ELSEVIER

Computers in Biology and Medicine 37 (2007) 670–679

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

Error bounds for data-driven models of dynamical systems

Nicholas O. Oleng', Andrei Gribok, Jaques Reifman*

Bioinformatics Cell, U.S. Army Medical Research and Materiel Command, Frederick, MD 21702, USA

Received 9 September 2005; received in revised form 2 June 2006; accepted 6 June 2006

Abstract

This work provides a technique for estimating error bounds about the predictions of data-driven models of dynamical systems. The bootstrap technique is applied to predictions from a set of dynamical system models, rather than from the time-series data, to estimate the reliability (in the form of prediction intervals) for each prediction. The technique is illustrated using human core temperature data, modeled by a hybrid (autoregressive plus first principles) approach. The temperature prediction intervals obtained are in agreement with those from the Camp–Meidell inequality. Moreover, as expected, the prediction intervals increase with the prediction horizon, time-series data variability, and model inaccuracy. Published by Elsevier Ltd.

Keywords: Physiologic measurement predictions; Bootstrap; Error bounds; Confidence interval; Prediction interval; Time-series data; Dynamical systems

1. Introduction

In most dynamical systems, it is not possible to know the exact characteristics of the underlying system dynamics. Moreover, the system parameters as well as the noise characteristics may vary with time, presenting additional challenges in constructing models that can provide accurate and reliable predictions. Various data-driven techniques for system identification and prediction have been developed in systems theory to deal with these problems. Another modeling issue, which is particularly important in time-critical as well as safety-critical applications, is the lack of a mechanism to assess the reliability of model predictions. In many applications, it is generally not useful to have the estimated state of the dynamical system being modeled unless a measure of reliability of such predictions is also provided [1].

Consider the predictions of human-body core temperature, where the human body is viewed as a dynamical system, with the objective of preventing heat-related injuries, such as heat stroke [2–4]. Individuals subjected to the same workload and environmental conditions may, in some cases, yield very different physiological responses. Such variation in physiologic response is especially critical at limiting thresholds of physiologic health, such as extreme values of core temperature, where small variations can make the difference between a suitable recovery and an irreversible pathological condition [5]. It is, therefore, imperative that predictive models be customized to specific individuals in order to account for inter-individual variability, and that such models explicitly provide error bounds in the form of confidence and/or prediction intervals about their predictions. Yet, to the authors' knowledge, little, if any, effort has been devoted to this end. This paper addresses this problem, by providing a method for estimating error bounds, in the form of confidence and prediction intervals, around the predictions of the future outputs of data-driven nonautonomous dynamical system models. The method employs the bootstrap technique in estimating these intervals.

The bootstrap, first introduced by Efron [6], is a well-known and widely used technique that has been applied, with considerable success, to problems of independent and identically-distributed (IID) data. This technique is used to assign measures of accuracy, such as the standard deviation, to statistics like the mean and the median of a distribution. The cornerstone of the technique lies in the creation of multiple replicate (bootstrap) samples by sampling, with replacement, from the available data set, which is assumed to be representative of the (unknown) underlying population. Its appeal stems from two main facts: no parametric structure needs to be assumed for the distribution of

* Corresponding author. Tel.: +1 301 619 7915; fax: +1 301 619 1983.
E-mail address: jaques.reifman@us.army.mil (J. Reifman).

0010-4825/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.compbiomed.2006.06.005

the data; and it can provide measures of accuracy for statistics for which closed-form solutions are not readily available.

In addition to IID data, the bootstrap has been applied to both linear and nonlinear regression tasks. In [7,8], paired bootstrap samples were used to train neural networks for nonlinear regression tasks. However, the use of the bootstrap technique for time series resulting from the outputs of autonomous and nonautonomous dynamical systems has been somewhat limited. This is primarily because the time dependence structure of the data in such systems has to be preserved in any re-sampling procedure, making it difficult to obtain independent replicate samples of the time series. Recently, there have been increased efforts in applying the bootstrap technique to time-series data derived from autonomous systems [9,10]. Of the techniques employed, the block bootstrap (moving blocks) technique, described in [10], has emerged as the most dominant. However, little, if any, progress has been made in implementing the bootstrap for prediction of nonautonomous dynamical systems, i.e., systems with exogenous inputs, representative of a large number of real-world applications, such as those encountered in the control of industrial processes and biological systems.

The approach adopted in this paper is based on the idea that the set of all possible data-driven models of a given dynamical system corresponds to a set of parameters with a particular statistical distribution in some model-parameter space. Here, we restrict this set to include ARX (auto-regressive with exogenous input) models of order less than a given (known) bound. By extension, the predictions formed by this set would also form a related, albeit unknown, distribution at each time instance. Since virtually any model is a candidate for predicting the outputs of a dynamical system, we should only choose a sample of models that we assume to be sufficiently representative of the "true" distribution of the candidate models. Having chosen such a sample, it is then possible to obtain error bounds, in the form of confidence and prediction intervals, for the estimates of dynamical system outputs by applying the bootstrapping algorithm to this sample of models.

In this paper, as in the traditional bootstrap applied to IID data, no assumptions are made about the nature of the distribution of the models, except that the estimates provided by these models at each instance in time may not be IID. The models, whose parameters are here assumed to be time invariant, are formed from random blocks of varying data lengths to capture a wide array of locally time-invariant properties over the entire data range. However, due to the potential limitations on the size of the original data, the blocks may contain overlapping data segments, causing the derived models to be dependent. Such dependency violates the IID assumption, which is, in general, difficult to verify in practical applications and, if violated, causes the bootstrap-calculated variance to be underestimated [11,12].

The paper is organized as follows: In Section 2, we describe the bootstrap algorithm as it applies to the case of IID data and its extension to regression tasks. This section is concluded by discussing the application of the bootstrap to autonomous system time-series data. In Section 3, we extend the bootstrap to the case of outputs of nonautonomous dynamical systems, where we describe how to construct a sample of models and how to compute confidence and prediction intervals from these models. Section 4 provides an illustration of the described procedure as well as additional discussions on the method, while Section 5 provides the conclusions.

2. The bootstrap algorithm

The bootstrap algorithm [6,7] is a computer-based method for assigning measures of accuracy to statistics, such as a sample mean and median. This technique is particularly appealing because it avoids the limitations of having to make parametric assumptions about the distributions involved and can be used when the parameter of interest is a complicated function of the underlying distribution. The general idea of the bootstrap is to create multiple secondary (bootstrap) samples by re-sampling, with replacement, from the original sample. It is based on the assumption that the available sample of size n is a particular realization of some unknown probability distribution F and forms a discrete empirical distribution F_n, which is a good representative of F. Therefore, the relationship between the empirical distribution F_n and a secondary (bootstrap) sample drawn from it should be similar to the relationship between the "true" unknown distribution function F (illustrated in Fig. 1) and the original sample F_n of size n.

