Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Mathematics and Computers in Simulation 65 (2004) 273–288

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

A *m*-fold-decimation-based technique for model validation using a single system output data set

Dimosthenis Anagnostopoulos[a,*], Vassilis Dalakas[b], Mara Nikolaidou[b]

[a] *Harokopio University of Athens, 70 El. Venizelou Str, 17671 Athens, Greece*
[b] *University of Athens, Panepistimiopolis, 15771 Athens, Greece*

Received 29 September 2003; received in revised form 5 January 2004; accepted 7 January 2004

Abstract

Using simulation to support making decisions imposes that models are thoroughly validated. We discuss widely adopted techniques for testing the validity of simulation results against real observations, as various techniques may be employed depending on the statistical features and the number of the available system/model data sets. Confidence-interval-based comparison provides certain advantages, but cannot be employed when a single system output data set is available. We propose an effective validation technique, performing confidence-interval-based comparison using a single system data set and multiple model data sets. The system data set is considered as a discrete signal and *m*-fold decimation is used, according to signal processing theory, fulfilling the essential requirements imposed to retain the statistical features of the original data set. We have evaluated both the effectiveness and the applicability of this technique, and thus have experimented with various data sets (both test and real-world). Results and conclusions are discussed to indicate the potential of the technique.

Keywords: Modelling and simulation; Model validation; Comparison techniques

1. Introduction

One of the most important steps in the development of a simulation model is determining whether this is an accurate representation of the system under study [1]. As a model is developed for a specific purpose (or application), its validity is examined with respect to this purpose. A widely accepted definition for model validation is thus “the process of determining whether a simulation model is an accurate representation of the system, for the particular objectives of the study” [2].

Various kinds of validation can be identified, e.g. conceptual model validation, structural validation and behavioral validation [3,4]. Conceptual model validation is the evaluation of the conceptual model with

[*] Corresponding author. Tel.: +30210-9549171; fax: +30210-9514759.
E-mail addresses: dimosthe@hua.gr (D. Anagnostopoulos), vdalakas@di.uoa.gr (V. Dalakas), mara@di.uoa.gr (M. Nikolaidou).

0378-4754/$30.00 © 2004 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2004.01.018

respect to the system, where the objective is primarily to evaluate the realism of the conceptual model with respect to the goals of the study. Structural validation is the evaluation of the structure of a simulation model with respect to the perceived structure of the system. Behavioral validation is the evaluation of the simulation model behavior. Model validation can also be supportive to system identification, usually in the context of monitoring and fault detection. Parameter estimation and inverse modeling capabilities may then be provided. Our concern in this paper is with behavioral validation, that is, the process of demonstrating that the behavior exhibited by the model is consistent with the expected and/or measured behavior of the system.

As one cannot establish the "absolute" validity of the model, the goal in the validation process is to gain a reasonable level of confidence so that inferences drawn from the model can be accepted with confidence, i.e. they are genuinely applicable to the real phenomenon that is being studied [3]. The most definite test for model validity is to establish that its output data closely resembles the output data that would be expected from the actual system.

When the system under study is an existing one, model validation is considerably enhanced. A number of statistical tests have been described in the literature for comparing the output from a simulation model and the corresponding real-world system. This comparison is not as simple as it might appear, as the output processes of almost real-world systems and simulations are nonstationary (i.e. the distributions of successive observations change over time) and autocorrelated (i.e. observations are correlated with each other) [5]. Classical statistical tests based on independent and identically distributed (IID) observations are thus not directly applicable. Hypothesis tests, on the other hand, are far from appropriate in this case, as a null hypothesis that the system and the model are exactly the same will be false due to the stochastic nature of simulation. It is thus more meaningful to discuss about the differences between the model and the system, to determine how close they are, and thus if the model realistically represents the system under study.

The most trivial approach to accomplish system-model data comparison is to compute one or more statistics from the real-world observations and corresponding statistics from the model data, and then to compare the two sets of statistics without the use of a formal statistical method (inspection approach) [5]. This approach is vulnerable to the inherent randomness of the observations from the system and, especially, from the model due to its stochastic nature. A better approach for system-model data comparison uses the system historical input data for running simulation experiments. This is called the correlated inspection approach, in which both the model and the system are subjected to the same input data [5]. The randomness caused by sampling from distributions to represent model input data is thus eliminated. However, this approach is totally inappropriate in the case of production runs, where the model is used to investigate the system behavior under various, yet of unknown effect conditions.

A more effective approach is to use confidence intervals to give an indication of the magnitude by which the system differs from the model. Specifically, we try to reach a reliable approximate confidence interval for the difference in the two expectations (from the system and the model). There is more than one technique that can be employed, depending on various criteria, such as the number of available data sets from the system and the model. Time series may also be employed for system/model data comparison. In most cases, such approaches require only one set of output data from the system and the model, but have other constraints (e.g. require that system output is independent from model output, or that a steady state can be reached).

Concerning the latter approaches, deriving n simulation output data sets is, in general, not difficult to accomplish. There are domains, however, where a single output set is only available, as for most military and manufacturing systems, due to the paucity of real-world data. Another such case is real-time simulation, i.e. when simulation reaches conclusions for systems behavior in real time. The term *real time*, as it relates to simulation, denotes that advancement of simulation time must occur in the real-world time (i.e. not faster or slower). In faster-than-real-time simulation (FRTS), results are delivered earlier than real-time. In FRTS, we are capable of using system observations and model results to both test model validity and, in case of a valid model, to reach predictions for the system future states. However, in both RTS and FRTS, it is most probable that, due to the limited time interval in which system and model data are collected, only one system output data set will be available.

