Calculating Error Tradeoffs in Weapon Simulation for Live Training

Robert J. Hall
AT&T Labs Research
Florham Park, NJ
bob-3Tradeoffs-@channels.research.att.com

Matthew R. Janisz
Applied Research Associates, Inc.
Orlando, FL
mjanisz@ara.com

ABSTRACT

Next generation instrumentation systems for live military training will simulate weapon engagements (shots) using geo-pairing: matching shooter with target by geometric computations that determine whether the trajectory or blast of the round intersects the target. Such calculations depend critically on accurate sensor readings, including positions of shooter and target(s), the pointing angle of the weapon at the time of the simulated shot, and positions of terrain obstacles. However, *perfect* sensors are unattainable, so a key question for system engineers is just how accurate do the sensors need to be? After all, physical weapons themselves have inherent inaccuracies (weapon "dispersion" or "spread"); for training purposes, it is not necessary to simulate better accuracy than exists in the weapons themselves. Moreover, a less accurate point angle sensor can to some degree be compensated for by a more accurate position sensor, and vice versa. Thus, ultimately we would like to understand this tradeoff quantitatively in order to support cost effective system engineering decisions.

This paper describes an iterative statistical approximation method, implemented as a set of computational tools, used by the U.S.Army's One Tactical Engagement Simulation System (OneTESS) Project for supporting these design tradeoffs. In particular, we show how to compute the probability of correct sensor-based shot adjudication for given sensor combinations, as well as tradeoff diagrams showing which combinations of sensors can be used together to achieve realism matching the physical accuracy of actual weapons systems. A key part of this method is building approximate computational models of aim, weapon spread, and sensor error distributions based upon input gleaned from subject matter experts in the areas of live training and instrumentation. We illustrate the method with case studies involving direct fire, indirect fire, and terrain problems.

ABOUT THE AUTHORS

Robert J. Hall earned the SM, EE, and PhD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology. Since then he has been a Principal Investigator at AT&T Laboratories Research, working in the areas of automated software engineering, requirements engineering, and modeling and simulation. He serves on the Steering and Program Committees for the ACM/IEEE International Conferences on Automated Software Engineering, as well as the editorial board of the Automated Software Engineering Journal (Springer) and I.F.I.P. Working Group 2.9 on Requirements Engineering. He has nine issued patents and over 40 publications.

Matthew R. Janisz is a Software Engineer with Applied Research Associates, Inc., currently involved with modeling and simulation on the One Tactical Engagement Simulation System (OneTESS) program. He works in the areas of network event simulation, constructive simulation, and error (quantitative) analysis simulation. Mr. Janisz received his Bachelor of Science degree in Computer Science from the University of Central Florida in 2005 and is currently pursuing a Master of Science degree in Modeling and Simulation from the University of Central Florida.

Calculating Error Tradeoffs in Weapon Simulation for Live Training

Robert J. Hall
AT&T Labs Research
Florham Park, NJ

Matthew R. Janisz
Applied Research Associates, Inc.
Orlando, FL

INTRODUCTION AND GOALS

The military wishes to train its troops and test its systems in situations as realistic as possible short of using live fire and thereby risking serious injuries. To support this, an instrumentation system replaces live fire with simulated fire. The U.S. Army's One Tactical Engagement Simulation System (OneTESS) is a next generation instrumentation system based upon accurate sensor technology and Mobile Ad Hoc Networking (MANET).

A critical capability of OneTESS (U.S.Army - OneTESS 2006) will be its ability to compute the outcome of simulated engagements based on sensor measurements. We term this process *adjudication*. Measurements are taken at weapon, shooter, and potential targets; these are concentrated to one location during engagement adjudication using the MANET wireless network (Hall 2005); and geometric calculations are performed using sensor data and a terrain database to compute who, if anyone, is damaged by the engagement. These calculations are termed *geopairing* (U.S.Army - OneTESS 2006; SRI International 2006) as they *pair* shooter with target(s) using *geo*metric calculations (Parry 1995) based on *geo*graphic data. This sort of adjudication is as opposed to laser-based pairing, where a laser is used to designate a shot target via a physical line of sight. The limitations of laser-based approaches are many, including susceptibility to obscuration by things (such as dust and tent flaps) that do not stop bullets, and inability to handle indirect fire (highly curved trajectories and area effects) at all. Geopairing has potential to improve on these aspects. Using wireless networking, which may relay around line of sight obstructions, messages can get from shooter to target even when the latter is visually obscured. Also, the system can simulate any sort of non-light-like trajectory, including curved, guided, and even intelligently controlled.

However, the Achilles' Heel of geopairing is sensor accuracy. If sensors or terrain information are inaccurate, say reporting a target's position as a few meters to the left of the body's actual position, or reporting the pointing angle of the rifle a few fractions of a degree off from the true aim, the outcomes of engagements will be incorrect and useless for training. Many human shooters, particularly the more highly skilled ones, can tell when they are likely to hit their target, and if the system reports the opposite result too often, they will lose confidence in it or, worse, find ways to cheat the system to improve scores without actually becoming trained for real battle.

Perfect sensors do not exist; all real-world devices have some error. Terrain cannot be measured exactly, either. Usually, by spending more money, one can purchase more and more accurate sensors, so the economic question becomes, how much money should a system developer spend to get a system that is accurate enough to meet its goals? To begin to answer this question, the present study addresses the slightly simpler question of how accurate sensors and terrain measurements need to be for the system to meet its goals.

Geopairing calculations involve inputs from multiple sources; inaccuracies in one sensor can, to some degree, be compensated for by increased accuracy in others. Thus, we are really attempting to study the *tradeoffs* possible among measurement inaccuracies, with *tradeoff diagrams* showing allowable combinations. Such diagrams can support economic decisions about which sensors to buy. Of course, costs of sensors will not correspond simply with position in the diagram; for example, a highly accurate position sensor coupled with a less accurate point angle sensor may be cheaper overall than a highly accurate point angle sensor coupled with a less accurate position sensor. But having the diagram to delineate the space of combinations "good enough" should be useful in making these economic decisions.

