To appear in: S. Narayanan and L. Rothrock (eds.) *Human-in-the-loop Simulations: Methods and Practice*

Determining the number of simulation runs: Treating simulations as theories by not sampling their behavior

Frank E. Ritter[1], Michael J. Schoelles[4], Karen S. Quigley[2], Laura Cousino Klein[3]

[1] College of Information Sciences and Technology, and the Department of Psychology, The Pennsylvania State University

[2] Department of Veterans Affairs, NJHCS, East Orange, NJ and Department of Psychiatry, UMDNJ-New Jersey Medical School

[3] Department of Biobehavioral Health, The Pennsylvania State University

[4] Department of Cognitive Science, Rensselaer Polytechnic Institute

Corresponding Author: Frank E. Ritter
College of Information Sciences and Technology
The Pennsylvania State University
316G IST B
University Park, PA 16802
+1 (814) 865-4453 (ph) +1 (814) 865-5604 (FAX)
frank.ritter@psu.edu

Text word count: 6,681 words (excluding figures, tables, and references)
172 words in abstract

Draft of 24 December 2010

Abstract

How many times should a simulation be run to generate valid predictions? With a deterministic simulation, the answer simply is just once. With a stochastic simulation, the answer is more complex. Different researchers have proposed and used different heuristics. A review of models presented at a conference on cognitive modeling illustrates the range of solutions and the problems in this area. We present the argument that because the simulation is a theory, not data, it should not so much be sampled but run enough times to provide stable predictions of performance and of the variance of performance. This applies to both pure simulations as well as human-in-loop simulations. We demonstrate the importance of running the simulation until it has stable performance as defined by the effect size of interest. When runs are expensive we suggest a minimum numbers of runs based on power calculations;

when runs are inexpensive we suggest a maximum necessary number of runs. We also suggest how to adjust the number of runs for different effect sizes of interest.

Keywords: power calculations, effect sizes, simulation methodology, cognitive modeling, human-in-the-loop simulations

INTRODUCTION

We provide guidance here for how many times to run a simulation, including a human-in-the-loop simulation or a cognitive model, to ensure that the simulation has converged on stable predictions. This advice is derived from power calculations.

One paradigm in which this heuristic guidance can be applied is when a model is constructed to perform and make predictions about some human task. Although this seems like a narrow topic to explain to a general simulation audience, the number of times to run a simulation is an important topic because in many cases simulations are being used incorrectly, and as a result, analysts and their audience do not truly understand the simulation's predictions. We begin with our view of simulation. This will require talking about several layers of a simulation taxonomy until we reach the level at which this chapter is aimed, and then illustrating the problem and quantifying the solution for an example simulation.

The methodology that we are prescribing provides suggestions for any simulation with random processes as components, including the development of human-in-the-loop simulations and stochastic cognitive models developed and run on computational cognitive architectures. In the next sections, we will elaborate on these terms.

The first term to notice in our taxonomy is "computational". What is the difference between computational modeling and statistical or mathematical modeling? We believe that the main difference is that computational models are computer programs rather than equations or distributions. The advantage of models as computer programs is that they can simulate complex behavior. For instance, a computational model of a human playing a computer-generated game of Tetris through a computer interface is feasible, but it seems it would be very difficult to develop a mathematical model of the integrated cognitive, perceptual and motor processes involved in this task. One important advantage of models as computer programs is that they can be process models, providing a theory of the information processes in cognition by processing information themselves.

Computer programs that are intended to model some cognitive process belong to a part of Artificial Intelligence (AI) called human-level AI or cognitive science, depending on the emphasis of intelligence level or being human-like in the processing. One approach being taken to achieve human-level intelligence is simulation of human behavior. The field of cognitive architectures has developed in the last 25 years to create high-fidelity simulations of human behavior. There are many definitions of the term cognitive architecture. Most definitions include some notion that the cognitive architecture contains the immutable functional machinery of cognition. For example, a definition by Ritter and Young (2001, consistent with Newell, 1990) is:

> "A cognitive architecture embodies a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task. "

That is, those information processing computations are not modified by changes in beliefs, goals, and so on.

Another important notion of cognitive architectures is that the architecture by itself cannot produce any behavior. Knowledge must be added to the architecture to achieve behavior, creating a model. The current state of the art in cognitive architectures is that the modeler must supply the knowledge. Therefore, many architectures equip the modeler with a modeling language to develop models. While being able to program models via a modeling language has benefits in terms of efficiency and complexity, Byrne (2003) points out that "individual modelers need to have solid programming skills". The advantage of architectures implemented as computer programs is that the programming language is not ambiguous, and therefore supports a more uniform interpretation of the theory.

Cognitive architectures also come in many flavors. Cognitive architectures as computer programs represent scientific theories such as those in ACT-R, Soar, and EPIC. Some like Soar (Laird, Newell, & Rosenbloom, 1987) and EPIC (Kieras, 2003; Kieras, Wood, & Meyer, 1997) are basically deterministic in the sense that in most models noise is not directly added to computations and the same predictions are made each time the model is

run. ACT-R is an example of a hybrid architecture that has stochastic components. The example model described later in this paper is an ACT-R model (Anderson, 2007).

To understand the dilemma presented in the next section a brief description of the ACT-R architecture is required. ACT-R is symbolic and rule-based, but also has a layer below the symbolic layer, called the sub-symbolic layer. One quantity computed by the subsymbolic layer is the activation of declarative memory elements. This activation determines the retrieval probability and latency for a memory element. The other important computed quantity is the rule utility, and it is based on Temporal Difference Reinforcement Learning. The computation of both these quantities involves the addition of noise to the calculation. These noise quantities are controlled by the modeler through parameters (i.e., one for activation and one for utility). Therefore, ACT-R models can be stochastic and most are, unless major components are removed by setting the noise to 0.

To develop a model in ACT-R, the model adds procedural knowledge in the form of production rules and background knowledge in the form of declarative memory chunks. This allows very complex models to be built, but these types of models can be difficult to analyze and evaluate because of the inherent complexity of the knowledge contained within them and the variability in processing the knowledge by the noisy architecture.

As discussed above, cognitive architecture-based models are often built to simulate human users of computer systems. So, to evaluate such a model it seems natural to want to compare model data to human data. The traditional way to do this is by hypothesis testing where the null hypothesis is that there is no difference between human data and model data. But hypothesis testing can only be used to reject the null hypothesis. Thus, we can show that model data does not match the human data, but cannot prove that it matches. (Grant, 1962, provides an argument showing how correlation helps provide an answer in this area.)