In model validation, it is acknowledged that confidence-interval-based comparison between the system and the model provides considerable advantages, being a formal method that may also give an indication of the magnitude by which the corresponding estimated means differ. Confidence-interval-based comparison takes under consideration not only the statistical means in system (model) data sets, but also the variance of these observations. Moreover, the process of calculating confidence intervals and reaching subsequent conclusions can be automated for real-time execution. However, to be applied, at least m model output data sets are required, which is not always feasible in practice.

In this paper, we review existing comparison methods and focus on how validation may be performed based on either multiple or a single system data set. Considering the disadvantages of the last case, we introduce a technique for deriving m data sets from a single original system data set, to perform system/model comparison using confidence intervals (an early approach towards this objective has also appeared in [6]). Specifically, we use m-fold decimation for producing m sets of system observations retaining the statistical features of the original sequence. Potential applications of this technique extend to all types of simulations where model validation must be performed using a single system data set (e.g. military and manufacturing systems), and especially to RTS, where validation must be accomplished as an automated process, with minimum time overhead.

The paper is organized as follows: in Section 2, alternative comparison methods for accomplishing model validation are described. In Section 3, we discuss m-fold decimation and the proposed technique for performing a confidence-interval-based comparison using a single system data set. In Section 4, we apply the technique to various data sets, both test and real-world, to test its effectiveness and applicability, and discuss the results obtained, while conclusions reside in Section 5.

2. Model validation

In the general case, validation is accomplished based on a number (k) of measures that are under monitoring in the system and the model. These values can be stored in k corresponding variables ($\mathrm{MV}_{1} \ldots \mathrm{MV}_{k}$) and to accomplish validation, k comparisons need to be made. The values of monitoring variable i for the system and the model are, respectively, denoted as $\mathrm{MV}_{i}r$ and $\mathrm{MV}_{i}s$. For example, simulating a single-server/single-queue system, the average delay in the queue can be used for model validation. In this case, there is a single monitoring variable MV_1. $\mathrm{MV}_1 r$ and $\mathrm{MV}_1 s$ are the corresponding values for the delay in the system and the model.

In each comparison between $\mathrm{MV}_{i}r$ and $\mathrm{MV}_{i}s$, a single comparison technique is used as more appropriate for the specific data. The acceptable deviation range (dr) must be defined for each comparison, so that, when model value deviation from the corresponding system values exceeds a predetermined range, invalidity is indicated. However, as later discussed, deviation range may as well be interpreted in a different way, depending on the comparison technique employed.

When the inspection approach is used, deviation range determines the lower and upper endpoints of the interval $[l(\mathrm{MV}_i r), u(\mathrm{MV}_i r)]$ and model validity is obtained when:

$$\mathrm{MV}_i s \in [l(\mathrm{MV}_i r), u(\mathrm{MV}_i r)]$$

$$l(\mathrm{MV}_i r) = \mathrm{MV}_i r(1 - \mathrm{d}r), u(\mathrm{MV}_i r) = \mathrm{MV}_i r(1 + \mathrm{d}r)$$

The above apply when one system and one model data set are available, deriving a single observation for the system ($\mathrm{MV}_i r$) and the model ($\mathrm{MV}_i s$). If m system data sets and n model data sets are, respectively, available, a single observation is derived from each data set and $\mathrm{MV}_i r$, $\mathrm{MV}_i s$ are calculated as follows:

$$\mathrm{MV}_i r = \frac{\mathrm{sum}(\mathrm{MV}_{i1} r, \mathrm{MV}_{i2} r, \dots, \mathrm{MV}_{im} r)}{m}$$

$$\mathrm{MV}_i s = \frac{\mathrm{sum}(\mathrm{MV}_{i1} s, \mathrm{MV}_{i2} s, \dots, \mathrm{MV}_{in} s)}{n}$$

where $\mathrm{MV}_{ij} r$ ($\mathrm{MV}_{ij} s$) is the statistical sample obtained from j data set when m (n) system data sets (replications) are available. Simulating a single-server/single-queue system, for example, we execute n replications to ensure statistical validity of the results. For the first monitoring variable (MV_1) (e.g. queue delay), each replication j produces a single value $\mathrm{MV}_{1j} s$, i.e. the first replication produces $\mathrm{MV}_{11} s$, the second one produces $\mathrm{MV}_{12} s$, etc. The final value for the queue delay in the model ($\mathrm{MV}_1 s$) is calculated as the average of all values produced by the n independent replications. Considering that m sets of observations are available for the system under study, the respective final value for queue delay in the system ($\mathrm{MV}_1 r$) is calculated as the average of all the values produced by the m data sets.

A more effective approach is to use confidence intervals to give an indication of the magnitude by which the system differs from the model [5]. Specifically, we try to reach a reliable approximate confidence interval for the difference in the two expectations (from the system and the model). There is more than one technique that can be employed, depending on various criteria, such as the number of available data sets from the system (m) and the model (n). Law and Kelton suggest the use of the paired-t approach for building a confidence interval when $m = n$ (for such a case, see [7]). This approach may be applied even when results from the respective system and model data sets are correlated with each other. On the other hand, if there is no such correlation, the approach proposed by Welch [8] may be employed. This approach is used for comparing two systems with unequal and unknown variances, called the *Behrens–Fisher problem*, when output data are normally distributed [8]. This classical approach is appropriate for building a confidence interval based on a different number of data sets, as it enables m to be different than n, but imposes that system data sets are independent from the corresponding model data sets. This approach has a wider applicability, as usually $m < n$. Applying the Welch approach:

$$\overline{\mathrm{MV}_i r} = \frac{\sum_{j=1}^{m} \mathrm{MV}_{ij} r}{m}, \overline{\mathrm{MV}_i s} = \frac{\sum_{j=1}^{m} \mathrm{MV}_{ij} s}{n}$$

$$S^2(\mathrm{MV}_i r) = \frac{\sum_{j=1}^{m} [\mathrm{MV}_{ij} r - \overline{\mathrm{MV}_i r}]^2}{m - 1}$$

$$S^2(\mathrm{MV}_i s) = \frac{\sum_{j=1}^{n} [\mathrm{MV}_{ij} s - \overline{\mathrm{MV}_i s}]^2}{n - 1}$$