To summarize, the goal of our study is to determine the tradeoffs possible among sensor and terrain inaccuracies that allow geopairing to attain an acceptably high level of accuracy for particular weapon types. The novel contributions of this paper are aspects of the computational techniques themselves and illustrative results obtained for particular weapons and situations. Note that this paper does not attempt to derive requirements for the OneTESS program; OneTESS must encompass results like these over many weapon systems and varied situations. Our study has only begun this work. In particular, this paper is limited as follows: 2-dimensional direct fire only (i.e., ignoring elevations); very simple weapon models, such as ignoring effects of turret acceleration; inferring elevation from the terrain database instead of using GPS z meaurements; and perfect aim (see below).

The next section discusses the computational tools, and the three succeeding sections overview the specific methods and results for direct fire, indirect fire, and terrain-related problems. Due to space restrictions, we can only summarize the full results of our study.

Note on weapon information and result data. In our OneTESS work, we analyzed and obtained results for real military weapon systems. Since the U.S. Government has not approved the public release of certain weapons capabilities and performance data, we are restricted from disclosing that information in this paper. In the interest of national security we have chosen to rename the weapon systems to fictional names (DF00, IF01, etc); any performance information relating to a fictional name has been fictionalized as well; direct fire result graphs apply to a large class of direct fire weapons and do not reveal any sensitive information; indirect fire result graphs (with and without elevation error) are *obfuscated* by being expressed in incompletely specified measurement units. Finally, some subsections, such as direct fire and foxhole problem, do not contain sensitive information and are unchanged.

COMPUTATIONAL MODELING TOOLS

Our approach is to model sensors' error distributions, aim distributions, and weapon spreads, and to build computational tools for combining these along with a computational model of each engagement type, and then compute tradeoff graphs showing which sensor error combinations are "good enough" in each situation. This section expands on this method.

Modeling Sensor and Terrain Errors

Sensor errors are modeled as probability distributions yielding the deviation from nominal. For this study, we have modeled the following.

Global Positioning System (GPS): A GPS sensor produces the x and y coordinates of the sensor's position on the surface of the Earth. According to vendor literature and corroborated by limited lab tests, each coordinate reading deviates according to its own independent 1-dimensional Gaussian normal distribution. We constructed computational representations of these distributions, parameterized by the *3-sigma radius:* the distance within which 99.7% of all errors occur. Thus, for example, if a node is positioned at (0,0), then 99.7% of the readings of a sensor with 3-sigma radius of 1 meter will have x value in [-1, 1]. Independently, 99.7% of the readings will have y value in [-1, 1]. Therefore, 99.4% will have both readings in those ranges.

Point Angle Sensor (PAS): A point angle sensor determines (typically, through magnetic and/or inertial means) the direction relative to North (or inclination angle relative to horizontal) that it is pointing. This is sometimes referred to as Weapon Orientation Measurement (WOM). Again according to vendor literature, these devices give readings whose errors obey normal distributions independently in each dimension (a dimension is either azimuth or elevation).

Terrain Database: the terrain database is not a sensor, as such, in that readings are taken off-line and stored for later use during geopairing calculations. However, we are interested in how accurate such stored data must be in order to support geopairing involving terrain occlusion. While we do not really know a lot about the distribution of errors during terrain measurement, once these measurements are made, they are "frozen" into the database. Thus, every geopairing calculation made near a particular point of the database will have exactly the same reading each time. Therefore, it is not appropriate to treat it the same way as we treat random GPS and PAS errors. In our studies of terrain, we will look at behavior near terrain edges and characterize the error simply by the edge's offset from reality.

Aim Points and Weapon Spreads

In a given engagement situation, there are two physical factors affecting what it means for sensor-based geopairing to be good enough, the aim point distribution and the weapon spread distribution.

The *aim point distribution* is the distribution of aiming angles at which the weapon (whether it be an instrumented sensor-based weapon or an actual weapon) is pointed during a representative set of scenarios.

The *weapon spread distribution* is the distribution of deviations inherent in the physical weapon itself. Each weapon is itself imperfect, in the sense that not every round will go to exactly the same place given a fixed aim (and environmental conditions). This inter-round dispersion is due to factors such as materials variations in the loads, mechanical or heat strain on the weapon, and other factors. Note that for this study, when we wish to model environmental factors, such as weather, we include these effects in weapon dispersion.

Our general goal is to adjudicate as accurately as the physical weapon, so sensor-based adjudication need not be perfect. If the physical weapon would hit $H\%$ of the time in a given class of situation, then the sensor-based weapon should adjudicate a hit $H\%$ of the time as well in that class. Thus, it need not be the case that the simulated "bullet" is calculated to hit at exactly the same place each time. In general, a target has equivalence classes of points, any member of which, if hit, yields the same outcome. For example, if we model only "hit" or "miss", then any point within the target body is good enough. If we have a richer model such as "catastrophic kill", "mobility kill", "near miss" , etc, then each of these represents an equivalence class of points on (or near) the target. All the sensor-based weapon has to do is score the same class of hit as the physical weapon would. This is a looser requirement than requiring that each round be adjudicated within the weapon's spread area.

As an example of this distinction, consider rifle fire at a 60 cm wide target. If the rifle guarantees to deliver all rounds within a circle of diameter 14cm, all such shots (perfectly aimed) would be hits. However, a sensor-based weapon need only register every shot a hit on the target, which can be achieved by hitting anywhere within the 60 cm wide body. For each engagement type evaluated below, we will define exactly what criteria are being used to compare sensor-based weapons to physical.

Computational Tools

The computational tool set we developed comprises a representation for error distributions, techniques for determining the probability of correct outcome for both physical weapon engagements and sensor-based engagements, and techniques for computing tradeoff diagrams. A *tradeoff diagram* is a graph whose axes correspond to the error radii of two sensors. We draw curves in this space to represent the boundary of the set of points representing sensor combinations that meet or exceed a particular criterion. For example, we will show combinations of GPS and PAS for which sensor-based DF00 engagements achieve at least 99.5% of the correct outcomes achieved by the physical weapon.

