Use of Control Variates in a Large-Scale Aggregated Combat Model

Earl M. Bednar

Kenneth W. Bauer, Jr.

J. O. Miller

Department of Operational Sciences
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
Earl.Bednar@rl.af.mil
[Kenneth.Bauer;John.Miller]@afit.edu

In early fall 2004, the Air Force Studies and Analyses Agency (AFSAA) forwarded a proposed research topic, endorsed by the Air Force Chief of Staff, to the Department of Operational Sciences at the Air Force Institute of Technology. This topic requested an examination of potential variance reduction techniques (VRTs) for use with the THUNDER combat model. THUNDER is a large-scale, highly aggregated, discrete-event simulation of campaign-level military operations used to examine issues involving the utility and effectiveness of air and space power in a theater-level, joint warfare context. A new, state of the art combined multiple recursive generator was incorporated into THUNDER to synchronize the random inputs for common random numbers (CRN) and antithetic variates (AV). This work discusses the application of CRN, AV, and control variates (CV), within THUNDER, and the resulting variance reductions obtained for an example scenario.

Keywords: Simulation, variance reduction techniques, combat modeling, THUNDER

1. Introduction

The Department of Defense (DoD) uses a variety of large, stochastic combat simulations to conduct various analyses of interest. These vary from acquisition to planning to training. One of the large, stochastic combat simulations used by the Air Force (AF) is THUNDER. THUNDER is a stochastic, two-sided, analytical simulation of campaign-level military operations developed in the 1980s under the guidance of the Air Force Studies and Analyses Agency (AFSAA). The simulation was designed and built expressly to examine issues involving the utility and effectiveness of air and space power in a theater-level, joint warfare context.

***JDMS*, Volume 4, Issue 1, January 2007 Pages 29–41**

Given the large number of stochastic components within THUNDER, the model often produces measures of effectiveness (MOEs) with significant variance. This variance often causes difficulties in evaluating alternative force structures, weapon systems, etc. With time sensitive analysis and long replication times, AFSAA desires an effective and efficient way to reduce the variance in the results from the THUNDER campaign-level model.

The objective of this research is to study the use of three variance reduction techniques on THUNDER. The three techniques are control variates (CV), common random numbers (CRN) and antithetic variates (AV). First, we identify a set of MOEs that are commonly used by THUNDER users. This helps ensure that the results will be of some interest to the users. Second, we modify the THUNDER source code to implement each of the techniques. For CV, this consists of identifying areas where random variates are drawn and adding logic to create an output file in which to accumulate the average values of the random inputs. For CRN and AV, this involves implementing a new random number generator. Third, we analyze the results from the variance reduction techniques to determine which, if any, are effective.

In section 2 we discuss THUNDER and the selected scenario as well as the MOEs used for our analysis. Section 3 provides details on the new random number generator (RNG) incorporated into THUNDER. In section 4 we briefly describe the implementation of CV and present our analysis results for CRN and CV, and in section 5 we present our main conclusions. Full details are available in Bednar [1].

2. THUNDER

There are many reasons why a system is modeled by computer simulation. For instance, a system may be too difficult or expensive to alter, the system may not exist, or it may be unacceptable to operate the system in real life. In combat simulations, the latter is probably the case. It would be unethical to start real combat merely to test a new weapon system. The goal in the design of combat models, such as THUNDER, is to represent warfare as accurately as possible, because results from combat simulations can guide national defense and other policy decisions.

There are many different levels of combat modeling. They range from engineering to campaign-level simulations; see Figure 1 [2]. A campaign level simulation encompasses a whole campaign and covers many different elements. The sheer volume of entities would make it difficult to simulate each entity interaction; therefore, they are performed with low resolution and a high level of aggregation. This high level of aggregation means that each entity is not simulated explicitly but the ensemble of such entities are replaced by probabilities and percentages. This allows a large campaign-level simulation to run in an acceptable time frame, at the cost of losing output resolution.

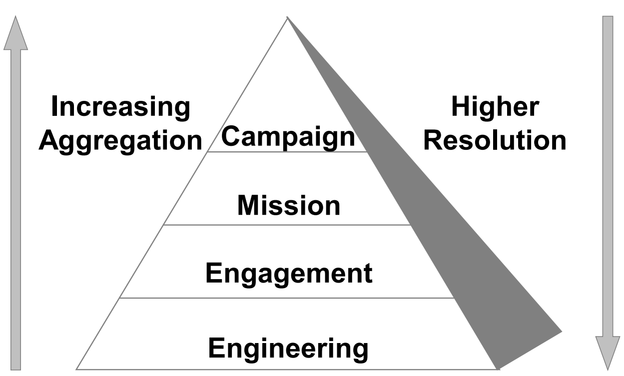

Figure 1. Combat modeling hierarchy [2]

THUNDER is a large-scale stochastic, two-sided, constructive simulation that has been operational since 1986 and was developed to examine the utility and effectiveness of air and space power at a campaign level and model joint military operations. The Air Force Studies and Analyses Agency (AFSAA) sponsored the development of THUNDER. The following description is adapted from the THUNDER web pages [3] and volume one of the THUNDER documentation [4].

THUNDER is designed for analytical flexibility, accessibility, and ease of use. It operates in a UNIX environment and is written in SIMSCRIPT II.5, a simulation programming language. SIMSCRIPT II.5 is ideal for the creation of large-scale, discrete-event simulations [5, 6]. To aid the analyst, THUNDER comes with in-depth documentation and source code, allowing modification at the programming level. The source code of THUNDER consists of around 360,000 lines of code spread over 1,500 routines. THUNDER operates at multiple levels of detail. It models air operations at a high level of detail, with the ability to simulate 27 different air missions. While it can simulate air operation at a high level of detail, it models ground operations at a medium level of detail, focusing on ground combat; mobility; logistics; terrain; and Command, Control, Communications, and Intelligence (C3I).