The estimated degrees of freedom are computed as

$$\hat{f} = \frac{[S^2(\mathrm{MV}_i r)/m + S^2(\mathrm{MV}_i s)/n]^2}{[S^2(\mathrm{MV}_i r)/m]^2/(m-1) + [S^2(\mathrm{MV}_i s)/n]^2/(n-1)}$$

The following interval is an approximate $100(1-\alpha)\%$ confidence interval for $\mathrm{MV}_i r$–$\mathrm{MV}_i s$

$$\overline{\mathrm{MV}_i r} - \overline{\mathrm{MV}_i s} \pm t_{\hat{f},1-a/2}\sqrt{\frac{S^2(\mathrm{MV}_i r)}{m} + \frac{S^2(\mathrm{MV}_i s)}{n}}$$

Evidently, for confidence intervals, the deviation range defines the value a, which means that we wish the confidence interval to cover $\mathrm{MV}_i r$–$\mathrm{MV}_i s$ with probability $1-a$. Suppose the upper and lower endpoints of the interval are marked as $u(a)$ and $l(a)$, respectively. If $0 \notin [l(a), u(a)]$, the difference between $\mathrm{MV}_i r$ and $\mathrm{MV}_i s$ is statistically significant at level a and we consider the model to be invalid. Based on the above, we acknowledge that confidence-interval-based comparison between the system and the model provides the following advantages.

1. It is a formal method that may also give an indication of the magnitude by which the corresponding estimated means differ.
2. It takes under consideration not only the means in system (model) data sets, but also the variance in these observations.
3. The process of calculating confidence intervals and reaching subsequent conclusions can be automated for real-time execution.

However, to apply this method with reliability, at least m model output data sets are required (according to the Welch approach)—note that it is not possible to increase reliability just by increasing n (i.e. the number of replications) when a single system data set is available. As already mentioned, is not possible to have m data sets for specific application domains. Furthermore, in both RTS and FRTS, it is most probable that, due to the limited time interval in which system and model data are collected, only one system output data set will be available.

3. *m*-Fold-decimation-based technique

We introduce a technique for deriving m data sets from a single original system data set, to perform model validation using the confidence interval approach. We use m-fold decimation for producing m sets of system observations retaining the statistical features of the original sequence. Potential applications of this technique extend to all types of simulations where model validation is accomplished using a single system data set (e.g. military and manufacturing systems), and especially to RTS, where validation must be accomplished as an automated process, with minimum time overhead.

Consider that $\mathrm{MV}_i r$ is calculated from a sequence of system values $x_1, x_2, \dots, x_n$, where $x(n)$ is a stochastic process and n is an integer. Having this single sequence $x(n)$, we consider it as a discrete signal and use well-established methods from digital signal processing to obtain a number of sequences that retain the statistical features of the initial sequence. A basic operation in multirate system digital signal processing is decimation. The m-fold decimator takes an input sequence $x(n)$ and produces the output sequence

$$x_m(n) = x(mn)$$

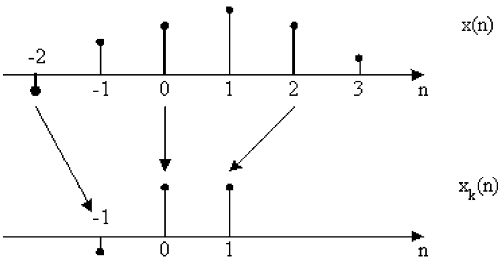

Fig. 1. m-fold decimation ($m = 2$).

where m is an integer [9]. Only these samples from $x(n)$ that occur at time equal to multiples of m are retained by the decimator. Fig. 1 demonstrates this idea for $m = 2$. As is substantiated, decimation results in aliasing, unless $x(n)$ is bandlimited [9]. To retain the statistical features of a sequence $x(n)$ using a sequence $x_m(n)$, we need to be able to reconstruct $x(n)$ from $x_m(n)$. In general, it may not be possible to recover $x(n)$ from $x_m(n)$ because of loss in information. To avoid this, m must satisfy the condition [9]

$$|\omega| < \frac{\pi}{m} \tag{1}$$

ω being the angular frequency of $x(n)$. To calculate $|\omega|$, we use the discrete time Fourier transformation (DTFT) on $x(n)$ [10].

$$X(\omega) = \sum_{n=-\infty}^{n=+\infty} x(n)\mathrm{e}^{-\mathrm{j}\omega n}$$

Fulfilling condition (1) is required since decimation causes an expansion to the Fourier transform $X(\mathrm{e}^{\mathrm{j}\omega})$ of the input signal $x(n)$. When a signal is being decimated by a factor of m, the stretched version of $X(\mathrm{e}^{\mathrm{j}\omega})$ may overlap with its shifted replicas and, in this case, $x(n)$ cannot be recovered from $x_m(n)$ (the overlap effect is called aliasing) [9].

