In D. Reitter & F. E. Ritter (Eds.), Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016). University Park, PA: Penn State.

Efficient Parameter Estimation of Cognitive Models for Real-Time Performance Monitoring and Adaptive Interfaces

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)
711^{th} Human Performance Wing, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA

Matthew M. Walsh (mmw188@gmail.com)
Tier1 Performance Solutions, Covington, KY, USA

Leslie M. Blaha (leslie.blaha@pnnl.gov)
Visual Analytics, Pacific Northwest National Laboratory, Richland, WA, USA

Glenn Gunzelmann (glenn.gunzelmann@us.af.mil) & Bella Veksler (bellav717@gmail.com)
711^{th} Human Performance Wing, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA

Abstract

Real-time monitoring provides an opportunity to examine the temporal dynamics of cognition, predict future behavior, and implement adaptive interfaces designed to mitigate declining performance. However, real-time monitoring poses a practical challenge because current parameter estimation methods are prohibitively slow, and real-time monitoring requires parameters to be estimated repeatedly as new data arrive. We developed a real-time parameter estimation method that involves storing pre-computed predictions in a distributed array and using it as a large look-up table. We term this method the Precomputed Distributed Look-up Table (PDLT). We applied the PDLT to an ACT-R model of the psychomotor vigilance test. PDLT estimates model parameters in just over 1 second with accuracy comparable to that of a much slower simplex method. We discuss methods for reducing the volatility of parameter estimates and the potential to scale up the PDLT method to more complex models and tasks.

Keywords: ACT-R; Psychomotor Vigilance Test; Fatigue; Parameter Estimation; Real-Time Monitoring

Introduction

Performance often declines as a result of various cognitive modifiers or stressors (fatigue, load, etc.; Gluck & Gunzelmann, 2013). In contrast to raw performance metrics, such as mean reaction time, cognitive models provide a principled method for understanding and predicting performance decrements because they formalize the cognitive processes that underlie performance. Moreover, because cognitive models specify the mechanisms underlying performance, predictions from cognitive models are more likely to show greater generalization across tasks compared to performance metrics. For these reasons, cognitive models have the potential to play a prescriptive role in the development of adaptive interfaces designed to mitigate declining performance. When a performance decrement is predicted, an adaptive interface may compensate, for example, by increasing the salience of cues, recommending a multitasking strategy, or prescribing a period of rest (e.g., Rouse, 1988).

Standard approaches to model fitting are ill-suited for implementing cognitive model-based adaptive interfaces, as they often involve pooling data across subjects and are conducted post-hoc. The practice of pooling data neglects multiple sources of variation in performance, resulting in poor individual user prediction and decreased efficacy of adaptive interfaces. For example, performance varies from individual to individual, as well as during a task performed by the same individual. Moreover, performance decrements might be attributable to several causes rather than a single cause, leading to additional variation across individuals and situations. Post-hoc model fitting is problematic because it precludes the ability to predict and mitigate cognitive modifiers.

In contrast, real-time monitoring provides the opportunity to examine variation in cognitive activity attributable to individual differences and changes caused by cognitive modifiers as they unfold (Wilson & Russell, 2003). Once parameterized, a cognitive model can make individualized performance predictions, which in principal could be used to anticipate future performance breakdowns. Individualized interventions could be administered to effectively mitigate performance decrements. With the exception of very simple models in intelligent tutoring systems (Corbett & Anderson, 1994), this application of cognitive models has not been realized because of computational limitations in model fitting.

An important challenge to overcome in using real-time monitoring is increasing the speed with which model parameters can be updated with incoming data. The primary reason for the reliance on post-hoc model fitting is practical: many cognitive models require computationally intensive simulations to generate predictions from a given set of parameters, and the best fitting parameters are often not known in advance. Thus, the parameters must be calibrated to the data using an exhaustive grid search or a search algorithm. In both cases, the process of calibrating the model to data can require hundreds or even thousands of processor hours on High Performance Computing resources (Harris, 2008), thereby rendering real-time monitoring impractical.

