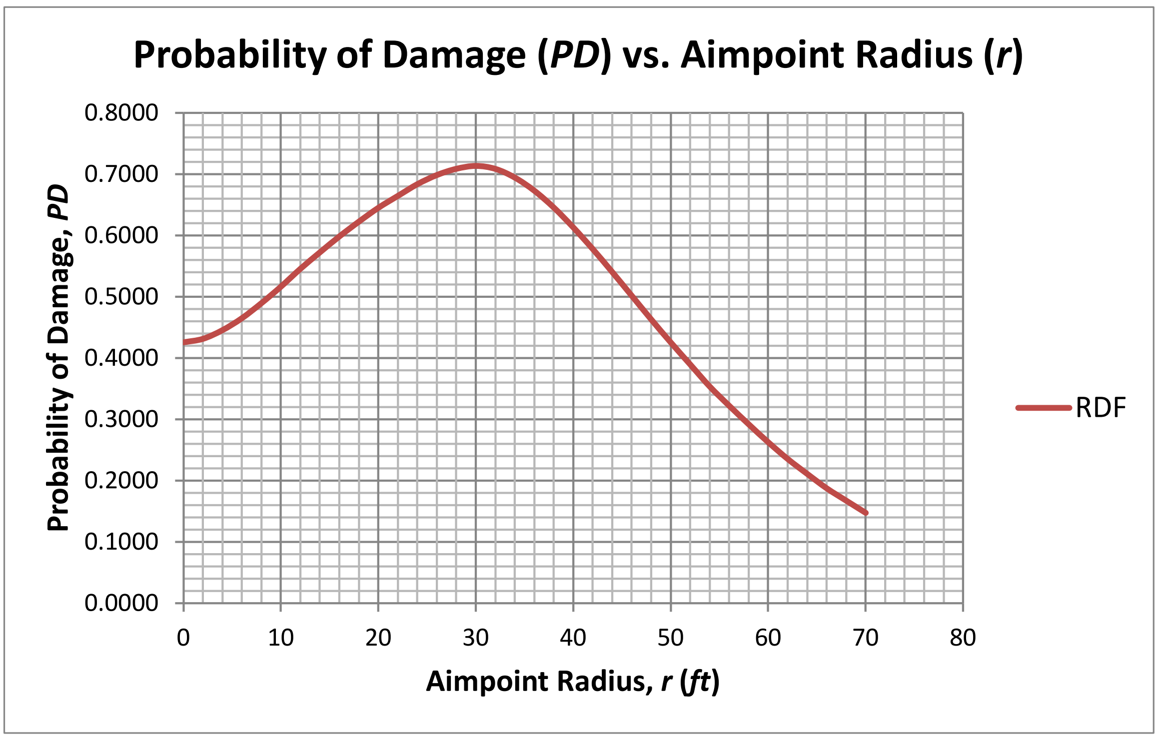

Figure 48. Effect of varying aimpoint radii on PD value using the RDF

From the graph in Figure 48, the PD value steadily increases from the aimpoint radius of 0ft until it hits a maximum value of 0.7127 at 30ft. Thereafter, the PD value steadily decreases until it reaches a low of approximately 0.15 at the aimpoint radius of 70ft. Again when $r = 0$ where we get the case of all weapons aimed at the assumed target, the PD value obtained is 0.42. Since $PD_{max} = 0.7127$, the potential benefit of spreading out the weapons is an increase in PD of 69.7%.

E. MC-SIMULATED LETHAL AREA MATRIX (LAM)

For the same input values listed in Table 1 for a four-weapon scenario, Figure 49 presents the PD values generated for the range of aimpoint radii from 0 to 70ft using the LAM. The MATLAB code used to generate this set of PD values using the LAM is found in Appendix C.3.

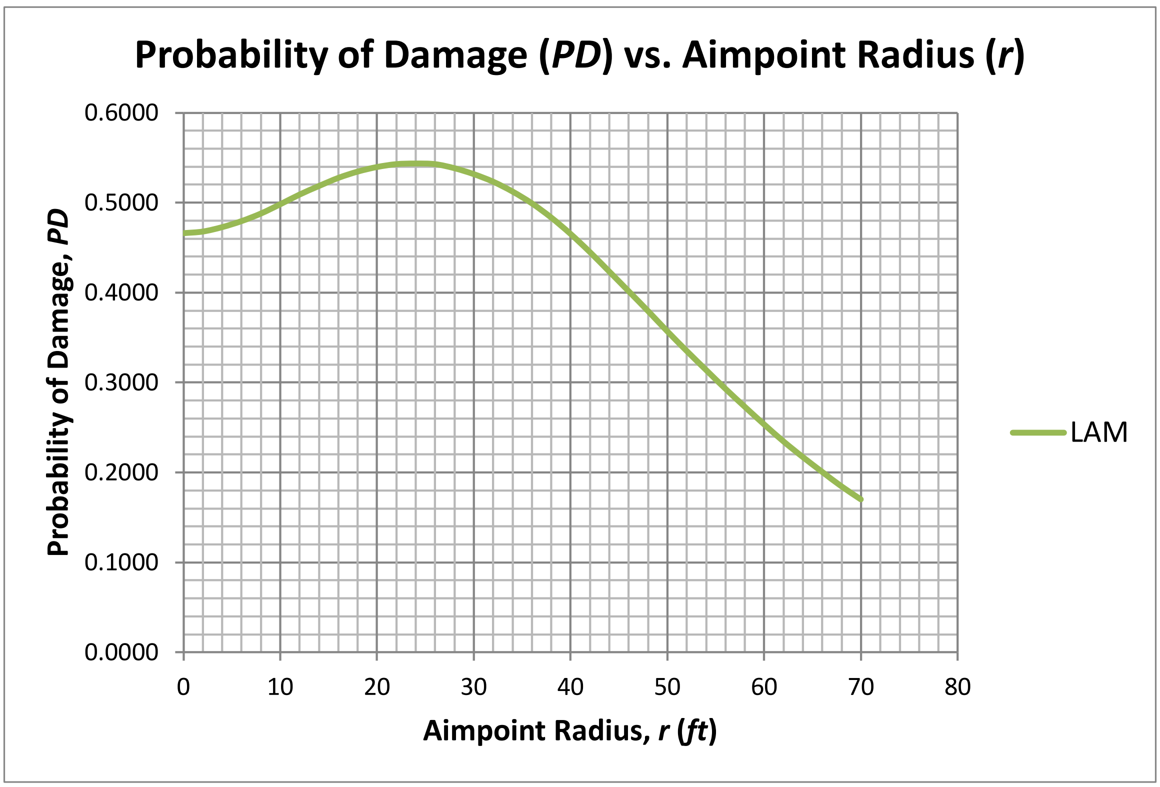

Figure 49. Effect of varying aimpoint radii on PD value using LAM

From the graph in Figure 49, the PD value steadily increases from the aimpoint radius of 0ft until it hits a maximum of 0.5435 at 24ft. Thereafter, the PD value steadily decreases until it reaches a low of 0.17 at the aimpoint radius of 70ft. Once more when *r* = 0 for the case of all weapons aimed at the assumed target, the PD value obtained is approximately 0.465. Since PD_{max} = 0.5435, the potential benefit of spreading out the weapons is an increase in the PD of 16.88%.

F. COMBINING PD VS. RADIUS PLOTS FOR ALL DAMAGE FUNCTIONS

For comparison purposes, the PD vs. *r* plots for the three damage functions are combined and presented in Figure 50.

