Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

TRACKING AND DETECTING SYSTEMATIC ERRORS IN DIGITAL TWINS

Luke A. Rhodes-Leader

Department of Management Science
Lancaster University
Lancaster, LA1 4YR, UK

Barry L. Nelson

Dept. of Industrial Engr. & Mgmt. Sci.
Northwestern University
Evanston, IL 60208-3119, USA

ABSTRACT

Digital Twins (DTs) have immense promise for exploiting the power of computer simulation to control large-scale real-world systems. The key idea is to evaluate or optimize decisions using the DT, and then implement them in the real-world system. Even with best practices, the DT and the real-world system may become misaligned over time. In this paper we provide a statistical method to detect such misalignment even though both the simulation and the real-world system are inherently stochastic. An empirical evaluation and a realistic illustration are provided.

1 INTRODUCTION

There has been a significant trend in recent years of using simulation models to tackle problems related to operational decision making and system control, moving beyond their traditional role in system design. This trend has been enabled through the development of technologies such as Digital Twins (DTs, Biller et al. 2022) and Symbiotic Simulation (Onggo et al. 2018). A DT is a simulation model running alongside a real-world or physical system. Real-time (or close to real-time) data, possibly collected through the Internet of Things, are fed into the simulation model to keep it updated. Using this data, the DT can predict the future evolution of the system or evaluate the performance of a variety of actions, often via a simulation-optimization algorithm applied at a decision point. The results of this simulation experiment can then be applied within the real-world system or used to alert a decision maker of impending problems. The implementation of an action completes a cycle of information exchange from the real-world system to the DT and back.

Evaluation of the efficacy of DTs has not received as much attention in the literature as their application. This paper considers the problem of assessing whether what happens in the real-world system is consistent with the predictions of the DT; if not, this may indicate that they are not sufficiently aligned. We propose a method to detect a particular type of misalignment.

Biller et al. (2022) distinguish between “asset twins," which are typically physics-based simulation models of equipment, and “process twins,” which model business processes. Process twins are more in line with the Operations Research/Management Science (ORMS) discipline’s use of simulation, and particularly stochastic simulation. Process twins are the focus of this paper.

As the real-world system evolves, real-time data are used to keep the simulation model up-to-date with the system conditions and also with any changes in the processes over time. Changes could require re-calibration of the simulation model, an area Biller et al. (2022) identify as needing further research.

Applications of DTs and symbiotic simulation span various sectors, from manufacturing and smart factories (Overbeck et al. 2021; Hua et al. 2022), to transportation (Ambra and Macharis 2020) and healthcare (Harper and Mustafee 2019). This is sparking various methodological developments to cope with the real-time nature of the setting, such as in synchronization (Tan and Matta 2022), validation

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

(Lugaresi et al. 2022), online optimization (Cao et al. 2021), partially offline optimization (Goodwin et al. 2022) and integrating with machine-learning methods (Greasley et al. 2022).

A DT, along with its associated dynamic updating, calibration and optimization methods, is a decision-support tool. As with any decision-support tool, it is important to evaluate and assess whether it is leading to good decisions. This is particularly important in the DT context as the DT will be used repeatedly, meaning that poor predictions could have significant consequences in the long run. Unlike in the traditional system design application of simulation—for which a final evaluation rarely occurs since there is no going back—there is a clear opportunity to utilize the feedback of what actually happened in the real-world system to evaluate the DT's decision-support effectiveness.

The setting we consider in this paper is when from time to time the DT is queried to recommend a decision for the real-world system; in the example later in the paper the decision is where to relocate fire engines to maintain rapid response times. These decisions are the result of a simulation optimization of some sort, perhaps only heuristic. Typically, the simulation optimization's objective function is the expected value of some system response. *Critically, we assume that the corresponding real-world response is also observable, perhaps after some time delay.*

We argue that the evaluation of DT-based decisions is non-trivial. We cannot simply ask "did things turn out as expected?" The real system will be subject to natural variability and uncertainty. We could find and implement the expected-value optimal decision, but then be unlucky in how reality unfolds, resulting in a poor outcome. A naïve evaluation could then lead us to decide that the DT is performing poorly even if it is perfectly aligned with the real-world system.

An appropriate analogy is the quality control problem in manufacturing: detect systematic "out of control" issues quickly, while not being fooled by the normal process variability. However, unlike quality control, real-time context or covariate information is a key part of a DT's ability to optimize the control decision (Goodwin et al. 2022). It is possible that each decision made using the DT comes with a unique context or covariate, in which case we should expect each real-world outcome to have its own unique conditional distribution. Thus, the tracking of real-world and simulation responses has to be context or covariate sensitive to have any meaning.

The aim of this paper is to highlight these issues, and propose a statistical method that can help to detect a particular type of systematic error or misalignment over time. We do this by using an approximate distribution of the performance measure outcome at each decision, which is generated by the simulation model itself, to standardize the real-world outcomes. Once standardized to an approximately homogeneous distribution, we can perform goodness-of-fit tests to detect discrepancies between the DT and the real world.

The paper is organized as follows. Section 2 reviews literature on the related topic of validation of DTs. Section 3 provides a problem statement and introduces some notation before the proposed method is discussed in Section 4. Results from a controlled experiment and a more realistic situation are shown in Section 5, with Section 6 concluding the paper with some further discussion.

2 RELATED WORK

The usefulness of a DT is greatly influenced by its ability to stay aligned with the real world. Marquardt et al. (2021) used a manufacturing example to demonstrate the potential issues when a DT is not sufficiently synchronized with the real world, whilst Mustafee et al. (2023) discuss some challenges with attaining this in practice. Tan and Matta (2022) deal with synchronization issues by formulating the problem as an optimal control problem.

Validation of DTs is a challenge. Traditional validation methodologies are performed offline, and compare the output of the simulation against the output performance of the real world. However, as pointed out by Oakley et al. (2020), validity depends on the initial system state, which could well be different at every decision point. The proposition of Oakley et al. was to validate against relative changes in the system, which was possible for predicting bed occupancy levels across multiple hospital wards in their

paper. However, this is not always an appropriate view of the output measures and it assumes that the underlying processes of the real-world system are not evolving.