2.1. The bootstrap for IID data

Based on the above assumption, the bootstrap algorithm can be applied to a sample of IID data through the following steps:

1. The unknown population distribution F, from which a single sample is drawn, is approximated by a discrete empirical distribution F_n from n observations of a single sample, $\{z_1, \ldots, z_n\}$ (see Step 1 of Fig. 1).
2. B[1] random bootstrap samples, $\{z_1^*, \ldots, z_n^*\}$ of size n, are drawn, with replacement, from the empirical distribution. From each sample b, with $b \in \{1, \ldots, B\}$, a bootstrap estimate p_b^* of the statistic is made (Step 2 of Fig. 1).
3. The bootstrap estimates p_b^* are then used to generate the bootstrap sampling distribution H_B of the statistic (Step 3 of Fig. 1).
4. With the unknown sampling distribution of the statistic estimated by the bootstrap sampling distribution H_B, we can make quantitative statements about the accuracy of the statistic (Step 4 of Fig. 1). For example, the bootstrap estimate of the standard deviation[2] σ_B of the statistic p is computed as

$$\sigma_B = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} (p_b^* - \bar{p}^*)^2 \right\}^{1/2}, \tag{1}$$

[1] In practice, because the bootstrap does not make any assumptions about the nature of the underlying distribution, B is often significantly larger than the number of samples required by standard parametric techniques for estimating confidence intervals.

[2] In the literature, the term "standard error" is often used interchangeably with standard deviation.

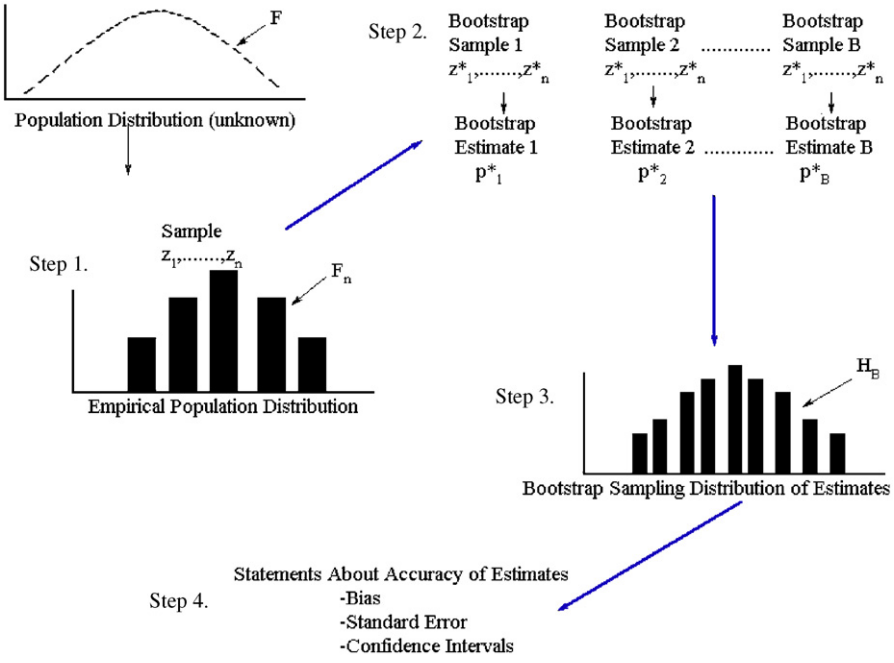

Fig. 1. The bootstrap method for estimating the accuracy of a statistic from a single IID sample. The pictorial representation follows the description given in Section 2.1.

where

$$\bar{p}^* = \frac{1}{B}\sum_{b=1}^{B} p_b^*. \tag{2}$$

Following the computations in Eqs. (1) and (2), the $(1-\alpha)\%$ confidence interval for the statistic p can be computed as

$$\bar{p}^* - c_{\text{conf}}\sigma_B \leqslant p \leqslant \bar{p}^* + c_{\text{conf}}\sigma_B, \tag{3}$$

where the factor c_{conf} can be obtained from the Student's t-distribution tables [13].

The bootstrap is especially useful for estimating statistics for which there are no apparent analytical expressions. Additionally, for statistics, for which there are analytical equations for computing measures of accuracy, bootstrap estimates have been shown to asymptotically converge to those values [13].

2.2. The bootstrap for regression tasks

The results described in Section 2.1 can be extended to regression tasks if we assume that the data are independent. For regression, we assume that the data are available in input–output pairs, for which a predictive model, such as a least-squares-linear or nonlinear fit, can be derived. The use of the bootstrap for nonlinear regression tasks is dealt with in [7,8], where neural networks are used as predictive models to capture the nonlinear fit. In the following, we summarize the method described in [7], adopting a similar nomenclature in this and the following sections. We assume that we are given a set of n input–output data pairs, $\{x, t\}$, generated according to equation

$$t(x) = f(x) + \xi(x), \tag{4}$$

where $t(x)$ is the target output, $\xi(x)$ denotes noise with zero mean and $f(x)$ represents the mean of the target-distribution, which is taken as the "true" regression, given the input x. From this original sample, we construct B bootstrap samples, where each sample is obtained by drawing input–output data pairs, with replacement. For each sample b, a single neural network is trained to obtain an output $o_b(x)$ as an estimate of the target output $t(x)$. As an estimate for the regression $f(x)$ the average

$$m(x) = \frac{1}{B}\sum_{b=1}^{B} o_b(x) \tag{5}$$

is computed from the ensemble of model output estimates.

Two fundamental measures for quantifying the statistical accuracy of a predictive model are confidence intervals and prediction intervals. Confidence intervals provide a measure of the uncertainty between the prediction and the expected (mean) value of the outcome. This, in turn, provides a way to quantify our confidence in our estimate $m(x)$ of the "true" regression $f(x)$, i.e., a characterization of the probability distribution $P(f(x)|m(x))$. The variance of this distribution can be estimated as

$$\sigma^2(x) = \frac{1}{B-1}\sum_{b=1}^{B}[o_b(x) - m(x)]^2. \tag{6}$$

Prediction intervals, on the other hand, indicate the expected error between the model prediction and the measured value of an individual outcome, i.e., they are based on estimates of the probability distribution $P(t(x)|m(x))$. Since they account for the data spread of individual outcomes, prediction intervals are

necessarily wider than confidence intervals. Considering the equation

$$t(x) - m(x) = [f(x) - m(x)] + \xi(x), \tag{7}$$

we observe that the confidence intervals are associated with the variance of the first term on the right-hand side of this equation, while prediction intervals are associated with the variance of the left-hand term of the equation.