The problem of how many times to run a model is one part of the bigger problem of model comparisons. One hoped for outcome of this handbook is to provide guidance for modelers developing complex models on how to show that the model data corresponds to

human data. In a symposium on "Model Fitting and Parameter Estimation" Ritter (2003) posited the following points on the problem of model validation and comparison for the type of models developed under cognitive architectures. (a) Task performance is more important than fit—more credit should be given to models that actually perform the task. (b) Enough detail should be given about the model fit to see if the model is worth taking seriously or not. If the model can fit any data, then it should be dismissed as a psychology theory (but may be useful as an AI model). (c) It should be reported where the model can be improved. In other words, let the reader know where the holes are and where the model can and will be improved. This view is that of Grant (1962) as applied to cognitive architecture-based models. But again, before a one can think about model comparisons, the model's predictions must be understood.

One of the strengths of ACT-R and architectures like it is the ability to interact with the same software as humans-in-the loop. It can do this because it has "eyes", "hands", "ears", and can speak. These perceptual and motor components of ACT-R are not only psychologically plausible but can interact with the stimulation and operating system software to manipulate input devices and read the computer screen. With these components ACT-R models perform human actions such as searching computer screens, listening to instructions, and manipulating a mouse or joystick.

The relevance to human-in-the-loop systems is that ACT-R can be a human-in-the-loop when human subjects are expensive. Imagine a team oriented simulated task environment where the team members are at workstations and communicate over a network. ACT-R models can be developed to work in such environments, replacing one or more of the team members, or, for some studies, all of the team (e.g., Ball et al., 2010). The issue for this paper is how many times do you need to run such a simulation to understand the implications of the simulation, with or without humans in the loop?

MODELER'S DILEMMA

Because cognitive models are really simulations, a common question facing creators of cognitive models, at least implicitly, is "How many times should we run the model to generate predictions for comparison across experimental conditions and for comparison with the data?" As we note below, authors have used a wide variety of answers: Some comparisons use a single run of the model, although this is somewhat uncommon with models that include stochastic effects. Some comparisons run the model once per subject. This is often just a handy heuristic as they look for a number to choose. Other researchers run it 10, or 20, or 50, or 1,000's of times. The dilemma is that you want to run a stochastic model enough times to understand its predictions without running it so many times as to waste resources. In completely human studies, this problem is addressed through power calculations. For other simulations including human-in-the-loop simulations, power calculations would be useful as well.

Figure 1 illustrates this problem. On the left hand side, if a model with random elements is run only a few times, the distribution of performance is not well known (shown with a shaded line indicating a less understood distribution). The mean and standard deviation are also less well known, and the standard error of the mean[1], is larger. On the right hand side of Figure 1, where the model is run more times, the distribution is better known (shown with a more complete histogram and a solid estimated distribution line). The mean and standard deviation become more stable and the standard error of the mean becomes smaller with additional runs. And yet, with further runs the improvement that each run provides decreases.

< Insert Figure 1 about here>

---

[1] The standard error of the mean is a standard statistical measure of how well known the mean is, and it is explained in more detail below.

The extent of the problem of knowing how many times to run a model can be illustrated by looking at a sample of models. There are many venues where this can be done. Table 1 provides just one example set, a summary of models presented at the 2004 International Conference on Cognitive Modeling (Lovett, Schunn, Lebiere, & Munro, 2004) where the papers are available online. Similar results are available for other sets of models[2]. The table includes each paper reporting a model to data comparison where the model appeared to have a stochastic component or where the task provided variance. The second and third columns note how many subjects were included and how many times the models were run[3].

Table 1, which is representative of other conferences and even journal papers, shows that more than a third (12.5/33) of the papers did not report how many times the model was run; and an additional 7.5 probably did not run their model enough to report stable predictions (20 or fewer runs). So, well over half did not run their model to get stable predictions or did not report that they did. In addition, none of the papers in Table 1 provided a rationale for the number of model runs beyond "to match the number of subjects" or "to provide stable performance." No paper mentioned effect sizes, although many included standard error bars on their graphs. The number of times the models should have been run is not known to us—it would depend on the effect size of interest, but we will see that it is most likely that the number of runs was too low. (The number of runs would also vary based on the number of parameters manipulated, but these models did not vary parameters or perform parameter sweeps.) This lack of reporting of the theories is alarming.

< Insert Table 1 about here>

---

[2] For example, http://acs.ist.psu.edu/nottingham/eccm98/home.html.

[3] Papers with two studies had each study counted 0.5. Papers that were not simple, that examined complex data, e.g., language corpora, or that presented only tools or theoretical points, are not included.

Of course, where models are deterministic, they only need to be run once. Where there are closed form solutions to obtain the predictions of models, these closed form solutions should be used. For example, we have run a Soar model for 100 hours to compute predictions, only to discover with a bit of mental effort that a closed form iterative function would provide the same data in 6 s on a slower machine (Ritter, 1988).

When runs are inexpensive, using a very large number of runs (e.g., 10,000 to 100,000) is a very satisfactory answer because it provides stable estimates of performance, and the power analyses below indicate why. However, there are an increasing number of cases when simply performing a large number of runs will not work. Performing a large number of runs is not possible when runs are expensive, numerous models must be run as in a network, or search in a combinatorial parameter space is required (where there may be 100,000 parameter sets to test, making 1,000 runs per setting turn into 100,000,000 runs). These situations include models that interact with or are based on robots that are both complicated to setup and cannot be run faster than real time, models that work with commercial software that can only run in real time, models that interact over a long time period, models that have multiple settings or parameters to be adjusted, models that interact with software too complicated to rewrite to run faster than real time (e.g., some process control models), and models that have to interact with people (i.e., human-in-loop simulations) or simulate group behavior with real time constraints (e.g., they cannot be run faster than real time).

Even when models can be run faster than real time there are cases when the modeler might wish to run as few as necessary. These include when there are multiple models to be considered or a combinatorial set of possible parameter sets (e.g., changes to working memory, changes to processing speed, and changes to representation). Even for models running faster than real time one should ask how many runs are needed to understand the model's predictions?

We will present the case here, using an example representative model, that suggests that researchers should run their model until it makes stable predictions (that is, the predictions obtained are representative of the model's predictions). We will also describe a way to compute stability. Our results suggest that some of the models in Table 1 may have been appropriately presented, but most could have been understood better and presented more clearly by following the suggestions we make below. We provide a rationale and a way to compute how many runs represent stable predictions based on effect sizes and the power with which the modeler wants to determine these effects. Our approach also shows that an answer of providing "an infinite number of runs" or "as many as possible" (which could also be put forward) are wasteful and unnecessary prescription for human-in-the-loop simulations. For illustration we use a medium-sized model we created to understand behavioral moderators. We analyze its behavior as an example—the calculations and implications apply to all user models with stochastic elements. We introduce that model next.

