A Statistical Method for Non-Laser-Based Force-on-Force Training Systems

Travis Hillyer
DEVCOM Soldier Center
Orlando, FL
travis.r.hillyer2.civ@army.mil

Gianluca Argentini, Guillermo Giannattasio, Brian Romot
University of Florida
Gainesville, FL

ABSTRACT

Force-on-force training is a live training event that Army Soldiers participate in as a final opportunity to apply all previous training in an environment that represents realistic combat as closely as possible. Currently, weapon effects are replicated using a laser-based system known as the Multiple Integrated Laser Engagement System (MILES). Unfortunately, lasers do not closely represent the behaviors of real munitions and often the systems can result in unbelievable outcomes or, in the worst case, cause negative training. To improve training effectiveness and address some of the limitations of lasers, the Army is looking for novel solutions that expand the training capabilities within force-on-force training exercises. One such approach is with the use of various sensor arrays and physics-based calculations in a technique referred to as geo-pairing. Under this scheme, the location and orientation of a weapon system are measured, and the data is fed into a digital environment or game engine. From there, physics-based ballistic models calculate the flight path of the projectile to estimate where the live round would have landed. If the result intersects with an instrumented target, then the result of the engagement is considered a hit. In practice, these sensors and supporting models have inherent errors that prevent replicating real-world outcomes with 100% accuracy. Here, a statistical approach using Monte Carlo analysis and an Analysis of Variance (ANOVA) is proposed to help overcome these errors, first by helping to understand which sensors have the greatest impact on the overall accuracy of the training system, assess how the errors impact the probability of a hit within the digital environment, and ultimately to make recommendations about what sensor parameters are required to achieve specific training effectiveness metrics. One such model is presented here based on a simple representation of direct fire weapon engagements.

ABOUT THE AUTHORS

Mr. Travis Hillyer is the Live Force-on-Force training lead at the DEVCOM Soldier Center's Simulation and Training Technology Center in Orlando, FL. He has led multiple efforts exploring methods to replace current laser-based training systems used by the Army during collective training exercises. He is currently exploring novel solutions for performing advanced ballistic simulations on Soldier or weapon-mounted devices. He also supports multiple prototyping efforts with the Program Executive Office for Simulation, Training, and Instrumentation to rapidly develop mature solutions for existing force-on-force training gaps. He holds a MS in Industrial and Systems Engineering and a BS in Mechanical Engineering, both from the University of Florida.

A Statistical Method for Non-Laser-Based Force-on-Force Training Systems

Travis Hillyer
DEVCOM Soldier Center
Orlando, FL
travis.r.hillyer2.civ@army.mil

Gianluca Argentini, Guillermo Giannattasio, Brian Romot
University of Florida
Gainesville, FL

BACKGROUND AND MOTIVATION

For over 240 years, the United States Army has defended the Nation and served the American people. From the United States Army's vision and strategy statement (2022), "the key to this success has been the skill and grit of the American Soldier, the quality of its leaders, the superiority of its equipment, and [its] ability to adapt to and dominate a complex and continuously changing environment (United States Army, 2022). Soldier training is critical to maintaining the Army's readiness and a variety of different training methods are employed to accomplish different objectives. Force-on-Force training is one such method and serves as a final test to apply skills training in an environment that represents realistic combat as closely as possible. Soldiers must fight against an opposing force comprised of fellow Soldiers to achieve a strategic objective, during which time they are in the field for days at a time, using their real weapon systems, and firing blank munitions at real humans who are also firing back. Currently, weapon effects (e.g., the result of a bullet impacting a target) are replicated using a laser-based system known as the Multiple Integrated Laser Engagement System (MILES). Unfortunately, lasers do not closely represent the behaviors of real munitions and often the systems can result in unbelievable outcomes (which detracts from the training experience) or, in the worst case, cause negative training. For example, when engaging a moving target, the appropriate technique is to apply an offset (often referred to as "leading the target") with the weapon sight so that the munition will intersect with where the target will be based on both object's direction and speed. However, a laser travels at the speed of light negating any need for this offset and, in fact, forces a Soldier to aim directly at the moving target in direct opposition to their training.

To improve training effectiveness and address some of the limitations of lasers, the Army is looking for novel solutions that expand the training capabilities within force-on-force training exercises (PEO STRI, 2021). One such approach is with the use of various sensor arrays and physics-based calculations in a technique referred to as geo-pairing. Under this scheme, the location and orientation of a weapon system are measured, and the data is fed into a digital environment or game engine. From there, physics-based ballistic models calculate the flight path of the projectile to estimate where the live round would have landed. If the result intersects with an instrumented target (which is also sending its location to the game engine), then the result of the engagement is considered a hit and the target is either wounded or killed (i.e., the target's ability to further participate in the battle would be limited or disabled). In practice, these sensors and supporting models have inherent errors that prevent replicating real-world outcomes with 100% accuracy, resulting in the need to determine what level of accuracy is necessary to achieve a realistic, effective training solution. In these scenarios, statistical analyses can help infer potential outcomes, their likelihoods, and lead to the development of models that enable fair engagement outcomes when information about the state of battlefield entities is incomplete or inaccurate. By evaluating the relationship between the performance of commonly used sensors to determine which errors have the greatest impact on the overall accuracy of the training system and how the errors impact the probability of a hit within the digital environment to ultimately make recommendations about what sensor parameters are required to achieve specific training effectiveness metrics.