As a first step toward using real-time monitoring and adaptive interfaces, we developed a real-time parameter estimation method for quickly and accurately obtaining maximum likelihood estimates for simulation-based cognitive models. We termed this method Pre-computed Distributed Look-up Table (PDLT). As its name implies, this method functions much like a large look-up table in which predictions are pre-computed and stored for later use, so they can be evaluated in parallel during a real-time monitoring task. As such, nearly all of the computational burden is offloaded prior to the experiment, allowing for the possibility of real-time monitoring.

In the following section, we provide a detailed description of the PDLT method. Next, we describe two simulations using an ACT-R model of the psychomotor vigilance test. The first simulation assessed the speed of the PDLT with different sampling resolutions and amount of data. The second simulation was designed to compare the speed and accuracy of the PDLT to the simplex method.

Pre-Computed Distributed Look-up Table

A wide variety of methods exist for parameter estimation, including grid search, the simplex algorithm (Nelder & Mead, 1965), and a large class of algorithms based on principles of biological evolution (Gen & Cheng, 2000). Each method can be defined as a point in trade-off space in which speed is sacrificed for accuracy. However, even with the benefit of distributed computing, these methods are prohibitively slow for real-time monitoring. The computational bottleneck can be attributed to (1) the reliance on computationally intensive simulation, and (2) the repeated evaluation of similar portions of the parameter space. Together, these factors greatly limit the speed of parameter estimation.

Our goal in developing the PDLT method was to minimize the speed-accuracy trade-off inherent in the aforementioned methods. Our solution was to pre-compute the predictions associated with plausible parameter combinations and store these predictions in a distributed look-up table for later use. A clear advantage of this approach is that it drastically reduces the computational burden during a real-time monitoring task. A related advantage is that pre-computation allows the PDLT to easily scale up to more complex models.

The PDLT method, displayed schematically in Figure 1, entails the following three steps:

Step 1. As represented by the top distribution, the first step is to define a distribution over the allowable parameter space Θ, from which parameter combinations θ are sampled. For illustrative simplicity, we display a unidimensional distribution over the parameter space. However, a distribution of any dimensionality can be used. The purpose of the parameter distribution is to sample plausible values of θ with a high probability. The distribution of parameters should be informed by a combination of theory and parameters obtained from previous studies. The choice of sampling resolution will depend on the the desired speed and accuracy as well as the goals of the application. Accuracy depends on the selected sampling resolution relative to the spread of the parameter distribution. A more granular sampling resolution will suffice if the goal is to identify qualitative patterns of behavior or cognition. However, a higher sampling resolution might be required if the goal is to generate predictions because small quantitative errors may become larger and larger the further out the predictions are extended.

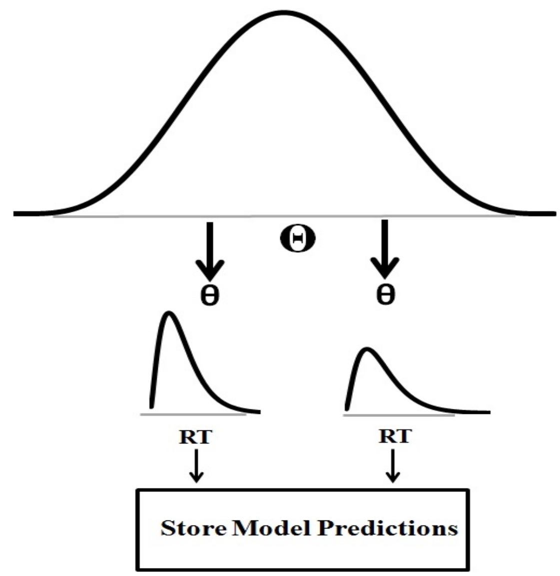

Figure 1: A schematic of the PDLT method. At the top, a distribution is defined over the parameter space. In the middle, predictions are simulated from the model using specific parameters. At the bottom, the predictions are summarized with a statistic and stored in a distributed look-up table.