In computing the confidence intervals, it is assumed that the ensemble of model estimates yields a more or less unbiased estimate for $f(x)$, i.e., the distribution $P(f(x)|m(x))$ is centered about $m(x)$. Since there is no way of knowing $f(x)$ for sure (the "true" distribution is unavailable), the variance of this distribution is estimated from the empirical distribution $P(o_b(x)|m(x))$ resulting in the estimate in Eq. (6). The confidence intervals can then be computed as

$$[m(x) - c_{\text{conf}}\sigma(x)] \leqslant f(x) \leqslant [m(x) + c_{\text{conf}}\sigma(x)], \tag{8}$$

where c_{conf} depends on some desired confidence level $1 - \alpha$. The factor c_{conf} can be chosen, as with the IID data, from the Student's t-distribution. Alternatively, the value of c_{conf} can be empirically obtained by computing c_{conf} so that $|o_b(x) - m(x)| \geqslant c_{\text{conf}}\sigma(x)$ for no more than $100\alpha\%$ of all predictions.

Since the two components, $f(x) - m(x)$ and $\xi(x)$, of the right-hand side of Eq. (7) are assumed to be independent, one can calculate the total variance in their sum as the sum of their variances. Thus, computing the variance of Eq. (7) results in

$$\begin{aligned} s^2(x) &\equiv \langle [t(x) - m(x)]^2 \rangle \\ &= \langle [f(x) - m(x)]^2 \rangle + \langle \xi^2(x) \rangle = \sigma^2(x) + \chi^2(x). \end{aligned} \tag{9}$$

The variance of the first component is the estimate for the confidence interval, already computed as $\sigma^2(x)$ in Eq. (6), and it remains to estimate the variance $\chi^2(x)$ of the second component.

Since, in general, $\chi^2(x)$ is a nonlinear function of the measurement x, it can be estimated as the output of a feed-forward neural network, for a given input x. The proposed neural network should have an exponential, instead of a linear, transfer function for the output layer to ensure that the estimate of $\chi^2(x)$ is positive. For the hidden layer, any nonlinear function, such as a hyperbolic tangent function, can be used.

The value of $\chi^2(x)$ is not known a priori. Therefore, in training the neural network, one cannot simply minimize the sum of the squared errors between the measured outputs and the predictions, as is commonly the case for feed-forward neural networks, given that the errors in estimating $\chi^2(x)$ cannot be computed directly. Hence, we need to indirectly train the network to predict $\chi^2(x)$.

Consider $\chi^2(x)$ to be the variance of the residual $r(x)$, whose square,

$$r^2(x) \equiv \max([t(x) - m(x)]^2 - \sigma^2(x), 0), \tag{10}$$

inferred from Eq. (9), is assumed to have a zero-mean Gaussian distribution.

By assuming that $\chi^2(x) = Var[r^2(x)]$, we can employ a maximum likelihood criterion to train a neural network to find, out of all possible functions, the one which minimizes the likelihood function given by the equation

$$L \equiv -\sum_i \log\left[\frac{1}{\sqrt{2\pi\chi^2(x_i)}} \exp\left(-\frac{r^2(x_i)}{2\chi^2(x_i)}\right)\right]. \tag{11}$$

Therefore, the neural network is trained with L in Eq. (11) as its cost function, instead of a sum of squared errors.

Having obtained an estimate for $\chi^2(x)$, and hence, an estimate for $s^2(x)$, the prediction interval of the output $t(x)$ can be computed as

$$[m(x) - c_{\text{pred}}s(x)] \leqslant t(x) \leqslant [m(x) + c_{\text{pred}}s(x)], \tag{12}$$

where the factor c_{pred} is chosen similar to c_{conf} above.

2.3. The bootstrap for autonomous dynamical systems

Much of the work conducted using the bootstrap algorithm has been focused on IID data, for which the method has proven to be quite effective. The extension to regression tasks, even nonlinear ones, can also be handled as seen in Section 2.2. However, the bootstrap has been much less effective when applied to time-series data, where sampling must be carried out in a way that suitably captures the dependent structure of time-ordered data streams [9,10]. A few efforts have been made to address this problem, with the "moving blocks" bootstrap technique emerging as the most common approach for applying the bootstrap to time-series data [10].[3] In using the moving blocks bootstrap for autonomous systems resulting in time-series data, the series of observations is divided into q blocks of l sequential observations. The blocks may be overlapping or nonoverlapping. Each bootstrap sample is constructed by randomly sampling q blocks, with replacement, and concatenating these into a series of $q \times l$ observations.[4] From these bootstrap samples, basic statistics can then be calculated. Most treatments of the moving blocks method assume that the time-series data are a regression of past outputs of the form

$$\begin{aligned} y(t) = &\sum_{k>0} a_k y(t-k) \\ &+ \sum_{k>0} d_k w(t-k), \quad k = 1, 2, \ldots, \end{aligned} \tag{13}$$

where a_k and d_k are constant coefficients, $y(t)$ denotes the output of the time series, $w(t)$ denotes an IID noise signal, and k denotes an integer index representing discrete time instances.

3. The bootstrap for nonautonomous dynamical systems

The output $y(t)$ at time t of a linear discrete-time dynamical system driven by some exogenous input[5] $u(t)$ can be

[3] An alternative to the moving blocks bootstrap technique is the *residuals bootstrap* method [10].

[4] The *stationary bootstrap* is a variant of this, in which random block lengths are used.

[5] This input is assumed to be deterministic, even though in practice it includes random measurement errors.

defined as

$$y(t) = \sum_{k>0} a_k y(t-k) + \sum_{k>0} d_k u(t-k) + \sum_{k>0} e_k w(t-k), \quad k = 1, 2, \ldots, \tag{14}$$

where $w(t)$ and k are defined as in Eq. (13), and a_k, d_k, and e_k are constant coefficients. Consider the problem of predicting, at time t, the output $y(t + H)$ at time $t + H$, where H is the prediction horizon. Given the current and past outputs as well as the past and future inputs of the system, the task is to compute the best estimate of the output of the system H steps ahead. This is often done by simply iterating a one-step-ahead predictor H times. It is evident that by concatenating blocks that are not necessarily adjacent in the original data, as with the moving blocks bootstrap, the resulting time series will inaccurately capture the dependency of the data, particularly if the exogenous input $u(t)$ varies significantly.