PROJECT DETAILS AND PRELIMINARY DESIGN

Objective

The overall objective is to apply statistical methods and to develop a model that will provide the probability of hit for a digital representation of a real-world weapon engagement given a combination of sensors with expected measurement tolerances. In other words, given a set of measurements of a weapon's position and orientation, what is an acceptable error for each measurement to ensure the simulated ballistic solution would match the corresponding live fire event with a certain probability. A simplified illustration of this challenge in two-dimensions is provided in Figure 1.

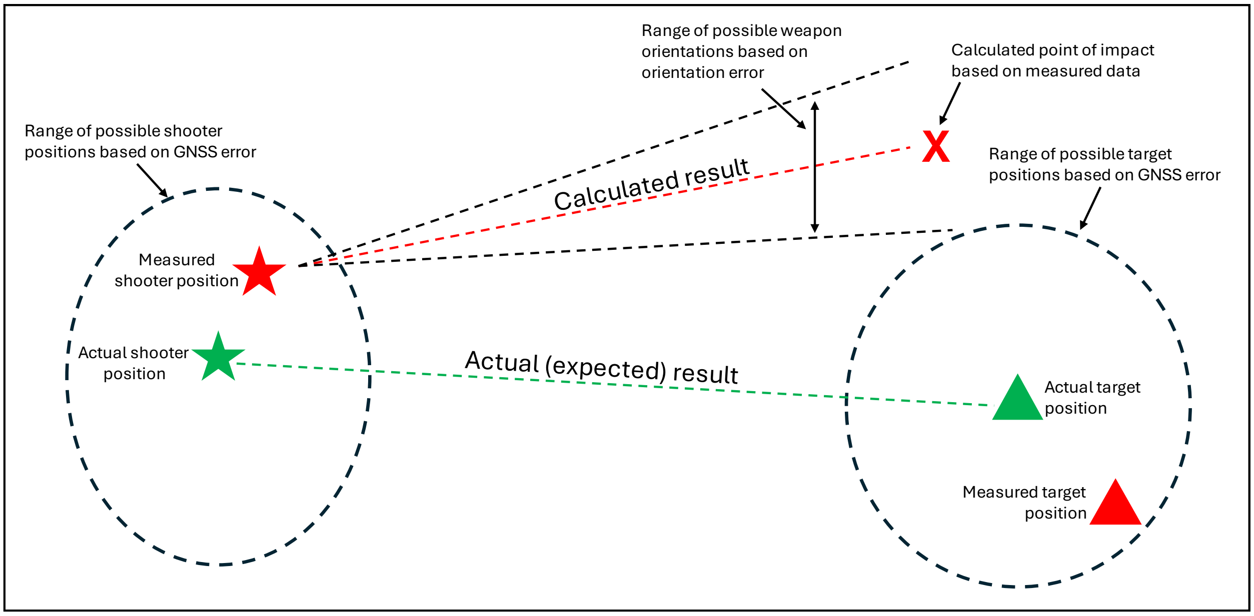

Figure 1. A 2-dimensional illustration of the geometric challenge associated with the proposed training system approach. The actual engagement (green) should result in a hit, however, the calculated result (red 'X') misses both the actual and measured target position (green and red triangles, respectively).

Ideally, the result of this project would be a model allowing system developers to:

a. understand which sources of errors have the greatest impact on performance,
b. evaluate allowable tolerances to increase probability of hit based on weapon type, and
c. make informed decisions regarding technology readiness and suitability to focus research investments.

This is accomplished through a three-phase modeling and analysis approach to address what sensor error has the greatest impact on the accuracy of the simulated engagement, how the errors impact the probability of a hit within the digital environment, and what capabilities are required out of each sensor to achieve a desired accuracy.

Modeling the Problem

To help build the model (and to eliminate the need for controlled information), the complex three-dimensional ballistic problem was simplified to a top-down view of a battlefield engagement where the effects of gravity, elevation, and ballistics can be ignored. Future efforts could expand on this model by adding more complexity back into the model; however, for the sake of preliminary analysis, the simplified model is considered sufficient to understand the effects of position and weapon orientation error on adjudication results. This limited scope (i.e. two-dimensional) geometric model is depicted in Figure 2 below and serves as the foundation for the statistical model and various analyses.