The online validation of DTs has also been considered. This is particularly important if the model is allowed to adapt whilst trying to match the changes in the real-world system. Hua et al. (2022) argue that this must be an ongoing process as the real-world system evolves in time, and highlight that availability of data can be a challenge in this instance. They suggest that if there are sufficient data then the validation should consist of trace-driven simulation; otherwise some "idealised input data" should be used. Their application considers a smart factory designed to know the position and status of every machine and part.

In a manufacturing and production setting, Overbeck et al. (2021) consider a Normalized Root Mean Square Error approach based on the throughput of the production line. Their measure considers the sum of squared differences between the amount produced in the simulation and the real world over a set of time intervals, normalized by the total amount produced in that period, and compares this to a predefined standard. This method makes no adjustment for the possibility that the variance of the production process may change during a time period. Some methods have been considered that look at the sample path of both the simulation model and the real world. Lugaresi et al. (2019) consider the validation of the input models by combining quasi trace-driven simulation (to correlate the random quantities generated within the simulation and the real system) with a signal processing approach. Morgan and Barton (2022) also compare the trajectory of the real system and the simulation model. Their suggestion is to perform a Fourier analysis on the trajectories of the system and the DT, and they propose a hypothesis test based on the Fourier coefficients of the two trajectories to discriminate between the real world and the DT. Their experimentation involves comparing the trajectories of single server queues over a long period of time. Both Lugaresi et al. (2019) and Morgan and Barton (2022) test their method using queueing examples, with thousands of jobs involved in the evaluation. Lugaresi et al. (2022) argue that quicker detection of problems is required in a real-time setting. They propose a dynamic data-warping approach, again based on quasi trace-driven simulation. Rather than statistical hypothesis testing their approach requires selecting a threshold for a quantity which is not easy to interpret.

Instead of considering the sample path of the simulation and real world, this paper takes a different perspective: we standardize the real-world system response to the decision, using the response distribution implied by the DT, and look for deviation from uniformity over many decisions.

3 PROBLEM STATEMENT

Suppose we believe that we have a well calibrated stochastic simulation DT. That is, we believe the DT is able to match the real-world system with sufficient detail to accurately produce a distribution of the output performance measure of interest that matches the real world; the calibration could be that we know whatever bias there is and can correct for it, or that we are able to dynamically update model parameters with real-world data. The DT acts as a decision-support system for decisions at various points in time. At these decision points specific context or state information from the real world is transferred into the DT so that the decision is based on current information.

For notation, the DT is used to make decisions at times t_j, $j = 1, 2, \ldots, J$, and the corresponding context or state of the real-world system is Ψ_j. The decision maker can choose one action to implement, denoted x, from a set of feasible actions, $\mathscr{X}(\Psi_j)$. The key performance measure of interest in the real world will depend on both the state of the system and the action taken. We treat this as a random variable $W(x, \Psi_j)$ whose distribution $F(\cdot\,|x, \Psi_j)$ is unknown. We assume that $W(x, \Psi_j)$ can be treated as a continuous-valued random variable, such as cost or delay. Further, these decision epochs are sufficiently far apart in time to consider the decisions to be independent of each other, by which we mean that, conditional on the state of the system at times t_j and $t_{j'}$ (Ψ_j and $\Psi_{j'}$), the random variables $W(x, \Psi_j)$ and $W(x', \Psi_{j'})$ are independent.

In our formulation, the DT is used at time t_j to evaluate a variety of actions in an attempt to optimize some property of $F(\cdot\,|x, \Psi_j)$. The current system state, Ψ_j, is loaded into the simulation model so that, as much as possible, the DT state matches the real world state. However, almost certainly the simulation

model can only capture an incomplete representation of Ψ_j, which we denote as $\widehat{\Psi}_j$. The simulation is capable of generating a performance measure corresponding to real-world outcome $W(x,\Psi_j)$, an output random variable we denote by $S(x,\widehat{\Psi}_j)$ with unknown distribution $G(\,\cdot\,|x,\widehat{\Psi}_j)$. The simulation is paired with some form of optimization algorithm to choose which $x \in \mathscr{X}(\widehat{\Psi}_j)$ should be implemented. This is done by minimizing a summary of the simulation output distribution, $H_{G(S|\cdot,\widehat{\Psi}_j)}[S(\cdot,\widehat{\Psi}_j)]$; that is, we choose the action to implement, $x_j^\star$, to approximately solve:

$$x_j^\star \approx \underset{x\in\mathscr{X}(\widehat{\Psi}_j)}{\operatorname{argmin}} \left\{ H_{G(S|x,\widehat{\Psi}_j)}[S(x,\widehat{\Psi}_j)] \right\}. \tag{1}$$

A common choice is the expectation, $H_{G(S|x,\widehat{\Psi}_j)}[S(x,\widehat{\Psi}_j)] = \mathbb{E}\left[S(x,\widehat{\Psi}_j)\right]$. However, the choice of H does not impact the procedure proposed in this paper, and we could consider optimizing a quantile or some other aspect of the distribution. Once the optimization has been performed, and an action $x_j^\star$ has been selected, it is implemented in the real-world system. This will result in the real-world performance being realized, and we record the real-world performance $W(x_j^\star,\Psi_j)$.

Notice that the optimization in (1) is with respect to the simulation model performance, not the real world. The DT assumption is that the simulation model is a sufficiently good approximation of the real world, i.e., $G(\,\cdot\,|x,\widehat{\Psi}_j) \approx F(\,\cdot\,|x,\Psi_j)$, for all x. If there are systematic errors in the distribution, this could lead to bad decision making, which could be very costly if the DT is used repeatedly for many decisions. However, due to natural variability in the real world, there is a chance that the actual outcome corresponds to poor performance, even if $x_j^\star$ was the optimal decision for the real-world problem. Thus, poor performance for a single decision could result from an actual discrepancy between the DT and the real world, or could just be bad luck. This paper proposes an approach for detecting systematic errors between the simulation and the real world by combining a *series* of decisions and real-world performance observations $\{W(x_j^\star,\Psi_j)\}_{j=1}^J$. As these outcomes are realizations from heterogeneous distributions (different contexts or covariates), we propose a method for standardizing them before the detection method is applied.

