Pacific Northwest National Laboratory

Operated by Battelle for the
U.S. Department of Energy

Power System Very Short-Term Load Prediction

RECEIVED
SEP 22 1997
OSTI

D. J. Trudnowski
J. M. Johnson
P. Whitney

September 1997

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

PNNL-11675

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831;
prices available from (615) 576-8401.

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161

The contents of this report are printed on recycled paper.

PNNL-11675
UC-350

Power System Very Short-Term Load Prediction

D.J. Trudnowski
J.M. Johnson
P. Whitney

February, 1997

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ph

Pacific Northwest National Laboratory
Richland, Washington 99352

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

Acknowledgments

The authors wish to thank Mr. Warren McReynolds of the Bonneville Power Administration (BPA) for guiding and supporting this work. Mr. Bart McManus of BPA also provided technical insight. Mr. Mark McLaughlin, a student at Montana Tech, conducted many of the data analyses related to the Kalman predictors.

Summary

A fundamental objective of a power-system operating and control scheme is to maintain a match between the system's overall real-power load and generation. To accurately maintain this match, modern energy management systems require estimates of the future total system load. Several strategies and tools are available for estimating system load. Nearly all of these estimate the future load in 1-hour steps over several hours (or time frames very close to this). While hourly load estimates are very useful for many operation and control decisions, more accurate estimates at closer intervals would also be valuable. This is especially true for emerging Area Generation Control (AGC) strategies such as look-ahead AGC. For these short-term estimation applications, future load estimates out to several minutes at intervals of 1 to 5 minutes are required.

The currently emerging operation and control strategies being developed by the BPA are dependent on accurate very short-term load estimates. To meet this need, the BPA commissioned the Pacific Northwest National Laboratory^a (PNNL) and Montana Tech (an affiliate of the University of Montana) to develop an accurate load prediction algorithm and computer codes that automatically update and can reliably perform in a closed-loop controller for the BPA system. The requirements include accurate load estimation in 5-minute steps out to 2 hours. This report presents the results of this effort and includes: a methodology and algorithms for short-term load prediction that incorporates information from a general hourly forecaster; specific algorithm parameters for implementing the predictor in the BPA system; performance and sensitivity studies of the algorithms on BPA-supplied data; an algorithm for filtering power system load samples as a precursor to inputting into the predictor; and FORTRAN 77 subroutines for implementing the algorithms.

Using BPA data spanning several months, significant improved predictions are demonstrated using the presented algorithms when compared to a standard hourly forecaster. Using spline interpolation on the hourly forecast, the root-mean-square error of the forecast 5-minutes ahead is 583 MW. Using the algorithm in the report, this error is reduced to 73 MW (the average value of the load is near 7000 MW).

^a Pacific Northwest National Laboratory is operated for the U. S. Department of Energy by Battelle Memorial Institute under Contract DE-AC06-76RLO 1830.

Contents

Acknowledgements	iii
Summary	v
1. Introduction	1.1
1. State of the Art	2.1
1. Prediction Strategy	3.1
1. BPA Load Evaluation	4.1
1. Zero-Phase Filter	5.1
1. Kalman Prediction	6.1
1.1. Adaptive Kalman Prediction	6.2
1. Hourly Predictors	7.1
1.1. 5th-Order Standard Hourly Predictor	7.1
1.2. 1st-Order Adaptive Hourly Predictor	7.4
1. 5-Minute Predictors	8.1
1. Conclusions	9.1
1. References	10.1
A. Appendix: FORTRAN Codes	A.1
A.1. KALMANA.F	A.1
A.2. PREDICTA.F	A.3
A.3. PREDICTA.CMN	A.5
A.4. PREHR1_4.CMN	A.6
A.5. PREHR5_3.CMN	A.6
A.6. PREMN1_1.CMN	A.8
A.7. PREMN3_2.CMN	A.9
A.8. DFIL2.F	A.10