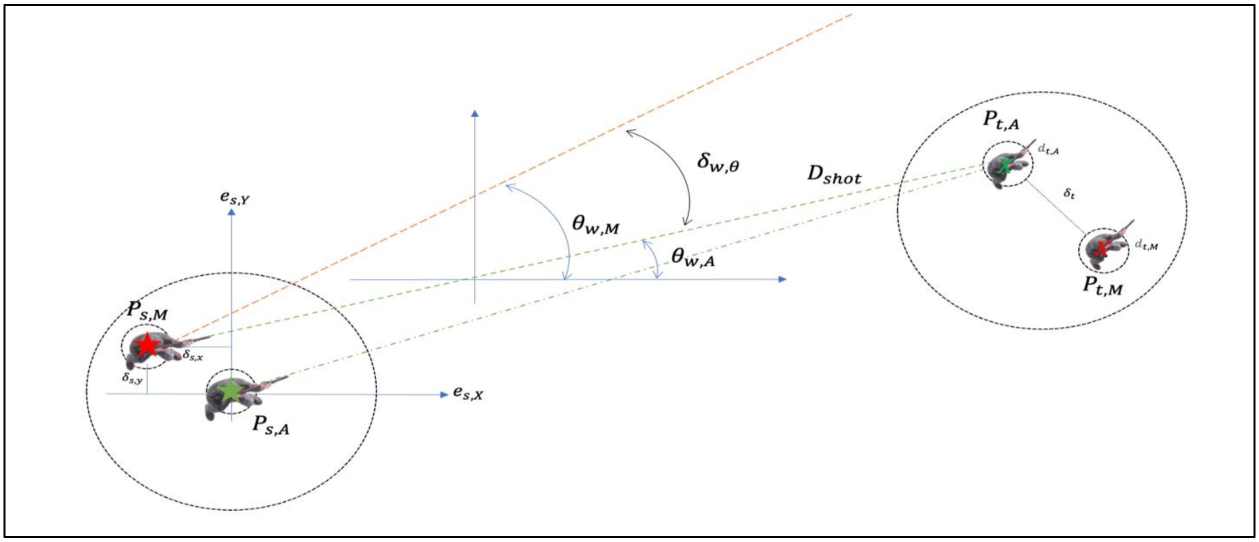

Figure 2. Graphical illustration of the two-dimensional geometric model used to represent the results of measurement error in the engagement outcome as seen from a top-down view. Here, the shooter, represented by stars, and the target, represented by *X*s, each have position offsets denoted by green (actual) and red (measured) versions of their respective symbol. Additionally, there is an angular measurement error of the weapons orientation indicated by the green line (actual point-of-aim) and orange line (measured aiming angle).

The geometric model calculates the bullet position (x_b, y_b) at some interval (e.g., $\delta_b = 1\text{mm}$) along a nominal shot distance of $D_{shot} = 500\text{m}$ according to equations 1 and 2,

$$P_{b,x_i} = P_{b,x_{i-1}} + \delta_b \cos(\theta_{w,m}) \mid P_{b,x_0} = x_{S,m} \quad (1)$$

$$P_{b,y_i} = P_{b,y_{i-1}} + \delta_b \sin(\theta_{w,m}) \mid P_{b,y_0} = y_{S,m} \quad (2)$$

where $\left(P_{b,x_i}, P_{b_{y_i}}\right)$ is the position of the bullet along the x and y axes, respectively for $i = \left\{0, \frac{D_{shot}}{\delta_b}\right\}$, the measured weapon orientation is $\theta_{w,m}$, and the measured position of the shooter, $P_{S,m}$, is defined as $(x_{S,M}, y_{S,M})$. The nominal value of D_{shot} was selected because it is the maximum effective range (MER) of the M4 rifle. The MER is the range where the weapon is able to inflict casualties and is generally the maximum range of interest for training systems. This range also has the lowest probability of hit for both real-world shooters and training systems compared to shorter distances. Under ideal conditions (i.e. the training system has zero measurement error), the measured weapon orientation is equal to the actual weapon orientation, $\theta_{w,A} = \theta_{w,m} \stackrel{\text{def}}{=} 0$, the measured position of the shooter is equal to the actual position of the shooter, $P_{S,M} = P_{S,A} \stackrel{\text{def}}{=} (0,0)$, and the measured position of the target is equal to the actual position of the target, $P_{T,M} = P_{T,A} \stackrel{\text{def}}{=} \left(D_{shot}\cos(\theta_{w,m}), D_{shot}\sin(\theta_{w,m})\right)$. Since the future Monte Carlo analysis will generate random error based on the ideal case, the actual shooter position and orientation are set to zero to eliminate the need to add a constant offset to the Monte Carlo simulation.

A baseline, theoretical training device was arbitrarily constructed using data from a set of sensors that might realistically be selected (based on size, performance, and cost) for an M4 training device. This theoretical system has GNSS and orientation sensor errors of 0.614m and 0.875°, respectively. This baseline will serve as the basis for an initial sensor sensitivity analysis in a Monte Carlo simulation and to help further verify the two-dimensional geometric model. The model assumes sensor measurement error is uniformly distributed. That is, the sensor output is equally likely to be any number within a fixed range equal to the nominal value plus and minus the maximum sensor error. However, this may not be the case and actual errors may be distributed differently (GNSS error may be normally distributed, for example, as suggested by Abbous and Samanta (2017)). Understanding an actual sensor's error distribution would require data collection for that specific sensor making this approach unfeasible for the current analysis. Regardless, the methodology presented here is still valid as a means to analyze relationships between sensors (in fact, small sample testing suggests the conclusions do not change if a normal distribution is selected instead of the presented uniform distribution).

Programming the Model

Selecting R as the preferred programming environment, the above model was replicated as a function that calculates the probably of hit for a user specified set of parameters and facilitates rapid iteration through multiple runs with different parameters simply by changing the input variables. Full source code is provided separately but critical elements are described below.