For ease of notation, unless it is otherwise stated, a subscript j will be used to denote the explicit dependence of all quantities on the time and state information for the jth decision, t_j, Ψ_j and $\widehat{\Psi}_j$, so we suppress these quantities to simplify notation; e.g., $W_j(x)$ instead of $W(x,\Psi_j)$.

4 A METHOD FOR DETECTING SYSTEMATIC ERRORS

Our approach is based on hypothesis testing. We start with the null hypothesis that the simulation matches the real world, that is $S_j(x) \overset{\mathscr{D}}{=} W_j(x)$. Under the assumption that we are dealing with continuous random variables, the properties of the cumulative distribution function (CDF), $G_j(\cdot|x)$, and the Probability Integral Transformation imply that $G_j(W_j(x)|x) \sim U(0,1)$. That is, regardless of the state or the action, transforming the real-world observation through the CDF of the corresponding simulation output produces a uniform random variable. Therefore, transforming the outcomes of each of the J real-world decisions, $\{W_j(x_j^\star)\}_{j=1}^J$, in this way produces a standardized set of uniform random variables. However, if there is a mismatch between the real world and the simulation model, $W_j(x) \overset{\mathscr{D}}{\neq} S_j(x)$, then $G_j(W_j(x)|x)$ will not be uniformly distributed. The fact that this transformation leads to independent and identically distributed random variables means that we can treat the detection of model mismatches as a goodness-of-fit test for uniformity. This is the core principle behind our method.

Whilst $G_j(\cdot|x_j^\star)$ is unknown, it can be estimated by simulating N replications at decision $x_j^\star$ and using the empirical CDF (ECDF)

$$\widehat{G}_{jN}(s) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left(S_{jn}(x_j^\star) \le s\right) \tag{2}$$

where $S_{jn}(x_j^\star)$ is the output of the nth replication of the simulation of decision j under action $x_j^\star$. These simulation replications are entirely separate from those used to reach the optimized decision $x_j^\star$ to avoid dependence between the decision and the assessment. The properties of the ECDF imply that this estimation can be done to arbitrary precision provided that one chooses a sufficiently large number of replications N. In fact, the Glivenko-Cantelli Theorem (Loève 1977, page 20) states that

$$\Pr\left\{\lim_{N\to\infty}\sup_{s\in\mathbb{R}}\left|\widehat{G}_{jN}(s)-G_j(s|x_j^\star)\right|=0\right\}=1.$$

Furthermore, the pointwise rate of convergence is exponentially fast (Massart 1990):

$$\Pr\left\{\sup_{s\in\mathbb{R}}\left|\widehat{G}_{jN}(s)-G_j(s|x_j^\star)\right|>z\right\}\le 2e^{-2Nz^2}.$$

To enable the real-world observations to be compared with the simulation prediction, we standardize the observations using the simulation ECDF:

$$U_{jN}=\widehat{G}_{jN}\left(W_j(x_j^\star)\right),\ j=1,2,\ldots,J. \tag{3}$$

These random variables are approximately i.i.d. $U(0,1)$ when the real-world and simulation distributions are the same. This motivates us to apply a goodness-of-fit hypothesis test to the set of transformed observations $\{U_{jN}\}_{j=1}^J$, specifically

$$H_0\colon \{U_{jN}\}_{j=1}^J \overset{i.i.d.}{\sim} U(0,1) \qquad \text{versus} \qquad H_1\colon \{U_{jN}\}_{j=1}^J \overset{i.i.d.}{\not\sim} U(0,1). \tag{4}$$

The case H_0 follows directly from the hypothesis that the simulation and the real-world observations follow the same distribution. The rejection of H_0 provides evidence towards a mismatch, suggesting that the simulation may need updating or modification before further use. There are several possible test statistics that detect deviations from uniformity.

Under the null hypothesis we want the test to be valid (correct Type I error) for any number of decisions, J. Thus, validity depends on $\widehat{G}_{jN}$ being a close approximation of G_j for each j, which the results cited above indicate will be true when the number of simulation replications N is large. When the alternative hypothesis holds, the power of the test to reject the null hypothesis will increase as J, the number of decisions, increases. Ideally the rejection happens quickly. Our empirical results in the next section address the impact of N and J.

We do not intend this method to be applied online as part of the decision-making process. Rather, it is an offline assessment. The large number of simulation replications used to create the ECDF can be completed between decision epochs or entirely separately from the DT used for decision making. The latter may be necessary if the frequency of decisions is high. The ability to perform an offline assessment is facilitated by the availability of parallel computing because replications are easily executed in parallel.

5 EVALUATION

This section summarizes two experiments that test our method. We use two goodness-of-fit tests, the Kolmogorv-Smirnov (KS) test (Thas 2010, pages 123-129) and the Anderson-Darling (AD) test (Anderson and Darling 1952). All tests are performed with a Type I error rate of $\alpha=0.05$.

We also apply a simple control chart method to the U_{jN} values: after the Jth decision we test all $j=1,2,\ldots,J$ decisions to see if any are outside a specific interval $[\xi_J,1-\xi_J]$. If any of the J observations are outside this interval, then the DT is considered to be out of alignment with the real world. Under the null hypothesis that $U_{jN}\sim U(0,1)$, the number of standardized values outside $[\xi_J,1-\xi_J]$ follows a Binomial distribution with success probability $2\xi_J$ and number of trials J. So the probability that at least 1 value is outside the limits is $1-\Pr(Y=0)$ where $Y\sim\text{Binom}(J,2\xi_J)$. Choosing $\xi_J=\alpha/2J$ approximates the Type I error rate of α (the true error at $J=100$ is 0.0488). This procedure is similar to the modification to a $k\sigma$ control chart when testing past data (Ryan 2011, pages 92–93).