EXAMPLE MODEL: COGNITIVE APPRAISAL AND SUBTRACTION

Serial subtraction commonly has been used to assess the relationship between task appraisals and resulting physiological changes. This task is regarded as an active coping task and has been used across many laboratories as a stressor task (e.g., Kirschbaum, Pirke, & Hellhammer, 1993; Quigley, Feldman Barrett, & Weinstein, 2002; Tomaka, Blascovich, Kelsey, & Leitten, 1993). In this task, subjects are given an arbitrary seed number and are asked to subtract repeatedly a single- or double-digit number. For example, a subject is given 1,457 as the seed number and is asked to repeatedly subtract 7 from the running total while speaking aloud each result. Mistakes are noted and the subject is asked to correct them before they can continue.

The type of appraisal made prior to the task affects performance on the serial subtraction task—subjects making challenge appraisals attempt more subtractions and have more correct responses than do subjects who make threat appraisals prior to the task (Tomaka,

Blascovich, Kelsey, & Leitten, 1993). A "challenge" appraisal occurs when, although stressfulness of the task is deemed high, coping ability is also deemed high. A "threat" appraisal occurs when stressfulness is high and coping ability is seen as low. Although serial subtraction may not appear stressful to everyone, it is typically challenging and often threatening to participants, probably due in large part to the highly evaluative and social nature of the task (e.g., the experimenter often is seated close to the subject and "knows the answers", and the subject is told that they are being recorded for "later analyses"). We know that these appraisals influence performance and are not entirely evaluations of knowledge because performance varies when the participant's appraisals are manipulated and knowledge held constant (Tomaka, Blascovich, Kibler, & Ernst, 1997).

The model

To illustrate the effect of increasing the number of model runs we chose a cognitive model of serial subtraction that was built using the ACT-R cognitive architecture. It is similar in size and complexity to many ACT-R models and models being developed in other architectures. ACT-R is a production rule-based cognitive architecture; that is, cognitive activity takes place through the successive firing of production rules that take an "if…then" format. The model includes several stochastic elements. The details are not important for this analysis, but are available in the descriptions of the model (Ritter, Reifers, Klein, Quigley, & Schoelles, 2004; Ritter, Reifers, Klein, & Schoelles, 2007) and of the architecture (Anderson & Lebiere, 1998).

The choice of which rule to fire from among those that match a particular situation (so-called conflict resolution) is thus a knowledge-based process, where higher valued rules represent more strongly held beliefs. It is also a stochastic process due to the presence of ACT-R's Expected Gain Noise (EGN) parameter. This noise allows the occasional firing of rules that are less than optimal. Adding noise to the decision process is consistent with several theories of stress indicating that high levels of stress negatively influence cognition, particularly decision making (e.g., Mathews, 2001), and as we shall show, consistent with

existing data. There are, of course, other possible approaches to modeling stress in ACT-R (Ritter, Reifers, Klein, & Schoelles, 2007). Testing all of them and all of their combinations would be a useful but non-trivial exercise, so this model is presented here for illustration. Indeed, the need to test these combinations (up to 200 possible variants) suggests the need to understand how many times we need to run each variant to understand its predictions.

Our current serial subtraction model contains the necessary procedural knowledge (i.e., 28 rules implementing subtraction) to perform the serial subtraction task as well as declarative knowledge about numbers and arithmetic facts (257 declarative memory elements made from 16 types, such as digits, columns, subtraction-facts, and comparison of number pairs). The model, the graphical interface, and movie demos of the model running are available (acs.ist.psu.edu/ACT-R_AC/).

Changes to the model examined

The capacity for the model to perform the task under threat or challenge appraisals is implemented by adjusting the value of the rule utility noise parameter[4] to simulate the effects of cognitive appraisal influencing the decision process about what knowledge to apply. When the model is set to challenge appraisal, the rule utility noise parameter is set to a small value (0.1) to model a "clear head", but one where errors can occur as they do in real subjects. When the model is set to threat appraisal, the default value of the rule utility noise is changed to a greater number (1.0) to simulate a state where the procedural knowledge is applied less accurately in the thought process of threatened individuals. Although appraisals are often dichotomized as challenge or threat, they fall along a continuum of appraisals and thus this parameter could vary across a continuum as well. An attractive feature of this type of modification based on modifying architectural parameters is that it is based on a cognitive architecture (Ritter, Reifers, Klein, & Schoelles, 2007). This allows the modification to be

[4] The parameter is EGN in ACT-R 5, and EGS in ACT-R 6.

easily borrowed and used by any other model built in ACT-R. We have applied a related change to a model of driving (Ritter, Van Rooy, St. Amant, & Simpson, 2006).

These changes to produce two conditions of the model, however, are used for illustrative purposes here. Our analysis applies to any model that makes predictions that include a stochastic component, and where closed form or infinite runs are not available.

COMPARISON OF THE MODEL WITH DATA

Results from the model performing the serial subtraction task under challenge and threat appraisals can be compared to human data obtained from an empirical study using the same task. The first three rows of Table 2 present the human data taken from Tomaka et al. (1993) of subjects performing the serial subtraction task who made pre-task appraisals. With more challenging appraisals, more subtractions were attempted and more attempts were correct. (These differences were reported by Tomaka et al. as being significantly different, but standard deviations were not reported in their paper. The ACT-R model predicts that the standard deviations were small with respect to their sample size and mean and that these differences are reliable.) The model's standard deviations are, however, much smaller than data from later studies where the SD (across subjects) is approximately 15 subtraction attempts (Ritter, Bennett, & Klein, 2006).

The model's predictions with the pre-task appraisal overlay are shown in the second set of rows of Table 2 (rows 4-6). In each case, the model makes predictions that are different from each other ($p < 0.01$) for each type of appraisal. The model for threat appraisals reproduces fairly accurately the average number of attempts and correct responses when performing under threat appraisal. However, in the case of a challenge pre-task appraisal, the model does not perform as many subtractions as in the human data, but it successfully matches the ratio of correct responses to subtraction attempts.

The model's performance was measured over multiple runs because its performance varied. The noise applied to the model is supplied by a pseudorandom number generator

based on the Mersenne Twister, which is designed to provide numbers with low autocorrelation, that is, runs of ACT-R are independent when taken in a series, and are samples from a single distribution (independent and identically distributed). When the rule utility noise is zero (EGS = 0), the model exhibits perfect performance because it applies rules completely accurately. When the rule utility noise is greater than zero, the model can make several kinds of mistakes based on applying a nearly appropriate but wrong rule, or applying the right rule at the wrong time. The rules chosen can vary slightly (at EGS = 0.1) to somewhat (EGS = 1.0) from optimal.