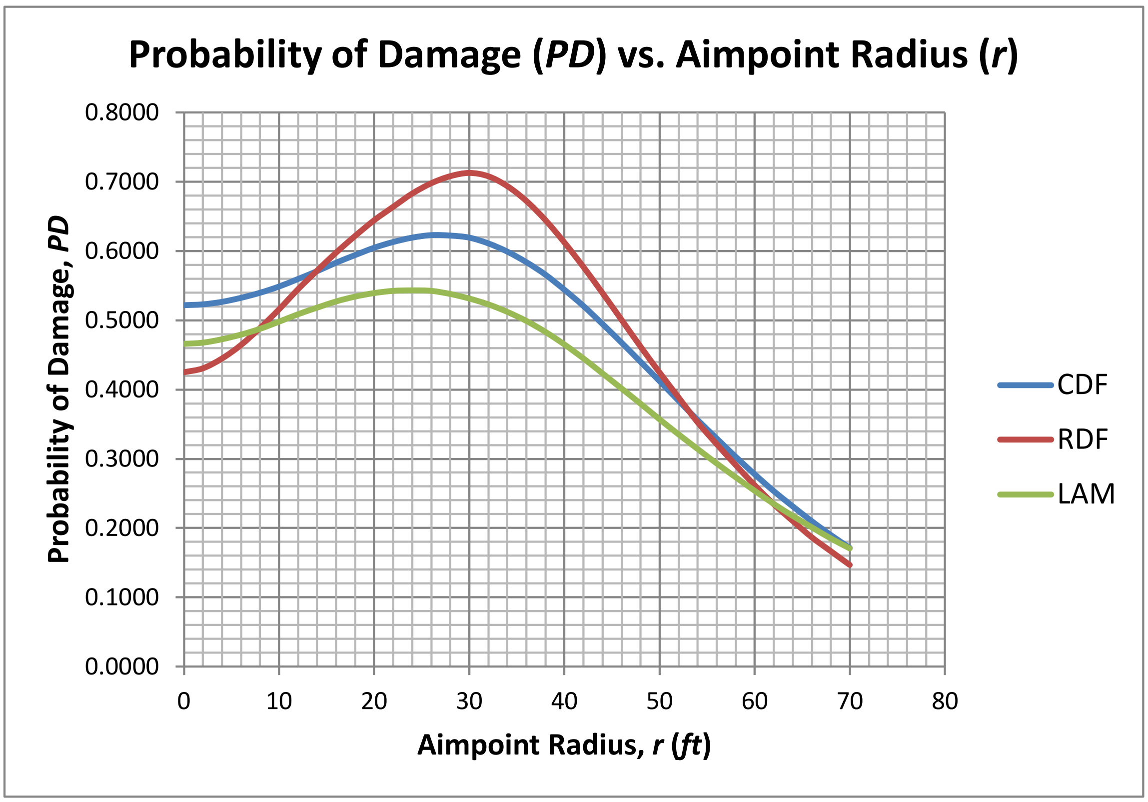

Figure 50. Comparison of PD values vs. *r* of the CDF, RDF and LAM

From Figure 50, where the aimpoint radius *r* = 0 for the case of all weapons aimed at the assumed target, the damage function that returns the highest PD value is the CDF, followed by the LAM and then the RDF. However, where the magnitude of the maximum PD obtained is concerned, the RDF returns the highest value, followed by the CDF and then the LAM.

In addition, the optimal aimpoint radii where the PD_{max} is returned for all three damage functions are different at 30ft, 26ft and 24ft for the CDF, RDF and LAM respectively. Table 2 presents the simulated PD_{max} values against their respective optimal aimpoint radii for each of the three damage functions of a four-weapon salvo.

Table 2. Maximum PD values (simulated) for multiple-weapons

Multiple Weapons		
Damage Function	max. PD	Optimal Aimpoint Radius
Rectangular	0.7127	30 ft
Carlton	0.6227	26 ft
Lethal Area Matrix	0.5435	24 ft

This observation matches the trend for the calculated PD1 values in Table 3.

Table 3. Single-weapon calculated PD1, from [4]

Single Weapon	
Damage Function	Calculated PD1
Rectangular	0.4655
Carlton	0.3901
Lethal Area Matrix	0.2467

THIS PAGE INTENTIONALLY LEFT BLANK

IV. WEAPON EFFECTIVENESS SIMULATIONS FOR VARYING DEPENDENT-INDEPENDENT ERROR RATIOS (*Q*)

A. INTRODUCTION

The program used to simulate the effectiveness for multiple weapons across a range of aimpoint radii for the damage functions utilizes the same set of input values as shown in Table 1. Based on this list, there are many combinations of inputs that can be further varied numerically with one another and their effects investigated based on the returned probability of damage (PD) values and optimal aimpoint radii within the simulation program.

However, due to time constraints within the scope of this thesis research, only one relationship between a pair of inputs was investigated, i.e. ratio of dependent error against independent error. With the exception of the dependent error, all other input variables were kept constant in order to observe the effects of this accuracy ratio.

Table 4 summarizes the list of input values that are either made variable or fixed for the purpose of the investigation.

Table 4. Summary of variable or fixed input values

Varying Dependent-Independent Errors	
Inputs	**Values**
Dependent Error (σ_{DEP})	**Variable**
Independent Error (σ_{INDEP})	*2ft*
Number of Weapons	4
Mean Area of Effectiveness, MAEF	2270*ft*^2
Impact Angle	65 degrees
Weapon Radii (Deflection)	*15.46ft*
Weapon Radii (Range)	23.36*ft*

The ratio of the dependent error to the independent error of a weapon is given by the variable Q, as shown in equation (4.1).

$$Q = \frac{\sigma_{DEP}}{\sigma_{INDEP}} \tag{4.1}$$

In this chapter, the effect of varying Q on the PD generated for varying aimpoint radii of n number of weapons across the three damage functions is investigated. The independent error was set at a constant value of 2, whilst the values of the dependent errors are varied in order to generate Q values of 1, 2, 5, 10 and 20. Table 5 gives an overview of the independent, dependent errors and subsequent Q values generated for the afore-mentioned investigation.

Table 5. Overview of independent, dependent errors and Q values

Independent Error (σ_{INDEP}, *ft*)	Dependent Errors (σ_{DEP}, *ft*)	Q Value
2	2	1
	4	2
	10	5
	20	10
	40	20

B. DEPENDENT-INDEPENDENT ERROR RATIO, *Q* = 1

Figure 51 presents the PD values generated for the range of aimpoint radii from 0 to 70ft of all three damage functions for the *Q* value of 1.