5.1 Controlled Experiments

The purpose of the controlled experiments is to assess the power of our method for detecting various ways in which the DT and real-world system could be out of alignment.

Suppose that for the jth decision under action x, the distribution of the real-world performance measure is

$$W_j(x) \sim \mathrm{N}\left(A_j x^2 + (B_j + \varepsilon)x + C_j, \delta D_j(x+1)\right)$$

where (A_j, B_j, C_j, D_j) represent the real-world context information that is exploited when the decision is made. In our experiments, we let $A_j \sim \mathrm{N}(-2,1)$, $B_j \sim \mathrm{N}(1,0.25)$, $C_j \sim \mathrm{N}(0,1)$ and $D_j \sim \mathrm{Exp}(1)$. The quantities ε and δ, however, are unknown and are not captured by the simulation model, representing a misalignment with the real-world system; these we control in this study.

At each decision point, we wish to select $x \in [0,1]$ that will maximize the expected outcome. We use a simulation model whose output is:

$$S_j(x) \sim \mathrm{N}\left(a_j x^2 + b_j x + c_j, d_j(x+1) \,|\, A_j = a_j, B_j = b_j, C_j = c_j, D_j = d_j\right)$$

and maximize $\mathbb{E}[S_j(x)]$ (exactly) to find the optimal solution $x_j^\star$. Notice that our assessment method is not dependent on actually finding an optimal decision, as whatever solution is implemented will result in a real-world observation of the performance measure, $W_j(x_j^\star)$. We then simulate solution $x_j^\star$ for N replications to obtain the empirical distribution of $S_j(x_j^\star)$, $\widehat{G}_{jN}(\cdot)$. The final step is to apply the goodness-of-fit tests to the sets $\left\{\widehat{G}_{jN}(W_j(x_j^\star))\right\}_{j=1}^{J}$.

The goal here is to evaluate the power to detect a difference as we vary the size of the deviation between the simulation and the real-world system: ε, which changes the mean and potentially the optimal solution; and δ, which changes the variance. The values of ε and δ were selected by considering a signal-to-noise-like ratio. As the impact of ε and δ change depending on the context, (a_j, b_j, c_j, d_j), and the solution, $x_j^\star$, which is dictated by the context, we treat both of these as random variables and average over the space of contexts. Let the random context be denoted by (A,B,C,D) and the corresponding optimal solution be $X^\star(A,B,C,D)$. For ε, we use

$$\mathrm{StN}_\varepsilon = \frac{|\mathbb{E}[W(X) - S(X)]|}{\sqrt{\mathrm{Var}(S(X))}} = \frac{\varepsilon \mathbb{E}[X^\star(A,B,C,D)]}{\sqrt{\mathbb{E}[X^\star(A,B,C,D)] + 1 + \mathrm{Var}(AX^2 + BX + C)}},$$

where the constants $\mathbb{E}[X^\star(A,B,C,D)]$ and $\mathrm{Var}\left(AX^2 + BX + C\right)$ are estimated through simulation. We chose values of ε such that StN_ε took values of approximately 0, 0.01, 0.025, 0.05, 0.075 and 0.1.

For δ, we used a similar ratio after setting $\varepsilon = 0$:

$$\mathrm{StN}_\delta = \frac{|\mathrm{Var}(W(X)) - \mathrm{Var}(S(X))|}{\mathrm{Var}(S(X))} = \frac{|\delta - 1|\left(\mathbb{E}[X^\star(A,B,C,D)] + 1\right)}{\mathbb{E}[X^\star(A,B,C,D)] + 1 + \mathrm{Var}(AX^2 + BX + C)}.$$

Our experience shows that detecting changes in variance is a considerably harder problem. We chose δ, both greater than 1 (increasing variance) and less than 1 (decreasing variance), so that the ratio took values approximately 0, 0.01, 0.05, 0.1, 0.25 and 0.5.

We also varied the number of replications used to create the ECDF, N, from 100 to 10,000. We found that the AD test only began to achieve the correct Type I error rates under the null hypothesis (i.e., the simulation matches the real world) when $N = 10{,}000$ replications. The control chart was similarly affected. Thus, only these results are presented. The KS test was much less sensitive to N, meeting the 5% Type I error even when $N = 100$.

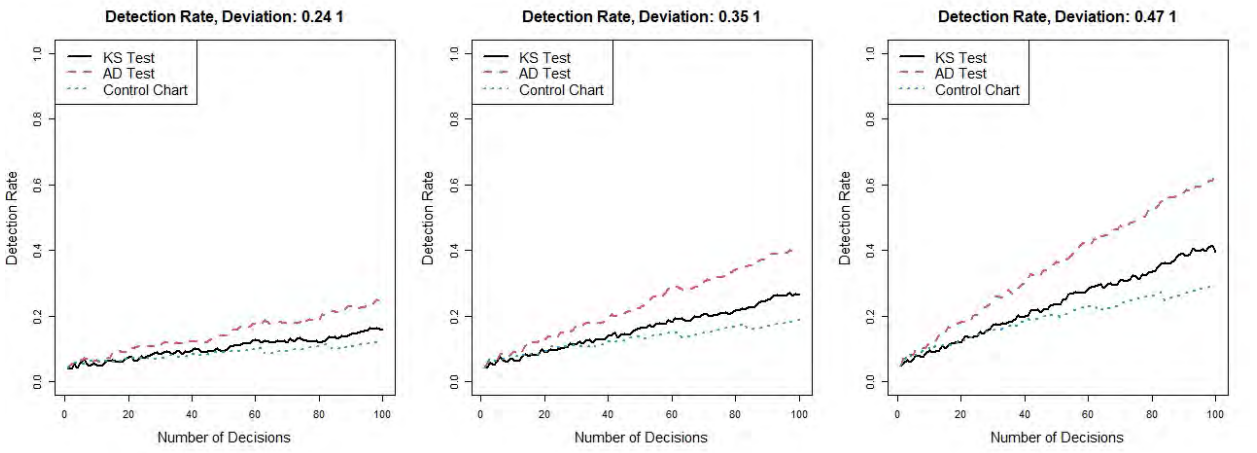

Figure 1: Detection rates for different values of $\varepsilon = 0.24, 0.35$ and 0.47 ($\text{StN}_\varepsilon \approx 0.05$, 0.075, and 0.1).