To extend the bootstrap technique to the prediction of nonautonomous dynamical systems, while avoiding the pitfalls of the moving blocks approach and retaining the dependent structure of the time-ordered data stream, we note that a set of candidate models for a given dynamical system forms a distribution in the model-parameter space. More precisely, the set of model-parameters forms a distribution in the parameter space. One way of obtaining a sample from such a distribution is by deriving models from sufficiently long individual blocks of contiguous time-series data, where the blocks are chosen randomly from the original data. Each of these models can then be used to provide predictions for the entire data range. By sampling with replacement from these models, rather than from the data, and forming bootstrap samples of their estimates at each time instance, we can obtain an estimate for the output of the dynamical system and its corresponding statistics. Each model assumes that the system can be modeled according to Eq. (14), or in compact form as

$$y(t) = \theta_1^{\mathrm{T}}(t)\varphi(t) + w_1(t), \tag{15}$$

where $y(t)$ is the output of the system, $w_1(t)$ is an IID noise signal with zero mean, $\varphi(t)$ is the regression vector of past inputs and outputs, and $\theta_1(t)$ is the parameter vector with an unknown distribution. We denote the mean of this distribution as the parameter vector $\theta_1^{\#}(t)$. Associated with this parameter vector, one may also define an output $y^{\#}(t)$, as

$$y^{\#}(t) = \theta_1^{\#\mathrm{T}}(t)\varphi(t). \tag{16}$$

We refer to $\theta_1^{\#}(t)$ and $y^{\#}(t)$ as the "true" parameter vector and the "true" output, respectively. In addition, the order of the system is unknown but it is assumed to have a known bound.

If we consider the value of the output H-time instances ahead of the present time t, i.e., $y(t + H)$, we note that Eq. (15) may be written as $y(t + H) = \theta_2^{\mathrm{T}}(t)\gamma(t) + w_2(t)$, except that the regression vector $\varphi(t)$ is replaced by $\gamma(t)$, which not only consists of past inputs and outputs as does $\varphi(t)$, but also of future inputs up to time $t + H$. Thus, as in Eq. (16), one could also define a "true" future output $y^{\#}(t+H) = \theta_2^{\#\mathrm{T}}(t)\gamma(t)$, where we assume that there already exists a finite past history of inputs and outputs as well as a profile of future inputs, of length H, for the prediction of future outputs at time $t + H$.

Next, we describe how to obtain a sample that would enable estimation of an empirical distribution of models representative of the "true" distribution. Then, in Section 3.2, we show how the predictions from these models can be used to compute both confidence and prediction intervals.

3.1. Computation of the models and their estimates

The set of candidate models for estimating the output of a given dynamical system can be argued to form a distribution in the model-parameter space. More tangibly, these models yield, at each instance in time, a distribution of the predictions of the output of the dynamical system. Here, we exploit this notion by deriving a number of models that constitute a random sample from the true distribution of models in the model-parameter space. The method adopted for deriving these models is aimed at utilizing prior knowledge about the system to obtain a sufficiently representative sample of the true distribution of models. It should be pointed out, however, that a user may implement a different strategy for obtaining the models from the one suggested here. The bootstrapping algorithm can then be applied, at each time instance, to the estimates provided by these models as described in the following steps:

1. n different models are constructed from the training data of the given dynamical system. To obtain the rth ($r = 1, 2, \ldots, n$) model, an arbitrary data interval of length N_r, starting from an arbitrary time t_r, is selected. The interval length N_r belongs to a discrete uniform distribution $[L_{\min}, T]$, where T is the length of the entire time series and $L_{\min}$ is the minimum data length necessary to derive a model [14]. Likewise, the starting time t_r is chosen from a discrete uniform distribution $[0, T - N_r]$. The uniform distribution was selected because, based on the Principle of Maximum Entropy, it is the least biased assignment among discrete distributions. An estimate of parameters θ for a model developed with time series of size N_r is obtained by minimizing a mean square error performance measure [15], to find the best fit for the data, resulting in a model M_r.[6] Corresponding to each model M_r, we can obtain a time series of estimates $o_r^*(t)$ for the system. This step corresponds to Step 1 of Fig. 1, where, instead of a data sample, we have a "model sample." Associated with these n models is, thus, an empirical distribution of model-parameter values.
2. At each time instance t, B bootstrap replicate samples of the model estimates are obtained as follows: for each bootstrap replicate sample, we re-sample n times, with replacement, from the model set, thereby producing a replicate sample

[6] Since we assume that the order of the system is unknown, the regression vectors $\varphi(t)$ in Eq. (16), and, therefore, parameter vectors, may have different lengths/orders, as determined by the Akaike Information Criterion or Minimum Description Length Criterion [14].

of n estimates $o_r^*(t)$. From these, we obtain a bootstrap estimate $y_b^*(t)$, $b=1, 2, \ldots, B$, of the measured output $y(t)$, given by the mean of the estimates $o_r^*(t)$, according to the equation

$$y_b^*(t) = \frac{1}{n}\sum_{r=1}^{n} o_r^*(t). \tag{17}$$

This corresponds to Step 2 in Fig. 1, where $y_b^*(t)$ takes the place of p_b^*. In the case of the prediction, we obtain the estimate as $y_b^*(t+H)$, where H is the desired prediction horizon. Since the prediction is assumed to take place online, the prediction $y_b^*(t+H)$ is computed by iterating a one-step-ahead predictor $y_b^*(t+1)$. The unavailable future outputs $y_b(t_m)$, where $t \leqslant t_m \leqslant t+H$, are replaced in the regression vector by the predictions $y_b^*(t_m)$, as in [3,4]. For simplicity, the explanations below concentrate on the estimation of $y_b^*(t)$, although they are equally applicable to $y_b^*(t+H)$, unless otherwise noted.