The function can accept the following six inputs:

1. `E_g`: the absolute value of the error associated with GPS/GNSS in meters
2. `E_m`: the absolute value of the error of the orientation sensor in degrees, converted to radians for compatibility with R by multiplying the input by $\pi/180$
3. `num_runs`: (optional) the number of runs to simulate the model. The default value is 10 to minimize the default computation size
4. `step`: (optional) the interval distance of the bullet. The default value is 0.1m to minimize the default computation size
5. `t_size`: (optional) the size of the target. The default value is 0.5m, based on a commonly used paper target size
6. `D_max`: (optional) max nominal distance to consider. The default value is 500m plus twice the GPS error to allow for errors that might place the target outside the 500m maximum effective range of an M4

The first part of the function simply initializes variables to improve execution of future calculations. This includes randomizing the shooter location, target location, and weapon angle. In R, this is achieved using the `runif` function which generates pseudorandom data from a uniform distribution. For example, the shooter's x-position is defined as `r_sx = runif(num_runs,-E_g,E_g)`. The same approach is done for the shooter's y-position (`r_sy`), weapon angle (`r_a`), target x-positon (`r_tx`), and target y-position (`r_ty`). Note that the target position also includes a 500m offset to account for its separation from the shooter. Additionally, the function establishes several variables for the bullet trajectory that are used later to determine if the bullet intersects the target. Specifically, `Pb_x_list` and `Pb_y_list` stores the bullet path in a list for each run.

Once the initial setup is complete, the function executes a Monte Carlo simulation. A `for` loop runs the simulation for the specified number of runs, calculating the path of the bullet and the engagement result for each run. The method for calculating the bullet's position is identical to the methods described in the geometric model and equations 1 and 2 above.

There are two approaches within the function to calculate the simulation's probability of hit. They are illustrated in Figure 3. The first is to check if each individual run resulted in a hit to the corresponding randomized target location. In practice, this is equivalent to considering whether shot one intersected target one, shot two intersected target two, and so on. This is referred to as the individual probability of hit. The second method is to calculate the trajectory of all runs and count how many of those runs intersect all possible target locations. Practically speaking, this would be the same as shot one intersecting any of the possible targets, shot two intersecting any of the possible targets, etc. This is referred to as the "windowed" or "total" probability of hit. Each of these methods considers the target to be a square of size `t_size` centered about the random position `(r_tx,r_ty)` and calculates probability of hit by dividing the sum of all hits by the total number of runs. The difference between these two methods is important to consider since there will be differences in the hit rate and total probability of hit. A discussion regarding these differences is presented in the Analysis Methodology and Preliminary Results section; however, for the purposes of the remainder of this analysis, the individual probability of hit will be used in

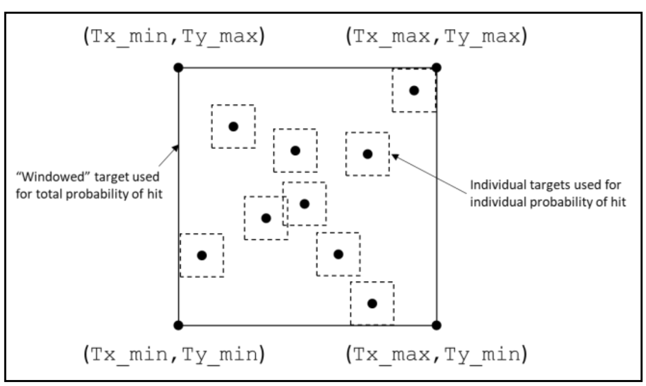

Figure 3. The two different target areas considered for probability of hit are defined by the individual targets or the total area encompassing all possible targets.

all cases, unless otherwise stated, since this is viewed as the more accurate version of the simulation (i.e. its closer to a real shot engagement).

The individual targets in Figure 3 are defined as:

```
r_tx ± (t_size/2)
r_ty ± (t_size/2)
```

Conversely, the windowed target boundaries in the figure are:

`Tx_min = min(r_tx)`: sets lower bound for target x position
`Tx_max = max(r_tx)`: sets maximum bound for target x position
`Ty_min = min(r_ty)`: sets lower bound for target y position
`Ty_max = max(r_ty)`: sets maximum bound for target y position

Two counters within the function keep track of run results and are used to calculate hit percentage. One counter, `count` and `result`, tracks the individual runs (i.e., Run 1, Run 2, etc.) for calculating individual probability of hit and the second, `countT` and `resultT`, counts the total hits for all runs used in calculating windowed probability of hit. The final step is to calculate the two probably of hits and return the following results:

`P_Hit = (result/num_runs)*100`: Calculates the individual probability of hit as a percentage
`P_HitT = (resultT/num_runs)*100`: Calculates the total probability of hit as a percentage