A.9. DFLSEG2.F	A.11
A.10. DZFL2.F.....	A.13
A.11. FILTER.CMN.....	A.15

Figures

1. Overall Prediction Strategy.	3.1
2. Sample of 1995 BPA 1-hour load averages (y_1) and 1-hour forecasts (f_0) versus time. Top plot shows load and forecast; bottom plot shows load - forecast (e_1).	4.2
3. Lag plots for a sample of 1995 BPA 1-hour load averages minus 1-hour forecasts (e_1). Top left plot shows a 1-hour lag progressing to a 4-hour lag in the bottom right plot.	4.2
4. Power Spectral Density (PSD) of approximately 1 year of BPA 1-hour load averages minus 1-hour forecasts (e_1) taken in years 1994 and 1995.	4.3
5. Zero-phase filtering process.	5.1
6. Magnitude frequency responses of the 30th-order MA filter and the zero-phase Butterworth-Elliptic filter.	5.2
7. Filter results from a 1000 MW/hr ramp.	5.3
8. Typical results obtained from filtering actual load data.	5.4
9. 5th-order standard hourly Kalman predictor model. ZOH = zero-order hold. Open switches represent samplers sampling at a rate of 1 sample per hour. Time base is hours.	7.2
10. Power Spectral Densities (PSD) of 5th-order hourly predictor input and output for $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30)-f_1(t+30t)\}$, 2-hour = $\text{PSD}\{y_1(t+90)-f_1(t+90t)\}$, 3-hour = $\text{PSD}\{y_1(t+150)-f_1(t+150t)\}$, and 4-hour = $\text{PSD}\{y_1(t+210)-f_1(t+210t)\}$	7.2
11. Power Spectral Densities (PSD) of 5th-order hourly predictor input and output for $A_1 = 100$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30)-f_1(t+30t)\}$, 2-hour = $\text{PSD}\{y_1(t+90)-f_1(t+90t)\}$, 3-hour = $\text{PSD}\{y_1(t+150)-f_1(t+150t)\}$, and 4-hour = $\text{PSD}\{y_1(t+210)-f_1(t+210t)\}$	7.3
12. Time-domain of 5th-order hourly predictor input and output for $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30)-f_1(t+30t)$	7.3
13. Time-domain of 5th-order hourly predictor input and output for $A_1 = 100$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30)-f_1(t+30t)$	7.4
14. Power Spectral Densities (PSD) of 1st-order adaptive hourly predictor input and output. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30)-f_1(t+30t)\}$, 2-hour = $\text{PSD}\{y_1(t+90)-f_1(t+90t)\}$, 3-hour = $\text{PSD}\{y_1(t+150)-f_1(t+150t)\}$, and 4-hour = $\text{PSD}\{y_1(t+210)-f_1(t+210t)\}$	7.5
15. Time-domain of 1st-order adaptive hourly predictor input and output. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30)-f_1(t+30t)$	7.6

16. 5-minute standard Kalman predictor model..... 8.1
17. Power spectral densities (PSD) of 3rd-order 5-minute predictor input and output. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; hourly = $\text{PSD}\{e_2(t)\}$; 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$ 8.2
18. Power spectral densities (PSD) of 3rd-order 5-minute predictor versus 1st-order predictor. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$ 8.3
19. Time-domain sample 1st-order 5-minute predictor used in Figure 18 ($p_3(t+5|t)$) versus the original forecast ($f_0(t)$) and the actual load ($y_2(t)$)..... 8.3
20. Power spectral densities (PSD) of 1st-order 5-minute predictor using the 5th-order standard hourly predictor versus one using the 1st-order adaptive hourly predictor. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$ 8.4
21. Power spectral densities (PSD) of 1st-order 5-minute predictor with $A_1 = 0.03$ versus $A_1 = 0.01$. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$ 8.4
22. Power spectral densities (PSD) of 3rd-order 5-minute predictor with $A_1 = 0.4$ versus $A_1 = 0.04$. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$ 8.5