To assist the user and speed up replications, THUNDER automatically generates its own Air Tasking Orders (ATOs) and Intelligence Tasking Orders (ITOs), based on theater-level apportionment and target priorities. It also uses a deterministic time stepped ground operations model based on the Center for Army Analysis Concept Evaluation Model (CEM) and its attrition-calibration (ATCAL) methodology.

THUNDER can be run in two modes: analytical and wargame. The analytical mode focuses on the contribution of systems, capabilities, forces, and employment concepts. It is run in a batch mode and delivers results that can be identified by replication or as an average across replications with confidence intervals. When run in batch mode, THUNDER starts each random number stream for successive replications where each previous stream ends for the previous replication. The wargame mode supports user interaction in a game-like atmosphere, where a player dynamically influences the outcome of a run.

As a member of the Air Force Standard Analysis Toolkit (AFSAT), THUNDER supports senior decision makers throughout the Air Force, Department of Defense, and other national-level organizations in matters of policy, acquisition, and operations. Typically, these customers require precise and timely analysis. The analysis applications include: exploration of readiness, modernization, force structure, and sustainability issues; comparison of alternative courses of action (ACA); assessments of evolving capabilities, alternative strategies, and potential operational concepts; and the facilitation of senior staff training through wargaming.

There are three notional model databases that are used as examples and come with THUNDER. They are called Databig, Datasmall, and ME. Most THUNDER model databases are classified due to the amount and the nature of the data in the model. Therefore, one of the notional models was selected for the THUNDER analysis so that this research would be unclassified. Since the THUNDER introductory course uses the ME, a Middle Eastern scenario, it was selected as the baseline for this research. It is a five-day scenario consisting of blue forces from Saudi Arabia and red forces from Iraq. Running just the standard ME scenario is applicable for both CV and AV, since they both are single-model variance reduction techniques (VRTs). On the other hand, the application of CRN requires at least two configurations or models to be compared. (See the appendix for a short tutorial on the CV, AV, and CRN VRTs.) Therefore, a modification to the original ME configuration is needed to test and demonstrate a CRN application. We elected an easily implemented modification to a single input file, *squadron.dat*, that should have a measurable impact on the selected MOEs. To achieve a significant effect, 204 F-111 aircraft were taken from the blue forces. This eliminated all of the F-111s from the scenario. No modifications were made to any other ME input files.

The MOEs for the ME scenario were selected after discussion with expert THUNDER users [7]. Three types of MOEs identified were forward line of troops (FLOT) movement, sortie generation, and aircraft attrition. These MOEs can be extracted from standard THUNDER reports. The two types of reports needed are Ground Combat Cycle (CC) reports and the Air War (AW) reports. The output data for the FLOT movement is located in the CC-4 (FLOT movement) reports, the data for sorties generated is located in the AW-5 (sortie generation) reports, and the data for aircraft attrition is located in the AW-3 (aircraft loss

Table 1. Measures of effectiveness

MOE	Aggregation	Force	Report
A-10 Losses per Sortie (LPS)	System Level	BLUE	AW-3
A-6 Losses per Sortie (LPS)	System Level	BLUE	AW-3
BLUE Flown per Planned Sortie (FPS)	Force Level	BLUE	AW-5
BLUE Losses per Sortie (LPS)	Force Level	BLUE	AW-3
BLUE Square Miles Gained (SMG)	Force Level	BLUE	CC-4
DHAHRAN Flown per Planned Sortie (FPS)	Base Level	BLUE	AW-5
F-111 Losses per Sortie (LPS)	System Level	BLUE	AW-3
F-15 Losses per Sortie (LPS)	System Level	BLUE	AW-3
FA-18 Flown per Planned Sortie Total (FPS total)	Squadron Level	BLUE	AW-5
FA-18 Flown per Planned Sortie DCA (FPS DCA)	Squadron Level	BLUE	AW-5
MIG-23 Losses per Sortie (LPS)	System Level	RED	AW-3
MIG-29 Losses per Sortie (LPS)	System Level	RED	AW-3
MIRAGE Flown per Planned Sortie Total (FPS total)	Squadron Level	RED	AW-5
MIRAGE Flown per Planned Sortie INT (FPS INT)	Squadron Level	RED	AW-5
MIRAGE Losses per Sortie (LPS)	System Level	RED	AW-3
MUDAYSIS Flown per Planned Sortie (FPS)	Base Level	RED	AW-5
RED Flown per Planned Sortie (FPS)	Force Level	RED	AW-5
RED Losses per Sortie (LPS)	Force Level	RED	AW-3
RIYADH Flown per Planned Sortie (FPS)	Base Level	BLUE	AW-5
SHAIBAH Flown per Planned Sortie (FPS)	Base Level	RED	AW-5

Note: Defensive Counter Air (DCA), Interdiction (INT) summary) reports. All three reports are contained within a single output file named *reports.std*. A set of 21 specific MOEs were selected from the three reports. Table 1 identifies all MOEs, aggregation levels, forces, and the reports from which they are derived.

3. Random Number Generation

To use the CRN technique most effectively, each pseudo–random number call needs to have a unique random number stream. This allows for synchronization of these calls between replications for different simulation configurations. THUNDER already has a way to assign different random number streams. The problem with THUNDER's current random number generator (RNG) is that there are only ten streams to choose from and THUNDER makes at least 254 different random number calls. Therefore, a new random number generator was needed to allow each random number call to have an assigned stream. In this research, the random number generator suggested by Pierre L'Ecuyer, Richard Simard, E. Jack Chen, and W. David Kelton [8] was added to THUNDER.