3. At each time instance, the means $y_b^*(t)$ from each bootstrap sample, obtained in Step 2, form an empirical distribution, corresponding to Step 3 in Fig. 1, whose mean value is calculated as

$$m(t) = \frac{1}{B}\sum_{b=1}^{B} y_b^*(t), \tag{18}$$

as in Eq. (5). The variance of this distribution can be estimated, similar to Eq. (6), as

$$\sigma^2(t) = \frac{1}{B-1}\sum_{b=1}^{B} [y_b^*(t) - m(t)]^2. \tag{19}$$

3.2. Computation of confidence and prediction intervals

Having obtained the mean and variance of the model predictions at time t in Eqs. (18) and (19), respectively, one can now compute the confidence and prediction intervals for the predictions of the dynamical system outputs at each time instance using an approach quite similar to that described in Section 2.2. This corresponds to Step 4 of Fig. 1. The regression vector of $\varphi(t)$ in Eq. (15) corresponds to the input x in Eq. (4). In the case of prediction at time $t+H$, we augment $\varphi(t)$ to include the next H inputs to the system resulting in the vector $\gamma(t)$.

The output $y(t)$ in Eq. (15) and the "true output" $y^{\#}(t)$ in Eq. (16) are equivalent to $t(x)$ and $f(x)$, respectively, in Eq. (4). Given these parallels, the variances $\sigma(t)$ and $s(t)$ associated with the prediction error at each time instance can be computed following the procedure described in Section 2.2. The confidence and prediction intervals for the output predictions can, therefore, be computed as in Eqs. (8) and (12), respectively. As with the regression models, the factors c_{conf} and c_{pred} are chosen so that $|y_b^*(t) - m(t)| \geqslant c_{\text{conf}}\sigma(t)$ and $|y(t) - m(t)| \geqslant c_{\text{pred}}s(t)$ for no more than $100\alpha\%$ of all predictions. Alternatively, they can be chosen from bounds provided by an inequality, such as the Camp–Meidell inequality, which only requires that the distribution be uni-modal, or the Chebychev inequality, which applies to any distribution. These, in general, lead to more conservative (larger) values for the confidence and prediction factors c than would be computed from the Student's t-distribution. The use of confidence factors from the Student's t-distribution would require an assumption that the models be independent of each other—and therefore, their predictions at each time instance—be normally distributed, one that may be too strict.

While we discuss both confidence and prediction intervals, we point out that prediction intervals are more relevant for our application to prediction of dynamical system outputs because we are only interested in estimating prediction uncertainty about a specific observation. Thus, in the illustration that follows, we only present prediction intervals.

4. Illustration and discussions

In this section, we explore the applicability of the technique illustrated in Section 3 by employing it to data obtained from a laboratory study on core temperature of human subjects [16]. The study entailed nine volunteer subjects in a treadmill walking experiment in two environmental conditions: (i) CONTROL (20 °C/68 °F temperature and 50% relative humidity); and (ii) HUMID (27 °C/81 °F temperature and 75% relative humidity). The wind speed was 1.1 m/s (2.5 mph) for both environments. On the morning of test days, the subjects, dressed in air permeable battle dress uniform, were instrumented for the collection of various physiological measurements, including core (rectal) temperature. Then they sat on a chair for 10 min just before starting to walk at 3 mph on level treadmills. The walking paused after every 30 min for 10 min of sitting. There were four 30-min walking periods per test, so that the entire experiment lasted a total of 170 min, including 10-min rest periods at each end. The activity profile (excluding the resting periods at each end) is shown at the bottom of Fig. 2. At the end of each 10-minute pause, the subjects were given 150 ml of water before walking again. Rectal temperature (assumed to be representative of the core temperature) was collected continuously and recorded every minute. Fig. 2 also illustrates the measured core temperature of one individual (marked with the "●" symbol) which rises, after a delay, with increase in activity and falls with the onset of rest.

The simulations conducted for this illustration are aimed at constructing models that capture the individual variability amongst the subjects. To this end, separate models are constructed for each subject, taking into account the individual traits of that subject, and employing model feedback to predict future values of temperature. In this paper, however, we only present the results of the simulation for one subject (the subject with median-best results) under the CONTROL condition.

Hybrid models, consisting of a first-principles model [2] in parallel with a data-driven ARX (auto-regressive with exogenous input) model [4], are employed for each subject. Because the first-principles model is a fixed model, the proposed technique applies only to the data-driven, ARX part of the hybrid model, which is designed to estimate the offset between the actual temperatures and the predictions provided by the

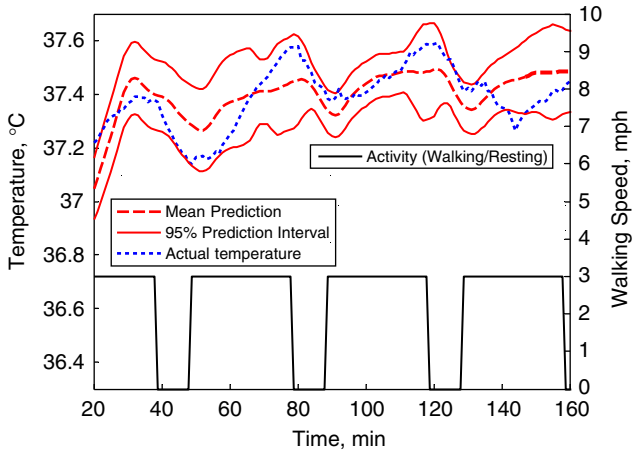

Fig. 2. 10-min-ahead prediction of core temperature (CONTROL condition) with 95% prediction intervals.

first-principles model. Predictions, at time t, for the temperature at time $t + H$ are carried out by iterating the one-step-ahead predictor for the ARX part of the model H times and adding the resulting predictions to those provided by the first-principles model at time $t + H$.

The data-driven part of the model predicts a single output time series as a function of seven corresponding input time series. At each time instance (each minute in this case), seven inputs—age, height, weight, body fat percentage, mean radiant temperature, relative humidity and walking speed—are used to predict the core temperature following the procedures described in Sections 3.1 and 3.2.

In constructing the ARX models for a specific subject, random blocks of that subject's data from each of the two environmental conditions were used as training data. The resulting models were used in conjunction with the first-principles model to predict the temperature for the entire experiment under both environmental conditions. Although the entire data set was used to develop the models and compute $\sigma^2(x)$ in Eq. (6), $\sigma^2(x)$ is not made arbitrary small because each model prediction $o_b(x)$ is based only on part of the data, the corresponding moving block. In this example, for the sake of simplicity, all ARX models are chosen to have the same order. The regression vector $\varphi(t)$ consists of inputs and outputs for the past four time intervals. Hence, the regression vector $\varphi(t)$ consists of 32 components (4 past values for each of the 7 inputs + 4 past outputs). Two hundred ($n = 200$) ARX models are constructed from the data, and the bootstrap technique (with $B = 1000$ bootstrap replicates) described in the previous section is employed to provide H-min-ahead predictions.