To find the appropriate frequency $\omega_{\max}$ to bandlimit the signal spectrum in the range $|\omega| < \omega_{\max}$, we estimate the power spectral density (PSD) of $x(n)$ using the Welch method [11]. In the PSD of $x(n)$, the Fourier coefficients determine the distribution of power at the various discrete frequency components. The range of ω is $[0, \omega]$. The area below PSD(ω) in the power spectrum density is a measure of the amount of information of $x(n)$ (Fig. 2). Bandlimiting $x(n)$ in the range $|\omega| < \omega_{\max}$, frequencies above $\omega_{\max}$ are excluded. To avoid a significant loss of information, the following condition is applied:

$$\sum_{i=0}^{\omega_{\max}} \mathrm{PSD}(\omega_{\mathrm{i}}) \geq (1-l)\sum_{i=0}^{\pi} \mathrm{PSD}(\omega_{\mathrm{i}}) \tag{2}$$

Using (2), we determine the frequency range and may accordingly apply a bandlimit filter to preserve $x(n)$, depending on the choice of l. The maximum frequency preserved is $\omega_{\max}(l)$. The smaller the value of l, the smaller is the information loss. Bandlimiting the signal is achieved through a digital filter, i.e. a filter that takes a sequence of numbers (the input signal) and produces a new sequence of numbers (the filtered output signal). Applying a simple low-pass filter is required, i.e. one that does not affect low frequencies

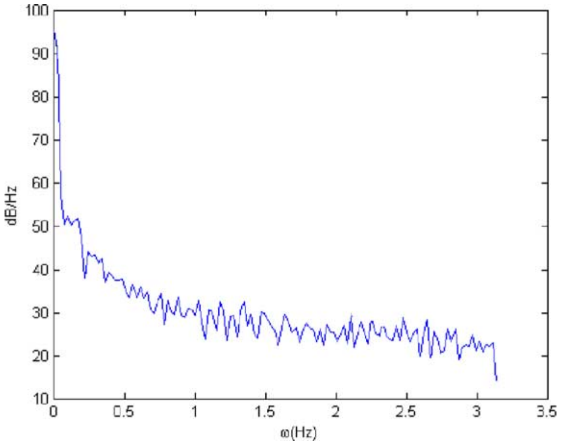

Fig. 2. Example PSD of $x(n)$.

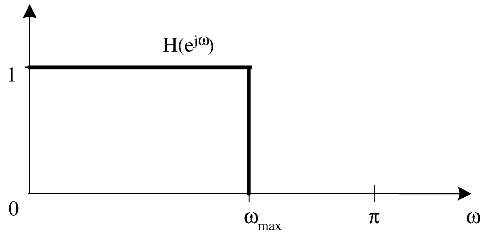

Fig. 3. Digital filter.

and rejects high frequencies. The gain of the ideal low-pass filter is equal to 1 for frequencies between 0 Hz and the cutoff frequency f_c Hz, and 0 for all higher frequencies ($\omega_{\max} = 2\pi f_c$) [12], as depicted in Fig. 3. The output spectrum is produced by multiplying the spectrum obtained from the PSD of $x(n)$ by the transfer function, and contains the information included in [0, PSD($\omega_{\max}$)].

Having ensured (1), it is possible to obtain $x_m(n)$ through m-fold decimation. Signals $x_k(n)$, $1 \leq k \leq m$, also preserve the statistical features of $x(n)$, using a lower degree of decimation. Decimation signals can be produced using a digital filter bank which splits $x(n)$ into m sequences $x_k(n)$, $1 \leq k \leq m$, typically called subband signals [9]. Independence between subband signals is enabled when a subset of them is selected, with various (as different as possible) degrees of decimation.

Summarizing, m data sets can be derived from the original signal using a technique comprising the following steps (Fig. 4).

1. Calculate the power spectral density of the data set.
2. Calculate the maximum frequency contribution $\omega_{\max}$ where $\sum_{i=0}^{\omega_{\max}}\mathrm{PSD}(\omega_{\mathrm{i}})(1-l) \geq \sum_{i=0}^{\pi}\mathrm{PSD}(\omega_{\mathrm{i}})$. Then, calculate m using condition (1)
3. Construct m sequences $x_k(n)$, $1 \leq k \leq m$, using m-fold decimation
4. Select m' sequences using as different as possible degrees of decimation. As sequences $x_1(n)$ and $x_m(n)$ are evidently selected (having decimation degrees equal to 1 and m, respectively), $m' - 2$ sequences

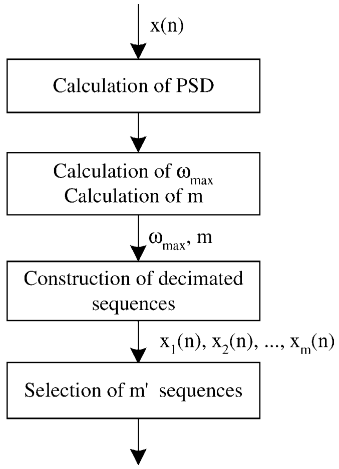
x(n)
Calculation of PSD
Calculation of ω_max
Calculation of m
ω_max, m
Construction of decimated sequences
x_1(n), x_2(n), ..., x_m(n)
Selection of m' sequences

Fig. 4. Deriving m' data sets from $x(n)$.

need to be selected from the remaining $m-2$ ones. This is optimally accomplished when the following sequences are ultimately selected:

$$x_1(n), x_{1+\mathrm{div}(m-1,m'-1)}(n), x_{1+2\,\mathrm{div}(m-1,m'-1)}(n), \ldots , x_{1+(m'-2)\mathrm{div}(m-1,m'-1)}(n), x_m(n)$$

where div(x, y) is the integer quotient of the division of the numerator x by the denominator y.

To give an example, for $m = 128$ and $m' = 10$, $\mathrm{div}(m-1, m'-1) = 14$ and, thus, sequences $x_1(n)$, $x_{15}(n)$, $x_{29}(n)$, $x_{43}(n)$, $x_{57}(n)$, $x_{71}(n)$, $x_{85}(n)$, $x_{99}(n)$, $x_{113}(n)$ and $x_{128}(n)$ are selected.