Currently, most programming software use small linear congruential generators (LCGs) as random number generators. It is recognized by simulation experts that small LCGs, with moduli around 2^{31}, should not be used for most purposes. In fact, the period on most any LCG can be exhausted in a matter of minutes on a standard PC, and LCGs are known to have poor point structure, both of which can bias simulation results. L'Ecuyer proposed to replace the more common LCGs with a more robust RNG method called a combined multiple recursive generator (CMRG). L'Ecuyer et al. [8] have produced a CMRG RNG package, which is readily available on the Internet. A parameter set for a CMRG, from a previous paper by L'Ecuyer [9], was selected as the backbone of the proposed RNG package. The parameter set is called MRG32k3a [10] and was one of many sets that were discovered after a year's worth of computer processing time. MRG32k3a has a long period of approximately 2^{191} or 3.1×10^{57}.

To create multiple streams, the full period needed to be broken into smaller sections that are equal in length. The goal was to find a size for the streams and substreams that are relatively long and have good statistical properties. In essence each stream and substream should resemble a stand-alone random number generator. To do this L'Ecuyer et al. [8] examined the lattice structure of substreams with lengths ranging between 2^{51} and 2^{150} numbers. The combination of each stream consisting of 2^{127} numbers with 2^{51} substreams consisting of 2^{76} numbers was chosen because it has a sound statistical structure. A CMRG can be viewed as an LCG in matrix form, which for MRG32K3a allows use of four specially structured 3×3 matrices to advance the state of the RNG by either 2^{127} (length of a stream) or 2^{76} (length of a substream) numbers with a single matrix multiplication. This allows the user to immediately jump to the next stream or substream. These innovations by L'Ecuyer et al. greatly facilitated the implementation of MRG32K3a into THUNDER [8].

4. Implementation of VRTs and Analysis

In the late 1980s, an initial effort was undertaken to identify and allow synchronization of random variates generated in THUNDER. Reviewing this work through examination of the source code, 245 of the 254 points where random variates are drawn were already identified. The *control.dat* file of THUNDER's required input files contains the list of random inputs (code requiring a random number draw) and allows the user to assign random number streams to each of the random inputs. The nomenclature used to label the random input identifies which module the random input is from, and a number is added to account for more than one random input from a module. For example, *AIR070.2* is a random input from module *AIR070.SIM*, and it is the second random input from that module. Implementing CRN and AV simply required specifying a unique stream for each random number draw (designated as synchronized in Table 2) and setting a switch for AV in the new RNG.

The method of CV relies on the output of a random variate where the expected value is known. In simulations, the random variates generated throughout a replication are effectively random and have a known expected value. Therefore, all the random variates generated in a simulation are potential control variates and should be identified. When creating an output file of random variates, the variates should be labeled and produced with the label, type, parameters, and result. Therefore, the variates can be identified by the labels, the average can be calculated from the result, and the expected value can be determined by the type and parameters.

Since the random variate draws had been identified, the problem involved understanding how the variates are drawn, what they are used for, and how to output them. There are two methods used in THUNDER to draw a random variate. The first method is where a random variate is called in a logical expression. To capture the result, the random variate is drawn immediately prior to its use in THUNDER and placed in a variable called *EARL.RESPONSE*.

Following the random variate draw, any required probability distribution parameters (none required for a uniform (0,1)) are placed in variables *EB.P1*, *EB.P2*, and *EB.P3*. *EARL.RESPONSE* is then sent to an output routine *EARL.STORE*, along with the variate name, variate type (distribution family), and all three parameters. *EARL.STORE* was specifically built to organize the random variate output for ease of post-processing. After calling the output routine, THUNDER uses *EARL.RESPONSE* in place of the original random variate draw.

The second method to draw a random variate in THUNDER is through a subroutine. This method allows the user to set the random variate distribution in an input file. Therefore the subroutine identifies the parameters and which distributions are needed and then draws from that distribution. The results from these random variate draws are returned from the subroutine and stored in a variable to be used by THUNDER. To collect the results, code is added after

Table 2. THUNDER input sets

Set	RNG	Synchronized	Antithetic	Collect CV	Configuration
A	Old	No	No	Yes	Original
B	New	No	No	No	Original
C	New	No	No	Yes	Modified
D	New	Yes	No	Yes	Original
E	New	Yes	No	Yes	Modified
F	New	Yes	Yes	Yes	Original
G	Old	No	No	No	Original
H	New	Yes	No	No	Original

Table 3. CRN results (30 replications)

MOE	Base Half-width	CRN Half-width	Percent Change
A-10 LPS	0.00194	0.00159	**-18.01%**
A-6 LPS	0.00518	0.00386	**-25.66%**
BLUE FPS	0.01024	0.00854	**-16.67%**
BLUE LPS	0.00077	0.00066	**-14.36%**
BLUE SMG	401.95764	357.53075	**-11.05%**
DHAHRAN FPS	0.02990	0.02736	**-8.51%**
F-15 LPS	0.00140	0.00161	14.81%
FA-18 FPS Total	0.01610	0.01673	3.87%
FA-18 FPS DCA	0.00550	0.00459	**-16.48%**
MIG-23 LPS	0.03233	0.02409	**-25.48%**
MIG-29 LPS	0.02765	0.03097	11.98%
MIRAGE FPS Total	0.01711	0.02523	47.47%
MIRAGE FPS INT	0.02960	0.03048	2.99%
MIRAGE LPS	0.01957	0.01501	**-23.30%**
MUDAYSIS FPS	0.01687	0.01767	4.73%
RED FPS	0.00627	0.00684	8.95%
RED LPS	0.00591	0.00727	22.99%
RIYADH FPS	0.00636	0.00582	**-8.49%**
SHAIBAH FPS	0.03402	0.04037	18.66%

the subroutine returns the random variate, but before it is used. The random variate result, along with the random variate name, type, and parameters are sent to the same *EARL.STORE* as in the first method.