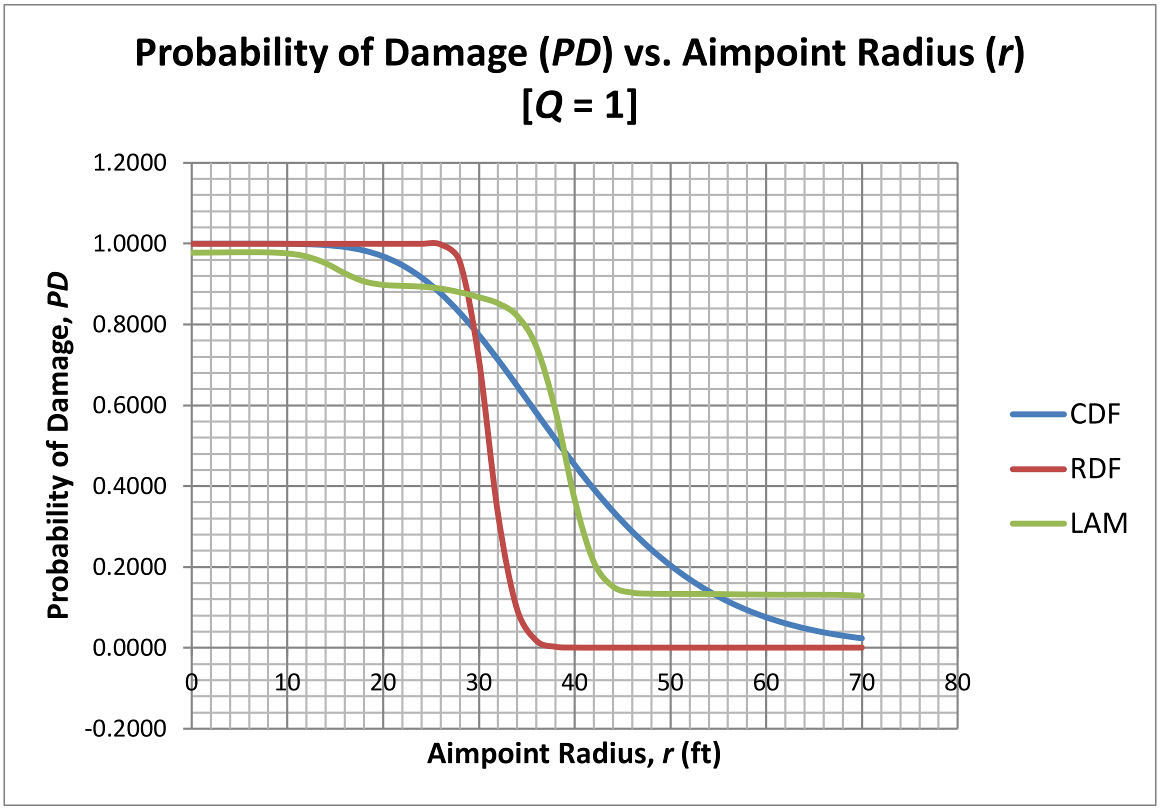

Figure 51. PD vs. aimpoint radii plot for (*Q* = 1)

Based on the graph in Figure 51 for equal magnitudes of both the dependent and independent errors, the PD values remain high (1.0 ≥ PD ≥ 0.9) for the range of aimpoint radii from 0 to 26ft. After that point, the PD values decrease steeply beyond the aimpoint radius of 26ft.

C. DEPENDENT-INDEPENDENT ERROR RATIO, *Q* = 2

Figure 52 presents the PD values generated for the range of aimpoint radii from 0 to 70ft of all three damage functions for the *Q* value of 2.

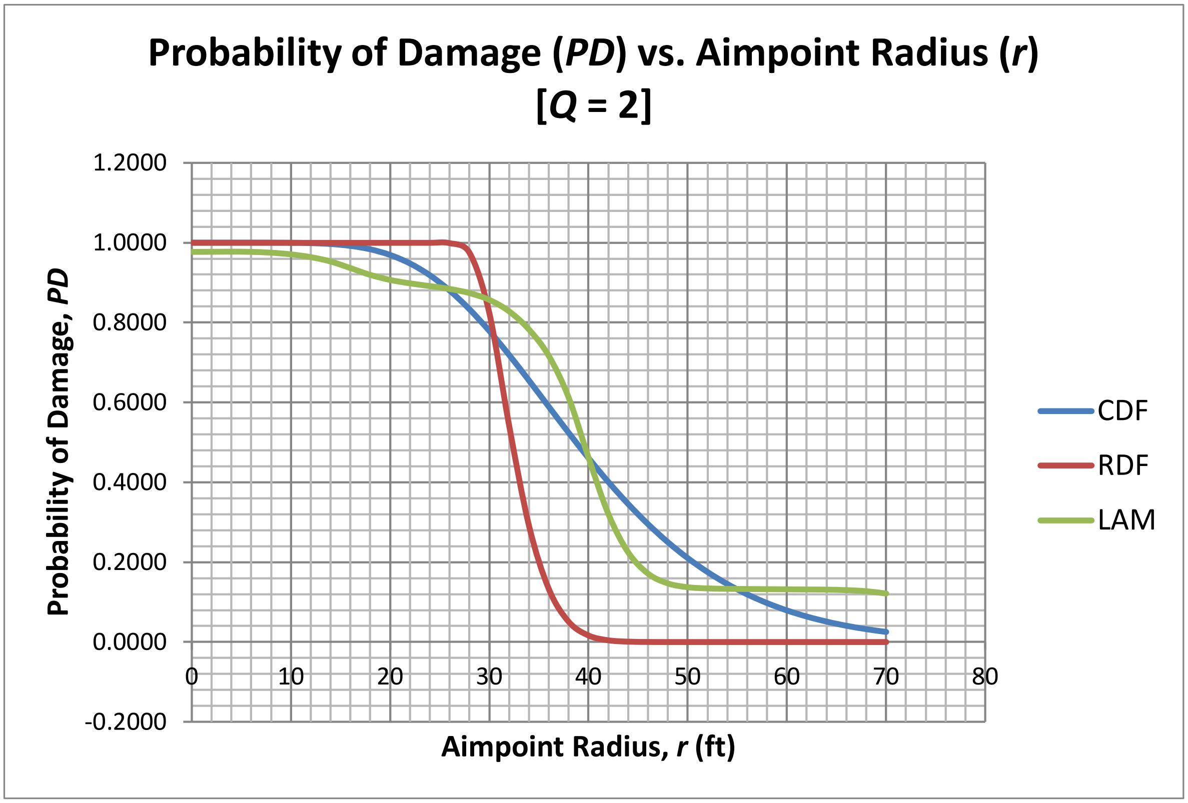

Figure 52. PD vs. aimpoint radii plot for (*Q* = 2)

From the graph in Figure 52 where the dependent error is twice that of the independent error, the PD value remains high (1.0 ≥ PD ≥ 0.9) for the range of aimpoint radii from 0 to 26ft. After that point, the PD values decrease steeply beyond the aimpoint radius of 26ft. This is similar to what was observed for the ratio *Q* = 1 in the previous section.

D. DEPENDENT-INDEPENDENT ERROR RATIO, *Q* = 5

Figure 53 presents the PD values generated for the range of aimpoint radii from 0 to 70ft of all three damage functions for the *Q* value of 5.