The results summarized here are based on 1000 macro-replications of $J = 100$ decisions. When $\varepsilon = 0$ and $\delta = 1$, we found that the Type I error rates are quite close to the desired levels of 0.05, although the AD test did have slightly higher average error rates.

As we change ε, keeping $\delta = 1$, the power of the tests increase, although all tests struggle to detect any errors when $\text{StN}_\varepsilon < 0.05$. Figure 1 shows the detection rates for ε set to $0.24, 0.35$ and 0.47. As expected, the more observations (decisions) included, the greater the power, and as we increase the shift, ε, we also see increased power. The AD test outperforms the other two tests.

Figure 2 shows the power and the median number of observations tested before a misalignment is detected for each test as we alter the difference in variance, but with no shift. The KS and AD tests' power increases as $|\delta - 1|$ grows, although not as quickly when $\delta > 1$. The AD test is still the best, both in terms of power and in time to detect the misalignment. The control chart is not able to perform well for $\delta < 1$; a variance-based control chart may be required for that.

Table 1 shows the power of each of the tests in a wider range of experiments. We vary the misalignment parameters, ε and δ, as well as the number of decisions used to detect the misalignment, J. From this, we can see the relationship between the misalignment and the number of decisions it takes to detect it. The three tests seem to have different strengths and weaknesses, determined mostly by the variance change, δ, and the number of decisions, J. The strength of the KS test is when the variance is reduced, $\delta < 1$, and there are not many decisions, where it outperforms the other tests. However, it performs poorly when the variance is increased, $\delta > 1$. The control chart appears to perform best when the variance is increased and there are not many decisions, while it is very poor when $\delta < 1$. The AD test is competitive under most scenarios, but is particularly strong whenever the number of decisions is large or when the variance does not change that much; it also improves the most as the shift in the mean, ε, increases.

5.2 Realistic System

In this section we include a simple simulation model that could be applied in operational decisions, based on a problem discussed by Cheng (2007). Consider a fire service in a city, modelled as a unit square, that contains 10 fire stations, each with 5 fire engines; see Figure 3. When an incident occurs, according to a spatial Poisson process, one or more fire engines from the nearest fire station are sent to the incident. The key performance measure is the percentage of incidents in which the response time is less than 15 minutes (the target is 90%). Occasionally, the fire service must deal with a major incident, in which case several fire engines are needed to respond and are unavailable until the incident is over. The cover-moves

Table 1: Detection rates for each test, at different values of ε, δ and varying the number of decisions J. Values in bold show the best performing test.