Since there were three types of variance reduction techniques to analyze and a new RNG, different sets of THUNDER input files were needed to generate the required data for the desired comparisons. A total of eight different configuration settings were identified as shown in Table 2, with 300 replications run at each setting for 2400 total THUNDER replications. The number of replications for each set of input conditions was selected to ensure at least one run for every potential control variate. Our initial analysis showed at most seven statistically significant (at α = 0.05 level) controls for any of the individual MOEs. Therefore, the results presented here for CRN and CV are for thirty replications at each configuration, which is more in line with the actual number of replications that AFSAA would use in one of their studies.

To analyze the potential contribution of each VRT, the 95% confidence interval half-widths of the base case, no VRT, are compared with the 95% confidence interval half-widths generated by each variance reduction technique. Access databases were built to manipulate the data into the MOEs for each technique and then exported to Excel for ease of comparison. The variance reduction was calculated by dividing the difference between the base half-width and the VRT half-width by the base half-width. We present results for CRN (Table 3) and CV (Table 4) here. In Table 3, the F-111 LPS MOE is not included since all F-111s were removed from the modified scenario. For AV results (for 300 total replications – 150 antithetic pairs) see Table 7 in the appendix or Bednar [1].

For CRN results, three sets of runs were accomplished. The first set of runs was the original scenario. The second set of runs was independent runs of the "no F-111 scenario." Finally, the third set of runs was the "no F-111 scenario" using the same random number inputs used in the first set of runs. Base half-widths are computed from the first two

Table 4. CV results (30 replications)

MOE	Base Half-width	CW Half-width	Percent Change	Equivalent Replications
A-10 LPS	0.00075	0.00016	**-78.85%**	607
A-6 LPS	0.00349	0.00212	**-39.24%**	77
BLUE FPS	0.00906	0.00724	**-20.09%**	46
BLUE LPS	0.00056	0.00034	**-40.17%**	79
BLUE SMG	268.07053	236.17667	**-11.90%**	38
DHAHRAN FPS	0.03193	0.02463	**-22.85%**	49
F-11 LPS	0.00171	0.00082	**-52.18%**	121
F-15 LPS	0.00097	0.00063	**-35.25%**	68
FA-18 FPS Total	0.01693	0.01541	**-8.97%**	36
FA-18 FPS DCA	0.00343	0.00331	**-3.41%**	32
MIG-23 LPS	0.02011	0.02033	1.08%	30
MIG-29 LPS	0.02635	0.02206	**-16.29%**	42
MIRAGE FPS Total	0.01572	0.01336	**-14.97%**	41
MIRAGE FPS INT	0.02255	0.01396	**-38.09%**	74
MIRAGE LPS	0.01444	0.01305	**-9.63%**	37
MUDAYSIS FPS	0.01311	0.01204	**-8.10%**	35
RED FPS	0.00432	0.00327	**-24.34%**	51
RED LPS	0.00590	0.00515	**-12.74%**	39
RIYADH FPS	0.00605	0.00408	**-32.58%**	63
SHAIBAH FPS	0.02297	0.01691	**-26.40%**	53

sets of runs and the CRN half-widths are computed from the first and third sets of runs. Table 3 shows a reduction in half-width for ten of nineteen MOEs, with three of these posting over a 20% reduction. However, the *MIRAGE FPS total* MOE shows an increase of 47%. We believe that one of or some combination of the following three phenomena is occurring: 1) There are non-monotonic generators in the model. Some of the random variates, for example the normal variate, do not use an inverse transform and may lose synchronicity in variate generation. 2) Synchronicity is defeated through complicated branching or feedback. 3) Some sections of the model are synchronized perfectly while others are not. In the case of the first two possibilities, it seems reasonable that one would observe about one half of the CRN half-widths as larger and the other half smaller (relative to the base half-width); which is what we observe. So CRNs appear to work quite well, but are not successful in providing more accurate results for all MOEs. AV achieved similar results to CRNs. The AV results are presented in Table 7 of the appendix.

Looking at the CV results in Table 4 we see a reduction in half-width for all but one of the twenty MOEs, with eleven of these posting a reduction of over 20%. The only increase in half-width is for the *MIG23 LPS* MOE with just over a one percent increase. The implementation of CV requires the added output of

Table 5. Control variate statistically significant controls

MOE	Weight	Control	MCE	Weight	Control
A-10 LPS	-0.0002	__AIR060_2	MIG-23 LPS	-6.6526	__ADF121_2
A-6 LPS	-0.0071	__AIR050_2		0.1433	__AIR101_1
	-0.0759	__ISR000_4	MIG-29 LPS	0.6471	__AIR528_1
BLUE FPS	-0.0722	__AIR050_1		-1.4314	__BSE050_1
	-0.0659	__AIR070_2		-0.2311	__GRD095_1
	-0.0035	__AIR800_3		-0.7147	__ISR000_4
	0.3282	__BSE400_1	MIRAGE FPS total	-0.3242	__ADF150_1
	46.8237	__PLA443_1		0.5198	__AIR561_1
	0.6659	__UTL104_1		0.0855	__AIR602_1
BLUE LPS	0.0101	__ADF150_2	MIRAGE FPS INT	-3.3544	__ADF121_1
BLUE SMG	-23825.1823	__ADF105_1		7.9362	__ADF121_2
	40.5388	__AIR060_2		0.0101	__AIR800_3
	3649.5981	__AIR101_2		-1.0679	__BSE003_2
	-1408.6753	__AIR561_2		106.4336	__PLA443_1
	-20765.0668	__AIR840_1	MIRAGE LPS	-2.3615	__ADF121_1
	-27766.1967	__AIR840_2	MUDAYSIS FPS	-0.2438	__ADF150_2
	-8446.2005	__BSE400_1		-0.0089	__AIR800_3
DHAHRAN FPS	-0.1779	__AIR050_1		0.1621	__GRD095_2
	-0.2230	__AIR070_2	RED FPS	0.6949	__BSE200_5
	0.9677	__BSE400_1	RED LPS	0.0922	__ADF150_1
F-111 LPS	0.0213	__ADF150_2		-0.0629	__AIR070_2
	-0.0120	__AIR060_1		0.6117	__UTL104_1
	-0.1221	__BSE200_5	RIYADH FPS	-0.2501	__AIR007_1
F-15 LPS	0.0069	__AIR050_1		-0.0395	__AIR101_1
	0.0019	__AIR050_2		-0.0621	__AIR101_2
FA-18 FPS total	0.0252	__AIR050_2		-0.0505	__GRD095_1
	-0.0889	__AIR070_2	SHAIBAH FPS	-0.1675	__AIR101_1
	-1.3057	__AIR810_2		1.3115	__BSE003_1
FA-18 FPS DCA	1.4195	__ADF121_3			
	0.0006	__AIR060_2			