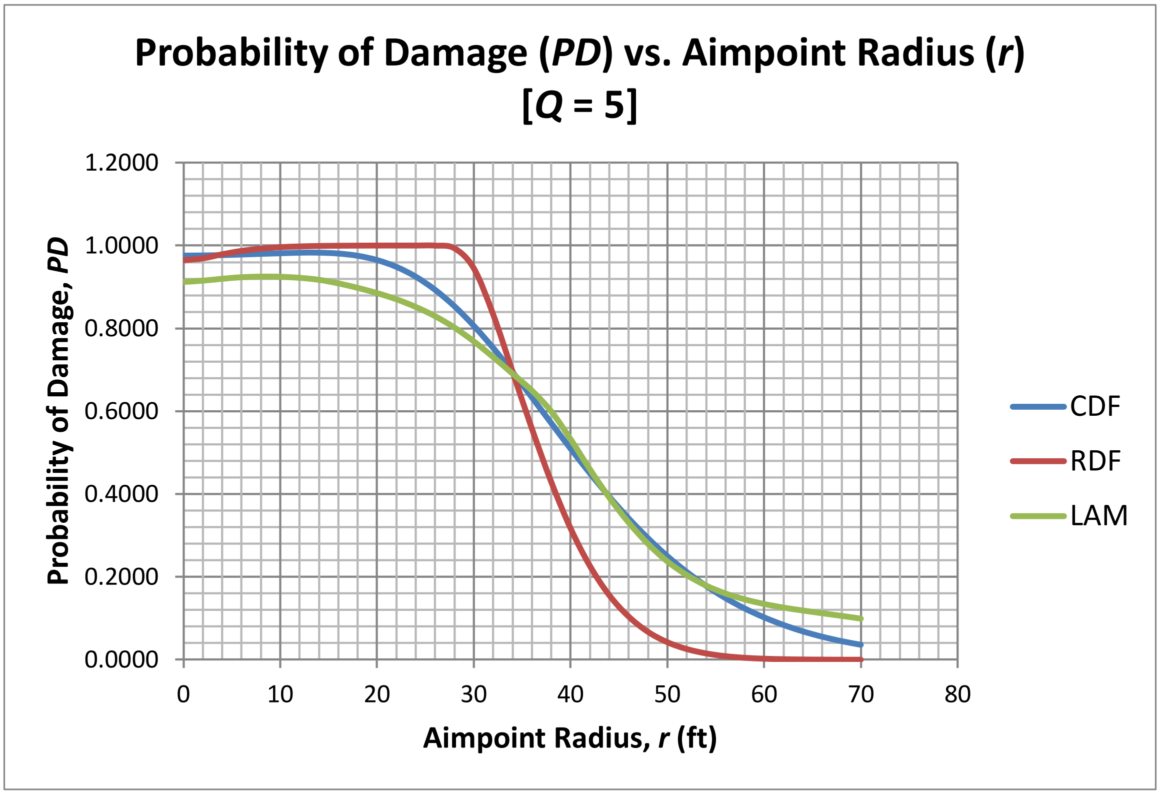

Figure 53. PD vs. aimpoint radii plot for ($Q = 5$)

Based on the graph in Figure 53, the initial PD values are close to but not as high as the initial values observed for both ratios $Q = 1$ and $Q = 2$. This is expected that as the dependent error increases, the weapon impact points would more likely fall at a coordinate farther away from the assumed target location. Hence, the PD values would start to decrease for a higher *Q* value.

E. DEPENDENT-INDEPENDENT ERROR RATIO, *Q* = 10

Figure 54 presents the PD values generated for the range of aimpoint radii from 0 to 70ft of all three damage functions for the *Q* value of 10.

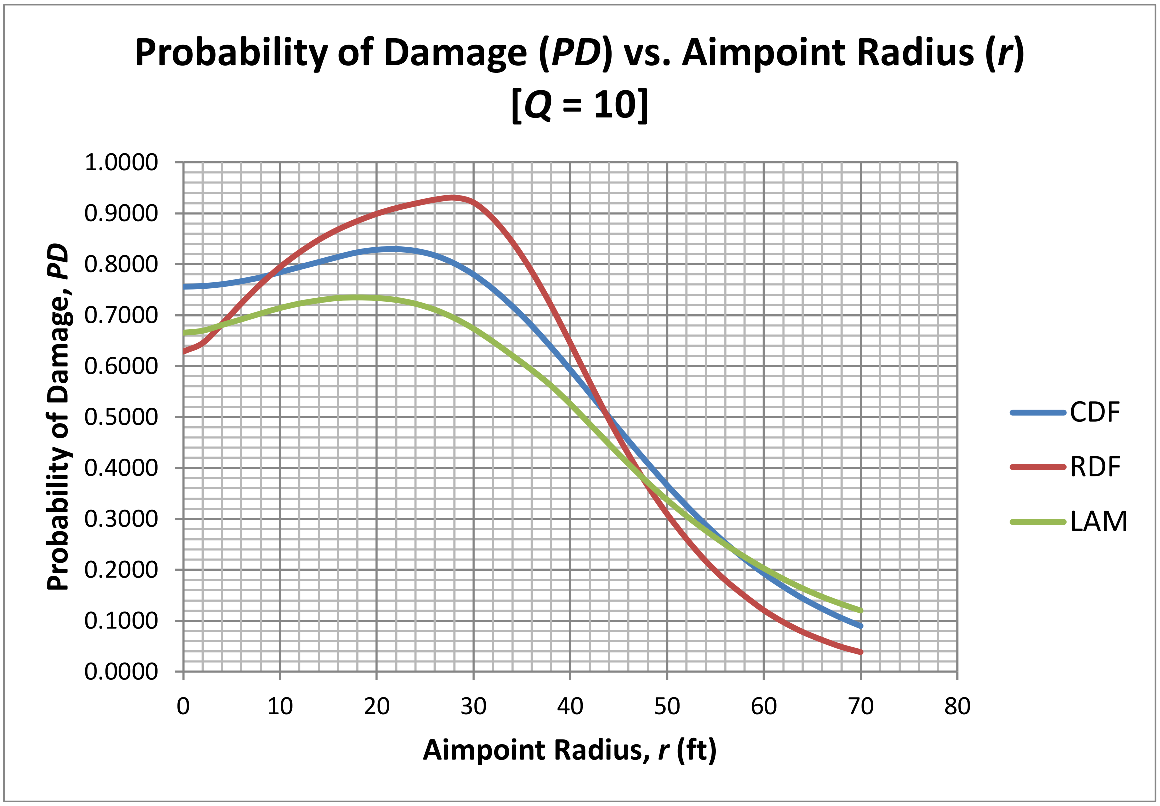

Figure 54. PD vs. aimpoint radii plot for (*Q* = 10)

From the graph in Figure 54, it is observed that there is a general decrease in the PD values across the damage functions as compared to the PD values obtained in ratios *Q* = 1, 2 and 5 for the same range of aimpoint radii. At this point, the high dependent errors (high ratio *Q*) meant that most of the weapon impact points would have landed at a location much farther away from the assumed target location as compared to the impact points for the previous three ratios. As such, their weapon effectiveness on the target would have been reduced. In addition, compared to the PD vs. *r* graphs obtained from the previous three Q values, there are now three distinctive peaks of the plots for all three damage functions.

F. DEPENDENT-INDEPENDENT ERROR RATIO, Q = 20

Figure 55 presents the PD values generated for the range of aimpoint radii from 0 to 70ft of all three damage functions for the Q value of 20.