Analysis Methodology and Preliminary Results

Statistical methods used for the model analysis are a Monte Carlo simulation and an Analysis of Variances (ANOVA). The first, Monte Carlo simulation, enables a large set of pseudo-random events to be simulated to better understand trends in outcomes when the inputs are uncertain. This type of approach is well suited to the geo-pairing system use case since all the sensor measurements have inherent errors that make predicting accurate engagement results difficult. For this simulation, 5,000 runs are calculated (five trials of 1,000 runs each), each with randomized input for shooter position, target position, and weapon orientation. The result of each run is recorded as a hit or a miss and used to determine, in general, the likelihood that a set of randomly recorded measurements would result in a hit. From this information, it's possible to infer a sense of how realistically a combination of sensors in a geo-pairing training system could replicate an actual weapon engagement. After developing the algorithm to simulate the training engagement and predict the probability of hit, the initial parameters were set to the theoretical system's GPS error (`0.614cm`) and orientation sensor error (`0.875°`). Additionally, the scenario used the default values for step size, target size, and maximum distance (`0.1m`, `0.5m`, and `500m`, respectively). The function was run five times using these parameters, with 1000 simulations per trial (`num_runs=1000`), to show the variability inherent in our Monte Carlo simulation. Figure 4 provides a visualization of the results from one of the trial runs. Each of the black lines represents an individual bullet flyout starting from a randomized location. The program also draws the concept described in Figure 3 using green boxes to represent the individual target for each run and a red box surrounding them to represent the windowed target. Not only does this image provide a convenient depiction of the results, showing how many bullet paths intersect with the target space, but it also helps

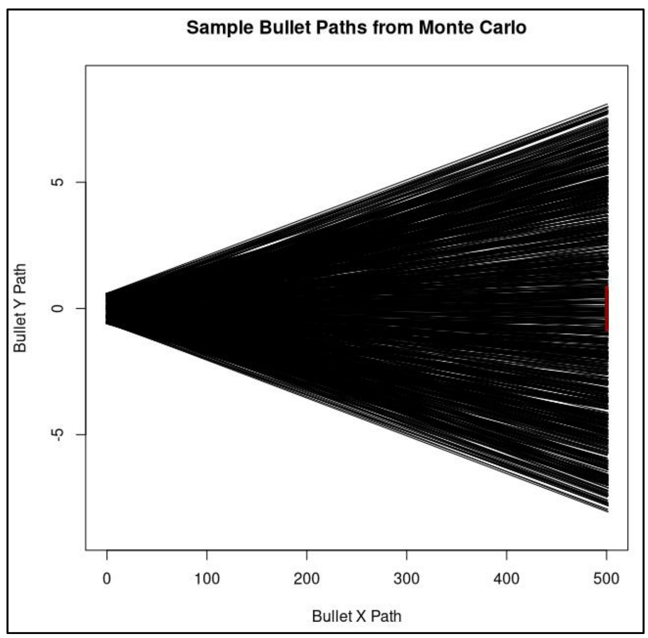

Figure 4. Sample plot from one of the runs. Barely visible on the right-hand side is the red box representing the windowed group of randomized targets used to calculate the windowed probability of hit. The individual targets are drawn as green boxes but are not visible in this image.

visualize the changes in performance as sensor inputs are changed. Table 1 presents the individual results and the overall average for these trials—an average individual probability of hit equal to 3.2% and windowed target probability of hit equal to 11.2%.

Table 1. Results from five trials of five trials of the Monte Carlo simulation.

Trial Number	Individual Probability of Hit	Windowed Probability of Hit
Trial 1	3.4	11.2
Trial 2	3.3	11.8
Trial 3	3.0	9.3
Trial 4	2.9	11.0
Trial 5	3.4	12.9
Average	**3.2**	**11.2**

The second analysis tool, ANOVA, helps determine which sources of error most impact system performance. In this a case a three factor, two level (2^3) experiment with replication is sufficient to look at all the driving factors that affect probability of hit. The three factors are GNSS error (factor A), orientation error (factor B), and target size (factor C). The data for factors A and B are the average minimum and maximum values from arbitrarily selected, commonly used sensors. The third factor, target size, was included in the experiment to see how much the probability of hit could be influenced by adjusting the impact area. To keep it somewhat realistic however, the minimum and maximum values were defined as 50% (0.25 meters) or 150% (0.75 meters) of the default target size, respectively. All the values used for these factors are summarized in Table 2.

Table 2. Minimum and maximum values for sensor errors used as factors for the ANOVA experiment.

Factor	**-**	**+**
Factor A: GPS Error, E_g (meters)	0.614	2.21
Factor B: Orientation Error, E_m (degrees)	0.875	11.5
Factor C: Target Size, t_size (meters)	0.250	0.750

The 2^3 experiment has the format found in Table 3. Note that runs AB, AC, BC, and ABC, represent the interactions between each of the independent factors. Since the experiment is using replication, the model was run four times with the values from Table 1. In addition to the target size, GPS, and orientation error above, the R function ran with `D_max = 500` meters and `num_runs = 1000`.

Table 3. The three-factor, two-level experimental setup.

		Factors			**Results (Probability of Hit)**			
Run		**A**	**B**	**C**	**1**	**2**	**3**	**4**
1	(1)	-	-	-	###	###	###	###
2	a	+	-	-	###	###	###	###
3	b	-	+	-	###	###	###	###
4	ab	+	+	-	###	###	###	###
5	c	-	-	+	###	###	###	###
6	ac	+	-	+	###	###	###	###
7	bc	-	+	+	###	###	###	###
8	abc	+	+	+	###	###	###	###

The data in Table 2 was imported into R (as a variable named `data`) and loaded into a linear model using the built-in function `lm()`. An ANOVA was run on the model using the function `anova()`.

Generate a linear model using the simulation data for the ANOVA

```
> model=lm(x ~ a + b + a*b + c + a*c + b*c + a*b*c , data=data)
> anova(model)
```

These results are sufficient to draw conclusions about the training system; however, as an additional check, a second model was created using the three most significant factors (B, C, and BC). The functions and input data are the same as before, only the least significant factors are removed from the model. This second ANOVA does not alter the conclusion of the analysis.