random variates, which does increase running time. For our model, the CV increased the execution time less than one percent. So our CV results show a much more consistent increase in accuracy across all of the MOEs, with a small increase in execution time. In addition, the method of control variates has an added benefit to the analyst. It generates a series of random inputs that are statistically significant in their correlation to the selected MOE. Knowledge about these random inputs can give insight into how the simulation works. First, knowing the weight and sign of the coefficient can inform the analyst how much and in which direction a shift in the control can alter the selected MOE. Table 5 identifies the significant controls for each MOE and gives the weight and sign of the corresponding control.

Table 5 can also be used in conjunction with knowledge of the simulation code. If the analyst looks at how the controls are used, he may be able to identify systems or actions that affect the MOE. For example, take MOE BLUE SMG (square miles gained). The controls ADF105_1, AIR060_2, AIR101_2, AIR561_2, AIR840_1, AIR840_2, and BSE400_1 are all significant. By looking at the code, the use of each of the random variates can be identified by examination of the module purpose, the comments in the code and the nomenclature of the interacting variables. Table 6 outlines information gleaned from the code.

Using Table 6 a number of system characteristics can be predicted. First, by increasing the saturation delay on the shooter there will be a small increase in BLUE SMG. Second, it can be determined that increasing the number of random draws greater than the engagement probability will greatly increase the BLUE SMG. Another way to look at this is that decreasing the air-to-air engagements will increase the BLUE SMG. Third, increasing the success of a mid-course update will increase the BLUE SMG. Fourth, increasing the probability of hitting a live target will greatly increase the BLUE SMG. Fifth, the greatest increase in BLUE SMG is related to the number of sorties. If the degradation in the number of sorties is large, then the number of sorties are reduced and the BLUE SMG is also reduced. An overall look at this implies that the more sorties that are generated and the fewer enemy engagements along with better strikes will lead to more BLUE SMG. However, the second observation above, relative to the engagement probability, becomes worrisome in the light of our general observation and the way in which this probabilistic draw appears

Table 6. Control variate purpose of statistically significant controls

Control	Weight	Module Purpose	Specific Draw Purpose
ADF105_1	-23825.18	Used to calculate the flight group's position, altitude, speed, average location, and delivery profile	Used to check for point on battlefield (distribution)
AIR060_2	40.54	Manages defensive anti-tactical ballistic missile mission detections and engagements	Used in setting saturation delay on shooter (distribution)
AIR101_2	3649.60	Sets up an air engagement by creating an air-to-air engagement for each flight to be used for storing computations	Used in determining the engagement probability (.prob.engage < *U*(0.1)
AIR561_2	-1408.68	Determines if, having arrived at the target's estimated coordinate, the flight group can find the target	Determines the success of a mid-course update (*U*(0,1) <= .best.prob)
AIR840_1	-20765.07	Determines the number of target elements that are destroyed when a given weapons area of effect fully or partially covers a target	Determines whether a live target element is hit (*U*(0,1) < .prob.hit.live)
AIR840_2	-27766.20	Determines the number of target elements that are destroyed when a given weapons area of effect fully or partially covers a target	Determines whether a live target element is hit (*U*(0,1) < .exp.live.hits)
BSE400_1	-8446.20	Determines the availability of aircraft, establishes the munition configurations, allocates fuels, and determines whether to cancel or run the mission as scheduled	Determines the cancellation of a sortie due to degrade (*U*(0,1) < .eawp.sortie.degrade)

to be coded (see Table 6). Since one would expect "(U(0,1) < .prob.engage)" rather than "(.prob.engage < U(0,1))"as the appropriate coding, the analysis here suggests a hard look at the model's code. This said, a similar type of analysis can be applied to the control variates related to other MOEs.

5. Conclusions

Control variates consistently realized a reduction in the half-widths for all but one of the MOEs studied is relatively easy to implement, can be used to reduce the required number of replications, and give valuable insight into the model. The consistency in the half-width reduction allows a postprocessor to automatically apply control variates and give a reduced half-width without worry of possibly increasing the half-width. The ease of implementation also allows a postprocessor to apply control variates without the direct monitoring of the analyst. If there is an established half-width goal, use of control variates will reduce the number of replication needed and in turn reduce the time required to analyze the model. Use of the insight given by the weight and sign of the controls can help identify inconsistencies of the code or methodology. It can also aid in understanding how elements of the model affect desired MOEs.

6. References

[1] Bednar, E.M. "Feasibility Study of Variance Reduction in the THUNDER Campaign-level Model." Master's thesis, AFIT/FOR/ENS/05-01. Air Force Institute of Technology, Wright-Patterson AFB, OH, 2005.

[2] Miller, J.O. Class power point slides, OPER 671, Combat Modeling I. Department of Operations Research, Air Force Institute of Technology, Wright-Patterson AFB, OH, 2004.

[3] Group W Inc. "What is THUNDER?," *THUNDER Overview*, 2005. Available from: http://group-w-inc.com/AFSAAV&V/thunder.htm

[4] L3 Communications. *THUNDER Analyst Manual, Version 6.9.* Vol. 1, *THUNDER Overview.* 2003.