3.1. An example

An example system/model data comparison using the above technique is the following: the assembly process of a manufacturing system consumes time equal to d_i. A single data set is only available from the manufacturing system, while n data sets ($n > 1$) are available from experimentation with its respective simulation model.

Model validation objective is to reach reliable conclusions for the quality of the model. We use the average delay (*avgdelay*) for comparing the system and the corresponding model. A single monitoring variable MV_1 is thus required. We apply the proposed technique to enhance the effectiveness of the inspection-approach-based model validation.

Considering the original sequence $d(r) = d_1, d_2, \ldots , d_r$, we derive the power spectral density of this spectrum and then calculate the maximum frequency contribution $\omega_{\max}$ where the amount of information is greater than at least $(1-l)$% of the overall information of $d(r)$. The maximum m is calculated according to condition (1). System observation data $d(r)$ and $n = 10$ model data sets $y_1(r), y_2(r), \ldots , y_{10}(r)$ are depicted in Fig. 5 for $\mu = 10$ and $\sigma^2 = 2.0$. The time consumed (d_i) follows the normal distribution and the model is an accurate representation of the system.

Using a value of 0.001 for l parameter, the power spectral density of $d(r)$ and $\omega_{\max}$ are depicted in Fig. 6. In this case, m is calculated to be equal to 128. Then, m sequences $d_k(r)$ are constructed using

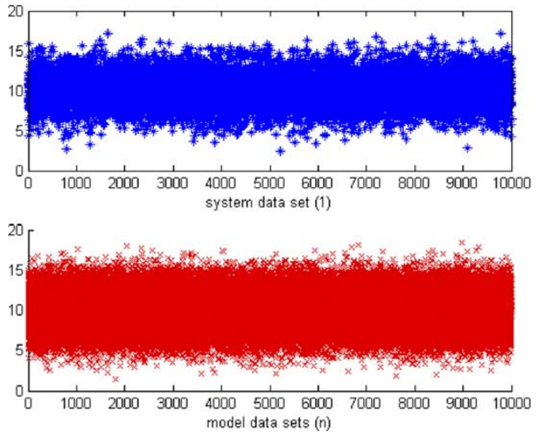

Fig. 5. System-model initial data sets.

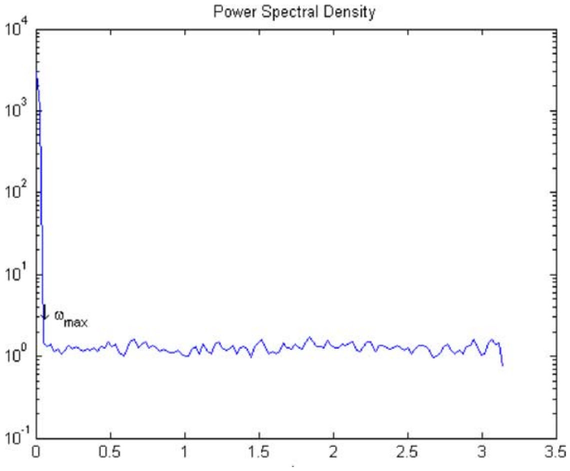

Fig. 6. Power spectral density.

m-fold decimation. Sequences $d_k(r)$ for various values of k are depicted in Fig. 7. Then, m' sequences are finally selected. As $m = 128$, sequences $x_1(n)$, $x_{15}(n)$, $x_{29}(n)$, $x_{43}(n)$, $x_{57}(n)$, $x_{71}(n)$, $x_{85}(n)$, $x_{99}(n)$, $x_{113}(n)$ and $x_{128}(n)$ are selected. Obtaining m' system observation data sets and n model data sets, we build a $100(1-a)\%$ confidence interval for MV_1r–MV_1s (a single monitoring variable is used). We first calculate $\overline{\mathrm{MV}_1 r}$, $\overline{\mathrm{MV}_1 s}$, $S^2(\mathrm{MV}_1 r)$, $S^2(\mathrm{MV}_1 s)$ and $\hat{\mathrm{f}}$. Then, we construct the confidence interval $[l(a), u(a)]$. The difference between MV_1r and MV_1s is statistically significant at level a if $0 \notin [l(a), u(a)]$.

4. Experimental evaluation

To evaluate the proposed technique, we have made numerous experiments with various data sets (both test and real-world). We also experimented with both exact and approximate model representations of the system, to determine if the model seems to be (a) valid, when actually is an accurate representation of

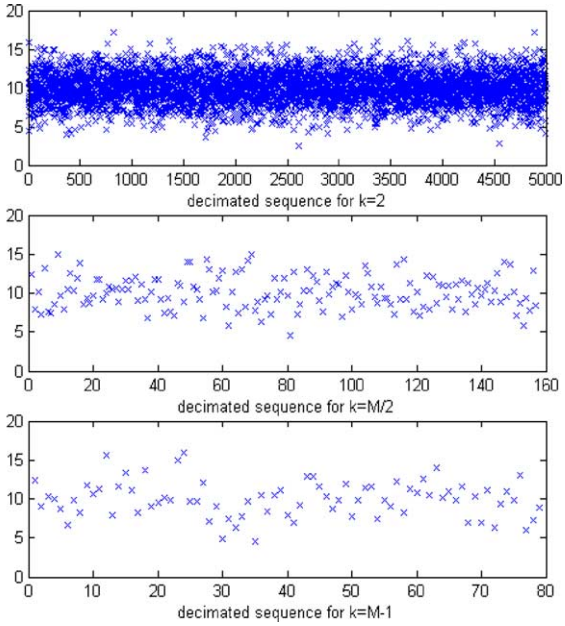

Fig. 7. Decimated system data sequences.

the system and (b) invalid, when actually is an approximate representation. For the case of hypothetical data, we have experimented with data sets from the following distributions: uniform, normal, poisson, lognormal and gamma. Results from system/model comparisons, for both accurate and approximate representations, are depicted in Table 1. All steps of this process were automated using Matlab.