1. Introduction

A fundamental objective of a power-system operating and control scheme is to maintain a match between the system's overall real-power load and generation. To accurately maintain this match, modern energy management systems require estimates of the future total system load. Several strategies and tools are available for estimating system load. Nearly all of these estimate the future load in 1-hour steps over several hours (or time frames very close to this). While hourly load estimates are very useful for many operation and control decisions, more accurate estimates at closer intervals would also be valuable. This is especially true for emerging Area Generation Control (AGC) strategies (such as look-ahead AGC – e.g., see Schulte (1996)). For these short-term estimation applications, future load estimates out to several minutes on intervals of 1 to 5 minutes are required. This report describes a methodology for estimating a power system load in 5-minute steps out to 2 hours. The methodology is applied to data supplied by the Bonneville Power Administration (BPA) and specific algorithms (and FORTRAN subroutines) are presented for filtering system load samples and forming short-term future estimates.

An overall or total power system load represents the net effect of all customer loads and tie-line interchanges. In a large system such as the BPA, the total load is representative of thousands of customers. In general, it is well accepted that the net total load has a stochastic component and a component driven by several global inputs such as time-of-day, temperature, etc. As an example, typically a person turns on their house lights at approximately the same time every night plus or minus some random amount of time. Mathematically, the load can be modeled as

$$y(t) = y_d(t) + y_s(t) \quad (1)$$

where y is the system load, y_d is the "deterministic" component of the load, and y_s is the "stochastic" component; the assumed units is MW. Each component in Equation (1) is dependent on time t . The deterministic component of the load (y_d) is primarily a result of the cause-effect dependence on measurable inputs such as time-of-day, day-of-week, temperature, and other factors. The stochastic component is primarily driven by the random variations in customer and interconnection loading. The majority of load estimation methods assume a cause-effect relationship and use the inputs such as time-of-day and temperature to estimate the future system load in 1-hour steps over several hours. If such a method performs accurately, it will primarily estimate the y_d component of the load. Using industry nomenclature, we call these "short-term load forecasting" methods. If one desires to estimate the load more accurately, then the y_s component must also be estimated. The estimation methodology presented in this report incorporates a general 1-hour load forecaster with a stochastic prediction algorithm to estimate both the y_d and y_s components of the future load in 5-minute steps over 2 hours. For purposes of clarity, we will call such a method a "very short-term load prediction" method.

The currently emerging operation and control strategies being developed by the BPA are dependent on accurate very short-term load estimates (this includes a look-ahead AGC system). To meet this need, the BPA commissioned the Pacific Northwest National Laboratory (PNNL) and Montana Tech (an affiliate of the University of Montana) to develop an accurate load prediction algorithm and computer codes that automatically update and can reliably perform in a closed-loop controller for the BPA system. The requirements for the BPA include accurate load estimation in 5-

minute steps out to 2 hours. This report presents the results of the effort and includes: a methodology and algorithms for short-term load prediction that incorporates information from a general hourly forecaster; specific algorithm parameters for implementing the predictor in the BPA system; performance and sensitivity studies of the algorithms on BPA supplied data; an algorithm for filtering power system load samples as a precursor to inputting into the predictor; and FORTRAN 77 subroutines for implementing the algorithms.

2. State of the Art

In surveying the open literature, one finds few published results available on estimating future power system loads over a several minute time-frame. A literature search revealed only two publications that directly address the problem. The most recent result is in Liu et al. (1996). Liu et al. compares three algorithms based on fuzzy-logic, neural networks, and auto-regressive modeling to predict the system load 30 minutes into the future in 1-minute steps. Each algorithm uses the previous load samples as an input, and none of them incorporates information from an hourly forecaster. The performance of the three methods is compared using a 24-hour data set. For the conducted tests, it is concluded that the fuzzy-logic and neural-network approaches are superior to the auto-regressive method. The error for a 1-minute-ahead prediction is shown to be consistently below 1%.