For fitting the residuals r^2 in Eq. (10), we employ a feed-forward neural network with a single hidden layer, where the input vector to the network consists of the components of the augmented regression vector $\gamma(t)$ for a total of 102 components (4 past values for each of the 7 inputs + 4 past outputs + 10 projected (into the future) values of each of the 7 inputs) for a 10-min-ahead prediction. To reduce the dimensionality of this vector, we employ the principal components analysis (PCA) algorithm [17] and use, as inputs to the network, the principal components that account for 96% of the variance in $\gamma(t)$. This reduces the dimensionality of the network inputs from 102 to 15 components, which significantly reduces the complexity of the network training.

Fig. 2 shows the 10-min-ahead predictions of core temperature and the corresponding 95% ($\alpha = 0.05$) prediction intervals, at each time instance, for a single subject, during the CONTROL conditions. Predictions for the subject are given by the dashed line and the solid thin lines illustrate the corresponding upper and lower prediction interval levels of the prediction. Note that the prediction at time t is actually computed at time $t - 10$ using only the information available at that time instance. Consequently, to avoid artifacts that result from the lack of information prior to the start of the treadmill activity at $t = 10$ min, the simulation shows the prediction starting from time $t = 20$. As can be seen, the method provides reasonable predictions and prediction intervals.

In the following, we describe two approaches that we use to semi-quantitatively validate the prediction intervals provided by our method. As discussed in the previous section, the confidence/prediction intervals are determined by two components: the two measures of standard deviation $\sigma(t)$ and $s(t)$ at each time instance t, and the corresponding confidence factor c_{conf} and prediction factor c_{pred}. While the measures of standard deviation are inherent in the nature of the data sample (the model set in this case), the confidence and prediction factors c can be chosen in a variety of ways, depending on the assumptions made about the underlying distribution of the sample.

4.1. Validation of the computed confidence and prediction factors

There are, in practice, a number of ways to compute the factors c_{conf} and c_{pred}. The most straightforward one is to conduct a numerical count of the actual existing data, i.e., obtain c (c_{conf} and c_{pred}) directly from the empirical distribution of the predictions so that no more than $100\alpha\%$ of all the predictions fall out of the desired region around the actual output of the system. Alternatively, the factors can be obtained from established parametric methods in statistical theory. The conservativeness of any of these methods is naturally dictated by the assumptions made about the underlying distribution of the data for which the confidence and prediction intervals are computed. For example, the Chebychev inequality [18], which makes no assumptions about the data distribution, yields the most conservative method with an upper bound of $c = 4.47$ for a 95% confidence, i.e., $\alpha = 0.05$. On the other extreme, the Student's t-distribution, which assumes normality of the data distribution, provides a fairly liberal factor of $c = 1.96$ for the same 95% confidence. The Camp–Meidell inequality relaxes the normality assumption only requiring that the resulting distribution be uni-modal, yielding an upper bound for the confidence factor of $c = 2.98$.

For the illustration in this paper, we checked the distribution of estimates $y_b^*(t)$ for normality, computed the confidence and prediction factors by numerical count, and proceeded to compare the values obtained with those provided by the three

parametric methods discussed above. In so doing, the count resulted in a confidence factor ranging from $c = 2.01$ to 2.50, across all nine subjects, slightly less than that provided by the Camp–Meidell inequality ($c = 2.98$) and in line with a uni-modal assumption about the distribution of the model estimates. Concurrent numerical tests conducted on the empirical distributions of the model estimates at each time instance also indicated that they were uni-modal, but not normal. This provides one way to validate our approach and the opportunity to select a confidence/prediction factor c in the range between 2.50 and 2.98. In the figures shown in this paper, we choose a factor of $c = 2.74$.

4.2. Relating the standard deviation of model predictions to input/output variability

Another way to validate our approach is to check for correlation between the variability of the data and the model uncertainty. We expect the models to be more certain, i.e., yield smaller standard deviation, when the data being predicted and that being used in prediction is less variable. Accordingly, the standard deviation (or uncertainty) of the model predictions can be explained from two different perspectives. The first is to consider how the variability of the measured variable being predicted (the *target output* of the dynamical system)[7] affects the standard deviation of the model predictions. We expect a larger standard deviation of the model prediction in regions where the variable being predicted varies the most and vice versa. The second approach is to consider how the variability of the *inputs* to the models[8] affects the standard deviation of the model predictions. Again, we expect the model predictions to be more variable in regions where the inputs to the model are more variable.

There are two main obstacles to evaluating the variability of the data in this example. The first is encountered in evaluating the effect of the model inputs on the standard deviation of the model predictions. Given that the augmented regression vector consists of 102 elements, at each time instance, while the standard deviation is a scalar, it is difficult to compare the variations in the model inputs with the variations in the standard deviation of the model predictions, since we do not have enough data to construct a 102-dimensional probability density function. Thus, a transformation of the regression vector to a scalar quantity, which can then be compared to the standard deviation, is desired. To accomplish this, we take the first 15 principal components of the regression vector that account for 96% of the input variance,[9] and assume a weighted average[10] of these to be representative of the entire regression vector.

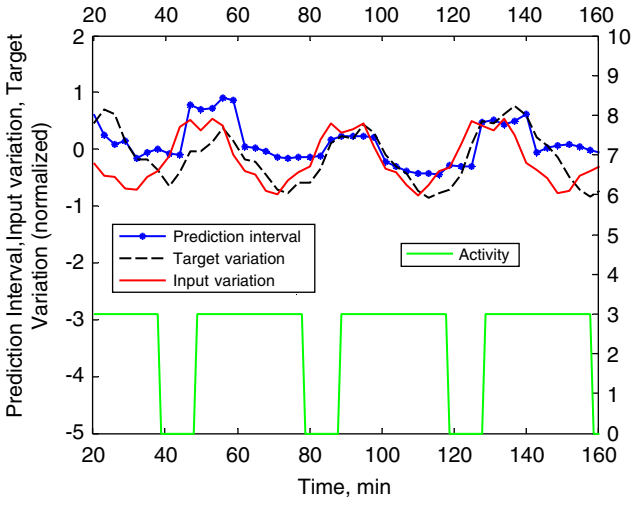

Fig. 3. Variability of the prediction interval in relation to the regression vector (model input) variability and the target (model output) variability. To eliminate confounding effects due to time delay in the predictions, the illustration represents results obtained with a 1-min-ahead prediction.