This section will overview the computational techniques we use to compute both probabilities of correct outcome for particular sensor combinations as well as the tradeoff diagrams themselves.

Discrete Probability Distributions

The basic tool we need for computing tradeoff diagrams is the ability to approximate an integral over a probability distribution. Suppose we have a random variable X whose value represents some quantity in a calculation. For example, X could be the difference between the actual x-value of the position of a body and the x-coordinate reading of the body's GPS unit. We model X using a probability distribution, which (roughly speaking) represents the probability that if we take a GPS measurement we will get a particular measured value. In this study, we will use only 1-dimensional distributions, treating each dimension independently.

Our goal is to compute for given types of physical situation, whether the sensor-based weapon will compute the correct outcome for the engagement with probability at least as high as (or within a small tolerance of) the probability that the physical weapon will itself produce the correct outcome. The weapon's outcome can be "incorrect" if, for example, a well-aimed shot misses the target due to random variation within the spread. To compute these probabilities (both for physical weapon and for sensor-based weapons), we need to look at every possible situation-variant that could arise due to particular errors and variations, grade the outcome as correct or incorrect, and then add up the grades weighted by the probability of that situation-variant arising. For example, for rifle fire, we need to look at all possible choices of (a) aim point relative to center of target, (b) shooter's position error, (c) target's position error, and (d) point angle sensor error. Using these values, we compute the outcome geometrically and compare it to the outcome as it would be deduced from a perfect weapon aimed at the aim point. This basic method is used for all types of engagements. To compute physical weapon outcome correctness probability, we look at all possible choices of (a) aim point relative to center of target and (b) weapon spread deviation value. We then compare geometrically the perfect-weapon outcome (ignoring spread) and the actual-weapon outcome (including spread). Note that the *accuracy* of a weapon (or sensor) refers to characteristics of the probability distribution of its outcomes over time, while *correctness* refers to how well a single trial matches with reality.

Of course, it is impractical to look at *all* possible combinations of error values, as there are infinitely many. Instead, we will *discretize* the distribution, looking systematically at a finite sampling of its domain and looking at all combinations of the sample values of all distributions. We establish a finite, uniformly spaced *mesh* over the domain of the distribution, as shown below.

Figure 1 shows a Gaussian Normal distribution (Patel & Read, 1996) of 4-sigma radius R. All distributions in this study are centered at 0. We have discretized it using mesh 12, meaning we have chosen 12 evenly spaced points of the domain: -11R/12, -9R/12, ... 11R/12. We then do our calculations once for each of these points. Then, to compute the final probability, each point's result is weighted by the probability value at that point. Probability values are computed as follows: sum the values of the curve at all 12 points and then divide the value at each point by this sum. This produces 12 numbers between 0 and 1 whose sum is 1 and whose values are proportional to the height of the Gaussian at that point. For example, the diagram shows the point -3R/12 is weighted with probability 0.1613. I use mesh 12 here for illustration; our data is based on higher mesh values.

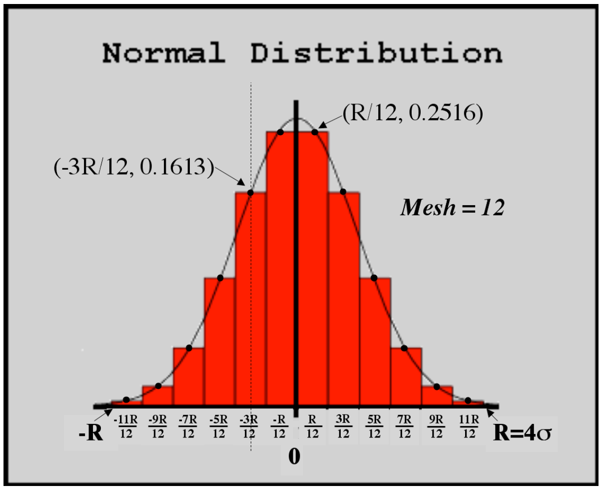

Figure 1: Discretized Normal Distribution (mesh 12).

This method ignores the infinite tails of the normal distribution, effectively assuming that *all* samples lie between $+/-4\sigma$. Our parameterization of sensor families will be in terms of the 3σ *radius* of the distribution; however, we represent and iterate over values across $+/-4\sigma$ for increased accuracy.

By *iterating over a (discretized) distribution,* we will mean this process of successively choosing each of the discrete points defined by the mesh, each of which is termed a *mesh point with associated probability weight,* performing some calculations, and then adding up results by weighting according to the probability weight of each mesh point for that case. This fundamental operation will underlie all of our studies below. Note that we can *nest* such iterations: after choosing a mesh point for a first distribution, we then iterate over all values of a second distribution. The weights used to combine the results from all such combinations are simply the products of the individual probability weights. For example, combining two distributions identical to the above, one case would have X1 = -3R/12 and X2 = R/12. The probability weight of this case is 0.1613 x 0.2516 = 0.04058308. For two mesh-12 distributions, there are 144 cases; in general, for n distributions with meshes m_i, there are $\Pi\ m_i$. It is easy to show that this method of combining distributions is again a (multi-dimensional) discrete distribution whose probability weights also sum to 1.

Thus, for nested distributions, the computational cost is proportional to the product of the meshes (and potentially exponential in the nesting depth). A number of optimization tricks are quite useful in reducing the constants of the complexity function, such as precomputing the mesh points and their probabilities and cleverly managing intermediate results, but fundamentally complex engagement geometries involving many error sources will lead to long computational runs to get results. Obviously, there is a natural desire to keep meshes small, but if they are too small, the results are not very accurate. Thus, part of the art of this study is choosing mesh values large enough for accuracy and small enough for tractability. This generally involves multiple runs to be sure the meshes are large enough to give stable results.