		KS test				AD test				Control Chart			
ε	δ	J=10	25	50	100	10	25	50	100	10	25	50	100
0	0.09	**0.168**	0.968	**1.000**	**1.000**	0.088	**1.000**	**1.000**	**1.000**	0.000	0.000	0.000	0.000
	0.54	**0.029**	**0.058**	0.122	0.291	0.010	0.042	**0.156**	**0.568**	0.002	0.002	0.001	0.001
	0.82	**0.035**	0.041	**0.044**	0.058	0.030	**0.043**	0.040	**0.062**	0.017	0.011	0.010	0.013
	1.00	0.043	0.046	0.050	0.044	0.046	0.059	0.056	0.068	0.046	0.042	0.049	0.055
	1.18	0.051	0.054	0.062	0.063	0.066	0.094	0.105	**0.163**	**0.096**	**0.102**	**0.110**	0.125
	1.46	0.066	0.080	0.106	0.135	0.127	0.176	0.252	**0.466**	**0.202**	**0.262**	**0.288**	0.354
	1.91	0.090	0.143	0.207	0.446	0.258	0.434	**0.663**	**0.908**	**0.355**	**0.489**	0.602	0.709
0.05	0.09	**0.173**	0.971	**1.000**	**1.000**	0.092	**1.000**	**1.000**	**1.000**	0.000	0.001	0.002	0.002
	0.54	**0.029**	**0.064**	0.128	0.308	0.010	0.043	**0.161**	**0.567**	0.002	0.003	0.004	0.004
	0.82	**0.036**	**0.043**	0.047	0.072	0.030	0.039	**0.049**	**0.076**	0.017	0.012	0.013	0.017
	1.00	0.043	0.042	0.051	0.053	**0.047**	**0.057**	**0.063**	**0.077**	0.044	0.045	0.052	0.057
	1.18	0.048	0.053	0.065	0.067	0.070	0.099	**0.114**	**0.175**	**0.095**	**0.102**	0.110	0.133
	1.46	0.069	0.084	0.110	0.149	0.125	0.186	0.258	**0.472**	**0.200**	**0.262**	**0.298**	0.362
	1.91	0.087	0.142	0.217	0.445	0.259	0.440	**0.660**	**0.902**	**0.363**	**0.493**	0.605	0.711
0.12	0.09	**0.200**	0.967	**1.000**	**1.000**	0.099	**0.999**	**1.000**	**1.000**	0.001	0.004	0.008	0.016
	0.54	**0.032**	**0.076**	0.147	0.359	0.017	0.058	**0.195**	**0.594**	0.003	0.007	0.009	0.019
	0.82	**0.042**	0.049	**0.064**	0.092	0.034	**0.050**	**0.064**	**0.116**	0.022	0.018	0.022	0.035
	1.00	0.044	0.054	0.061	0.081	**0.049**	**0.068**	**0.077**	**0.124**	0.048	0.049	0.061	0.078
	1.18	0.050	0.059	0.074	0.087	0.071	0.102	**0.128**	**0.212**	**0.097**	**0.117**	0.125	0.150
	1.46	0.061	0.083	0.118	0.189	0.126	0.195	0.285	**0.516**	**0.201**	**0.271**	**0.312**	0.375
	1.91	0.086	0.146	0.238	0.474	0.256	0.439	**0.658**	**0.918**	**0.365**	**0.503**	0.616	0.728
0.24	0.09	**0.247**	0.972	**1.000**	**1.000**	0.097	0.999	**1.000**	**1.000**	0.006	0.014	0.026	0.047
	0.54	**0.046**	**0.090**	0.229	0.502	0.029	0.088	**0.272**	**0.676**	0.013	0.022	0.034	0.055
	0.82	**0.048**	0.071	0.109	0.187	0.047	**0.080**	**0.130**	**0.255**	0.031	0.034	0.049	0.071
	1.00	0.052	0.072	0.094	0.158	**0.065**	**0.105**	**0.139**	**0.248**	0.062	0.069	0.090	0.123
	1.18	0.059	0.089	0.111	0.171	0.085	**0.137**	**0.188**	**0.344**	**0.109**	0.136	0.171	0.204
	1.46	0.077	0.116	0.154	0.292	0.151	0.236	**0.355**	**0.639**	**0.221**	**0.290**	**0.355**	0.427
	1.91	0.099	0.166	0.285	0.558	0.266	0.471	**0.709**	**0.946**	**0.371**	**0.528**	0.641	0.744
0.35	0.09	**0.309**	0.982	**1.000**	**1.000**	0.103	**0.996**	**1.000**	**1.000**	0.018	0.035	0.056	0.095
	0.54	**0.070**	0.138	0.337	0.662	0.055	**0.143**	**0.379**	**0.782**	0.020	0.040	0.062	0.114
	0.82	0.066	0.094	0.174	0.305	**0.072**	**0.126**	**0.224**	**0.412**	0.043	0.062	0.082	0.139
	1.00	0.069	0.102	0.165	0.264	**0.089**	**0.150**	**0.226**	**0.408**	0.080	0.106	0.137	0.189
	1.18	0.076	0.114	0.168	0.272	0.114	**0.181**	**0.283**	**0.514**	**0.122**	0.172	0.219	0.279
	1.46	0.085	0.148	0.213	0.399	0.173	0.275	**0.451**	**0.742**	**0.234**	**0.317**	0.392	0.503
	1.91	0.109	0.198	0.355	0.645	0.294	0.503	**0.755**	**0.967**	**0.382**	**0.545**	0.669	0.774
0.47	0.09	**0.378**	0.991	**1.000**	**1.000**	0.131	**0.997**	**1.000**	**1.000**	0.026	0.058	0.114	0.178
	0.54	**0.094**	0.210	0.445	0.788	0.078	**0.218**	**0.502**	**0.883**	0.032	0.068	0.120	0.204
	0.82	0.094	0.146	0.258	0.451	**0.103**	**0.193**	**0.348**	**0.635**	0.061	0.095	0.151	0.239
	1.00	0.097	0.140	0.236	0.394	**0.119**	**0.203**	**0.366**	**0.628**	0.104	0.142	0.206	0.294
	1.18	0.100	0.149	0.241	0.432	**0.142**	**0.239**	**0.423**	**0.684**	0.140	0.214	0.279	0.377
	1.46	0.107	0.174	0.283	0.539	0.207	0.340	**0.563**	**0.843**	**0.254**	**0.352**	0.447	0.568
	1.91	0.130	0.232	0.421	0.747	0.320	0.552	**0.810**	**0.978**	**0.399**	**0.576**	0.703	0.811

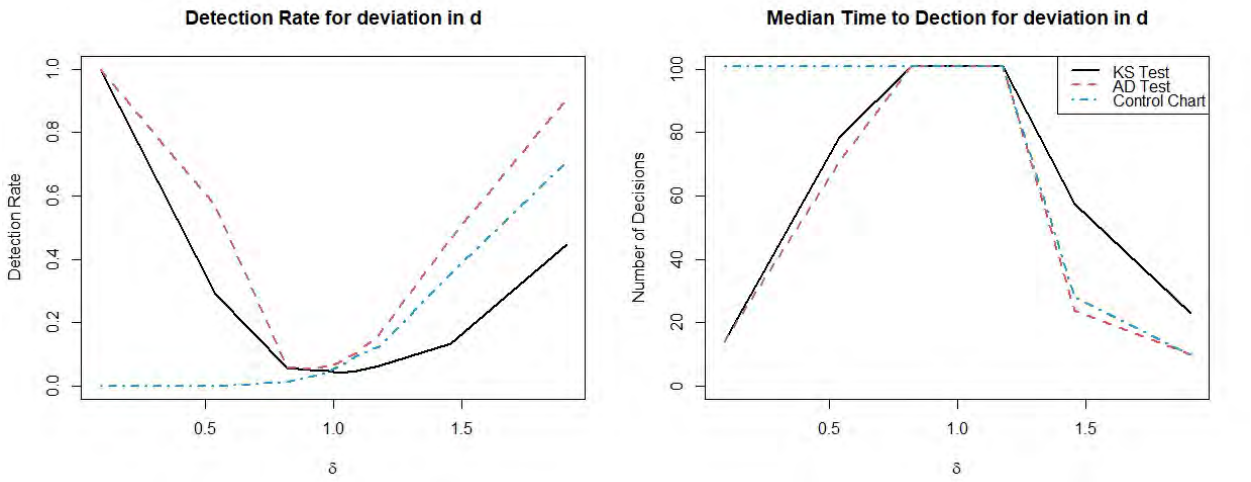

Figure 2: The power based on $J = 100$ decisions (left) and median time to detect an error (right, 101 represents not detecting within 100 decisions) of each test as we vary the variance difference δ ($\varepsilon = 0$).

problem described by Cheng (2007) involves moving some of the remaining fire engines to other stations, to cover those involved in the incident, preserving the desired response times as much as possible.