[5] CACI. *SIMSCRIPT II.5 Programming Language.* CACI, Arlington, VA, 1997.

[6] CACI. *SIMSCRIPT II.5 Users Guide.* CACI, Arlington, VA, 2002.

[7] Thompson, D.C., and T.W. Ewart. Interview with author, ASC/ENMM, Wright-Patterson AFB, OH, 2004.

[8] L'Ecuyer, P., et al. "Online Companion for: An Object-Oriented Random Number Package With Many Long Streams and Substreams," *Operations Research* 50 (6, 2002). Available from: http://or.pubs.informs.org/Media/L'Ecuyer.pdf

[9] L'Ecuyer, P. "Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators," *Operations Research* 47 (1, 1999): 159–164.

[10] L'Ecuyer, P. "MRG32k3a," *Random Number Generator Code in C Directory*, 2004. Available from: http://www.iro.umontreal.ca/~lecuyer/myftp/streams00/c/

[11] Lavenberg, S.S., and P.D. Welch. "A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations" *Management Science* 27 (3, 1981): 322–335.

[12] Lavenberg, S.S., T.L. Moeller, and P.D. Welch. "Statistical Results on Control Variables with Application to Queueing Network Simulation" *Operations Research* 30 (1, 1982): 182–202.

[13] Law, A.M., and W.D. Kelton. *Simulation Modeling and Analysis* 3rd ed. New York: McGraw-Hill, 2000.

Appendix: Variance Reduction Techniques

A stochastic simulation, such as THUNDER, relies on input that is random and therefore produces output that is also random. This means that, since there are inputs that vary randomly from replication to replication, there will also be random variation in the output from replication to replication. Often this variation is so large it obscures the estimates of the models MOEs. Therefore, analysts have developed techniques to reduce this random variation in the output. These techniques are called *variance reduction techniques* (VRTs). There are many different types of VRTs, and the effectiveness of each technique is dependent on the particular model and application. Three common VRTs are control variates (CV), common random numbers (CRN), and antithetic variates. CV and AV are single-model techniques—meaning they are applied to a single configuration of a simulation—while CRN is a multiple-model technique—meaning it is applied when two or more configurations are compared.

Control Variates (CV)

The method of control variates attempts to take advantage of any correlation between random variates in a simulation and the selected output. A control variate is some random variable that is observed during the execution of the simulation model that is exploited due to its correlation to some response of interest. Its values can be used to "explain away" some portion of the variability found in the response. The method is effected through a regression of the response onto a set of control variables. Good references for control variate techniques are [11, 12].

Using regression theory an interval estimate for μ_Y can be obtained. By making the assumption of joint normality for X and Y, the conditional distribution of Y given X will be normal by equation (A.1):

$$Y \mid X = x \sim \mathrm{N}\left(\mu_Y + \beta\left(x - \mu_X\right), \sigma_\varepsilon^2\right) \qquad \text{(A.1)}$$

where

$$\sigma_\varepsilon^2 = \sigma_Y^2\left(1 - \rho_{XY}^2\right) \qquad \text{(A.2)}$$

and

$$\sigma_Y^2 = \mathrm{Var}(Y) \ . \qquad \text{(A.3)}$$

Since the values of the control variable X and its mean μ_X are known, then it can be seen that the conditional mean of Y given X has two terms. The terms are broken into the parameter to be estimated, μ_Y, and a correction term. From equation (A.1), equation (A.4) can be formed.

$$Y_i = \mu_Y + \beta(X_i - \mu_X) + \varepsilon_i,\ 1 \le i \le K \quad \text{(A.4)}$$

where ε_i are the residuals and are of the form in equation (A.5).

$$\{\varepsilon_i : i = 1,\ldots,K\} \overset{\text{i.i.d.}}{\sim} \mathrm{N}\left(0, \sigma_\varepsilon^2\right) \ . \quad \text{(A.5)}$$

Since the values of μ_Y and β are unknown, the method of least squares can be applied. The statistic $\hat{\mu}_Y(\hat{\beta})$ will be the estimator of the intercept and is normally distributed as in equation (A.6):

$$\hat{\mu}_Y\left(\hat{\beta}\right) | X_1,\ldots,X_K \sim \mathrm{N}\left(\mu_Y, \sigma_\varepsilon^2 s_{11}\right) \ . \quad \text{(A.6)}$$

The value of s_{11} in equation (A.6) is the upper left-hand entry in the matrix $\left(D^T D\right)^{-1}$ where D is of the form in equation (A.7).

$$D = \begin{bmatrix} 1 & X_1 - \mu_X \\ 1 & X_2 - \mu_X \\ 1 & X_3 - \mu_X \\ \vdots & \vdots \\ 1 & X_K - \mu_X \end{bmatrix} \ . \quad \text{(A.7)}$$

To generate a confidence interval about $\hat{\mu}_Y\left(\hat{\beta}\right)$, σ_ε^2 must be estimated. Since σ_ε^2 represents the variability in Y given X, the formula for the residual mean square error is used.

$$\hat{\sigma}_\varepsilon^2 = \frac{\sum_{i=1}^{K}\left(Y_i - \hat{Y}_i\right)^2}{K-2} \quad \text{(A.8)}$$

where

$$\hat{Y}_i\left(\hat{\beta}\right) = \hat{\mu}_Y\left(\hat{\beta}\right) + \hat{\beta}(X_i - \mu_X),\ 1 \le i \le K \ . \quad \text{(A.9)}$$

From regression theory we have

$$\left.\frac{\hat{\mu}_Y\left(\hat{\beta}\right) - \mu_y}{\sqrt{\hat{\sigma}_\varepsilon^2 s_{11}}}\right| X_1,\ldots,X_K \sim t_{K-2}, \quad \text{(A.10)}$$

where t_ν denotes Student's t-distribution with ν degrees of freedom. Therefore, the confidence interval for μ_Y is given by

$$\hat{\mu}_Y(\hat{\beta}) \pm t_{K-2,\left(1-\frac{\alpha}{2}\right)}\sqrt{\hat{\sigma}_\varepsilon^2 s_{11}} \ . \quad \text{(A.11)}$$