Another approach to integrating over error distributions we did not explore here is the use of *Monte Carlo Simulation* (Robert & Casella, 2004). The idea is that instead of systematically looking at weighted values as we do, one simply repeatedly selects randomly from each distribution and computes the situation result from the particular choices made. By repeating this for a large number of samples, a reasonably accurate estimate of the results can be obtained. While this may be less computationally intensive, since one can always choose the total number of samples to be any amount desired, it may lead to less accuracy for a given amount of computation, since the choosing process is less systematic. Thus far, the computational burden of our approach has not been too great, but it is possible we will need to move to Monte Carlo methods in future work.

Computing Probability of Correct Outcome

For each engagement type discussed below, we identify the geometric calculation that needs to be performed to compute the probability of correct outcome. Such calculations involve sensor readings, such as shooter and target positions and weapon pointing angle. For a given particular choice of sensors, we compute the probability of correct outcome of a sensor-adjudicated shot as follows. Represent each sensor by a discretized normal distribution of appropriate 3-sigma radius and choose a mesh. We then nest iterations over all of the selected distributions, perform the geometric calculation within the nested iteration, and compute from it whether or not the outcome is adjudicated correctly. For each correct outcome, we add the probability of this nested-sensor case (i.e. the product of the probability weights associated to the mesh points chosen at the iteration step) to the result sum. After the entire nested iteration is complete, this result sum is the desired probability of correct outcome given the selected combination of sensors.

Computing Tradeoff Diagrams

Tradeoff diagrams represent significant computational effort, since in theory each point in the space must be evaluated using the approach above. However, we can speed this up quite a bit using the following approach, explained for concreteness for the case of 2-D rifle fire. (Similar techniques apply to tradeoff diagrams for other types of engagement.)

Assume the x-axis of the graph will denote the 3-sigma radius of the 1-D GPS error. (Thus, for example, in 2-D Direct Fire, there will be four instances of this in each calculation, corresponding to x and y errors of Shooter and Target.) Assume the y-axis of the graph is the 3-sigma radius of the angular error distribution for the PAS. We will iterate over the x axis starting from 0 and increasing in fixed (e.g. 10 cm) steps. At $x = 0$, we start from some initial large y value (e.g. 10 milliradians). For $x > 0$, we start with y equal to the same y value of the curve point found for the previous x value. We compute the probability of correct outcome for the current x, y pair. If this probability value is greater than or within a small tolerance of the target value (typically, the probability of correct outcome for the physical weapon), then this is the next curve point and we increment x. If the probability value is significantly less than the target value, we decrement y by a fixed amount (e.g. 0.5 milliradians) and continue the iteration. Thus, instead of computing all the points in a 2-D area of the graph, we are following the curve down and to the right. This is possible because, as we increase the error in GPS, we must decrease (or keep the same) the error in the PAS.

The outputs of this iterative curve following are points in the 2-D tradeoff space. We plot these and fit them with a curve to get the final diagram. *The meaning of this graph is that all points either on or below/left of the curve represent sensor-error radius combinations that allow sensor-based adjudication to meet the defined accuracy requirement with respect to the physical weapon system.* Examples appear in the following sections.

RESULTS FOR DIRECT FIRE

Direct fire is typically used to mean munitions that travel on a linear trajectory from shooter to target (e.g. rifle fire). A hit is adjudicated if and only if the target's body intersects the trajectory prior to any other body (or terrain) intersecting it. This contrasts with indirect fire (next section), in which the round first travels via a curved trajectory to a detonation point, at which time and place a blast affects targets within an effect radius of the detonation point. This section describes error tradeoffs for sensor-based adjudication of direct fire rounds following 2-dimensional linear trajectories.

We will address three cases. First, we ask what it would take to achieve perfect simulation (a simulated shot scores a hit if and only if the corresponding physical shot could have). Next, since that turns out to be impossible, we look at what it takes to achieve perfect simulation for all targets beyond a given range. Finally, we examine the most practical case, where we limit the set of targets for which the outcomes must match. Throughout, we will use DF00 rifle fire, with a maximum range of 580 meters and weapon spread corresponding to 0.123 m groups at 100m.

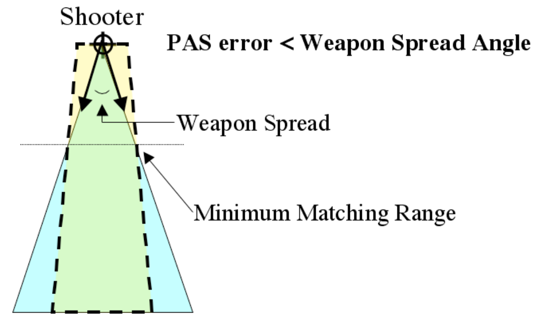

Figure 2: Sensor vs Physical for Direct Fire

While our tool set can handle non-perfect aim distributions, after conferring with Subject Matter Experts (SMEs), we determined that the most important case is matching weapon accuracy for well aimed shots. The reasoning is that shooters who can aim well are also those most likely to notice when sensor-based adjudication gives a wrong answer. Thus, for the rest of this paper, we will make the *perfect aim assumption:* all shots are aimed dead center of the target. Future study may determine a need to look at imperfect aim scenarios.

Perfect Direct Fire

Direct fire is impossible to mimic perfectly using sensor-based adjudication with imperfect GPS. (See Figure 2.)

The Shooter's position is indicated. The set of positions that the physical weapon can hit (labeled "weapon spread" in the figure) is the triangle whose vertex is at the Shooter. However, if the GPS sensors have nonzero error radii, then the points which can be hit by the sensor-based adjudication lie in the trapezoid indicated. Thus, there are points near the upper corners (yellow) which may be "hit" by sensor-based adjudication but which may never be hit by the physical weapon, no matter how small the PAS sensor's error radius may be. Note that this assumes that the target is arbitrarily narrow, and we are neglecting the width of the bullet itself.

Perfect Direct Fire at Minimum Range

While we can never mimic direct fire perfectly (with imperfect GPS), we *can* mimic it perfectly for shots beyond a minimum range, using a good enough PAS. If the sensor-hit trapezoid has a narrower angle than the weapon spread triangle, targets beyond the *Minimum Matching Range (MMR)* will be correctly adjudicated at least as often by the sensor-based system as by the weapon itself.