Step 2. Simulated data are generated from the model for each θ, and predictions are summarized with a statistic. The middle level of Figure 1 indicates that the behavioral output of our model is a reaction time (RT) distribution. In general, a variety of statistics can be used, including means, proportions, quantiles, and kernel density functions.

Step 3. The statistics are stored for later use in any data structure necessary for a given application. Once the distributed arrays are stored, steps 1 and 2 can be omitted in future applications unless the parameter distribution requires modification due to poor model fit.

Evaluation

We performed two simulations using an ACT-R model of the psychomotor vigilance test (PVT). In the first simulation, we examined the speed of the PDLT method as a function of sampling resolution and the number of trials to which the model was fit. In the second simulation, we validated the PDLT method with a parameter recovery study comparing its accuracy to that of the simplex method. Note that although we demonstrate the PDLT on the PVT, the method will extend to most simulation-based cognitive models and tasks.

Common Methods

Hardware

Simulations were performed on a cluster of four Mac Pro computers, each with 16 3.0 GHz cores and 32 GB RAM.

Software

Parameter recovery simulations were programmed in Julia 0.4, a fast, high-level, scientific computing language available via an open source MIT license (Bezanson et al., 2014).

Task

We used the PVT (Dinges & Powell, 1985) to assess the adequacy of PDLT for real-time parameter estimation. The PVT is an ideal task for testing real-time parameter estimation as a proof of concept because of its relative simplicity and its sensitivity to changes in fatigue occurring within 2 to 5 minutes (Loh et al., 2004). The PVT is a simple 10-minute detection task that requires subjects to respond as quickly as possible once the stimulus appears on a test device. The stimulus appears after a random 2-10 sec inter-stimulus interval (ISI). RT distributions from the PVT are empirically rich and are often used to examine the effects of sleep deprivation and fatigue on sustained attention (Lim & Dinges, 2008). A hallmark of fatigue is an increase in the mean, variance, and skew of the RT distribution. Furthermore, an increase in both false starts (RTs before or within 150 ms of stimulus onset) and lapses (RTs > 500 ms) is typically observed as fatigue increases.

Model

We used an ACT-R model of the PVT (Gunzelmann et al., 2009) to demonstrate the use of the PDLT method for two reasons. First, the PVT is of practical relevance for real-time monitoring as it is sensitive to fatigue due to sleep deprivation (Walsh et al., 2014) and time on task (Veksler & Gunzelmann, Under Review). Second, the model has been validated as a plausible account of fatigue decrements in the PVT (Gunzelmann et al., 2015). The model posits that the PVT performance can be characterized with three productions: (1) wait for the stimulus to appear; (2) attend to the stimulus; and (3) respond to the stimulus. During each production cycle, a production is selected stochastically according to partial matching between production rules and the internal and external conditions represented by the model. This partial matching between productions and conditions allows false starts to occur with some low probability. Two parameters are principally responsible for the probability of production selection–a utility scalar (U_S) and a utility threshold (U_T). Eq. 1 provides a formal representation of the production utility:

$$U_{ij} = U_S(U_i - MMP_{ij}) + \varepsilon \tag{1}$$

U_{ij} is the utility of production i in state j, U_S is the utility scalar, U_i is the stored utility for production i, MMP_{ij} is the mismatch penalty for production i in state j, and ε is logistically distributed noise. A value of U_{ij} is assigned to matches, and a value of 0 is assigned to mismatches, yielding a symmetrical payoff matrix. An important feature of the payoff matrix is that the mismatch penalty ensures that the incorrect production is enacted with low probability. The production with highest utility is selected and enacted if its utility exceeds the utility threshold, U_T:

$$\text{Production} = max(U_{ij}) \text{ if } max(U_{ij}) > U_T \tag{2}$$

In the event that no production utilities exceed the utility threshold, no production is enacted, resulting in a microlapse. When a microlapse occurs, utility in Eq. 1 is decremented by a scalar, FP_{dec}: $U_S = U_S \cdot FP_{dec}$. The likelihood of microlapses increases in subsequent production cycles. This leads to behavioral lapses (RTs > 500 ms), and generally lengthens the right tail of the RT distribution. The difference between U_S and U_T, denoted as *Diff*, is an important indicator of fatigue. Relatively low values of *Diff* are associated with greater fatigue. The parameter *cycle time* controls the duration of conflict resolution at the start of each production cycle. The summed duration of each of these processes constitutes the observed RT. Importantly, the duration of each process is stochastic, giving rise to the characteristically right-skewed RT distribution. In summary, the ACT-R model uses four free parameters: U_S, U_T, FP_{dec}, and *cycle time*.

PDLT Specification

The first step in implementing the PDLT method is to define a distribution over the plausible parameter space. We opted for multivariate Gaussian distributions because they account for the central tendency and covariance structure in the empirical parameter estimates, thereby ensuring that unlikely parameter combinations are not needlessly evaluated (see Table 1). The model parameters were derived from reported parameters of related models (Walsh et al., 2014; Veksler & Gunzelmann, Under Review). The parameters were based on well-rested and fatigued subjects to capture a wide range of behavior. One multivariate Gaussian distribution was based on 33 well-rested subjects (Walsh et al., 2014; Veksler & Gunzelmann, Under Review) and the other was based on 13 subjects who underwent 72 hours of sleep-deprivation (Walsh et al., 2014).

We used a high sampling resolution of 150,000 parameter combinations to achieve a relatively high degree of accuracy. During parameter estimation, this required about 2.90 GB of RAM on the primary core and about 325 MB for the remaining cores. The parameter combinations were evenly sampled from multivariate Gaussian distributions associated with well-rested subjects and sleep-deprived subjects.

We used kernel density functions as our fit statistic because they can be used to find maximum likelihood estimates, which have desirable statistical properties, such as consistency and efficiency (Van den Bos, 2007). A kernel density function uses empirical or simulated data to approximate a continuous probability density function. The estimation process involves weighting existing data points as a decreasing function of distance from the target point.

For each parameter combination, a kernel density function was estimated from 64,000 simulated trials. A large number of simulations were used with a small bandwidth (.008) to prevent distortion of the distribution where the false starts end and alert responses begin. Distortion can otherwise occur because the kernel density estimator will be weighted heavily

Table 1: Mean and standard deviations (SD) of parameters used in the PDLT method.

	Utility	Threshold	FP_{dec}	Cycle Time	Diff
Well-Rested	5.86 (1.27)	5.04 (1.00)	.99 (.01)	.05 (.01)	.83 (.63)
Fatigued	3.50 (.51)	3.73 (.50)	.98 (.01)	.04 (.01)	-.23 (.16)

toward the more frequent alert RTs. An object was stored in a distributed look-up table, so the kernel density could be efficiently reconstructed in real time. We used the DistributedArrays package in Julia to spread the computational burden over a cluster of four desktop computers. The look-up table required approximately 35 hours to generate.

Simulation 1

Simulation 1 examined the speed of the PDLT method with different sampling resolutions and number of RTs. The size of the look-up table was varied from 1 to 150,000 in 10 equally spaced increments. The number of trials was either 100 or 1,000. The results were averaged over 100 repetitions for each combination of sampling resolution and number of RTs to produce a stable estimate.

Results

Figure 2 shows the mean completion time for the PDLT method as a function of sampling resolution and number of RTs. Across all combinations, the mean completion time was well below the minimum ISI of 2 seconds in the PVT, indicating that the PDLT method is suitable for trial-by-trial monitoring of task performance on the PVT.

Results indicate that increasing the sampling resolution produces a modest linear increase in completion time. Additionally, increasing the number of trials results in a small overall increase in completion time. By comparison, the simplex method requires about 1-4 minutes to fit the model using the same hardware and software, depending on the number of iterations, number of starting points, and number of simulations per evaluation.