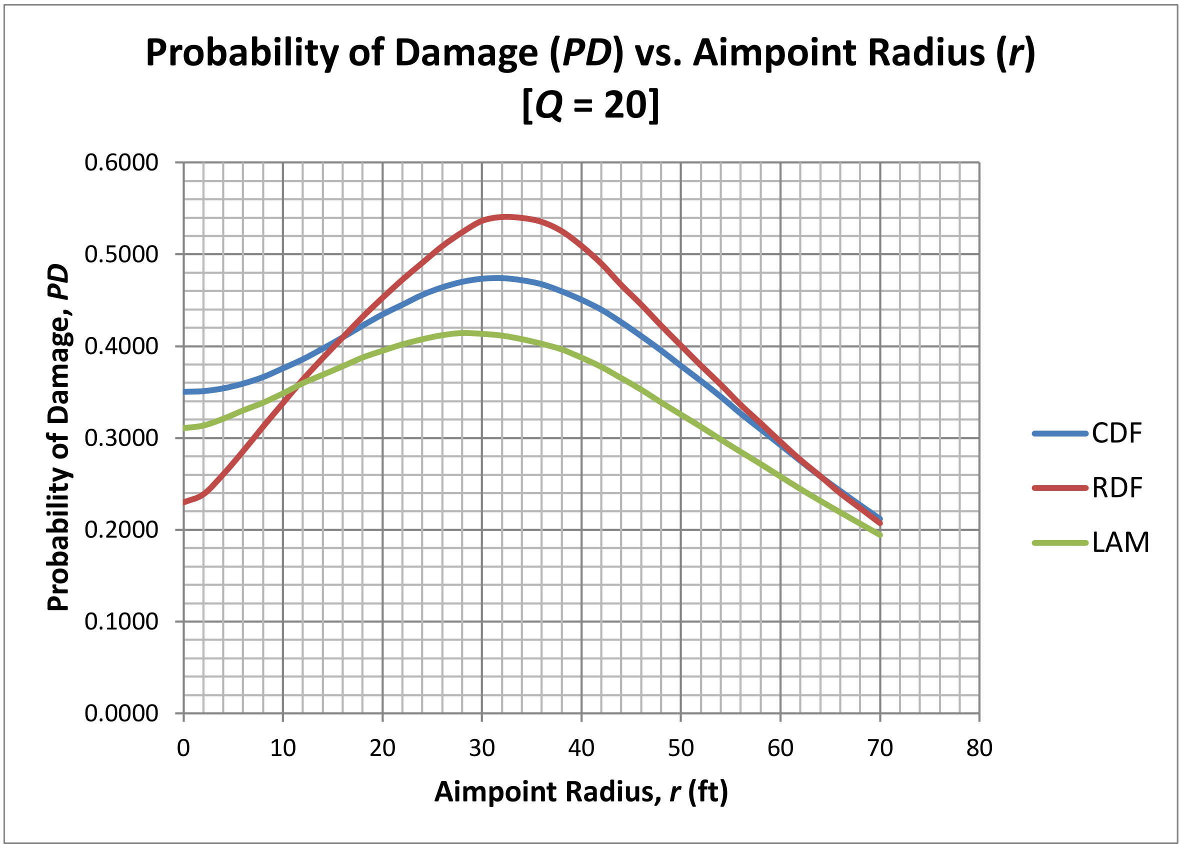

Figure 55. PD vs. aimpoint radii plot for (Q = 20)

The results reflected in the graph shown in Figure 55, further reinforces the observations and deductions made in the explanation for ratio Q = 10 that the high dependent errors result in the weapon salvo impacting the ground plane at even further distances from the assumed target. Therefore, the overall PD values are lowered across all damage functions. Additionally at the ratio Q = 20, the shape of the peaks are more defined for the plots across the range of aimpoint radii for each of the three damage function.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSION

- A tool has been produced to plan strike missions to optimize the pattern of aimpoints for a known number of weapons and weapon accuracies such as dependent and independent errors. Other inputs needed for the model include the mean area of effectiveness, impact angle, weapon radii and LAM.
- For the range of accuracies studied when the dependent error is small, the maximum probability of damage (PD_{max}) occurs at aimpoint radius $r = 0$. This is to be expected because the weapon impact points would land close enough to the actual target such that the full effects of these weapons would be experienced by the target. Therefore, there is no advantage in spreading out the weapons in cases of small dependent error.
- When the dependent error is large, spreading out the weapons increases the PD up to some optimum value of r, after which the PD decreases steadily, i.e. an optimum PD and corresponding r can be identified.
- The maximum PD and corresponding r depend on the damage function used, see Table 2. The Rectangular Damage Function obtained for the multiple-weapon salvo returns the highest PD value, followed by the Carlton Damage Function and then the Lethal Area Matrix.
- This trend matches the trend for single-weapon calculated PD1s as referenced from Table 3.
- The time taken to obtain the optimal aimpoint radius r result is longest for the damage function that has a higher fidelity. If speed or time to obtain a solution is an issue, then the approximate optimal aimpoint radius can be obtained using the simplest damage function.

B. RECOMMENDATIONS

- The program tool can be further developed into generating the PD values against a given range of aimpoint radii *r* for areas containing multiple, identical, unitary targets. This would provide a maximum Fractional Damage (FD) value.
- Further comparisons and investigations in varying the input values against each other can be explored for the weapon effectiveness simulations for multiple weapons across all three damage functions, e.g., variations in impact angle and mean area of effectiveness (fragmentation) for different target types.

APPENDIX. MATLAB CODES.

A. MONTE-CARLO SIMULATION FOR WEAPON IMPACT POINTS

1. Example of Monte-Carlo Simulation

```
%% Monte-Carlo Method Demonstrator

% This code plots the weapon impact points around the assumed target
using Monte Carlo simulations of (n=100) iterations for 1 weapon
without any errors.

%% Inputs:

sig_x = 15; % weapon accuracy in x-direction
sig_y = 15; % weapon accuracy in y-direction

aim_x = 0; % weapon aimpoint in x-direction
aim_y = 0; % weapon aimpoint in y-direction

%% Monte Carlo Simulation:

% number of Monte Carlo iterations
n = 100;
i = 1;

while i<n
    i = i+1;

    % sample the weapon accuracy in x-direction
    x1(i) = aim_x+sig_x*randn;
    % sample the weapon accuracy in y-direction
    y1(i) = aim_y+sig_y*randn;
end

%% Coordinate Plots of Target & Weapon:

tgt_x = [0];            % point target at origin
tgt_y = [0];            % point target at origin

figure(1)
plot(tgt_x,tgt_y,'ro',x1,y1,'bx')
grid on
xlabel('X (ft)')
ylabel('Y (ft)')
title('Weapon Distribution Around Target')
axis([-50, 50, -50, 50])
```

2. Single-Weapon Impact Point Generator

```
%% Single-Weapon MC-simulation Generator

% This code plots the weapon impact points around the assumed target
using Monte Carlo simulations of (n=100) iterations with dependent &
independent errors for 1 weapon.

%% Inputs:
aim_x = 0; % weapon aimpoint in x-direction
aim_y = 0; % weapon aimpoint in y-direction

sig_x = 5; % independent error in x-direction
sig_y = 5; % independent error in y-direction

sx = 15; % dependent error in x-direction
sy = 15; % dependent error in y-direction

%% Monte Carlo Simulation:

% number of Monte Carlo iterations
n = 100;
i = 1;

while i<n
    i = i+1;

    % sample the dependent errors in x-direction
    mu_x(i) = aim_x+sx*randn;
    % sample the dependent errors in y-direction
    mu_y(i) = aim_y+sy*randn;

    % sample the independent errors in x-direction
    x_imp(i) = mu_x(i)+sig_x*randn;
    % sample the independent errors in y-direction
    y_imp(i) = mu_y(i)+sig_y*randn;

%% Coordinate Plots of Target & Weapon:

tgt_x = [0]; % point target at origin
tgt_y = [0]; % point target at origin

figure(1)
plot(tgt_x,tgt_y,'ro',x,y,'bx')
grid on
xlabel('X (ft)')
ylabel('Y (ft)')
title('Weapon Distribution Around Target')
axis([-100, 100, -100, 100])
hold on

end
```

3. Multiple-Weapon Impact Points Generator

```
%% Multi-Weapon MC-simulation Generator

% This code plots the weapon impact points around the assumed target
using Monte Carlo simulations of (n=100) iterations with dependent &
independent errors for 4 weapons

%% Inputs:
aim_x = [25 -25 25 -25]; % weapon aimpoints in x-direction
aim_y = [25 25 -25 -25]; % weapon aimpoints in y-direction

sig_x = 5; % independent error in x-direction
sig_y = 5; % independent error in y-direction

sx = 15; % dependent error in x-direction
sy = 15; % dependent error in y-direction

%% Monte Carlo Simulation:

% number of Monte Carlo iterations
n = 100;
i = 1;

while i<n
    i = i+1;

    % sample the dependent errors in x-direction
    mu_x = aim_x+sx*randn;
    % sample the dependent errors in y-direction
    mu_y = aim_y+sy*randn;

    % sample the independent errors in x-direction
    x_imp = mu_x+sig_x*randn;
    % sample the independent errors in y-direction
    y_imp = mu_y+sig_y*randn;

end

%% Coordinate Plots of Target & Weapon:

tgt_x = [0]; % point target at origin
tgt_y = [0]; % point target at origin

figure(1)
plot(tgt_x,tgt_y,'ro',x_imp(1),y_imp(1),'bx',x_imp(2),y_imp(2),'kx',x_i
mp(3),y_imp(3),'gx',x_imp(4),y_imp(4),'mx')
grid on
xlabel('X (ft)')
ylabel('Y (ft)')
title('Weapon Distribution Around Target')
axis([-100, 100, -100, 100])
```