To calculate the MMR for a given sensor combination, denote the 3-sigma error radius of GPS by g, that of the PAS by p, and that of the weapon spread by w. Simple trigonometry shows

$$\text{MMR} = 2g/(\tan w - \tan p)$$

Now, a DF00 has $\tan w = 0.000615$. Given a GPS with 1.5m radius, for example, the smallest possible MMR is 3/(0.000615 - 0) = 4878m. The effective range of the DF00 is only 580m, so it is impossible to match DF00 fire for all shots and ranges using 1.5m GPS.

To see how good GPS and PAS must be to get an MMR value of v, we set $\text{MMR} = v$ and graph p versus g. Since w and p are very small, $\tan w \approx w$ and $\tan p \approx p$. This produces linear graphs (suppressed here for space). As an example, to match all targets beyond $\text{MMR} = 100$m, sensors must perform within a line whose PAS-intercept is 0.000615 radian and whose GPS-intercept is 0.031m. These are tight tolerances, indeed.

Direct Fire with Minimum Target Width

We can gain more leeway in sensor accuracy by only agreeing to match weapon outcomes for targets of a minimum width. That is, part of our problem has been that matching weapon outcomes for extremely narrow targets (for example, hitting a bullseye in the middle of a target) is much more difficult than matching outcomes for larger targets (hitting or missing the entire target).

Battlefield targets come in many sizes, from humans whose width can range from roughly 0.5 meter when standing up to 2 meters when lying prone (side view). Vehicles and buildings are, of course, even larger.

For this study, we will determine a tradeoff diagram for meeting weapon accuracy for targets 1 meter wide. We will discuss at the end how the results vary for smaller or larger width targets.

To compute these curves, we apply the techniques of Section 2 to the following geopairing calculations. First, to compute the probability of correct outcome for the physical weapon, we iterate over aim and spread distributions (recall, however, our perfect aim assumption for this study) and compute hit or miss according to the geometry in Figure 3. Here, d is the nominal range of the shot, θ is the resultant angle obtained by adding aim deviation to spread deviation (which is just the spread deviation, since aim deviation is assumed 0 here). Common rifles (including DF00) at the ranges we considered have small enough spreads that the physical correctness probability is essentially 1.0.

To compute probability of correct outcome for sensor-based adjudication, we use the slightly more complex

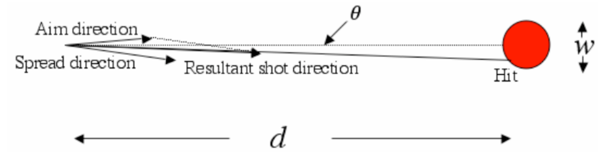

Figure 3: Physical Shot Geometry for Direct Fire.

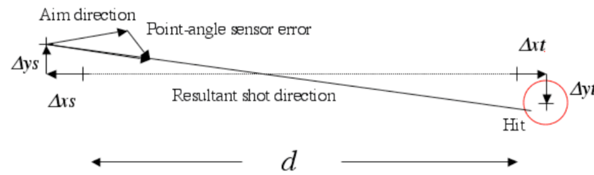

Figure 4: Sensor Shot Geometry for Direct Fire.

geometry shown in Figure 4. Here, we have both aim and PAS errors to determine the resultant shot direction. We also have the four deviations due to GPS (Δxs, Δys, Δxt, Δyt, two each for shooter and target). Again, the aim deviation will be zero in this study.

The resulting tradeoff curves are shown in Figure 5 for different nominal engagement ranges. They represent *all possible* direct fire weapons having near-100% accuracy at up to 400m range, so need not be obfuscated.

These data points are only approximate but, we believe, accurate within 5%. Each was produced with a particular mesh level and then run again with significantly higher mesh level. If the two values matched, the point is recorded. Additionally, iteration was done across four sigma for each distribution. Note if we were using ideal normal distributions, we would expect the data points at GPS=0.0 to be slightly higher than 20, 10, 5, 2.5, and 1.25 respectively. But our discretization results in the stable approximated values of PAS error when GPS error is 0.0 to be about 5% higher than expected. On the other hand, a heuristic argument indicates the (common) x-intercept of all five curves should be 0.375, which is

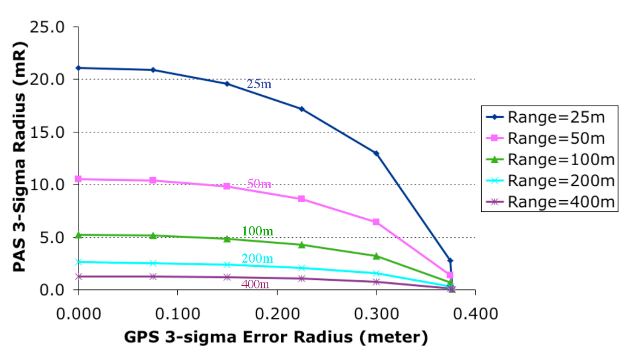

Figure 5: Direct Fire Tradeoff Diagram (1m target)

extremely close to the approximated value of 0.377. If we wish higher accuracy, we would use the triple-mesh-average technique described in the next section.

Other Target Widths

The results above apply directly only to engagements involving targets 1 meter wide. It turns out that we can easily infer from the 1m graphs new graphs for any target width large enough that the weapon is 100% accurate. Suppose we are interested in target width w instead. We simply multiply each coordinate of each data point by w to obtain a data point on the same range curve but for target width w. For example, if we are interested in width 0.5 meter, we would obtain a Range=100m curve whose y intercept is 2.63 milliradians (mR), whose x intercept is 0.189 meter, and each of whose points is the scaled image of one of the points shown in the 1m (Range=100m) curve. Compared with the results for MMR=100m, for example, this is much more attainable (GPS: 18.9 cm vs 3.1cm; PAS: 2.6 mR vs 0.6 mR). Proof of this *multiplicative property for target widths* is suppressed here.