< Insert Table 2 about here >

The good fit of the model with the pre-task appraisal overlay to the human data suggests that our choice of how to implement cognitive appraisal was sensible. The model offers one plausible and very simple hypothesis to explain the impact of cognitive appraisal on task performance. It encourages more work to determine if the way appraisal affects performance is indeed by influencing the level of noise present in the thought process of humans.

But is this a fair and sufficient comparison of the model's performance with the data? How many times should we have run our model to confidently report its predictions? When we have multiple possible changes to our theories, how many runs do we need to test each of these modifications?

COMPUTING HOW MANY RUNS TO PERFORM

Figure 2 starts to answer the question of how many times a model should be run. It shows the individual number of subtraction attempts across 100 runs (light points) as well as the running, cumulative average values (dark points). The error bars are the cumulative standard deviation at each point, that is, the standard deviation for the points up to that run. Figure 2 illustrates the range of possible values, the problem of using just a few runs, and how with an increasing number of runs the true average is approached.

We propose two possibilities for a criterion for stable predictions. The first is based on the standard error of the mean. Figure 3 shows how the standard error of the mean (SEM) decreases over a series of 100 runs for our model.

Equation 1 shows how the SEM is based on the variance and the size of the sample.

$$SEM = Variance / N = Standard\ deviation / sqrt(N) \quad \text{(eq. 1)}$$

The SEM represents the error in predicting the mean of a distribution. Assuming that the values are independent and identically distributed, the SEM indicates that the true mean has a 95% chance of being within a range of the estimated mean +/- 1.96 * SEM. Thus, one way to determine how many times to run a simulation is to run it until the estimated range of the mean is small enough for your purposes.

Figure 3 also shows how the standard deviation stabilizes with additional runs. This figure is interesting because it also shows how the predictions of the mean and variance become more accurate with additional runs. The mean (Figure 2) and the variance (Figure 3) are initially unstable with a small sample of runs. With additional runs, the SEM basically decreases from run 4 on. With 100 runs the SEM is at 0.35 and decreasing rather slowly (related to the square root of the number of model runs, per Eq. 1). Figure 4 shows how the change in SD between runs decreases across the 100 runs.

In this case, if we wanted to know how many subtraction attempts the model predicted for a 4-minute block, +/- 0.5 subtractions with 95% confidence, based on Eq. 1 we would have to have a SEM of 0.5/1.96 or a SEM of 0.255 (0.5 = 1.96 * SEM, or 0.5/1.96 = SEM = 0.255). If we use an estimate from Figure 3 of the standard deviation as being 3.6 (it is probably slightly less), then 3.60/sqrt(N) = 0.255. Solving for N gives us 199 runs.

< Insert Figures 2, 3, & 4 about here>

Together, Figures 2 and 3 demonstrate that reporting a single run of our model, in particular the first run in our series, 61 attempts in a 4-minute block, would have over predicted the number of attempts by about 10%. Other single runs would be more or less

accurate. Some papers have reported one run of a model as an example. With deterministic models, this is appropriate. For models with stochastic components, these figures show that one run is clearly not enough.

Other reports have run the model once per subject. For this data set, the model would be run 22 times. Figure 3 suggests that the first 22 runs in our series would provide a fairly reasonable prediction of the mean total attempts from the model, 54.86. This prediction is still slightly high, however. Figure 3 goes on to show that with more runs, the model's average number of attempts drops slightly to 54.4 attempts. Figure 2 also shows that the SEM at 22 runs is 3.54/sqrt (22) = 0.75, and thus that other sets of 22 runs could more or less accurately represent the model's performance.

The heuristic of one run per subject ignores that model runs are typically much less expensive than subject time. Moreover, different sets of 22 runs could lead the modeler to a wide range of different conclusions, which is clearly not desirable. Most importantly, if one takes the model to be a theory, then the choice of "number of runs = number of subjects" reports a sample of the theory rather than reports the theory's predictions and thus is not at all appropriate.

Figures 2 and 3 show that increasing the number of runs improves the quality of the model's predictions in that they are more accurate. Namely, the cumulative averages are more stable, the mean standard error decreases, the standard deviation stabilizes, and the corresponding power to find differences between model conditions and between the model's predictions and the data increases. The two figures suggest that the best number of runs for the model is simply the largest number possible, as more runs provide more stable and more accurate predictions, although there are decreasing returns with more runs.

If one is using a simulation where running the simulation is not free or even inexpensive, one will have to choose a cutoff, however. For instance, using the model here, the SEM drops to 0.5 by about 40 runs and then drops slowly with additional runs. So, Figures 2 and 3 might lead to a different conclusion, which is to do runs until the changes to

the mean and SD from run to run become negligible—where negligible is defined by the modeler and the size of differences of interest. For human-in-the-loop simulations, this might be 10, or 40 or more but you can see the trade-offs in the figures.

But, when simulation runs are not easy to obtain, how can we choose an appropriate number of runs? Power calculations are a way to compute how likely a study is to find effects based on their size and the size of the sample (Cohen, 1988). This is the same computation that experiment designers use to determine how many subjects to run. It is a simple equation that can be used to compute the probability of finding a given effect size given the number of times a variable is measured. An effect size is the ability to see a difference between two means using the standard deviation as a unit. Thus, an effect size of 1 is observed when the difference between two means is separated by 1 standard deviation; 0.5 is a difference of half a SD, and so on. Because effect sizes are represented in terms of standard deviations, it does not matter what the source of noise is in the model, or if the standard deviation is large in relation to the mean. A disadvantage to using effect sizes is that they are in terms of standard deviations, not in the raw measure. For example, an effect size on reaction time is in standard deviations rather than milliseconds, which is slightly harder to reason about.

The equation for computing power is used in Table 3 and shown here as equation 2.

$$\delta = \text{effect size} * \text{sqrt}(N/2) \qquad \text{(eq. 2)}$$

In this equation the noncentrality parameter (δ) is based on two components, effect size and sample size. For a given power value, small effect sizes require correspondingly larger sample sizes.

The SD of model performance (e.g., shown in Figures 2 and 3) and sample size can be used to compute a measure of (statistical) power (using the formula in Table 3, taken here from Howell, 1987, Ch. 9), to find medium differences (effect size = 0.5 SDs) with a probability of 0.94, and small differences (effect size = 0.2 SDs) with a probability of 0.29. The use of standard deviations as a measure allows this calculation to be unitless and to apply to all differences between models and subjects and also between model conditions. The

differences between model conditions here have an effect size of more than 2—in this case, the difference in means of the challenge and threatened model's subtraction attempts divided by the (pooled) standard deviation (54.5-46.8)/3.5, is an effect size of 2.14—so there is more than adequate power to find reliable differences between the model conditions of threat and challenge. Practically, for our model, 100 runs provides more than enough power (0.94) for the example model's effect size of interest (e.g., medium = 0.5).