In all cases presented in Table 1, we used models producing output following the same distribution with system data. Distribution parameters for both the system and the model are presented along with the corresponding statistical properties (mean, variance), which are determined by the specific distribution parameters. For instance, using parameters 6.0 and 10.0 for the gamma distribution, the theoretic mean is 6.0 and variance is 1.9. Values of n and m and the lower and upper endpoints [$l(a)$ and $u(a)$] of the confidence interval formed are also depicted. Calculation of m was performed for $l = 5 \times 10^{-2}$. In case $m > 5$, we set $m' = 5$ and respectively, selected only 5 system data sets. In all cases, we used five model data sets and built 90% confidence intervals.

Based on the above results, the technique was successfully employed in all test cases, indicating model validity (invalidity) when the model was an accurate (approximate) representation. This is indicated by success (s) in the last column of Table 1, when $0 \in [l(a), u(a)]$ for accurate representations and $0 \notin [l(a), u(a)]$ for approximate models. There was also a case the technique was not applied (i.e. decimation was not possible) (second row of Table 1). Such cases, however, should not be considered as failures, as no misleading results are obtained.

We also experimented with real-world data sets. In such a test case, we used traces of HTTP requests—a set from the University of Saskatchewan's www server is available at http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html. We simulated the responses issued by web servers (heading to client processes) and built a single data set using the HTTP request *reply_size* field. Data analysis indicated that *reply_size* follows the normal distribution with $m = 4805$ and $\sigma^2 = 18{,}653$. Applying m-fold decimation with

Table 1
Data sets and comparison results

Distribution type	Parameters		Statistical properties: mean/variance		Data sets		Confidential interval		Success/fail
	System	Model	System	Model	n	m	l(a)	u(a)	
Gamma	6/10	7/1	6/1.9	7/7	5	2	−1.75	−0.60	s (invalid)
Gamma	950/6	1000/11	950/387.84	1000/301.51	5	–	–	–	na
Gamma	1000/11	1000/11	1000/301.51	1000/301.51	5	128	−13.52	54.31	s (valid)
Lognormal	2/6	5/7	6/2	7/5	5	2	−2.61	−0.29	s (invalid)
Lognormal	350/1050	300/1000	1050/350	1000/300	5	4	−68.61	−15.74	s (invalid)
Lognormal	300/1000	300/1000	1000/300	1000/300	5	4	−34.29	66.28	s (valid)
Normal	6/2	7/5	6/2	7/5	5	2	−2.63	−0.50	s (invalid)
Normal	1050/350	1000/300	1050/350	1000/300	5	2	3.95	118.66	s (invalid)
Normal	1000/300	1000/300	1000/300	1000/300	5	2	−34.08	64.65	s (valid)
Poisson	12	13	12/3.46	13/3.61	5	2	−2.37	−0.76	s (invalid)
Poisson	13	13	13/3.61	13/3.61	5	2	−1.40	0.14	s (valid)
Poisson	960	950	960/30.98	950/30.82	5	128	9.60	16.94	s (invalid)
Poisson	950	950	950/30.82	950/30.82	5	128	−0.40	6.94	s (valid)
Uniformreal	3/9	−2/16	6/1.73	7/5.20	5	3	−1.37	−0.12	s (invalid)
Uniformreal	500/1400	400/1600	950/259.81	1000/346.41	5	4	−68.61	−15.74	s (invalid)
Uniformreal	400/1600	400/1600	1000/346.41	1000/346.41	5	4	−18.76	34.08	s (valid)

Table 2
Applying *m*-fold decimation for various data sets

Distribution type	Parameters	Statistical properties: mean/variance	M_1	M_2	Figure
Normal	$10^3/0.001$	$10^3/0$	128	16	8(b).1
Normal	$10^3/0.005$	$10^3/0.01$	128	16	8(b).2
Normal	$10^3/0.05$	$10^3/0.05$	128	14	8(b).3
Normal	$10^3/0.01$	$10^3/0.01$	128	16	8(b).4
Normal	$10^3/10^2$	$10^3/100$	128	na	8(b).5
Gamma	1/1	1/1	na	na	8(c).1
Gamma	1/10	1/0.32	2	na	8(c).2
Gamma	$1/10^2$	1/0.10	128	na	8(c).3
Gamma	$1/10^3$	1/0.03	128	na	8(c).4
Gamma	$1/10^4$	1/0.01	128	na	8(c).5
Poisson	100	100/10	128	na	8(d).1
Poisson	200	200/14.14	128	na	8(d).2
Poisson	300	300/17.32	128	na	8(d).3
Poisson	400	400/20	128	na	8(d).4
Poisson	500	500/22.36	128	na	8(d).5
Lognormal	$10^6/10^6$	$10^6/10^6$	na	na	8(e).1
Lognormal	$10^3/10^6$	$10^6/10^3$	128	na	8(e).2
Lognormal	$1/10^6$	$10^6/1$	128	16	8(e).3
Lognormal	$10^6/10^3$	$10^3/10^6$	na	na	8(e).4
Lognormal	$10^6/1$	$1/10^6$	na	na	8(e).5
Uniform	−10/10	0/5.77	128	na	8(f).1
Uniform	0/20	10/5.77	64	na	8(f).2
Uniform	0/40	20/11.55	3	na	8(f).3
Uniform	0/60	30/17.32	2	na	8(f).4

$l = 5 \times 10^{-2}$, m was calculated to be equal to 128, as depicted in Fig. 8(a). We selected $m' = 5$ system data sets and also derived 5 model data sets. We then compared the actual data set with two diverse model data sets, using either an accurate model ($m = 4805$ and $\sigma^3 = 18{,}653$) or an approximate model ($m = 4500$ and $\sigma^2 = 20{,}000$). For the accurate model, the confidence interval $[-204.93, 45.90]$ was built. As $0 \in [-204.93, 45.90]$, the model is considered to be valid, which is correct. For the approximate model, the confidence interval $[87.27, 356.12]$ was built. As $0 \notin [87.27, 356.12]$, the model is considered to be invalid, which is also correct.