RESULTS FOR INDIRECT FIRE

Indirect fire (IF) rounds detonate on contact, affecting targets within a given radius of the detonation point. (We are not considering air burst weapons here.) For this type of weapon, we need to account for this area effect. That is, we would like to evaluate the accuracy of the weapon with respect to the quantities of (a) correct hits, (b) incorrect hits, (c) correct misses, and (d) incorrect misses. In the direct fire case, there can only be 1 hit. For each indirect fire engagement, however, there can be several correct hits as well as several incorrect hits (i.e. targets adjudicated to be affected even though they would not have been affected by the physical round) and several incorrect misses (i.e. targets adjudicated to be unaffected even though they would actually have been). This section defines our methods and evaluates sensor tradeoffs for both trajectory targeted, such as rifle-mounted grenade launchers, and position targeted IF weapons, such as larger gun platforms.

Trajectory Targeted Indirect Fire

We use the *circle overlap* method for computing accuracy of an IF shot. Consider the physical weapon first. If it performs perfectly, a circle around the aimed detonation point contains all affected targets and no unaffected targets. However, weapon spread moves the affected circle somewhat (see Figure 6). The *accuracy score* for an IF shot is the area of the intersection divided by the area of the union, giving a number between 0 and 1, with a perfect shot being the only way to score 1.0:

$$\text{IF-Accuracy} = \text{Area(CH)}/\text{Area(IH} \cup \text{CH} \cup \text{IM)}$$

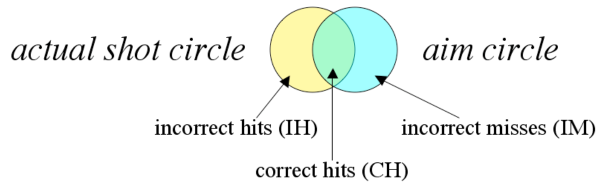

Figure 6: Physical Shot Geometry for Indirect Fire.

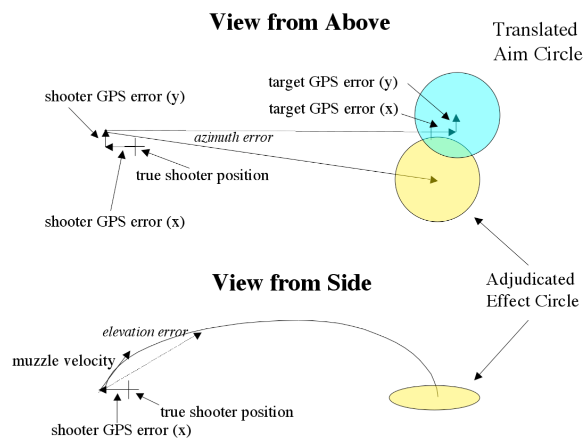

Figure 7: Sensor Shot Geometry for Indirect Fire.

We use the same basic technique for sensor adjudicated IF, but instead of weapon spread, we use combined sensor readings. We consider here *trajectory targeted IF,* by which we mean that the round's trajectory is determined from two point-angle sensors (one for azimuth and one for elevation) and the GPS sensors of shooter and target. We will discuss *position targeted IF* later in this section. The translated aim circle is simply the perfect-aim detonation point (calculated from simple Newtonian physics without air drag) translated by the target GPS error. The adjudicated effect circle is computed from the sensor-based position of shooter and values of the shooter's point angles, again based on 3-D Newtonian physics ignoring air drag. The accuracy score for the sensor-based adjudication is then the circle overlap score as computed above between the translated aim circle and the adjudicated effect circle. (See Figure 7.)

We applied the computational techniques discussed previously to approximate tradeoff curves at various nominal ranges. (Nominal range is the physical range, before perturbation by GPS errors.) Here we used the standard that the accuracy score of sensor-based must meet or exceed that of the physical IF weapon.

Using a single mesh value for PAS distributions results in spurious curve fluctuations, so we used instead the *triple-mesh-average method:* each data point's PAS

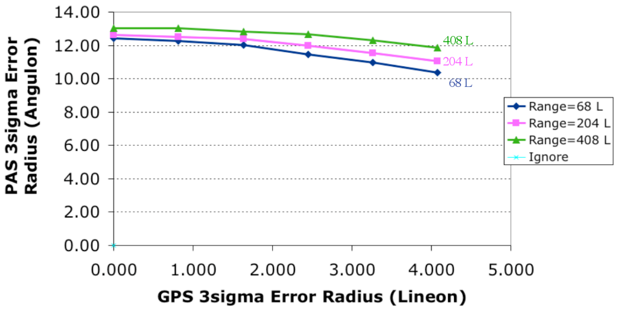

Figure 8: IF01 Sensor Tradeoffs.

value is no more than 1% above the minimum PAS value (for that GPS value) having the average of its sensor accuracy scores (obtained with meshes 60, 70, and 80) exceed the physical accuracy score. Intuitively, we traced out the average of the curves for meshes 60, 70, and 80.

We show in Figure 8 the results for the IF01 IF weapon. Our accuracy model of the IF01 is abstracted from military standards, but we have obfuscated the data here as follows. We define a new angle measure termed the "angulon" (abbreviated "A") which we fix at a particular *Angulon conversion factor ACF* relative to the milliRadian; i.e. 1 mR = *ACF* A. We do not disclose here the value of *ACF*. Similarly, we define a new linear distance measure termed the "Lineon" (abbreviated "L") with a fixed *Lineon conversion factor LCF* relative to the meter. We then multiplied the computed study data by *ACF* and *LCF* to produce the numerical data disclosed here.

The weapon spread of the IF01 amounts to an effective angular radius of 12.4 Angulons (A) in elevation and 15.5 A in azimuth. The maximum effective range of the IF01 is 435.2 Lineons. Interestingly, the most stringent conditions are those for shortest range, while for direct fire, the opposite is true. (This is due to the fact that shorter range IF shots are more horizontal and, hence, more sensitive to overshoot and undershoot due to PAS errors.) Again, using the triple-mesh-average method, we believe the data points are within 1% of true values.