< Insert Table 3 about here>

We suggest that a power of 0.90 for the expected effect size can provide a suggestion of how many runs are required when runs are expensive, and a power of 0.99 when runs are inexpensive. Table 4 thus provides a bracketing of number of runs based on a range of power. We choose 0.99, a relatively high number, because model runs are usually inexpensive, and because we wish to understand our model clearly and completely. Table 4 provides example values for runs assuming t-tests between means are used. Other values of alpha, other types of measures, and other tests are possible, but other choices for these values do not change the conclusions that increasing the number of runs is desirable to increase power and stabilize the mean and SD, and that power calculations can be used to suggest the number of runs to perform.

Table 3 shows the power for a range of effect sizes with 100 runs, which we used here. Table 4 provides the number of runs to achieve a power of 0.9 for the same expected effect sizes. This provides a set of reasonable minimum times to run a model where the models runs are expensive.

Table 5 provides the number of runs required to achieve a power of 0.99 for the same expected effect sizes. This provides a set of reasonable maximum runs for various effect sizes. If we expected an effect size of 0.8, then 56 runs would provide a power of 0.99 to differentiate predictions from different settings of the model. If we would like to differentiate an effect size between model conditions of 0.2 (Cohen's small effect) then 882 runs would be

required for a power of .99, and if an effect size of 0.1, then 3,528 runs. This last effect is but 10% of a standard deviation, but if we are interested in that difference, and the model predicts such differences, we can have the statistical power to detect it.

< Insert Table 4 about here>

< Insert Table 5 about here>

DISCUSSION AND CONCLUSIONS

We have presented an example of how many times to run a simulation to understand its predictions. This model's behavior represents theoretical predictions. Therefore, the theory's predictions should be as stable as possible. The results of our example model illustrate that models, where possible, should be run until their predictions are stable. This is particularly important when the model's performance includes predictions of variance in behavior. With a stochastic model implemented as a computer program, we do not wish to sample its behavior, but to report its predictions accurately. Thus, we recommend reporting performance based on a larger number of runs than appears to be typically done, and reporting the variance in the predictions. The results shown in Figures 2, 3, and 4 show that running a model once, or twice, or even several times per human experimental subject, typically will not provide completely accurate predictions and will sometimes provide uncharacteristic predictions.

The power calculations presented here provide a rational way to choose the number of runs. The rationale uses the size of the differences between model conditions and the desired probability of finding these differences to choose the number of model runs to report. This calculation is based on a simple equation included in most introductory statistics books. The calculations are based on standard deviations, which means that the model's standard deviation or mean does not have to be known before the model is run.

Although we used 100 runs for our serial subtraction model, we recommend 150 runs as a reasonable number that provided very stable predictions for medium to large effects.

Power calculations support the use of 150 runs as a useful number for most effect sizes and phenomena of interest to this subtraction model. If one is interested in smaller effect sizes (e.g., Cohen's small effects), then more runs will be required. If one is exploring how a model works or the model runs are expensive, then fewer runs may be appropriate, allowing that there will be less power to see differences between model conditions or between model and data, and a greater likelihood that the predictions are not stable. Other effect sizes and power requirements than the ones reported here can be used as well.

These results suggest that most of the papers in Table 1 did not report stable predictions for their model. While none of the papers in Table 1 reported effect sizes per se, large effects are relatively rare, and some of the models were examining what appear to be small to medium effects. On the other hand, the model that was run 7,200 times was almost certainly run too many times, although we agree that if resources are not an issue, then it is best to err on the side of caution.

This use of power analysis particularly helps when model runs are expensive. For example, humans-in-the-loop simulations (e.g., Thiruvengada & Rothrock, 2007), models of hour-long experiments that run in real-time (e.g., Schoelles & Gray, 2001), models that work with physical robots (Ritter, Kukreja, & St. Amant, 2007), or models run over large number of parameter settings (Best, Fincham, Gluck, Gunzelmann, & Krusmark, 2008; Lovett, Daily, & Reder, 2000; Ritter, Kase, Klein, Bennett, & Schoelles, 2009) become difficult to run many times. In these cases, this calculation lets modelers know how many runs are sufficient given the effect size of interest. The power analyses and graphs of the model's output can provide guidance of how many is enough.

These analyses also encourage modelers to think about effect sizes. These are not always known, however, it is useful to consider the effect size of the effect of interest. Where the effect size is small, more subjects need to be run and the model needs to be run more times to get stable predictions. Where the effect is large, less work generally has to be done. This should encourage researchers to look at large effects first.

Would this admonition apply to other models or other aspects of models? Absolutely. These results are not dependent on the specific architecture, but rather on the fact that the predictions have a distribution of outcomes. Soar models that include stochastic elements, for example, Miller and Laird's (1996) categorization model and Soar models with stochastic memory would similarly benefit from multiple runs, and could use the same tables. Psychology experiments already use these types of calculations, or should.

These results would also apply to different statistical tests for different measures, for example, Chi-square on categorical outputs, or different analyses, such as regression, although the power calculations would be different. If the comparison of interest was another measure, such as types of errors, then the percentage and types of errors (which this model makes) becomes clearer when more of its behavior has been examined. As models become more complex, the number of runs may need to be adjusted because of the additional cost of running the model, however, the cost of running the model additional times is typically much less expensive than not accurately representing and understanding its predictions.

What does this approach not answer? It does not tell you what effect size you will find interesting, or how many times to adjust your model (related to overfitting). It does not tell you what to do if the model does not fit the data; indeed, it suggests that if you run your model long enough, your significance tests will get accurate enough to find even small differences between model and data. These remain interesting and important problems, but at least we can hope that simulations are run enough to be thoroughly understood.

Acknowledgments

Earlier versions of this work have been presented at the US Air Force Workshop on ACT-R Models of Human-System Interaction, and ONR Workshops on Cognitive Architectures. Participants there provided useful comments. This project was supported by ONR award N000140310248 and DTRA HDTRA1-09-1-0054. Axel Cleeremans, Andrew Reifers, and Lael Schooler provided comments to

improve this paper. The views expressed in this paper do not necessarily reflect the position or the policies of the US Government, and no official endorsement should be inferred.