Now in simulations there are possibly more than one control for a response. Therefore, equation (A.4) is modified to be

$$Y_i = \mu_Y + \sum_{j=1}^{Q}\beta_j\left(X_{ij} - \mu_{X_j}\right) + \varepsilon_i,\ 1 \le i \le K \quad \text{(A.12)}$$

where

$$K > Q + 2 \quad \text{(A.13)}$$

and D (see (A.15)) is assumed to be of rank $Q + 1$. Now equation (A.8) is reformed into

$$\hat{\sigma}_\varepsilon^2 = \frac{\sum_{i=1}^{K}\left(Y_i - \left(\hat{\mu}_Y + \sum_{j=1}^{Q}\hat{\beta}_j\left(x_{ij} - \mu_{X_j}\right)\right)\right)^2}{K-Q-1}, \quad \text{(A.14)}$$

and s_{11} is the upper left-hand entry in the matrix $\left(D^T D\right)^{-1}$ where D is of the form

$$D = \begin{bmatrix} 1 & X_{11} - \mu_{X_1} & X_{12} - \mu_{X_2} & \cdots & X_{1Q} - \mu_{X_Q} \\ 1 & X_{21} - \mu_{X_1} & X_{22} - \mu_{X_2} & \cdots & X_{2Q} - \mu_{X_Q} \\ 1 & X_{31} - \mu_{X_1} & X_{32} - \mu_{X_2} & \cdots & X_{3Q} - \mu_{X_Q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{K1} - \mu_{X_1} & X_{K2} - \mu_{X_2} & \cdots & X_{KQ} - \mu_{X_Q} \end{bmatrix} . \quad \text{(A.15)}$$

Then the $100(1 - \alpha)\%$ confidence interval in equation (A.11) becomes

$$\hat{\mu}_Y\left(\hat{\beta}\right) \pm t_{K-Q-1,\left(1-\frac{\alpha}{2}\right)}\sqrt{\hat{\sigma}_\varepsilon^2 s_{11}} \ . \quad \text{(A.16)}$$

An important factor in the proper use of the CV technique is the selection of controls. Since the correlation between the control candidates and the MOE is unknown, a stepwise regression can be performed to identify which control candidates are significantly correlated to the selected MOE.

Common Random Numbers (CRN)

The strategy behind CRN is to induce a positive correlation in the random inputs between the outputs of interest from two or more configurations of a study.

The positive correlation will reduce the variance in the difference between the outputs. To create the positive correlation, the two configurations should use the same random variable inputs. The following theory behind CRN was adapted from Law and Kelton [13: 582–598].

To understand CRNs, consider two alternative configurations of a single simulation, where X_{1j} and X_{2j} are values from the first and second configuration on the j^{th} independent replication. Also consider that the estimate in question is equation (A.17).

$$\zeta = \mu_1 - \mu_2 = E\left[X_{1j}\right] - E\left[X_{2j}\right] \quad . \tag{A.17}$$

Let there be n replications of each system and let

$$Z_j = X_{1j} - X_{2j},\ j = 1,2,\ldots,n \quad ; \tag{A.18}$$

then

$$E\left[Z_j\right] = \zeta \quad . \tag{A.19}$$

Therefore, equation (A.20) is an unbiased estimator of ζ.

$$\overline{Z}(n) = \frac{1}{n}\sum_{j=1}^{n} Z_j \quad . \tag{A.20}$$

Since the Z_j are independent, identically distributed (IID) random variables,

$$\begin{aligned}\mathrm{Var}\left(\overline{Z}(n)\right) &= \frac{\mathrm{Var}\left(Z_j\right)}{n} \\ &= \frac{\mathrm{Var}\left(X_{1j}\right) + \mathrm{Var}\left(X_{2j}\right) - 2\mathrm{Cov}\left(X_{1j}, X_{2j}\right)}{n} .\end{aligned} \tag{A.21}$$

If X_{1j} and X_{2j} are independent, then

$$\mathrm{Cov}\left(X_{1j}, X_{2j}\right) = 0 \quad . \tag{A.22}$$

If then X_{1j} and X_{2j} are not independent and are positively correlated,

$$\mathrm{Cov}\left(X_{1j}, X_{2j}\right) > 0 \quad , \tag{A.23}$$

then $\mathrm{Var}\left(\overline{Z}(n)\right)$ will be reduced.

One key to the implementation of CRN is synchronization. The random number streams between the two configurations should match up as much as possible. This means that a random number drawn should be used for the same purpose in both configurations. Ideally, all the random variate draws should be applied in exactly the same way (implying the use of inverse CDF methods to generate random deviates) and in the same order for both configurations. To synchronize the simulations, some initial coding is involved. First, all of the points in the simulation where a random number or variate is drawn must be identified. Then, each point is assigned its own random number stream.

A primary concern should be how to keep track of all the different random number streams. Some large simulations, like THUNDER, could require hundreds of streams. Ideally, each of the streams should be independent and not overlap. This could be difficult, because these simulations could have millions or even billions of random number draws. Therefore, a large RNG with built in functions that keep track of random number streams is beneficial.

Antithetic Variates

The method of antithetic variates (AV) is similar to CRN. As in CRN, correlation is induced between separate runs, but this time the object is to induce a negative correlation. This method is applied to a single configuration of the system rather than, in the case of CRN, differing configurations of the system. The following is adapted from Law and Kelton [13: 598–604].