Only then, do we have a scalar time series that can be compared to the resulting standard deviation of the model predictions.

The second obstacle has to do with the time dependence of both regression vector and target variables. In [7], the variability in the data could be captured by estimating the density function of the input data. This could easily be done since the data were assumed to be IID. However, the time dependence in this case implies that such a simple density function can no longer be constructed. As a result, we use an approximation of the variability, computed as the second-order time difference of the data at each instance.

Fig. 3 compares the variability in the standard deviation of the model predictions reflected in the prediction intervals with the variability in both the regression vector (model input) and the target (model output). The values shown in the graph represent a mean value over a three-minute window for all three variables. This, results in some smoothing which, nevertheless, helps us discern the trend better, at least semi-quantitatively. To better illustrate the relationship between the variables, without the confounding effects of delay in prediction, the figure shows the relationship of the input and target variability to the prediction interval when the prediction horizon is only 1 min. The prediction interval is illustrated by the symbol (●), while the variation in the target output is given by the dashed line, and the variation in the regression vector is given by the solid line. In the figure, to provide a better visual comparison, the three variables have been normalized to have a zero mean. Note the significant impact of the activity level of the subject on these variables. It appears that the transition in activity, from walking to resting, results in an increase in uncertainty of the model predictions, which seems to be well correlated with the variations in the model inputs and the target outputs. Observe the 10-minute window after the time of transition from one activity to another. This uncertainty tends to decline as activity moves beyond the transition phase. In addition to Fig. 3, the

[7] In this case, this is the offset of the first-principles model prediction from the actual temperature. Hence, this illustration primarily evaluates the residuals of the first-principles model.

[8] We consider the entire extended regression vector $\gamma(t)$ as the "input".

[9] These same components are used to train the neural network that estimates the prediction intervals.

[10] The weighting is chosen to be proportional to the eigenvalue associated with each principal component.

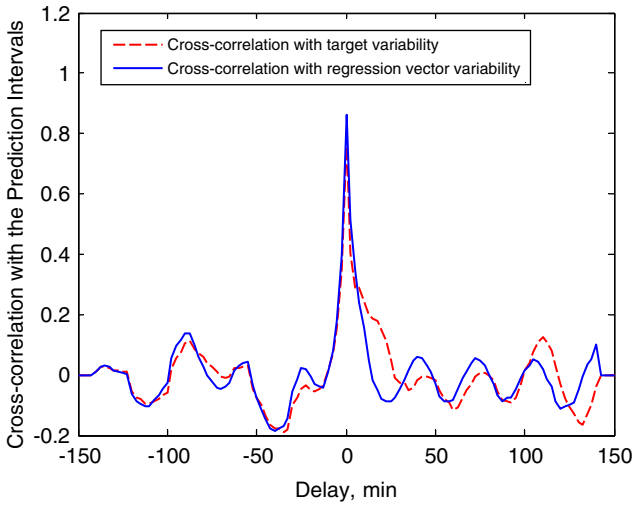

Fig. 4. Cross-correlation of variability measures (input/regression vector and target) with the prediction intervals.

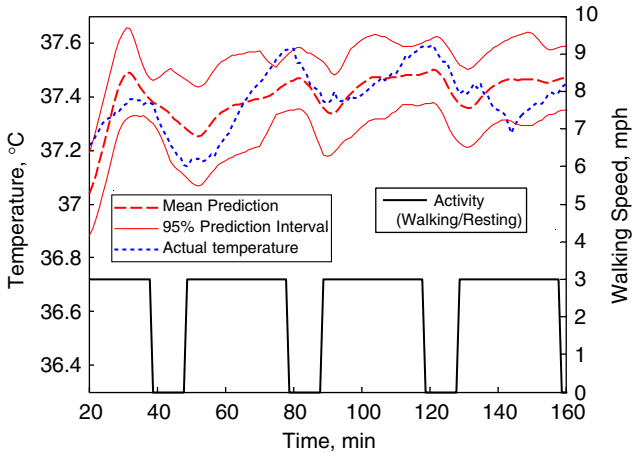

Fig. 5. 20-min-ahead prediction of core temperature (CONTROL condition) with 95% prediction intervals.

cross-correlation between the prediction interval and the two variables is plotted in Fig. 4. This figure shows a high correlation between the variables and the prediction interval with a peak correlation of almost 0.9. Note the peaks corresponding to the periodicity of the core temperature profile. Thus, Figs. 3 and 4 serve to confirm the hypothesis that the uncertainty of model predictions, and therefore, the prediction interval, is related to the variability of the data used in prediction as well as the variability in the output being predicted.

4.3. Impact of the quality of the sample of models

It is evident that the size of the standard deviation of model predictions will also be influenced by the quality of the sample of models used in the predictions. If the sample of models is not sufficiently representative of the "true" distribution of candidate models, then, as in the case of a poor data sample, one would expect the confidence and prediction intervals provided by the bootstrap technique described in this paper to be less accurate (larger in this case). As an exercise, a different sample of models was obtained to yield a less accurate set of models. This was done by perturbing the parameters of each of the models in the prior sample, in effect, resulting in a "poor" sample. The bootstrap technique described in this paper was then employed on this sample using the same data. As expected, the simulations resulted in a wider standard deviation of model predictions, and hence, in this case, a wider prediction interval.

While we suggest a particular method for obtaining the models, it should also be emphasized that there could be many other efficient ways of obtaining a desirable model set, particularly if the user has some prior knowledge about the nature of the system.

4.4. Extensions to larger prediction horizons and other subjects

The procedure described above for a 10-min-ahead prediction was tested for larger prediction horizons. It is expected that the predictions of all the models would deteriorate due to accumulation of modeling error caused by the iterative nature of the prediction process. Hence, their standard deviation would, in general, also increase. A simulation for 20-min-ahead predictions and corresponding 95% prediction intervals for the same subject, under the same conditions, confirmed a prediction interval that was 8% larger than that obtained for the 10-min-ahead predictions. The results of this simulation are shown in Fig. 5. For this simulation, as in the case of the 10-min-ahead prediction, a conservative confidence/prediction factor of $c = 2.74$ was used in the simulations, even though the numerical count indicated that one as low as $c = 2.5$ could have been used for this data.

While, in this paper, we show the results for only a single subject, simulations carried out for other subjects produced similar results with the confidence factor chosen as $c = 2.74$ for all the subjects.