Following a randomly generated major incident, we use a DT to evaluate a number of feasible cover-moves (limited to moving 2 fire engines) with the objective function of maximizing the percentage of future incidents for which the response is within 15 minutes over a fixed time window. The Ranking & Selection algorithm KN (Kim and Nelson 2001) is used for the simulation optimization.

Suppose that, unbeknownst to the fire service or the DT, traffic signals are altered within a particular area (see Figure 3), adding a delay to travel times of up to 4 minutes from some stations to respond to some calls. This could impact to which stations the fire engines should be moved.

Figure 4 shows the results of the KS and AD tests applied to this setting, based on 100 macro-replications. The left plot is the detection rate as the number of cover-move decisions increases. The right plot is a histogram of the number of cover-move problems until the error is first detected. The AD test has the edge, but both tests detected problems within 40 decisions in 92 of the macro-replications. We also applied the control chart approach in this setting, but its performance was poor.

6 CONCLUSIONS AND OPEN QUESTIONS

In this paper we have highlighted the difficulty of evaluating the efficacy of a DT based on a stochastic simulation model. The key challenges are that natural variability of a system can lead to poor performance even under the "optimal" decision, and that the heterogeneity of decisions makes a comparison across different decisions non-trivial. Our proposed method utilizes the ability to intensively simulate the selected decision offline to estimate the full distribution of realized system performance for each of a sequence of decisions.

There are many questions about how to do this in practice, and how this method could perform or be developed to tackle issues that occur in real DTs. Firstly, our tests have not considered the case when the simulation starts off matching the real world and then the real-world processes begin to change (either as a discrete change point or a gradual divergence). This slows detection considerably, and suggests that we might only want to use a fixed number of previous decisions for testing.

The choice of which output to monitor is not necessarily straightforward, and it interacts with the time horizon for realizing the real-world response. The output has to be something that will be reliably observed in reasonable time. In some cases, this can be easily selected. As an example, consider a short-haul airline

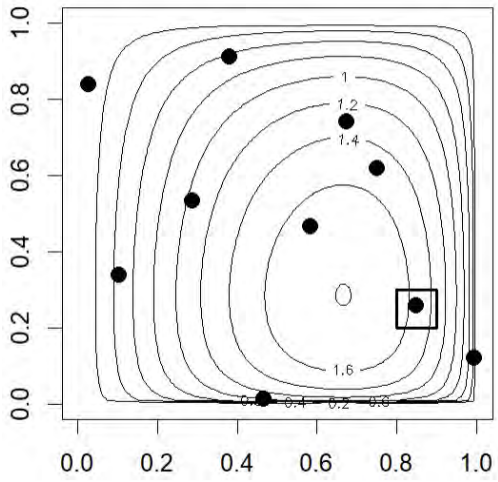

Figure 3: The station locations, contours of the incident location distribution, and the area effected by altered traffic lights.

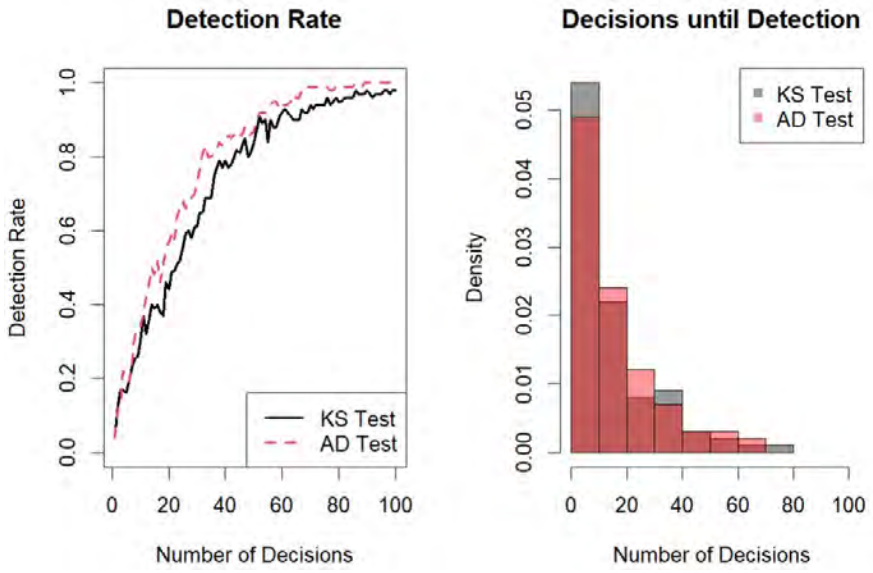

Figure 4: The detection rate (left) and time to detection (right) in the fire service simulation.

using a DT to recover from disruptions (Rhodes-Leader et al. 2022). In this case, there is a natural break overnight giving a horizon and reset point (most disruptions will not impact multiple days). Equally, any resulting delays should be known by the end of the operating day. Costs caused by the disruption may be less useful, as compensation claims may take weeks to come in.

One of our assumptions is independence among decision epochs. If the simulation state is complete, then conditional on the system state, the decision epochs should also be independent. However, if decision epochs overlap (e.g., a second major incident occurs within a decision epoch) then that would force dependence among the decisions and also call into question the relevance of the simulation output for the interrupted decision epoch. An incomplete simulation state description might not share this property, introducing correlation into the observations. This issue and its impact on the proposed method should be explored further.

The reader may be concerned that many real-world decisions might be required to detect misalignment. This is, of course, a consequence of system variability. In our experiments we used a Type I error level of 0.05, implicitly taking the position that incorrectly rejecting an aligned DT is of most concern. A larger Type I error level changes that emphasis toward incorrectly failing to reject a misaligned DT.