B. AIMPOINT GENERATION

1. *"All-Around"* Aimpoint Arrangement

```
% 'all-around' aimpoint arrangement:
theta = (2*pi)/n; % angular separation of aimpoints (rad)

i = 0;

while i<n

    i = i+1;
    d_theta = theta+i*theta;

    % resolve the x-coordinate aimpoint using theta & r
    aim_x(i) = r*cos(d_theta);

    % resolve the y-coordinate aimpoint using theta & r
    aim_y(i) = r*sin(d_theta);

end
```

2. *"Centered"* Aimpoint Arrangement

```
% 'centered' aimpoint arrangement:
theta = (2*pi)/(n-1); % angular separation of aimpoints (rad)

% sets 1st aimpoint at target position
aim_x(1) = 0;
aim_y(1) = 0;

% counters:
i = 1;

while i<n

    d_theta = theta+i*theta;
    i = i+1;

    % resolve the x-coordinate aimpoint using theta & r
    aim_x(i) = r*cos(d_theta);

    % resolve the y-coordinate aimpoint using theta & r
    aim_y(i) = r*sin(d_theta);

end
```

C. WEAPON EFFECTIVENESS FOR MULTIPLE WEAPONS

Figure 56 illustrates the general algorithm used to generate the weapon effectiveness for multiple weapons in the form of a flow diagram.

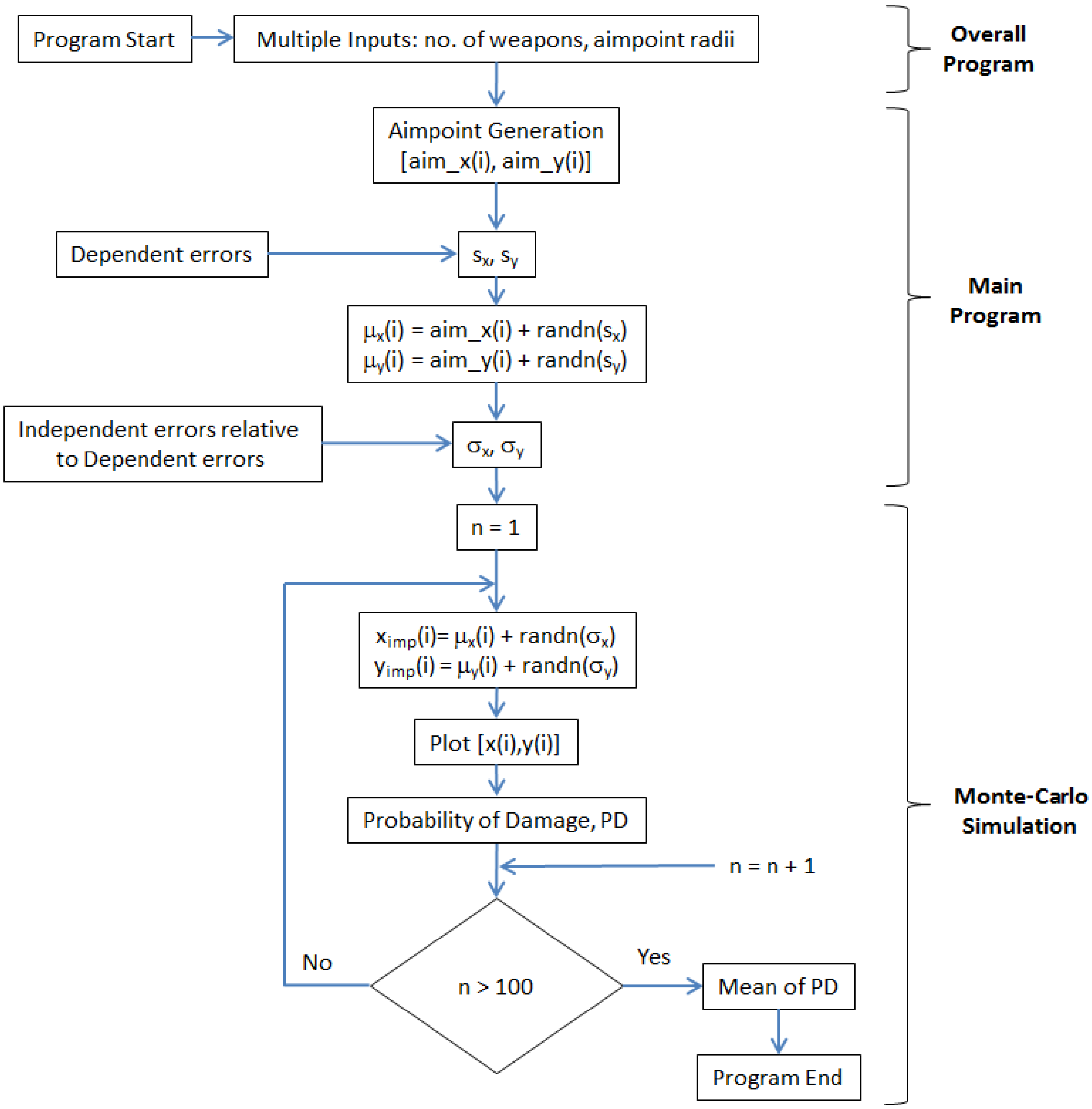

Figure 56. Overview of weapon effectiveness generation for multiple weapons

This algorithm is subsequently translated into MATLAB codes for each of the three damage functions. The 'overall program', 'main program' and 'Monte-Carlo simulation' sections of the flow diagram reflect how the code for each damage function is set up. These MATLAB codes are presented in the next three sections.

1. Carlton Damage Function

a. Overall Program File

```
clear all
close all
clc

n = 4;        % no. of weapons
r_min = 0;    % min. range of weapon
r_max = 70;   % max. range of weapon
r_step = 2;   % weapon range increment

for r = r_min:r_step:r_max

    [PD_mean] = CDF_Main1(n,r);

end
```

b. Main Program File

```
function[PD_mean] = CDF_Main1(n,r)
%% Carlton Damage Function for 'n' number of weapons

% Inputs:
sig_x = 5;  % [independent] errors in x-direction
sig_y = 5;  % [independent] errors in y-direction

sx = 30;  % [dependent] errors in x-direction
sy = 30;  % [dependent] errors in y-direction

MAEF = 2270;   % Mean Area of Effectiveness (Fragmentation)
I_deg = 65;                        % impact angle in (degrees)
I_rad = 65*pi/180;                 % impact angle in (radians)
a = max((1-0.8*cos(I_rad)),0.3);   % ratio of weapon radii
LET_p = 1.128*sqrt(MAEF*a);        % Effective Target Length (prime)
WET_p = LET_p/a;                   % Effective Target Width (prime)

% Weapon Radius in range direction, WRr or bx (ft)
bx = LET_p/(2*sqrt(2));
% Weapon Radius in deflection direction, WRd or by (ft)
by = WET_p/(2*sqrt(2));

bx_sq = bx^2;  % for ease of calculation in CDF_MCv1
by_sq = by^2;  % for ease of calculation in CDF_MCv1

% aimpoint generation:
theta = (2*pi)/n;      % angular separation of aimpoints (rad)

i = 0;                 % initial value of (i), for counter

while i<n
```

```
    i = i+1;
    d_theta = theta+i*theta;

    aim_x(i) = r*cos(d_theta);  % resolve the x-coordinate aimpoint
using theta & r
    aim_y(i) = r*sin(d_theta);  % resolve the y-coordinate aimpoint
using theta & r

end

% automation loop (t):
uv_counter = 1000000;
t = 0;

while t<uv_counter  % iteration of 'uv_counter'
   t=t+1;

% weapon error offsets (offset from aimpoint):
u = randn*sx;  % randomize [dependent] errors in x-direction, u
v = randn*sy;  % randomize [dependent] errors in y-direction, v

mu_x = aim_x+u;  % [independent] errors of aimpoint in x-direction
mu_y = aim_y+v;  % [independent] errors of aimpoint in y-direction

% call for Monte Carlo (MC) function:
[PD_total,x,y] = CDF_MCv1(mu_x,mu_y,sig_x,sig_y,bx_sq,by_sq);

PD(t) = PD_total;

end

PD_mean = mean(PD)

fprintf('The Probability of Damage at a weapon range of %2.1f (ft) is:
%2.4f\n', r, PD_mean)
```