REFERENCES

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Ball, J., Myers, C., Heiberg, A., Cooke, N. J., Matessa, M., Freiman, M., et al. (2010). The synthetic teammate project. *Computational and Mathematical Organization Science, 16*, 271-299.

Best, B., Fincham, J., Gluck, K., Gunzelmann, G., & Krusmark, M. A. (2008). Efficient use of large-scale computational resources. In *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation* 180-181. Simulation Interoperability Standards Organization: Orlando, FL.

Byrne, M. D. (2003). Cognitive architecture. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 97-117). Mahwah, NJ: Erlbaum.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review, 69*(1), 54-61.

Howell, D. C. (1987). *Statistical methods for psychology* (2nd ed.). Boston, MA: Duxbury Press.

Kirschbaum, C., Pirke, K.-M., & Hellhammer, D. H. (1993). The Trier Social Stress Test—A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology, 28*, 76-81.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, 33*(1), 1-64.

Lovett, M. C., Daily, L. Z., & Reder, L. M. (2000). A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Journal of Cognitive Systems Research, 1*, 99-118.

Lovett, M. C., Schunn, C., Lebiere, C., & Munro, P. (Eds.). (2004). *Proceedings of the Sixth International Conference on Cognitive Modelling, ICCM 2004*. Mahwah, NJ: Erlbaum. www.lrdc.pitt.edu/schunn/ICCM2004/proceedings/schedule.htm.

Mathews, G. (2001). Levels of transaction: A cognitive science framework for operator stress. In P. A. Hancock & P. A. Desmond (Eds.), *Stress, workload, and fatigue*. Mahwah, NJ: Erlbaum.

Miller, C. S., & Laird, J. E. (1996). Accounting for graded performance within a discrete search framework. *Cognitive Science, 20*, 499-537.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Quigley, K. S., Feldman Barrett, L. F., & Weinstein, S. (2002). Cardiovascular patterns associated with threat and challenge appraisals: A within-subjects analysis. *Psychophysiology, 39*, 292-302.

Ritter, F. E. (1988). Extending the Seibel-Soar Model: Presented at the Soar V Workshop held at CMU.

Ritter, F. E. (2003). Social processes in validation: Comments on Grant (1962) and Roberts and Pashler (2000). Comments as part of Symposium on Model Fitting and Parameter Estimation. In *ACT-R Workshop*, 129-130.

Ritter, F. E., Bennett, J., & Klein, L. C. (2006). *Serial subtraction performance in the cycling study* (Tech. Report No. 2006-1): Applied Cognitive Science Lab, College of Information Sciences and Technology, Penn State.

Ritter, F. E., Kase, S. E., Klein, L. C., Bennett, J., & Schoelles, M. (2009). Fitting a model to behavior tells us what changes cognitively when under stress and with caffeine. In *Proceedings of the Biologically Inspired Cognitive Architectures Symposium at the AAAI Fall Symposium. Keynote presentation*, Technical Report FS-09-01. 109-115. AAAI Press: Menlo Park, CA.

Ritter, F. E., Kukreja, U., & St. Amant, R. (2007). Including a model of visual processing with a cognitive architecture to model a simple teleoperation task. *Journal of Cognitive Engineering and Decision Making, 1*(2), 121-147.

Ritter, F. E., Reifers, A., Klein, L. C., Quigley, K., & Schoelles, M. J. (2004). Using cognitive modeling to study behavior moderators: Pre-task appraisal and anxiety. In *Proceedings of the Human Factors and Ergonomics Society*, 2121-2125. Human Factors and Ergonomics Society: Santa Monica, CA.

Ritter, F. E., Reifers, A. L., Klein, L. C., & Schoelles, M. J. (2007). Lessons from defining theories of stress for architectures. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 254-262). New York, NY: Oxford University Press.

Ritter, F. E., Van Rooy, D., St. Amant, R., & Simpson, K. (2006). Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *IEEE Transactions on System, Man, and Cybernetics, Part A: Systems and Humans, 36*(3), 592-601.

Ritter, F. E., & Young, R. M. (2001). Embodied models as simulated users: Introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human-Computer Studies, 55*(1), 1-14.

Schoelles, M. J., & Gray, W. D. (2001). Argus: A suite of tools for research in complex cognition. *Behavior Research Methods, Instruments, & Computers, 33*(2), 130-140.

Thiruvengada, H., & Rothrock, L. (2007). Time window-based performance measures: A framework to measure team performance in dynamic environments. *Cognition, Technology & Work, 9*(2), 99-108.

Tomaka, J., Blascovich, J., Kelsey, R. M., & Leitten, C. L. (1993). Subjective, physiological, and behavioral effects of threat and challenge appraisal. *Journal of Personality and Social Psychology, 65*(2), 248-260.

Tomaka, J., Blascovich, J., Kibler, J., & Ernst, J. M. (1997). Cognitive and physiological antecedents of threat and challenge appraisal. *Journal of Personality and Social Psychology, 73*(1), 63-72.

TABLES AND FIGURES

Table 1. Number of model runs compared to subject data for papers at the 2004 International Conference on Cognitive Modeling (Lovett, Schunn, Lebiere, & Munro, 2004). ng is not given. na is not applicable, as model results were presented for illustration only or the model was not stochastic.

Paper	Subjects	Model runs
Altmann & Burns, 2004	71	ng
Belavkin & Ritter, 2004	ng	ng
Brumby & Howes, 2004	20	100
Byrne et al., 2004	164	100
Chandrasekharan et al., 2004	3	10
Chartier et al., 2004	ng	100
Chavez & Kimbrough	48	20
Chong, 2004	ng	ng
Cox & Young, 2004	ng	ng
DelMisser, 2004	60	~ 8
Fu et al., 2004	32	ng
Fum & Stocco, 2004	ng	ng
Gray et al., 2004	54	48
Halverson & Hornof, 2004	24	2,520
Kushleyeva et al., 2004	10	10
Maka et al., 2004	45 essays	ng
Marnier & Laird, 2004	na	100
Martin et al., 2004	11	20
Matessa, 2004	ng	ng
Matusuka & Corter, 2004	14, ng	50, 500
Morita & Miwa, 2004	33	ng
Nason & Laird, 2004	na	500
Nellen & Lovett, 2004	160	180
Nuxoll et al., 2004	na, na	5, ng
Nuxoll & Laird, 2004	na	5
Peebles & Bothell	30	150
Rutledge & West, 2004	3	1,000
Salvucci et al., 2004	11	ng
Simen et al., 2004	3	ng
St. Amant & Ritter, 2004	6	20
Stewert et al., 2004	2,571	1,000
Taatgen et al., 2004	ng	ng
Wu & Liu, 2004	ng	7,200

Table 2. Comparison of the model's behavior for threat and challenge conditions to human data taken from Tomaka et al. (1993) per 4-minute block. Standard deviations of the model's performance are shown in parentheses.