If making optimal decisions is all we care about then perfect alignment between the output *values* of the real world and the simulation predictions is less important than the alignment of their *optimal solutions*. That is, some errors may be tolerated as long as the decision itself is not impacted. For example, a consistent additive bias between the DT and the real-world system, but no other discrepancy, implies that they still share the same optimal decision. An important question is whether tests could be designed that are insensitive to these sorts of errors.

ACKNOWLEDGMENTS

Nelson's work was partially supported by National Science Foundation Grant No. DMS-1854562.

REFERENCES

Ambra, T., and C. Macharis. 2020. "Agent-Based Digital Twins (ABM-DT) in Synchromodal Transport and Logistics: The Fusion of Virtual and Physical Spaces". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 159–169. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Anderson, T. W., and D. A. Darling. 1952. "Asymptotic Theory of Certain 'Goodness of Fit' Criteria Based on Stochastic Processes". *The Annals of Mathematical Statistics* 23(2):193–212.

Biller, B., X. Jiang, J. Yi, P. Venditti, and S. Biller. 2022. "Simulation: The Critical Technology in Digital Twin Development". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 1340–1355. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cao, Y., C. Currie, B. S. Onggo, and M. Higgins. 2021. "Simulation Optimization for a Digital Twin Using a Multi-Fidelity Framework". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cheng, R. 2007. "Determining Efficient Simulation Run Lengths for Real Time Decision Making". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 340–345. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Goodwin, T., J. Xu, N. Celik, and C. H. Chen. 2022. "Real-time Digital Twin-based Optimization with Predictive Simulation Learning". *Journal of Simulation*:1–18.

Greasley, A., G. Panchal, and A. Samvedi. 2022. "The Use of Simulation with Machine Learning and Optimization for a Digital Twin - A Case on Formula 1 DSS". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 2198–2209. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Harper, A., and N. Mustafee. 2019. "A Hybrid Modelling Approach Using Forecasting and Real-Time Simulation to Prevent Emergency Department Overcrowding". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 1208–1219. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hua, E. Y., S. Lazarova-Molnar, and D. P. Francis. 2022. "Validation of Digital Twins: Challenges and Opportunities". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 2900–2911. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kim, S.-H., and B. L. Nelson. 2001. "A Fully Sequential Procedure for Indifference-Zone Selection in Simulation". *ACM Transactions on Modeling and Computer Simulation* 11(3):251–273.

Loève, M. 1977. *Probability Theory I*. 4th ed, Volume 45 of *Graduate Texts in Mathematics*. New York: Springer-Verlag.

Lugaresi, G., G. Aglio, F. Folgheraiter, and A. Matta. 2019. "Real-time Validation of Digital Models for Manufacturing Systems: A Novel Signal-processing-based Approach". In *IEEE International Conference on Automation Science and Engineering*, 450–455: Institute of Electrical and Electronics Engineers, Inc.

Lugaresi, G., S. Gangemi, G. Gazzoni, and A. Matta. 2022. "Online Validation of Simulation-based Digital Twins Exploiting Time Series Analysis". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 2912–2923. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Marquardt, T., L. Morgan, and C. Cleophas. 2021. "Indolence is Fatal: Research Opportunities in Designing Digital Shadows and Twins for Decision Support". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–11. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Massart, P. 1990. "The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality". *The Annals of Probability* 18(3):1269–1283.

Morgan, L. E., and R. R. Barton. 2022. "Fourier Trajectory Analysis for System Discrimination". *European Journal of Operational Research* 296(1):203–217.

Mustafee, N., A. Harper, and J. Viana. 2023. "Hybrid Models with Real-Time Data: Characterising Real-Time Simulation and Digital Twins". In *Proceedings of the Operational Research Society Simulation Workshop 2023 (SW23)*, edited by C. Currie and L. Rhodes-Leader, 261 – 270. Birmingham, UK: The Operational Research Society.

Oakley, D., B. S. Onggo, and D. Worthington. 2020. "Symbiotic Simulation for the Operational Management of Inpatient Beds: Model Development and Validation using Δ-method". *Health Care Management Science* 23:153–169.

Onggo, B. S., N. Mustafee, A. Smart, A. A. Juan, and O. Molloy. 2018. "Symbiotic Simulation System: Hybrid Systems Model Meets Big Data Analytics". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 1358–1369. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Overbeck, L., A. Le Louarn, O. Brützel, N. Stricker, and G. Lanza. 2021. "Continuous Validation and Precise Updating for High Accuracy of Digital Twins of Production Systems". In *Simulation in Produktion und Logistik 2021*, edited by J. Franke and P. Schuderer, 609–618. Göttingen: Cuvillier Verlag.

Rhodes-Leader, L. A., B. L. Nelson, B. S. Onggo, and D. J. Worthington. 2022. "A Multi-fidelity Modelling Approach for Airline Disruption Management using Simulation". *Journal of the Operational Research Society* 73(10):2228–2241.

Ryan, T. P. 2011. *Statistical Methods for Quality Improvement*. 3rd ed. Hoboken, N.J.: Wiley.

Tan, B., and A. Matta. 2022. "Optimizing Digital Twin Synchronization in a Finite Horizon". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. G. Corlu, L. H. Lee, E. P. Chew, T. Roeder, and P. Lendermann, 2924–2935. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Thas, O. 2010. *Comparing Distributions*. NY: Springer Science+Business Media.

AUTHOR BIOGRAPHIES

LUKE A. RHODES-LEADER is a Lecturer in Management Science at Lancaster University. His research interests include applications of simulation optimization and methodological aspects of digital twins. His email address is l.rhodes-leader@lancaster.ac.uk, and his website is https://www.lancaster.ac.uk/lums/people/luke-rhodes-leader.

BARRY L. NELSON is the Walter P. Murphy Professor Emeritus in the Department of Industrial Engineering and Management Sciences at Northwestern University. He is a Fellow of INFORMS and IISE. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems. His e-mail and web addresses are nelsonb@northwestern.edu and http://users.iems.northwestern.edu/~nelsonb/.