c. *Monte-Carlo Simulation File*

```
function[PD_total,x,y] = CDF_MCv1(mu_x,mu_y,sig_x,sig_y,bx_sq,by_sq)

% Monte Carlo Simulation:

n = 100;  % number of Monte Carlo iterations
i = 0;

n_weap = length(mu_x);  % number of weapons
PS = 1;

while i<n                % this 'while' loop is the main overall loop
    i = i+1;             % counter for number of iterations

        j = 0;           % set (j) counter to zero
        while j<n_weap   % 'while' loop counts for n number of weapons
        j=j+1;           % counter for number of weapons up to (n) times

        % Carlton Damage Function for (j) weapon:

        % sample the input random variable 'x(j)'
        x(j) = mu_x(j)+sig_x*randn;
        % sample the input random variable 'y(j)'
        y(j) = mu_y(j)+sig_y*randn;

        % Carlton (Gaussian) Damage Function for (j) weapon
        PD(j) = exp(-0.5*(((x(j)^2)/(bx_sq))+((y(j)^2)/(by_sq))));

        PS = PS*(1-PD(j));  % 'Survivor Rule'

        end

    PD_iter(i) = 1-PS;
    PS = 1;                 % Reset PS
end

PD_total = mean(PD_iter);
```

2. Rectangular Damage Function

a. Overall Program File

```
clear all
close all
clc

n = 4;        % no. of weapons
r_min = 0;    % min. range of weapon
r_max = 70;   % max. range of weapon
r_step = 2;   % weapon range increment

for r = r_min:r_step:r_max

    [PD_mean] = RDF_Main1(n,r);

end
```

b. Main Program File

```
function[PD_mean] = RDF_Main1(n,r)
%% Rectangular Damage Function for 'n' number of weapons

% Inputs:
sig_x = 5;  % [independent] errors in x-direction
sig_y = 5;  % [independent] errors in y-direction

sx = 30;    % [dependent] errors in x-direction
sy = 30;    % [dependent] errors in y-direction

MAEf = 2270;   % Mean Area of Effectiveness (Fragmentation)
I_deg = 65;                        % impact angle in (degrees)
I_rad = 65*pi/180;                 % impact angle in (radians)
a = max((1-0.8*cos(I_rad)),0.3);  % ratio of weapon radii
LET = sqrt(MAEf*a);                % Effective Target Length
WET = LET/a;                       % Effective Target Width

% aimpoint generation:
theta = (2*pi)/n;      % angular separation of aimpoints (rad)

i = 0;                 % initial value of (i), for counter

while i<n

    i = i+1;
    d_theta = theta+i*theta;

    % resolve the x-coordinate aimpoint using theta & r
    aim_x(i) = r*cos(d_theta);
    % resolve the y-coordinate aimpoint using theta & r
    aim_y(i) = r*sin(d_theta);
```

```
end

% automation loop (t):
uv_counter = 1000000;
t = 0;

while t<uv_counter    % iteration of 'uv_counter'
   t=t+1;

% weapon error offsets (offset from aimpoint):
u = randn*sx;  % randomize [dependent] errors in x-direction, u
v = randn*sy;  % randomize [dependent] errors in y-direction, v

mu_x = aim_x+u;  % [independent] errors of aimpoint in x-direction
mu_y = aim_y+v;  % [independent] errors of aimpoint in y-direction

% call for Monte Carlo (MC) function:
[PD_total] = RDF_MCv1(mu_x,mu_y,sig_x,sig_y,LET,WET);

PD(t) = PD_total;

end

PD_mean = mean(PD)

fprintf('The Probability of Damage at a weapon range of %2.1f (ft) is:
%2.4f\n', r, PD_mean)
```

c. Monte-Carlo Simulation File

```
function[PD_total] = RDF_MCv1(mu_x,mu_y,sig_x,sig_y,LET,WET)

% Monte Carlo Simulation:
n = 100;                % number of Monte Carlo iterations
i = 0;

n_weap = length(mu_x);  % number of weapons
PS = 1;

while i<n               % this 'while' loop is the main overall loop
    i = i+1;            % counter for number of iterations

        j = 0;          % set (j) counter to zero
        while j<n_weap  % 'while' loop counts for n number of weapons
        j=j+1;          % counter for number of weapons up to (n) times

        % sample the input random variable 'x(j)'
        x(j) = mu_x(j)+sig_x*randn;
        % sample the input random variable 'y(j)'
        y(j) = mu_y(j)+sig_y*randn;

        % Rectangular Cookie-Cutter Damage Function for (j) weapon:

            if (abs(x(j))>LET/2)
                Pkx = 0;
            else Pkx = 1;
            end

            if (abs(y(j))>WET/2)
                Pky = 0;
            else Pky = 1;
            end

            PD = Pkx*Pky;

            PS = PS*(1-PD);  % 'Survivor Rule'

        end

    PD_iter(i) = 1-PS;
    PS = 1;                  % Reset PS
end

PD_total = mean(PD_iter);
```