Simulation 2

Simulation 2 was a parameter recovery study designed to compare the PDLT to the simplex method. Parameter recovery involves generating simulated data from a model with known parameters and fitting the model to the simulated data to assess the accuracy of the parameter estimates.

A total of 100 parameters were selected as the ground truth. Half were from the multivariate Gaussian distribution for well-rested subjects, and half were from the sleep-deprived subjects' multivariate Gaussian distribution (Table 1). For each parameter combination, 50 trials were simulated with the ACT-R model. The ACT-R model was fit to these simulated data sets using the PDLT and simplex methods.

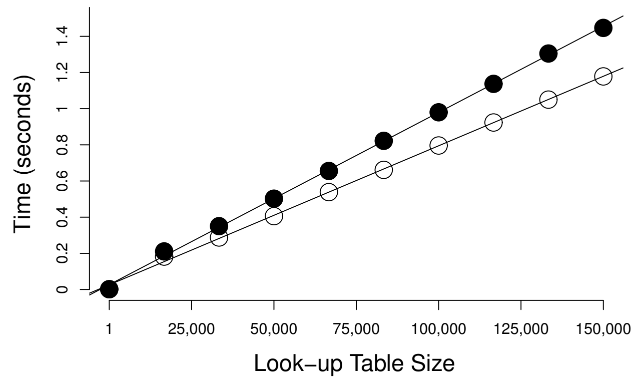

Figure 2: Mean completion time as function of look-up table size and number of trials. Filled circles denote 1,000 trials and unfilled circles denote 100 trials.

Simplex Method

We compared the PDLT to the simplex (Nelder & Mead, 1965) because it is commonly used, widely available, and applies to nonlinear models without tractable derivatives, such as ACT-R. Other algorithms will produce similarly long completion times due to their reliance on simulation. To ensure robustness to local maxima, we computed the likelihood of 50 candidate starting points and initialized the simplex with the three with the highest likelihood. Forty-nine candidate starting points were sampled from the multivariate Gaussian distributions, and the remaining candidate starting point was the best-fitting parameters from the PDLT for each corresponding data set. Initializing the simplex algorithm in this manner provides a rigorous test of the PDLT method because the simplex method has the benefit of fine-tuning the PDLT estimates in addition to using multiple starting points.

Next, we ran the simplex on three starting points with the highest likelihood. For each starting point the algorithm performed 100 iterations before recording the best-fitting parameters for that particular run. Upon each evaluation (five per iteration), 10,000 trials were simulated. Proposed parameter sets were evaluated with quantile maximum likelihood estimation, a discrete approximation to maximum likelihood estimation (Heathcote et al., 2002). Following Walsh et al. (2014), we binned false starts (RTs $<$ 150 ms) separately and binned the remaining RTs according to 20 quantiles. The parameters associated with the best fit were then selected.

Results

Compared to the simplex, the completion times for the PDLT were faster on average (1.32 vs 49.17 sec) and less variable (SD = .02 vs SD = 12.26 sec). We computed the correlation between the time to simulate the model using the recovered parameters and the corresponding completion times for each method. As expected, the correlation was nearly zero for the

PDLT ($r = .04$) but markedly higher for the simplex ($r = .94$). This underscores the inability of the simplex to scale up to slower models requiring more simulation time.

We assessed the ability of each method to accurately recover the parameters with two metrics: relative bias and correlation. Relative bias provides a standardized measure of the deviation between the true and recovered parameters using the following formula: $RB = \frac{(\hat{\theta}-\theta)}{\theta}$, where $\hat{\theta}$ is the estimated parameter and θ is the true parameter. The correlation measures the degree to which the true and recovered parameters are linearly related independent of systematic bias. Lower correlations are indicative of more noise in the parameter estimate. As shown in Table 2, mean relative bias was generally low for both methods, indicating an accurate parameter recovery. In general, the simplex method exhibited slightly less bias than the PDLT method.[1]

Table 2: Mean *RB* between true and recovered parameters.