4.5. Satisfying the independent and identically-distributed data assumption

Although a fundamental assumption of the bootstrap method is that the data are IID [6], in practice, this assumption is difficult to verify and to obey. In our case, the assumption of data independence may be violated if the original data set is of limited size. Limited data may cause the bootstrap moving blocks to contain overlapping data segments, causing the derived models to be dependent of each other and the variance to be underestimated. The assumption of IID data is, most likely, observed in our case as long as the order of the models is fixed throughout the approach. Even if this is not the case, more recent studies have shown that violation of the identical distribution assumption does not necessarily invalidate the bootstrap method [12].

5. Conclusions

The work presented in this paper demonstrates how the bootstrap method can be extended to estimate confidence and

prediction intervals for data-driven models of dynamical systems driven by exogenous inputs. The pitfalls of the moving blocks bootstrap, which result from having to concatenate data to form bootstrap samples, are avoided by sampling from a distribution of model estimates rather than the data itself. This approach then lends itself to an extension of the existing techniques (for estimation of prediction intervals for regression models) to estimation of prediction intervals for the outputs of dynamical systems. This contribution is appealing, in the context of identification and control of dynamical systems, since it is possible to use it in conjunction with any of the existing standard methods for data-driven models, hence, providing a measure of reliability of these methods. Without such a measure, predictions of the outputs of dynamical systems would, in many circumstances, be of little use.

The reader should realize, however, that the amount of available data to generate the bootstrap samples plays a key role in the accuracy of the estimated confidence/prediction intervals. Limited original data may cause the moving blocks to contain overlapping data segments, causing the derived models to be dependent of each other and underestimate the confidence/prediction intervals.

The approach described in this paper is illustrated using data from a laboratory study of human core temperature. The 95% confidence/prediction factors, c, are computed from a numerical count of the actual number of predictions that lie within the region predicted by these factors. The computed factors from this count are found to coincide with confidence/prediction factors derived from the Camp–Meidell inequality, therefore, validating the results obtained in our illustration. Moreover, the results show, as expected, that the prediction interval increases with increased variation in the input and output data, the prediction horizon, and the inaccuracy of the models.

Acknowledgments

The authors would like to acknowledge the support of Nela Zavaljevski for many useful suggestions and comments on the bootstrapping technique, Peg Kolka for comments on the manuscript, and William Santee for the data used in the illustration.

The authors were supported, in part, by the Combat Casualty Care and the Military Operational Medicine research programs of the U.S. Army Medical Research and Materiel Command, Ft. Detrick, Maryland. The laboratory study for the data used in the illustration was funded, in part, by the Natick Soldier System Center, Natick, Massachusetts.

Disclaimer. In collecting the data presented in this manuscript, the investigators of the study adhered to the policies for protection of human subjects as prescribed in Army Regulation 70-25, and the research was conducted in adherence with the provisions of 45 CFR Part 46. The subjects gave their informed consent to be studied during their field training exercise after being informed of the purpose, risks, and benefits of the study.

The opinions or assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the U.S. Army or of the U.S. Department of Defense.

References

[1] B.W. Lindgren, Statistical Theory, Chapman & Hall, London, 1993.
[2] K.K. Kraning, R.R. Gonzalez, A mechanistic computer simulation of human work in heat that accounts for physical and physiological effects of clothing, aerobic fitness, and progressive dehydration, J. Therm. Biol. 22 (1997) 331–342.
[3] N. Oleng', J. Reifman, L. Berglund, R. Hoyt, Alternative approaches to improve physiological predictions, in: Proceedings of the Army Science Conference, Orlando, FL, KO-05, 2004.
[4] N. Oleng', J. Reifman, Hybrid approaches to physiologic modeling and prediction, Conference on Bio-monitoring for Physiological and Cognitive Performance during Military Operations, in: Proceedings of the SPIE Symposium on Defense and Security, Orlando, FL, 2005, pp. 93–203.
[5] J.A.J. Stolwijk, J.D. Hardy, Control of body temperature, in: D.H.K. Lee (Ed.), Handbook of Physiology: Reactions to Environmental Agents, American Physiological Society, 1997, pp. 45–68.
[6] B. Efron, Bootstrap methods: another look at the jackknife, Ann. Statist. 7 (1979) 1–26.
[7] T. Heskes, Practical confidence and prediction intervals, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Proceedings of NIPS 9, 1997, pp. 176–182.
[8] D.A. Nix, A.S. Weigend, Estimating the mean and variance of the target probability distribution, in: Proceedings of the IEEE International Conference on Neural Networks, Orlando, FL (IEEE–ICNN'94), 1994, pp. 55–60.
[9] W. Hardle, J. Horowitz, J.-P. Kreiss, Bootstrap methods for time series, Int. Stat. Rev. 71 (2003) 435–459.
[10] D.N. Politis, The impact of bootstrap methods on time series analysis, Stat. Sci. 18 (2) (2003) 219–230.
[11] L. Felsenstein, Confidence limits on phylogenies: an approach using the bootstrap, Evolution 39 (4) (1985) 783–791.
[12] R.Y. Liu, Bootstrap procedures under some Non-I.I.D. models, Ann. Stat. 16 (4) (1988) 1698–1708.
[13] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, London, 1993.
[14] L. Ljung, System Identification—Theory for the User, Prentice-Hall, NJ, 1999.
[15] G.C. Goodwin, K.S. Sin, Adaptive Filtering, Prediction and Control, Prentice-Hall, NJ, 1984.
[16] W. Santee, L. Berglund, A. Cardello, C. Winterhalter, T.L. Endrusick, Physiological and psychological assessment of volunteers wearing air permeable battle-dress (BDU) uniforms during intermittent exercise. U.S. Army Research Institute of Environmental Medicine, Technical Report, in preparation.
[17] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.
[18] T.P. Ryan, Statistical Methods for Quality Improvement, Wiley, New York, NY, 2000.

Dr. Nicholas Oleng', was a Research Scientist with the Henry M. Jackson Foundation, supporting the U.S. Army Medical Research and Materiel Command's Bioinformatics Cell, Ft. Detrick, Maryland.

Dr. Andrei Gribok, Research Assistant Professor, University of Tennessee, Knoxville, is an IPA with the U.S. Army Medical Research and Materiel Command's Bioinformatics Cell, Ft. Detrick, Maryland.

Dr. Jaques Reifman, Senior Research Scientist, Department of the Army, is the Director of the U.S. Army Medical Research and Materiel Command's Bioinformatics Cell and Biotechnology High Performance Computing Software Applications Institute, Ft. Detrick, Maryland.