Though we have not computed curve points beyond those shown for nonzero PAS, we *have* determined the x-intercepts of the three curves: the maximum GPS error radius allowing sensor-based adjudication to meet weapon accuracy (i.e. assuming 0.0 PAS error). For the 68 L curve, this is 10.1 L GPS; for the 204 L curve, 10.5 L; and for the 408 L curve 10.9 meters.

In our study, the IF02 weapon is both longer range and more accurate than the IF01. Due to space restrictions, we cannot show the curves; however, they are much tighter than those of the IF01, lying between 2.32 A and 2.71 A for GPS radius between 0.0 and 4.08 L.

Position Targeted Indirect Fire

Some weapon systems, such as the IF03, are targeted by the operator specifying coordinates of the desired detonation point, rather than by pointing the gun tube himself as in the IF01 and IF02 cases. We term this type of IF weapon a *position-targeted* IF system. The instrumentation system's job here is much easier, because it can obtain the target point directly from the system. To adjudicate the shot, one need only decide whether the GPS position of the target is within the effect circle.

Any sensor errors within the platform itself (such as its own GPS and PAS) will increase weapon spread, so the instrumentation system is not "penalized" for these errors. It needs only to account for the error in the target's position. Our analysis of the accuracy of these systems is based on the simple observation that if the target's GPS error radius is smaller than the radius of the weapon spread, then the target is more likely to be adjudicated as hit than to be actually hit by the weapon. Conversely, if the target's GPS error is larger than the weapon spread, then the physical weapon is more likely to hit the target than is the sensor-based system likely to adjudicate a hit. Thus, the criterion for meeting the accuracy of position-targeted systems is *GPS error radius must be no larger than the corresponding radius of the weapon spread.*

For example, an IF03 IF system can place a Projectile04 projectile within 4.6 L of the target point. Assuming this means the 3-sigma weapon spread radius is 4.6 L, the GPS 3-sigma error radius must be no worse than 4.6 L in order to meet or exceed the accuracy of the IF03. Obviously, this criterion can be applied to any position-targeted IF system, once the spread is known.

RESULTS FOR TERRAIN

Geopairing must account for the presence of terrain features as well as shooter and target positions, because (a) targets can be shielded by terrain, and (b) terrain elevation must be taken into account in engagement adjudication, particularly for indirect fire; this is because elevation sensing is much less accurate than using the 2-D position to locate the elevation in a terrain data base. Thus, the third aspect of sensor error we will study here is the accuracy of the terrain database in representing features and elevations. We have studied three terrain problems: the *Foxhole Problem,* the *Direct Fire Edge Problem,* and the *Elevation Uncertainty Problem.*

The Foxhole Problem

The *Foxhole Problem* asks for the effect of terrain inac-

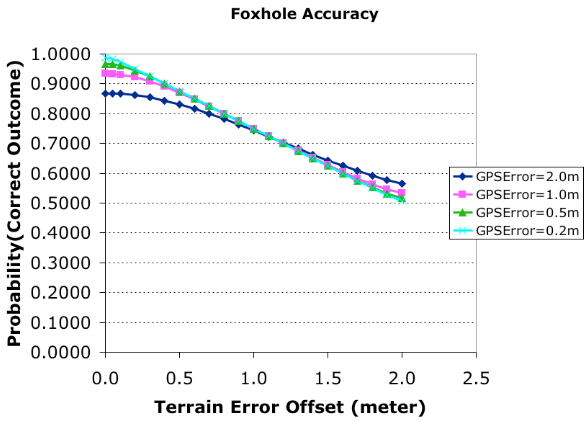

Figure 9: Foxhole accuracy with terrain error.

curacy on the ability of the system to decide whether an entity is on one side or another of a terrain edge. A critical instance of this capability is to determine whether a target is inside a foxhole (and hence shielded from a detonation) or outside (and hence vulnerable to damage from the detonation). We considered here a 1 dimensional problem, i.e. that of determining whether an entity is left or right of an edge positioned at $x = 0$. We discuss also some generalizations to multi-edge combinations, but this work is only beginning to explore the space of interesting edge configurations. (**Note:** since terrain edges are not weapon systems, this subsection's numerical results are not obfuscated.)

There are several interesting aspects of our result graphs. (See Figure 9.) **(1)** If GPS error is non-zero, then a GPS-based approach will never be able to adjudicate foxholes with 100% accuracy. This is because the GPS error may randomly move a person standing near the edge to the other side. **(2)** All curves cross at the same point, terrain error = 1.1m (probability 0.725). **(3)** For values of terrain offset approximately 0.25m and below, there is very little increase in accuracy for decrease in terrain error. **(4)** For small terrain errors, more accurate GPS is better, yet for terrain errors beyond 1.1 meter, the *less* accurate GPS is *better*. We believe this is because a more accurate GPS cannot offset the position enough to put the person back into the foxhole, while a less accurate GPS can do so sometimes, thereby increasing the success probability. This counterintuitive fact shows that while there is a limited tradeoff possible between GPS accuracy and terrain accuracy for terrain errors below 1.1m, when terrain errors are above that, the tradeoff is actually reversed: moving to worse GPS is compensated for by moving to worse terrain error! (In other words, to improve GPS, one must also improve terrain accuracy or else the effect is negative.) We also analyzed two-edge situations, finding that the probability of correct outcome is squared.

As one example of generalization to multiple edges, consider two perpendicular edges (i.e., a corner of a square foxhole). The entity's position is independently placed relative to each of the two edges, so we simply square the success probabilities, yielding graphs similar to those of Figure 9, but with each point's vertical coordinate squared. Clearly, more complex situations, such as opposite parallel edges, are not statistically independent and will need to be modeled using our techniques.