	Utility	Threshold	FP_{dec}	Cycle Time	Diff
PDLT	.09	.08	.00	.01	-.47
Simplex	.06	.06	.00	.00	-.45

Table 3 shows that the correlations were generally high for both methods, providing more evidence of accurate parameter recovery. In terms of correlation, the PDLT method performed slightly better than the simplex method. Taken together, these results indicate that the PDLT can achieve similar accuracy as the simplex in just a faction of the time. In contrast to the simplex, the independence between simulation time and fit time for the PDLT indicates that it can easily scale up to slower, more complex models.

Table 3: Correlations between true and recovered parameters.

	Utility	Threshold	FP_{dec}	Cycle Time	Diff
PDLT	.89	.74	.64	.89	.93
Simplex	.85	.73	.42	.84	.89

Discussion

Real-time monitoring using cognitive models provides an opportunity to examine changes in cognitive processes, predict performance decrements, and implement adaptive interfaces designed to mitigate performance decrements. A practical challenge in performing real-time monitoring is updating cognitive models with new data as they arrive. Established parameter estimation methods are unable to accurately estimate parameters of simulation-based models in real time. To overcome this limitation, we developed a real-time parameter estimation method, and validated it on a simulation-based ACT-R model of the PVT. PDLT pre-computes the models predictions in order to offload the computational burden associated with simulating model predictions. The results of our simulations demonstrate several important points. First, the PDLT is fast, even with a high sampling resolution and large number of RTs. Second, the PDLT can achieve a similar level of accuracy compared to the simplex method. Third, unlike other methods, the PDLT shows potential to scale up to more complex models due to the separation of simulation time and parameter estimation time. Together, these findings suggest that cognitive models can be updated on a trial-by-trial basis for many tasks.

Scaling Up

Although we used a relatively simple task and model for demonstration, there is reason to believe the PDLT is scalable to more complex models. For example, we demonstrated that the PDLT is invariant to the time required to simulate the model, unlike the simplex method. Other methods that require real-time model simulation for parameter estimation will likely show limited scalability. The ability of the PDLT to scale up can be attributed to the use of pre-computation.

There are some situations in which speed-accuracy trade-offs will be inevitable. For example, complex models that produce multiple responses will require the storage and evaluation of more predictions. In practice, this trade-off may be relatively small given that many tasks permit only a handful of responses. Similarly, dynamic models, whose parameters change during a task, require additional predictions to be stored. We recommend constraining the parameters changes to be some function of time or trial to reduce the parameter space and number of stored predictions. For example, our model could be extended to allow U_S or U_T to change as a function of trial. It may also be advisable to use quantiles rather than kernel density functions for dynamic models in order to reduce computation time. Additionally, some speed-accuracy trade-offs will occur with models that span large parameter spaces and have low correlations among parameters.

Parameter Volatility

A potential challenge with real-time monitoring is dealing with volatility in parameter estimates. One solution might be to incorporate data from previous sessions during real-time monitoring. Incorporating previous data would serve as a sort of quasi-prior, forcing the estimate to stabilize around an informed value. However, this may have the undesirable effect of making fatigue-relevant parameters less sensitive to changes in fatigue. An alternative approach might be to fix parameters that are invariant to changes in fatigue based on previous results. For example, Walsh et al. (2014) found evidence that *cycle time* and FP_{dec} vary across individuals but are invariant to the effects of fatigue. A drawback of this approach is that it would require a separate PDLT for each person. Alternatively, it might be possible to fix certain pa-

[1] The high relative bias for *Diff* is an artifact of its scale. Small values in the denominator of the relative bias formula tend to inflate the resulting value. Absolute bias for *Diff* is attenuated because U_S and U_T are biased in the same direction: $Diff = (U_S + bias_{U_S}) - (U_T + bias_{U_T})$.

rameters across individuals if they have low variance and/or do not contribute substantially to the model fit.

Another method for reducing parameter volatility is constraining the model with physiological data. Walsh et al. (2014) integrated a biomathematical model of fatigue with the ACT-R model, forcing U_S and U_T to vary according to participants' sleep/wake history and circadian rhythm. Physiological constraints serve to stabilize parameter estimates and allow for the detection of meaningful changes.