Generate a second linear model using the three most significant factors

```
> model2=lm(x ~ b + c + b*c , data=data)
> anova(model2)
```

Caution is merited with ANOVA studies since the significance of factors can be easily misinterpreted if the relationships between variables are poorly understood or assumptions about the model type are incorrect. This prompted an investigation into the relationship between orientation error and GNSS error. Additional trials (`num_runs = 1000, t_size = 0.5 meters`) were run using a fixed GNSS error (`0.614m`) and variable orientation error (`[1, 0.1, 0.01, 0.001, 0.0001, 0.00001] degrees`) to evaluate how probability of hit changed. A similar exercise was performed for different values of target size (zero through two, at one tenth meter increments) and fixed orientation and GNSS error. This was repeated for constant orientation errors (`[1, 0.1, 0.05, 0.01] degrees`) while steadily decreasing the GNSS error values (`[1, 0.5, 0.25, 0.125, 0.0625] meters`). Graphing these results helps to visualize the relationships between the different measurements on probability of hit and help to indicate how they affect the simulation across a wide range of possible values.

FINAL RESULTS AND DISCUSSION

A summary of the individual probability of hit results from the entire project is presented in Table 4. The first two rows represent results from an ideal control system and the hypothetical default training system based on a realistic potential set of sensors, from which the preliminary results were based on. Runs three through ten are the simulation results from the designed experiment. Most notable from these trials is that none of the tested sensor combinations yield a particularly desirable probability of hit—all less than ten percent. The results of the ANOVA indicate that GPS error (A) doesn't have as much impact on system performance as the orientation error (B) or the target size (C), which drive the greatest changes to probability of hit.

Table 4. A summary of the results of the model and ANOVA analysis. Significant factors, based on P-value less than 0.001, are indicated by asterisks (*). A is GNSS Error, B is orientation error, and C is target size.

Trial		Factors			Results				ANOVA
Run	Purpose	E_g	E_m	t_size	1	2	3	4	P Value
1	Control	0	0	0.5	1.000	1.000	1.000	1.000	N/A
2	Default System	0.614	0.875	0.5	0.031	0.032	0.032	0.029	N/A
3	DOE (-1)	0.614	0.875	0.25	0.010	0.014	0.020	0.015	N/A
4	DOE (a)	2.21	0.875	0.25	0.017	0.013	0.015	0.014	0.552
5	DOE (b)	0.614	11.5	0.25	0.001	0.000	0.004	0.001	2.2E-16*
6	DOE (ab)	2.21	11.5	0.25	0.001	0.001	0.000	0.000	0.663
7	DOE (c)	0.614	0.875	0.75	0.048	0.043	0.061	0.049	6.18E-12*
8	DOE (ac)	2.21	0.875	0.75	0.046	0.064	0.049	0.055	0.365
9	DOE (bc)	0.614	11.5	0.75	0.005	0.002	0.000	0.000	3.29E-11*
10	DOE (abc)	2.21	11.5	0.75	0.004	0.001	0.002	0.006	0.905

From Table 4, orientation error, target size, and the interaction between the two clearly have the greatest impact on probability of hit. Fundamentally, this result makes sense since a larger target would be easier to hit at any distance and angular error results in a linear increase in positional error proportional to total distance traveled. (Consider that the formula for the arc length, s, of a circular arc is calculated by $s = r\theta$, where r is the radius of the circle and θ is the angle of the arc in radians. In the context of the training problem, r is equivalent to the distance to the target, `d_max`, and θ is the orientation error, `E_m`. As `E_m` grows, so does the likelihood that the model will miss the target.) What is less clear, however, is why GNSS error has almost no impact on the outcome of the engagement. To better understand this, the plots in Figure 5 are useful tools to explain the GNSS-Orientation-target size interaction. These plots show the relationship between orientation error, GNSS error, and target size, respectively, against probability of hit. For each plot, one of the variables was changed over a wide range of values while the others were held constant.