		Cognitive Appraisal Conditions	
		Threat	**Challenge**
Human data (N=22)	**Attempts**	46	61
	Correct	42	56
	% correct	91%	92%

		Threat (EGS=1)		**Challenge (EGS=0.1)**		**ACT-R Default (EGS=0)**
Model (N=100)	**Attempts**	46.8 (3.6)	<	54.5 (3.5)	<	70.9 (1.3)
	Correct	42.5 (5.1)	<	50.2 (5.1)	<	70.9 (1.3)
	% correct	91%		92%		100%

Note. < denotes significant difference at $\alpha = 0.01$

Table 3. Power of t-tests (alpha=0.05, two-tailed) for a range of effect sizes. This table uses δ = effect size * sqrt (N/2) (Howell, 1987, p. 201-202, and associated appendix to compute power for the value of δ).

Mean Effect Size	N	δ	Power
0.1	100	0.71	< 0.17
0.2 (Cohen's small)	100	1.41	0.29
0.5 (Cohen's medium)	100	3.54	0.94
0.8 (Cohen's large)	100	5.66	> 0.99
2.14 (effect size reported here in the subtraction model)	100	15.13	> 0.99

Table 4. The required number of runs (N) to find the given effect sizes (for t-tests with alpha=0.05, two-tailed) for a range of effect sizes with power = .90. This table uses δ = effect size * sqrt (N/2) (Howell, 1987, p. 201-202, and associated appendix to compute power for the value of δ).

Mean Effect Size	N	δ	Power
0.1	2,178	3.30	0.90
0.2 (Cohen's small)	545	3.30	0.90
0.5 (Cohen's medium)	88	3.30	0.90
0.8 (Cohen's large)	34	3.30	0.90
2.14 (effect reported here)	5	3.30	0.90

Table 5. The required number of runs (N) to find the given effect sizes (for t-tests with alpha=0.05, two-tailed) for a range of effect sizes with power = .99. This table uses δ = effect size * sqrt (N/2) (Howell, 1987, p. 201-202, and Appendix Power to compute power for the value of δ).

Mean Effect Size	N	δ	Power
0.1	3,528	4.20	0.99
0.2 (Cohen's small)	882	4.20	0.99
0.5 (Cohen's medium)	142	4.20	0.99
0.8 (Cohen's large)	56	4.20	0.99
2.14 (effect reported here)	8	4.28	0.99

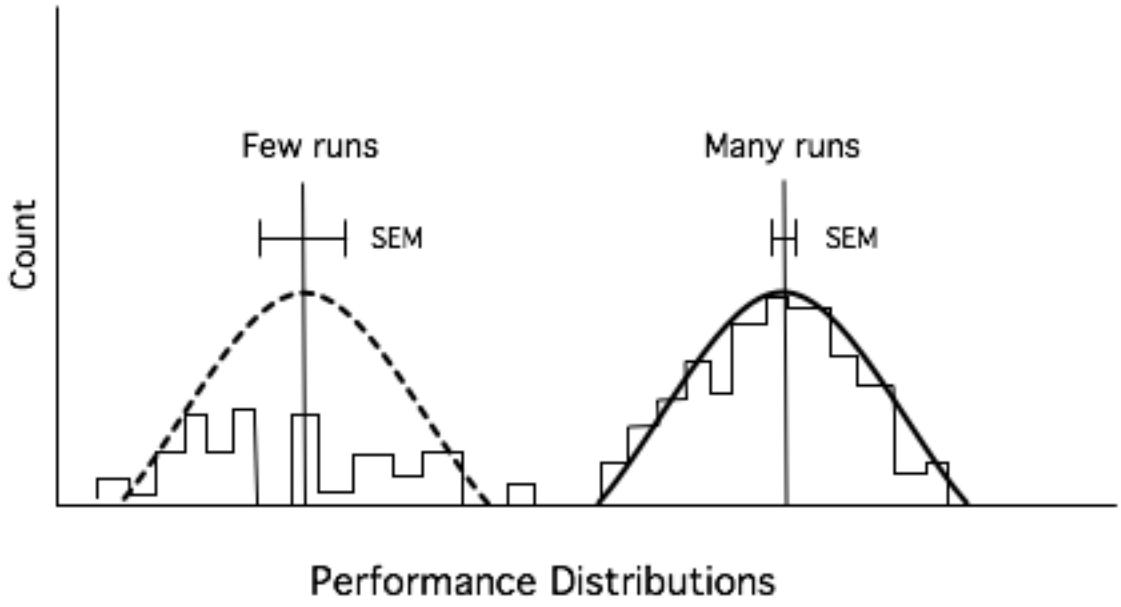

Figure 1. The distribution of performance, mean, and standard error of the mean for a model run a few times (left) and run many times (right). The distribution for the few runs is dashed to show that it is a less accurate representation.