The quality of an estimate may be improved by leveraging statistical properties of composite parameters. For example, *Diff*—the difference between U_S and U_T—is generally more important than the absolute values of U_S and U_T for understanding and predicting fatigue. From a statistical standpoint, the difference of two random variables has desirable properties. Bias in *Diff* is attenuated because U_S and U_T are biased in the same direction and tend to cancel out each other. In addition, the variance of *Diff* is attenuated because U_S and U_T are correlated. Thus, the volatility of the parameters can be mitigated by exploiting the statistical properties of the model and focusing on key relationships between parameters.

Conclusions

The real-time parameter estimation method that we developed and validated is an important advance toward real-time model-based monitoring. Prior to this effort, real-time monitoring was restricted to performance metrics, such as accuracy and mean RT, and very simple models, such as those used in intelligent tutoring systems (Wilson & Russell, 2003). The PDLT method can be used with any simulation-based model and easily scales to complex models due to the use of pre-computation. We believe this new methodology will enable the use of computational models for real-time monitoring of workload and fatigue and in the implementation of adaptive interfaces. (Rouse, 1988; Green et al., 2009).

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was supported by a 711th Human Performance Wing Chief Scientist Seedling grant to G.G. and L.M.B. Distribution A: Approved for public release; distribution unlimited. 88ABW Cleared 06/06/2016; 88ABW-2016-2869.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2014). Julia: A fresh approach to numerical computation. *arXiv preprint arXiv:1411.1607*.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction*, *4*(4), 253–278.

Dinges, D. F., & Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods, Instruments, & Computers*, *17*(6), 652–655.

Gen, M., & Cheng, R. (2000). *Genetic algorithm and engineering optimization*. New York: John Wily and Sons.

Gluck, K. A., & Gunzelmann, G. (2013). Computational process modeling and cognitive stressors: Background and prospects for application in cognitive. *The Oxford handbook of cognitive engineering*, 424.

Green, T. M., Ribarsky, W., & Fisher, B. (2009). Building and applying a human cognition model for visual analytics. *Information Visualization*, *8*(1), 1–13.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*, *33*(5), 880–910.

Gunzelmann, G., Veksler, B. Z., Walsh, M. M., & Gluck, K. A. (2015). Understanding and predicting the cognitive effects of sleep loss through simulation. *Translational Issues in Psychological Science*, *1*(1), 106.

Harris, J. (2008). Mindmodeling@home: A large-scale computational cognitive modeling infrastructure. In *Proceedings of the 6th annual conference on systems engineering research.*

Heathcote, A., Brown, S., & Mewhort, D. J. (2002). Quantile maximum likelihood estimation of response time distributions. *Psychonomic Bulletin & Review*, *9*(2), 394–401.

Lim, J., & Dinges, D. F. (2008). Sleep deprivation and vigilant attention. *Annals of the New York Academy of Sciences*, *1129*(1), 305–322.

Loh, S., Lamond, N., Dorrian, J., Roach, G., & Dawson, D. (2004). The validity of psychomotor vigilance tasks of less than 10-minute duration. *Behavior Research Methods*, *36*(2), 339–346.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*(4), 308–313.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *30*(4), 431–443.

Van den Bos, A. (2007). *Parameter estimation for scientists and engineers*. John Wiley & Sons.

Veksler, B., & Gunzelmann, G. (Under Review). Functional equivalence of sleep loss and time on task effects in sustained attention.

Walsh, M. M., Gunzelmann, G., & Van Dongen, H. P. (2014). Comparing accounts of psychomotor vigilance impairment due to sleep loss. In *Proceedings of the 36th annual conference of the cognitive science society* (p. 877-882). Pasadena, California.

Wilson, G. F., & Russell, C. A. (2003). Real-time assessment of mental workload using psychophysiological measures and artificial neural networks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *45*(4), 635–644.