The basic idea is to make pairs of runs of a model where if the first run observes a relatively "small" observation then the second run observes a "large" observation. Therefore, the two observations are negatively correlated. To obtain the negative correlation, complementary random numbers are used. For example, if U_k is a particular random number used for a particular purpose in the first run, then $1 - U_k$ is used for the same particular purpose on the second run. U_k and $1 - U_k$ form an antithetic pair. Note that like CRN, synchronization is extremely important to the successful use of AV. Further, as in CRN, the use of inverse CDF methods for generate random deviates is paramount.

To understand the basis for AV, suppose there are n pairs (formed by employing the antithetic pairs) of runs in a simulation resulting in observations

$$\left(X_1^{(1)}, X_1^{(2)}\right),\ \ldots,\ \left(X_n^{(1)}, X_n^{(2)}\right) \quad , \tag{A.24}$$

where $X_j^{(1)}$ is from the normal run of the j^{th} pair, and $X_j^{(2)}$ is from the antithetic run of the j^{th} pair. Both $X_j^{(1)}$ and $X_j^{(2)}$ are legitimate observations of the simulation; therefore,

$$E\left[X_j^{(1)}\right] = E\left[X_j^{(2)}\right] = \mu \quad . \tag{A.25}$$

Since each run is independent, each pair of runs is independent. Now let

$$X_j = \frac{\left(X_n^{(1)} + X_n^{(2)}\right)}{2} \qquad \text{(A.26)}$$

and let

$$\bar{X}(n) = \frac{1}{n}\sum_{j=1}^{n} X_j \qquad \text{(A.27)}$$

be the unbiased point estimator of

$$\mu = E\left[X_j^{(l)}\right] = E\left[X_j\right] = E\left[\bar{X}(n)\right]. \qquad \text{(A.28)}$$

Since the X_j's are IID,

$$\mathrm{Var}\left[\bar{X}(n)\right] = \frac{\mathrm{Var}\left[X_j\right]}{n} \qquad \text{(A.29)}$$

$$= \frac{\mathrm{Var}\left[X_j^{(1)}\right] + \mathrm{Var}\left[X_j^{(2)}\right] + 2\mathrm{Cov}\left[X_j^{(1)}, X_j^{(2)}\right]}{4n}.$$

If a negative correlation could be induced between $X_j^{(1)}$ and $X_j^{(2)}$, then

$$\mathrm{Cov}\left[X_j^{(1)}, X_j^{(2)}\right] < 0, \qquad \text{(A.29)}$$

which reduces $\mathrm{Var}\left[\bar{X}(n)\right]$.

Antithetic Variates: Empirical Results

Table 7. AV results (300 replications – 150 antithetic pairs)

MOE	Base Half-width	AV Half-width	Percent Change	Equivalent Replications
A-10 LPS	0.00028569	0.000304497	6.58%	-6.58%
A-6 LPS	0.000928243	0.001007279	8.51%	-8.51%
BLUE FPS	0.002036313	0.002105844	3.41%	-3.41%
BLUE LPS	0.000174156	0.000159844	-8.22%	8.22%
BLUE SMG	84.52936072	81.9763507	-3.02%	3.02%
DHAHRAN FPS	0.006733921	0.006852237	1.76%	-1.76%
F-111 LPS	0.000416308	0.000360925	-13.30%	13.30%
F-15 LPS	0.00031009	0.000295924	-4.57%	4.57%
FA-18 FPS TOTAL	0.003723047	0.003880073	4.22%	-4.22%
FA-18 FPS DCA	0.001389283	0.001180712	-15.01%	15.01%
MIG-23 LPS	0.006037934	0.005923777	-1.89%	1.89%
MIG-29 LPS	0.007942862	0.008588544	8.13%	-8.13%
MIRAGE FPS TOTAL	0.00449042	0.004841742	7.82%	-7.82%
MIRAGE FPS INT	0.005731393	0.005420474	-5.42%	5.42%
MIRAGE LPS	0.004501578	0.004337935	-3.64%	3.64%
MUDAYSIS FPS	0.00488572	0.004809609	-1.56%	1.56%
RED FPS	0.001772589	0.001555645	-12.24%	12.24%
RED LPS	0.001692612	0.001534979	-9.31%	9.31%
RIYADH FPS	0.001535138	0.001589555	3.54%	-3.54%
SHAIBAH FPS	0.006312332	0.006209099	-1.64%	1.64%

Acknowledgements

The views expressed in this article are those of the authors and do not reflect the official policy of the United States Air Force, the Department of Defense, or the U.S. Government.

Author Biographies

Captain Earl Bednar *received a Bachelor's of Science in Applied Mathematics from Western Washington University in 1998 and a Master's of Science in Operations Research from Air Force Institute of Technology (AFIT) in 2003. Currently he is assigned as a Scientific Computing Analyst to the Air Force Research Labs Information Directorate, Advanced Computing Technology branch in Rome, NY.*

Dr. Kenneth W. Bauer, Jr. *is a retired USAF Lieutenant Colonel and a Professor of Operations Research in the Department of Operational Sciences at AFIT. Dr. Bauer received a B.S. in Math from Miami University (Ohio) in 1976, an MEA from the University of Utah in 1980, an M.S. in Operations Research from AFIT in 1981, and a Ph.D. from Purdue University in 1987. Dr. Bauer's research interests include the statistical aspects of simulation, design of experiments, neural networks, and multivariate statistics.*

Dr. J.O. Miller *is a 1980 graduate of the U.S. Air Force Academy (USAFA). He retired from the Air Force as a Lt. Colonel in January 2003. He received an MBA from the University of Missouri at Columbia in 1983, an M.S. in Operations Research from the Air Force Institute of Technology (AFIT) in 1987, and a Ph.D. in Industrial Engineering from The Ohio State University in 1997. Dr. Miller is the Director of the Center for Operational Analysis and an Associate Professor in the Department of Operational Sciences at AFIT. His research interests include combat modeling, computer simulation, and ranking and selection.*