3. Lethal Area Matrix

a. Overall Program File

```
clear all
close all
clc

%% Lethal Area Matrix (LAM) with 26 rows x 20 columns (full-matrix)
damage_matrix =
[0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
0      0      0      0      ;
                 0      0      0      0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      ;
                 0.0001 0.0001 0      0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0001 0      0      0      0      0      0.0001 0.0001 0      0
0      0      0      0.0001 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0004 0.0003 0      0      0      0      0.0011 0.0011 0      0
0      0      0.0003 0.0004 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0004 0.0008 0.0009 0      0      0      0.0028 0.0028 0      0
0      0.0009 0.0008 0.0004 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0005 0.0009 0.0017 0.0029 0.0006 0.0001 0.0064 0.0064 0.0001 0.0006
0.0029 0.0017 0.0009 0.0005 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0005 0.0009 0.0019 0.0042 0.0099 0.0059 0.1402 0.1402 0.0059 0.0099
0.0042 0.0019 0.0009 0.0005 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0005 0.0009 0.0019 0.0045 0.0127 0.0459 0.5571 0.5571 0.0459 0.0127
0.0045 0.0019 0.0009 0.0005 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0005 0.0009 0.0019 0.0045 0.0156 0.0891 0.6794 0.6794 0.0891 0.0156
0.0045 0.0019 0.0009 0.0005 0.0002 0.0001 0.0001 ;
                 0.0001 0.0001 0.0002 0.0005 0.0012 0.0041 0.0116 0.0325 0.0927 0.1741 0.1741 0.0927 0.0325
0.0116 0.0041 0.0012 0.0005 0.0002 0.0001 0.0001 ;
                 0.0001 0.0002 0.0006 0.0016 0.0034 0.0063 0.0128 0.0258 0.0186 0.0060 0.0060 0.0186 0.0258
0.0128 0.0063 0.0034 0.0016 0.0006 0.0002 0.0001 ;
                 0.0003 0.0006 0.0010 0.0017 0.0032 0.0061 0.0118 0.0105 0.0050 0.0007 0.0007 0.0050 0.0105
0.0118 0.0061 0.0032 0.0017 0.0010 0.0006 0.0003 ;
```

```
                 0.0004 0.0006 0.0010 0.0017 0.0031 0.0056 0.0072 0.0015 0.0024 0      0      0.0024 0.0015
0.0072 0.0056 0.0031 0.0017 0.0010 0.0006 0.0004 ;
                 0.0003 0.0005 0.0009 0.0017 0.0028 0.0045 0.0011 0.0012 0.0010 0      0      0.0010 0.0012
0.0011 0.0045 0.0028 0.0017 0.0009 0.0005 0.0003 ;
                 0.0003 0.0005 0.0009 0.0015 0.0025 0.0012 0.0005 0.0009 0.0003 0      0      0.0003 0.0009
0.0005 0.0012 0.0025 0.0015 0.0009 0.0005 0.0003 ;
                 0.0003 0.0005 0.0009 0.0014 0.0011 0.0002 0.0004 0.0006 0      0      0      0      0.0006
0.0004 0.0002 0.0011 0.0014 0.0009 0.0005 0.0003 ;
                 0.0003 0.0004 0.0007 0.0009 0.0001 0.0002 0.0003 0.0003 0      0      0      0      0.0003
0.0003 0.0002 0.0001 0.0009 0.0007 0.0004 0.0003 ;
                 0.0003 0.0004 0.0006 0.0001 0.0001 0.0001 0.0003 0.0001 0      0      0      0      0.0001
0.0003 0.0001 0.0001 0.0001 0.0006 0.0004 0.0003 ;
                 0.0002 0.0004 0.0002 0      0.0001 0.0001 0.0002 0      0      0      0      0      0
0.0002 0.0001 0.0001 0      0.0002 0.0004 0.0002 ;
                 0.0002 0.0002 0      0      0.0001 0.0001 0.0001 0      0      0      0      0      0
0.0001 0.0001 0.0001 0      0      0.0002 0.0002 ;
                 0.0002 0.0002 0      0      0.0001 0.0001 0.0001 0      0      0      0      0      0
0.0001 0.0001 0.0001 0      0      0      0.0001 ;
                 0      0      0      0      0      0.0001 0      0      0      0      0      0      0
0      0.0001 0      0      0      0      0      ;
                 0      0      0      0      0      0.0001 0      0      0      0      0      0      0
0      0.0001 0      0      0      0      0      ;
                 0      0      0      0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      ;
                 0      0      0      0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0     ];


% area of each cell:
cell_width = 37.9;                      % length of cell in Deflection direction
cell_height = 14.3;                     % length of cell in Range direction
cell_area = cell_width*cell_height;     % each cell area of Lethal Area Matrix

% Mean Area of Effectiveness (Fragmentation), MAEF, calculation:
MAEF = sum(damage_matrix*cell_area);
MAEF_tot = sum(MAEF);
fprintf('The overall Mean Area of Effectiveness (Fragmentation) is %.4f square feet. \n\n', MAEF_tot)
```

```
n = 4;        % no. of weapons

r_min = 0;    % min. range of weapon
r_max = 70;   % max. range of weapon
r_step = 2;   % weapon range increment

for r = r_min:r_step:r_max

    [PD_mean] = LAM_Main3(damage_matrix,cell_width,cell_height,n,r);

end
```

b. Main Program File

```
function[PD_mean] = LAM_Main3(damage_matrix,cell_width,cell_height,n,r)

% Inputs:
sig_x = 5;  % [independent] errors in x-direction
sig_y = 5;  % [independent] errors in y-direction

sx = 30;    % [dependent] errors in x-direction
sy = 30;    % [dependent] errors in y-direction

% damage matrix offsets:

% base point in deflection direction is offset by 10 columns to the
right
X_offset = 10;
% base point in range direction is offset by 9 rows down
Y_offset = 9;

% aimpoint calculations:
theta = (2*pi)/n;  % angular separation of aimpoints (rad)

i = 0;             % initial value of (i), for counter

while i<n

    i = i+1;
    d_theta = theta+i*theta;

    % resolve the x-coordinate aimpoint using theta & r
    aim_x(i) = r*cos(d_theta);
    % resolve the y-coordinate aimpoint using theta & r
    aim_y(i) = r*sin(d_theta);

end

% automation loop (t):
uv_counter = 1000000;
t = 0;

while t<uv_counter  % iteration of 'uv_counter'
   t=t+1;

% weapon error offsets (offset from aimpoint):
u = randn*sx;   % randomize [dependent] errors in x-direction, u
v = randn*sy;   % randomize [dependent] errors in y-direction, v

% randomize [independent] errors of aimpoint in x-direction
mu_x = aim_x+u;
% randomize [independent] errors of aimpoint in y-direction
mu_y = aim_y+v;

% call for Monte Carlo (MC) function:
```

```
[PD_total] =
LAM_MCv3(damage_matrix,X_offset,Y_offset,cell_width,cell_height,mu_x,mu
_y,sig_x,sig_y);

PD(t) = PD_total;

end

PD_mean = mean(PD)

fprintf('The Probability of Damage at a weapon range of %2.1f (ft) is:
%2.4f\n', r, PD_mean)
```

c. Monte-Carlo Simulation File

```
function[PD_total] =
LAM_MCv3(damage_matrix,X_offset,Y_offset,cell_width,cell_height,mu_x,mu
_y,sig_x,sig_y)
% Monte Carlo Simulation:

n = 100;  % number of Monte Carlo iterations
i = 0;

n_weap = length(mu_x);  % number of weapons
PS = 1;

while i<n     % this 'while' loop is the main overall loop
    i = i+1;  % counter for number of iterations

        j = 0;          % set (j) counter to zero
        while j<n_weap  % this 'while' loop counts for n number of
weapons
        j=j+1;     % counter for number of weapons up to (n) times

        % sample the input random variable 'x(j)'
        x(j) = mu_x(j)+sig_x*randn;
        % sample the input random variable 'y(j)'
        y(j) = mu_y(j)+sig_y*randn;

        % Lethal Area Matrix Damage Function for (j) weapon:

        % divide aim_X by cell width (37.9ft) and round the x value to
the next whole integer (upper bound)
        col_X = X_offset+ceil(x(j)/cell_width);

            if (y(j)>=0)

                % divide aim_Y by cell height (14.3ft)
                row_Y = y(j)/cell_height;
                % offset row_Y by 9 rows and round the y value to the
next whole integer (upper bound)
```

```
                row_Y = ceil(Y_offset-row_Y);

            elseif (y(j)<0)

                row_Y = y(j)/cell_height;
                % offset row_Y by 9 rows and round the y value to the
next whole integer (lower bound)
                row_Y = ceil(Y_offset-row_Y);

            end

            % damage matrix placement:

            % outside damage matrix
            if ((col_X<=0)||(row_Y<=0)||(col_X>20)||(row_Y>26))

                PD(j) = 0;

            elseif((col_X<=20)&&(row_Y<=26))  % within damage matrix

                PD(j) = damage_matrix(row_Y,col_X);

            end

        PS = PS*(1-PD(j));  % 'Survivor Rule'

        end

    PD_iter(i) = 1-PS;
    PS = 1;                 % Reset PS

end

PD_total = mean(PD_iter);
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] General purpose bombs. (n.d.). Defense Systems. [Online]. http://aeqaeroespacial.com.br/i-defesa_MK.html. Accessed Aug. 02, 2015.

[2] Bombs, rockets & missiles. (n.d.). Aerospaceweb. [Online]. http://www.aerospaceweb.org/question/weapons/q0211.shtml. Accessed Aug. 02, 2015.

[3] $172.6M for GPD-guided Excalibur 155mm shells. (n.d.). Defense Industry Daily. [Online]. http://www.defenseindustrydaily.com/1726M-for-GPS-guided-Excalibur-155mm-Shells-06835/. Accessed Aug. 02, 2015.

[4] M. R. Driels, *Weaponeering: Conventional Weapon System Effectiveness* (2nd ed.). Reston, VA: American Institute of Aeronautics and Astronautics (AIAA), 2013.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California