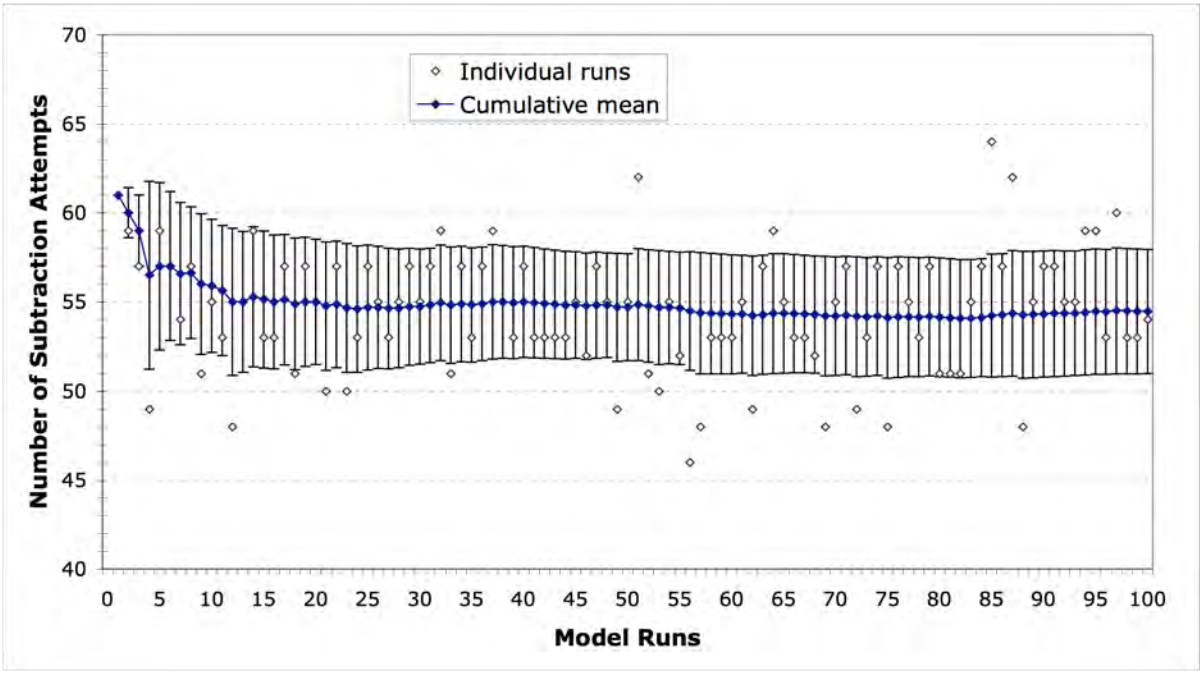

Figure 2. The predicted number of total attempts and cumulative standard deviation as error bars across the 100 runs of the model with a Challenge setting.

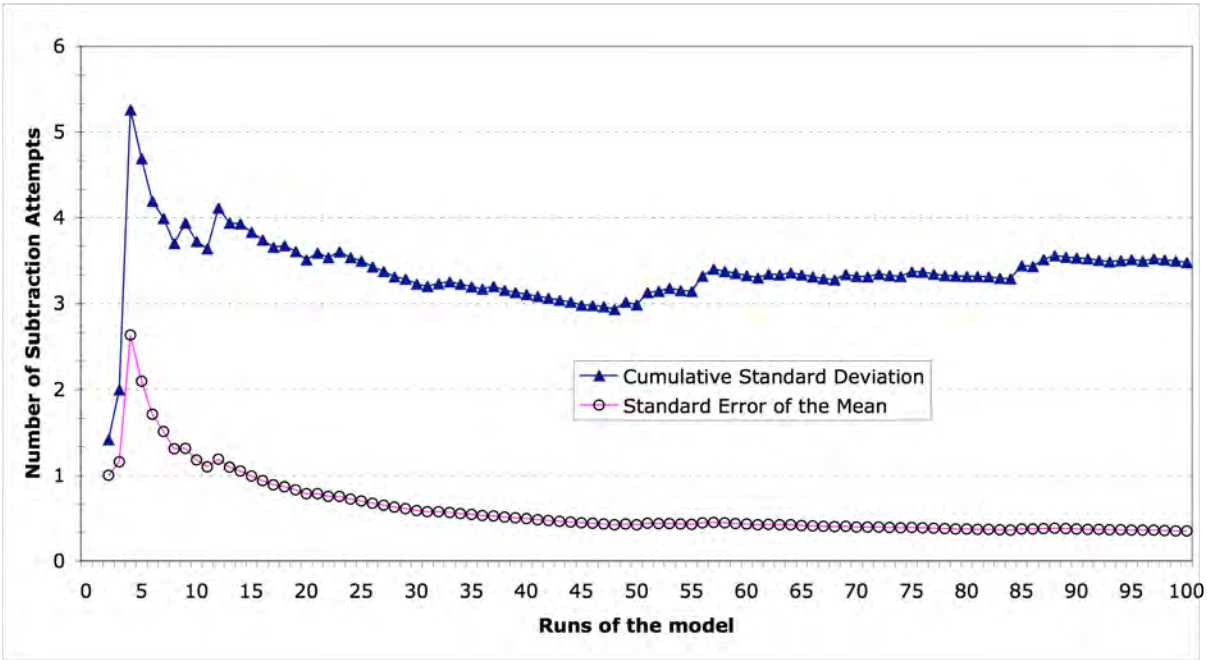

Figure 3. The cumulative standard deviation and the cumulative standard error of the mean across 100 runs of the model with a Challenge setting.

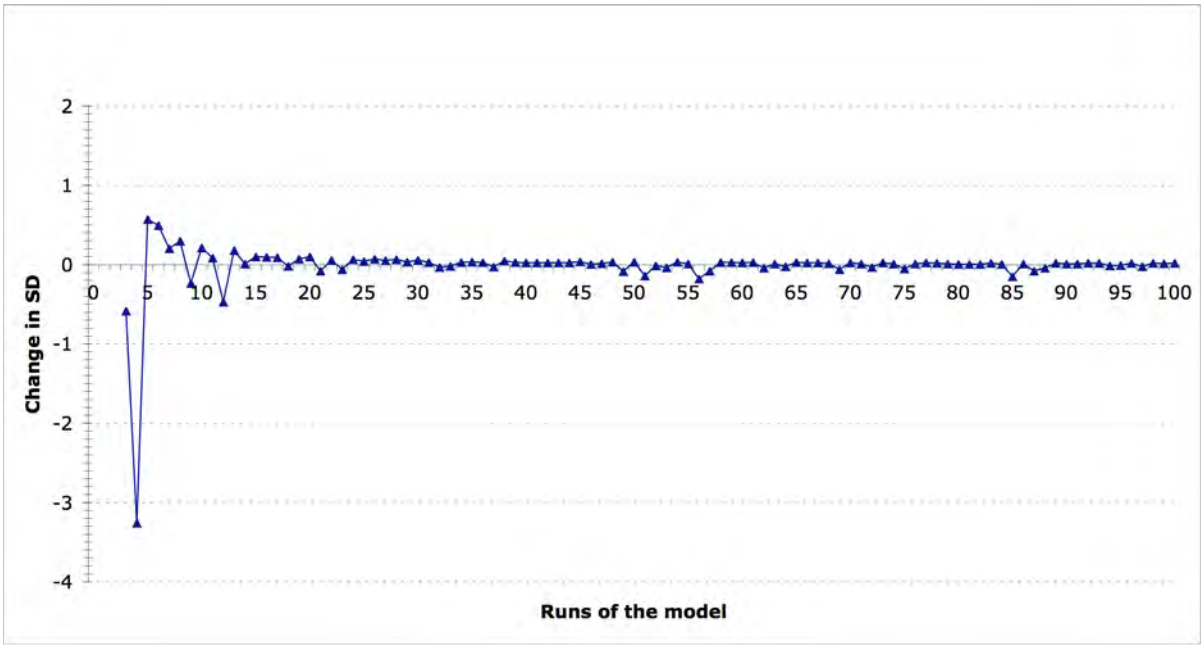

Figure 4. The change in standard deviation (between run N and N-1) across the 100 runs of the model with a Challenge setting.