The applicability of the technique introduced was also a matter of concern, as evidently, not all data sets can be subjected to *m*-fold decimation. To examine applicability, we have experimented with diverse data sets from the previously mentioned distributions: uniform, normal, poisson, lognormal and gamma. A subset of experiments is depicted in Table 2. Each data set contains 3,000 samples. We also examined if decimation is possible for two different values of $l{:}l_1 = 5 \times 10^{-2}$ and $l_2 = 10^{-8}$. Two values for m (m_1 and m_2) are, respectively, calculated. When decimation cannot be applied, na is used to denote it. Table 2 presents these results, including: the distribution type of system data, the parameters and statistical properties (mean, variance) determined by these parameters and the two respective values for m (m_1 and m_2) calculated for l_1 and l_2.

We used various values of l to explore the effect of this parameter, as evidently lower values reduce the applicability of *m*-fold decimation. The power spectral density and calculation of m for system data sets

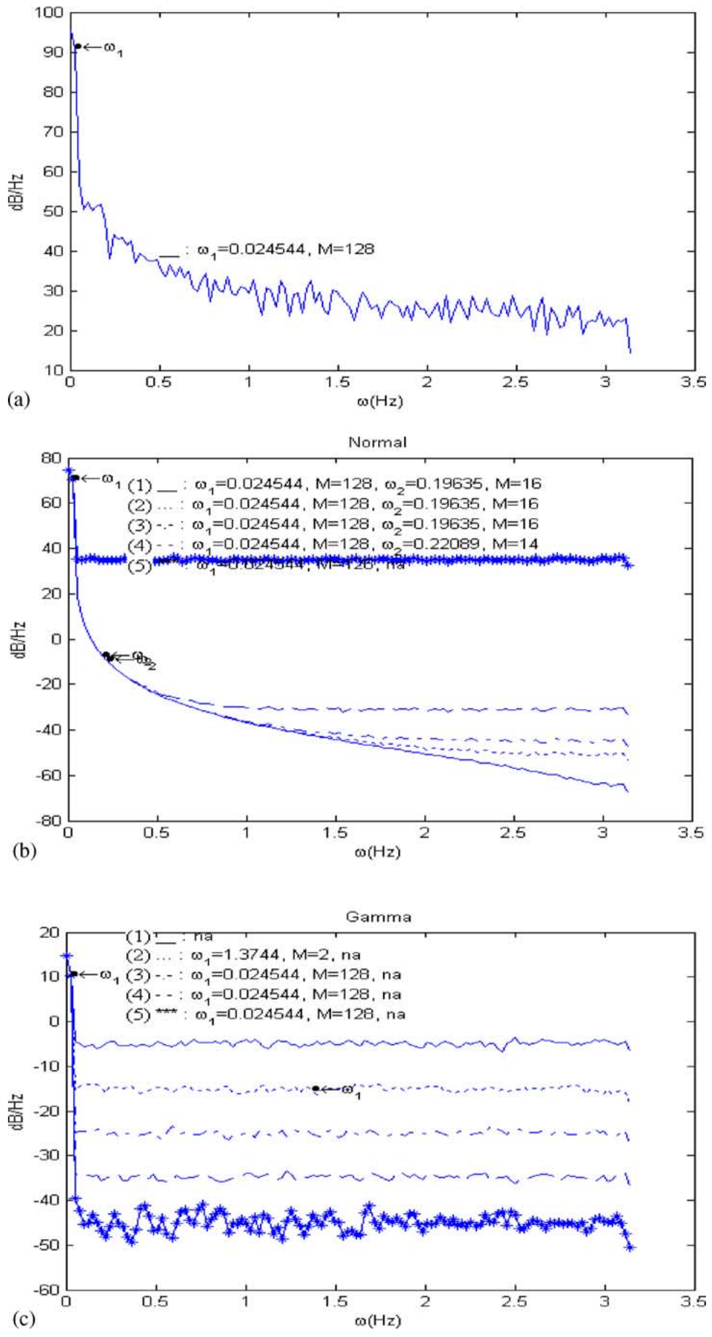

Fig. 8. Power spectral density: (a) real-world HTTP requests; (b) normal distribution data; (c) gamma distribution data; (d) poisson distribution data; (e) lognormal distribution data; (f) uniform distribution data.