The Direct Fire Edge Problem

The *Direct Fire Edge Problem* asks for the effect of terrain inaccuracy on the ability to correctly adjudicate whether a target is shielded from direct fire behind a vertical terrain edge. Here again we model a simple 1-D situation. The position of the player is uniformly distributed from -2 to +2 meters relative to the edge. The player's GPS is, as usual, normally distributed. The terrain offset is constant as in the foxhole study. We iterate over the nested distributions and in each case we evaluate the probability that a direct fire shot at range 100m at a 1-meter wide body placed relative to the terrain edge is adjudicated correctly with respect to reality. (Note that when an edge is present, the physical weapon will be imperfect as well, because even a small spread can lead to a round terminating on the unintended side of the edge.) The resulting graphs of correctness probability versus terrain error for representative choices of GPS/PAS show the remarkable fact that *there is no tradeoff of terrain accuracy with other sensor errors in direct fire.* The resulting graphs are linear and coincident from terrain error between 0.4 and 1.8 meters, with small variations outside that range. That is, while a more accurate GPS can to some degree compensate for a less accurate PAS, nothing can compensate for larger terrain errors, and we cannot gain any leeway with GPS or PAS by paying for better terrain accuracy.

The Elevation Uncertainty Problem

The *Elevation Uncertainty Problem* asks for the effect on sensor tradeoffs for trajectory targeted IF when the elevation (as recorded in the terrain database) is inaccurate. Assuming we use the 2-D GPS position to locate the player in the terrain map and then use the recorded elevation, such inaccuracies lead to *overshoot* and *undershoot.* (If instead GPS is used for elevation measurement, the terrain database does not affect indirect fire adjudication in this way.) If the terrain is, for example, actually 1 meter lower near the detonation point of the IF round than the terrain data base claims, we have an undershoot situation. Similarly, if it is 1 meter higher, then we get overshoot. (See Figure 10.)

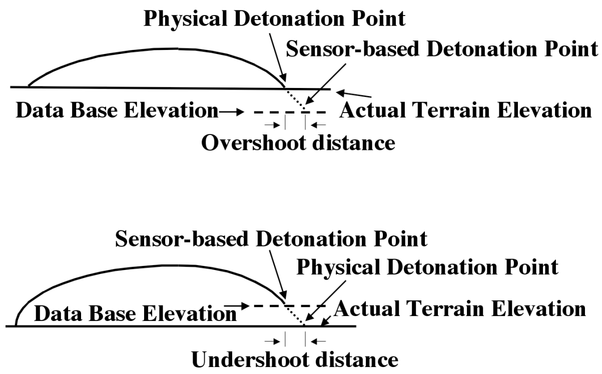

Figure 10: IF inaccuracy due to terrain error.

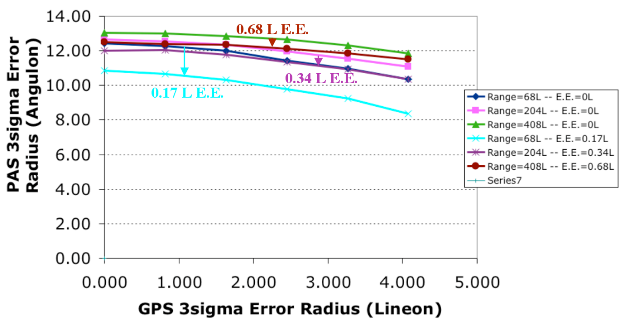

Figure 11: IF01 Tradeoffs with Elevation Error.

Overshoot can be approximated from the elevation angle as $O = E/\tan\phi$, where ϕ is the elevation angle of the shot, E is the elevation error (difference between the database-elevations of shooter and target minus difference between actual elevations of shooter and target). Thus, for shallow angle shots, which IF01 and IF02 shots must be if used within doctrinal ranges, there is a large multiplying factor. For example, one Lineon of elevation error translates to tens of Lineons of overshoot for the IF01 used at a range of 68 L.

So, if the overshoot is not too large, can we compensate for terrain errors with tighter bounds on the GPS and/or PAS? The answer is "yes", but there are limits. To compute tradeoff curves showing achievable compensations, we add in the (fixed) terrain elevation error offset into the IF calculations described previously and then use the triple-mesh-average method to discover curves for each weapon/range combination. Figure 11 shows one representative compensation tradeoff curve for each of three ranges for the IF01. The labels show the amount of elevation error the curve compensates. Each arrow associates a no-error curve with the corresponding with-error curve. For example, the light blue arrow shows that when 0.17 L of elevation error is added for 68 L shots, the GPS/PAS tradeoff curve drops down about 1.55 A.

For the weapon/range combinations shown, it *is* the case that lowering any of terrain error, GPS error, or PAS error values from those in the graph results in better sensor-based performance. However, counterintuitively, this *monotonicity* does *not* hold for *all possible* weapon/range combinations. For example, our techniques show that an IF weapon roughly one sixth as accurate as (i.e. much less accurate than) the IF01 could be simulated by a 20.4 L GPS radius at 0.567 meter elevation error, and yet could *not* be simulated by a *perfect* (0.0m) GPS at the same elevation error! We plan to document this phenomenon more fully in future work.

CONCLUSION

Meeting physical weapon performance with sensor-based adjudication is a major challenge for sensor and communications technologies. In this paper, we have introduced computational techniques for defining the tradeoffs among GPS, PAS, and terrain database errors. We have summarized here the results of applying these techniques in an initial study of direct fire (DF00), indirect fire (IF01, IF02, IF03), and terrain errors (foxhole, direct fire edge, and elevation error problems). We feel this study has taken significant initial steps in helping to define the space of allowable sensor combinations, as well as defining computational techniques that can be used in future studies.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of the U.S. Army's OneTESS program under contract N61339-00D-0706-0008.

REFERENCES

Hall, R.J. (2005) Combinatorial communications modeling of real-time tactical engagement adjudication architectures. In *Proc. 2005 IEEE Military Communications Conf.* IEEE

Parry, S. (ed) (1995) *Military Operations Research Analyst's Handbook*, Military Operations Research Society. Alexandria, Virginia.

Patel, J.K. & Read, C.B. (1996) *Handbook of the Normal Distribution*, New York: Marcel Dekker.

Robert, C.P. & Casella, G. (2004) *MonteCarlo Statistical Methods, 2e*. New York: Springer-Verlag

SRI International (2006) `www.dfirst.com`

U.S. Army (2006) OneTESS Program. `www.peostri.army.mil/PRODUCTS/ONETESS`