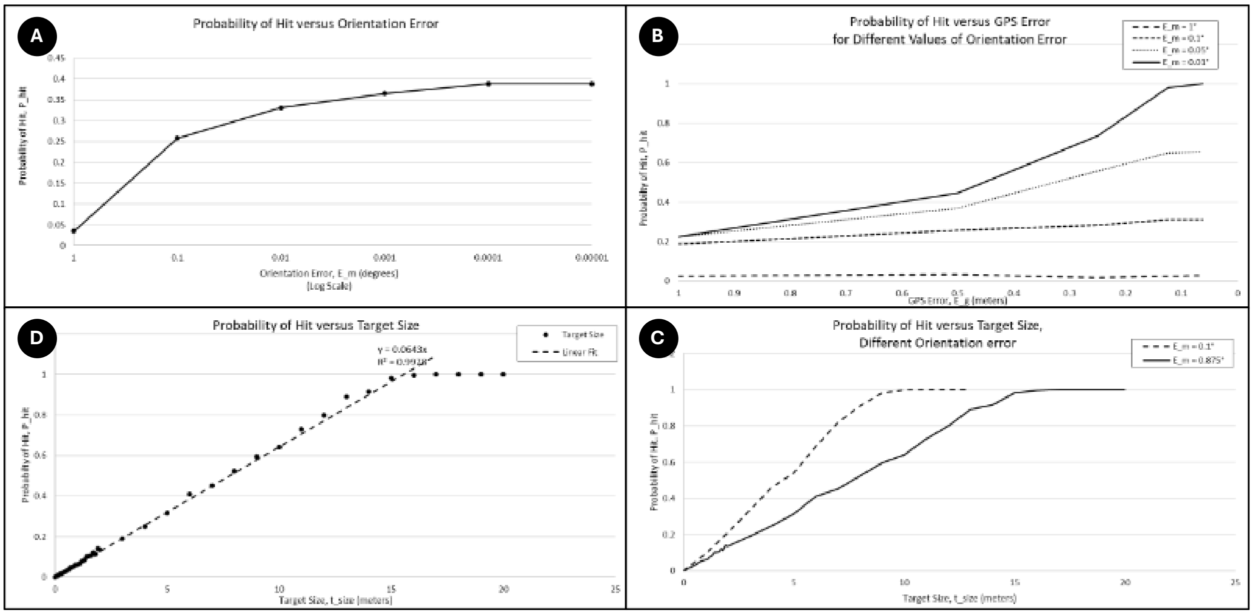

Figure 5. Clockwise from top left: (A) probability of hit versus orientation sensor error with GNSS error and target size constant, (B) probability of hit versus GNSS error for several constant values of orientation error and constant target size, (C) response of probability of hit versus target size for different orientation accuracies, and (D) probability of hit versus target size for fixed GNSS and orientation sensor error.

The plot in Figure 5(A) supports the ANOVA results for some values of orientation error but suggests that its impact on probability of hit becomes negligible after a certain accuracy is achieved, around 0.01 degrees for this system. Interestingly, at that same level of orientation accuracy, the GNSS error starts to have a greater impact on the probability of hit. In fact, Figure 5(B) suggests that the only way to achieve high probabilities of hit is to have both high orientation and GNSS accuracy. This interaction requires further study and is recommended for future expansion of this model as it may be particularly useful for informing future decisions when it comes to selecting sensors to achieve a certain probability of hit. The sensor packages required to achieve a desired probability can be found simply by locating the intersection of a horizontal line drawn at the desired value along the y-axis with one or more of the plotted lines. The x-value of the intersection point will indicate the required GNSS error, and the plotted line gives the necessary orientation error as illustrated in Figure 6. Target size, on the other hand, displays a strong ($r^2 = 0.998$) linear relationship with probability of hit and can be used as a scaling factor to improve overall system performance.

The chart in Figure 5(D) illustrates the linear relationship between probability of hit and target size, up until a 100% probability is reached. Specifically, probability of hit scales linearly with target size and is proportional to the slope of the fit. For example, in this scenario, increasing the target size by a half meter results in a roughly 0.034 (0.0674*0.5 = 0.034) increase in hit probability. The data in Table 6 are consistent with this result. This plot also helps visualize why target size is a significant factor per the ANOVA. While manipulations to GNSS or orientation error can improve overall probability of hit, it may be impossible to achieve 100% for certain combinations of accuracy (see Figure 5(B)). Manipulating target size, on the other hand, can achieve a 100% probability for any combination of sensors. This characteristic may be useful for fine tuning the simulation but is practically limited in its application to increase probability of hit, because there is a sensible limit to how much the target size can be increased. It is important to note, however, that the rate of change will be different for each unique combination of orientation and GNSS error, as illustrated in Figure 5(C). A more accurate system will have a larger slope and reaches a maximum probability of hit faster than a less accurate system, to the point where target size becomes negligible. Intuitively, this response is consistent with the idea that a larger target will be easier to hit to the point where it would be virtually impossible to miss.

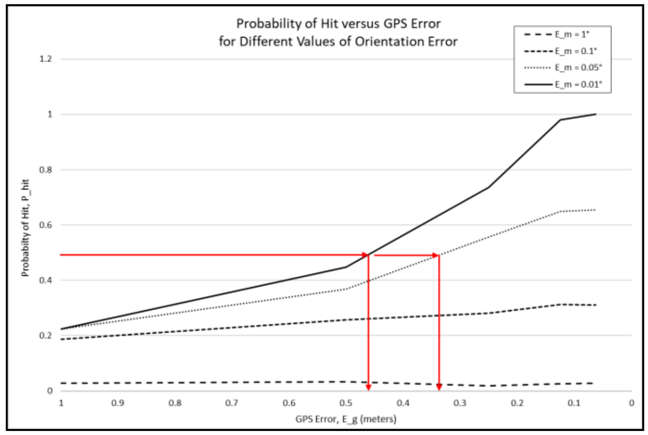

Figure 6. Using the plot to inform system design indicates two sensor combinations capable of producing a 50% probability of hit: an orientation accuracy of 0.01 degrees combined with a GNSS accuracy of approximately 0.46 meters or one with an orientation error of 0.05 degrees coupled with GNSS accuracy of 0.34 meters.