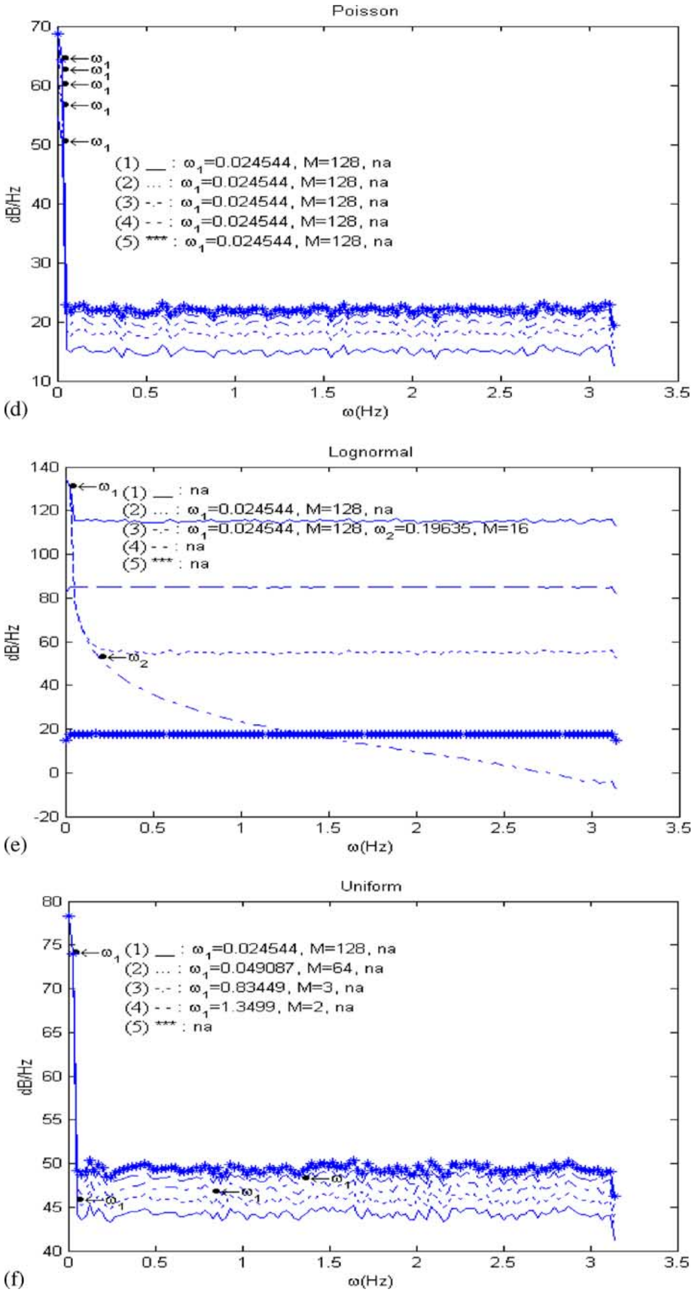

Fig. 8. (*Continued*).

of Table 2 are depicted in Fig. 8(b)–(f) (a reference to these results appears in the last column). The PSD and calculation of m for the real-world HTTP request data are depicted in Fig. 8(a). For $l = 5 \times 10^{-2}$, decimation was possible in nearly all cases. On the other hand, the technique could not be widely applied for very low values (close to 10^{-8}).

Overall evaluating the effectiveness and applicability of the technique introduced with both real-world and hypothetical data, the technique performed well, indicating that the model was valid—when accurately representing the system—and invalid otherwise. When approximate model representations were tested against the real system, confidence intervals clearly indicated this deviation, also providing an indication of the magnitude by which the model deviates from the system. Applying this technique, validation may be accomplished as an automated process. However, parameterization is critical, as low values of l reduce applicability (lower values for l resulted in the calculation of lower values for m). It is also evident that selection of a when constructing the confidence interval has a well-known impact on the reliability of results.

Comparing this technique with the inspection approach, in the latter approach there were numerous cases where an approximate representation seems to be valid due to the inherent randomness of simulation runs. This was almost eliminated in the experiments employing this technique.

As far as applicability is concerned, it depends on the specific data sets. Applicability is enhanced when (a) deviation in system data (s) is reduced; (b) l is increased; and (c) a high number of samples are available; in our experiments, we used at least 500 samples. However, not being able to apply decimation should not be considered as a failure, as no misleading results are obtained.

5. Conclusions

We introduced a new model validation technique using a single system data set, to perform system-model comparison considering not only the statistical mean of data sets, but also the variance of observations. In this way, validation is far more effective than simply examining the difference between statistical means, i.e. the currently used approach. The system data set is considered as a discrete signal and m-fold decimation is used, according to signal processing theory. Model validation is then realized as a confidence-interval-based comparison. Applications of this technique involve manufacturing and military system simulation due to the paucity of real-world data, and certainly RTS. The technique was effectively applied for both accurate and inaccurate model representations and validation was accomplished as an automated process. Appropriate parameterization is however critical, as may considerably affect the applicability achieved.

Acknowledgements

The authors wish to thank Prof. N. Kalouptsides for his initial comments on this work.

References

[1] O. Balci, R. Sargent, A methodology for cost-risk analysis in the statistical validation of simulation models, Commun. ACM 24 (11) (1981) 190–197.

[2] G.S. Fishman, P.J. Kiviat, The analysis of simulation generated time series, Manage. Sci. 13 (1967) 525–557.

[3] L. Birta, F. Ozmizrak, A knowledge-based approach for the validation of simulation models: the foundation, ACM Trans. Model. Comput. Simulation 6 (1) (1996) 76–98.

[4] T. Oren, B. Zeigler, Concepts for advanced simulation methodologies, Simulation 32 (3) (1979) 69–82.

[5] A.M. Law, W.D. Kelton, Simulation Modeling and Analysis, McGraw-Hill, New York, 2000.

[6] D. Anagnostopoulos, V. Dalakas, M. Nikolaidou, Guidelines and techniques for comparing system observations with multiple sets of simulation results, in: Proceedings of IEEE SMC'02, 2002.

[7] N. Runciman, N. Vagenas, T. Corkal, Simulation of haulage truck loading techniques in an underground mine using witness, Simulation 28 (1997) 291–299.

[8] B.L. Welch, The significance of the difference between two means when the population variances are unequal, Biometrica 25 (1938) 350–362.

[9] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, Englewood Cliffs, 1993.

[10] N. Kalouptsides, Signal Processing Systems: Theory and Design, Wiley, New York, 1997.

[11] P.D. Welch, The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms, IEEE Trans. Audio Electroacoust. AU-15 (1967) 70–73.

[12] J. Proakis, Digital Communications, McGraw-Hill, New York, 2001.