As described in the Programming the Model section, the function calculates two different probabilities of hit and the results of preliminary trials were presented in Table 3. As expected, the probability of hit will always be higher for the "windowed" situation when the GNSS error is greater than the target size. This can bring further opportunities to consider what constitutes a hit and may also provide an interesting opportunity to decompose the result into a near miss, wounding shot, or a fatal shot, similar to what the current MILES system does. In this scheme, developing the training system to consider multiple target sizes that serve as hit boxes, with increasing probabilities of hit would correlate to different engagement results. For example, a human sized target may have a kill hit box 0.4 meters wide with probability of hit equal to 50%, a wound hit box 0.5 meters wide with probability of hit equal to 75%, and a near miss hit box 0.6 meters wide with probability of hit equal to 90%. Additionally, the "windowed" probability of hit might be more useful for developing a training system where the difficulty of the engagement is controlled by the simulation software. In other words, the engagement could be made more or less difficult by adjusting the target size to decrease or increase probability of hit, respectively. Conversely, the individual probability of hit may be more useful for informing the training hardware design.

CONCLUSIONS

A method for modeling a simple ballistic engagement simulated using Monte Carlo experiments to predict outcomes in the face of sensor uncertainty was presented. Furthermore, an analysis of the results using ANOVA methods attempted to draw conclusions about the relationship between the performance of different sensors on the overall engagement outcomes. The results of this analysis support previous conclusions (Alombro, 2018) that there are limited, if any, sensor combinations suitable for dismounted, direct fire applications capable of achieving high probabilities of hit for a pure geo-pairing training system. The theoretical training device used during the Monte Carlo simulation, for example, was only able to achieve a 3.2% hit probability with a set of sensor errors that is not unrealistic for currently achievable training systems. However, there does appear to be a clear direction for future research and system development. First, focus on minimizing orientation error. The results of the ANOVA and subsequent plotting of results suggest that the priority should be on minimizing the system's weapon orientation error in order to have the greatest impact on probability of hit. This is especially true for long range weapons where distance has a linear scaling affect with angular error. Second, evaluate allowable tolerances to increase probability of hit based on weapon type. For the M4 direct fire application explored in this study, the threshold orientation error should be less than or equal to 0.05 degrees (0.89 mils), based on the chart in Figure 5(B). Only then will position error become an important factor. Fortunately, corrected GNSS methods, such as Real Time Kinematics (RTK), are already capable of providing sufficient accuracy under ideal conditions to achieve high probabilities of hit, provided the threshold orientation accuracy is reached. Additionally, exploring the interaction between target size and system error may offer an opportunity, if compromises are acceptable, to achieve higher probabilities of hit with less accurate systems. Based on Figures 5(C) and 5(D), increasing target size will result in a proportional increase in probability of hit—in the default sensor setup for this case, by approximately 0.06 per meter. This offsetting affect is still limited, however, if weapon orientation error remains large, since the target size must also be large to achieve high probabilities of hit. There is a functional limit to how large the target size can be increased before it starts to be absurd and cause other problems within the training scenario (e.g., a 20m wide soldier is nonsensical and targets may start to overlap). A more practical application of adjusting target size may actually be tuning the difficulty of the engagement. Small adjustments to the target size within the training simulation software could raise or lower the effective probability of hit while being imperceptible to the user. This offers leadership the ability to easily adjust training difficulty based on unit skills or the expected capabilities of the real-world threat.

FUTURE WORK

As noted throughout this report, there are several areas that merit further investigation to either improve understanding of the relationship between different variables or increase the realism of the model.

1) *Explore the effect of different distributions on the model performance. Attempt to validate the real-life distribution of each component*. The model presented here assumed sensor errors were uniformly distributed, e.g., the measurement had an equal probability of being anywhere within the sensor tolerances. While this is a valid assumption, it may not be the most accurate in all situations and for all sensors. For future work, it is worth applying different distributions (e.g., normal: $\mathcal{N}(\mu = 0, \sigma^2 = E_m)$) to the error to understand how that affects the analysis. Ideally, this would be taken one step further and different sensors would be characterized through real-world measurements to determine their actual distribution.
2) *Iterative experimentation to fine tune model parameters and account for unforeseen factors*. In the same vein as exploring additional probability distributions, there may be factors that were missed by this analysis. Using this model as basis for experimentation with real hardware could enable iterative design. Performing small-scale testing to provide feedback on sensor form, fit, and function can potentially generate data to update the current model with new accuracy or SWaP+C limits, or to inform further development of the training system. It is expected that this would be a long and costly process but may provide valuable information to help build next-generation technology that will maintain the Army's lead in fielding world-class training capabilities.

REFERENCES

United States Army. (2022, December 12). The Army's Vision and Strategy. Retrieved from U.S. Army: https://www.army.mil/e2/downloads/rv7/vision/the_army_vision.pdf
PEO STRI. (2021, Nov. 11). *Welcome to STE LTS Industry Day.* [Conference Session]. STE LTS Industry Day, Rollins College, Winter Park, FL, United States

A. El Abbous and N. Samanta, "A modeling of GPS error distributions," 2017 European Navigation Conference (ENC), Lausanne, Switzerland, 2017, pp. 119-127, doi: 10.1109/EURONAV.2017.7954200.

Alombro, M. N. (2018). Small Arms Weapons Orientation and Position Estimation Through Scenario-Based Simulations. Wright-Patterson Air Force Base: Department of the Army.