Ross et al. (1978) discusses a 5-minute prediction scheme using an auto-regressive algorithm. The algorithm operates on the difference between the current system load and the hourly forecast. Intermediate forecast data points are obtained using a cubic spline interpolation. A 2nd-order auto-regressive model driven by white noise is used to represent the load minus forecast data. The model is used to develop a recursive update algorithm for predicting the load in 5-minute steps. For the data set evaluated; the authors show a standard deviation 4.71 MW in the 5-minute-ahead estimated load. The 1-hour-ahead deviation is 80.7 MW.

Unlike very short-term prediction, several published results are available that address hourly forecasting. It is beyond the scope of this work to review all of them; therefore, a short overview is given here. Nearly all of the methods use inputs such as day of the week, time-of-day, temperature, and past loads to forecast future loads generally in 1-hour steps. Traditional methods are primarily based on regression and time-series models; for example, see Gross and Galiana (1987) and Papalexopoulos and Hesterberg (1990). Newer results have focused on using neural network approaches (see Highley and Hilmes (1993) and Bakirtzis et al. (1996)).

3. Prediction Strategy

As hypothesized in Section 1, the power system load consists of a stochastic component (y_s) and a component (y_d) driven by several global inputs such as time-of-day, temperature, etc. as represented by Equation (1). Because there are many short-term forecasting methods that provide accurate estimates to y_d , our goal is to develop a very short-term prediction strategy that can use information from any general short-term forecaster. If one is given an accurate forecast, then the difference between the load and forecast is dominated by y_s . Then the problem becomes one of predicting future values of y_s .

A strategy that incorporates a general short-term hourly forecaster is shown in Figure 1. There are two main steps to the strategy. The first is to improve the hourly forecast estimates using an hourly predictor and to interpolate it to a 5-minute sample rate. The second is to use the improved forecasts as inputs to a 5-minute predictor. In general, the hourly predictor provides a gross adjustment, and the 5-minute predictor provides a fine-tuned adjustment.

Figure 1. Overall Prediction Strategy.

The first step of the strategy starts with the current system load at time t (in units of minutes), which is represented by $y(t)$; in the BPA system, $y(t)$ is sampled every 10 seconds. Signal $y_1(t-30)$ represents the average system load over the hour $t-60$ to t ; the time argument for y_1 is $t-30$ because it represents the middle of the averaging window. Every hour the load estimate from a general forecaster $f_0(t-30)$ is subtracted from $y_1(t-30)$ to form an error $e_1(t-30)$. Residual e_1 represents the error of the general forecaster for the hour $t-60$ to t . This error is passed to an hourly prediction algorithm and ultimately used to form improved hourly forecast in f_1 . For example, $f_1(t+30t)$ is the system load forecast at time $t+30$ given information up to time t . To use the hourly forecasts in a 5-minute predictor, intra-hour interpolation must be performed in 5-minute steps. As suggested in Ross et al. (1978), this is done using cubic-spline interpolation (Kahaner et al. 1988) between $y_1(t-30)$, $f_1(t+30t)$, $f_1(t+90t)$, and $f_1(t+150t)$. The interpolated forecast is sampled every 5 minutes and termed f_2 .

The second step of the strategy operates every 5 minutes. First the load $y(t)$ is low-pass filtered to obtain $y_2(t)$. The goal of the filter is to have $y_2(t)$ represent a 5-minute average of the load at time t . Every 5 minutes the forecast $f_2(t|t)$ is subtracted from $y_2(t)$ to obtain an error $e_2(t)$. This error is passed to the 5-minute prediction algorithm, which forms predictions of future errors; for example, $p_2(t+5|t)$ is the predicted error at time $t+5$ given information up to t . The predicted errors are added to the 5-minute forecasts in f_2 to form the final predicted loads in p_3 .

Unlike the methods described in Liu et al. (1996), our strategy allows any general hourly forecaster to be used in f_0 . In developing the hourly and 5-minute predictors, it is assumed that errors e_1 and e_2 are primarily random in nature; thus providing the motivation for Kalman filtering. The sections that follow discuss development of the "Filter" block, "Hourly Predictor" block, and the "5-min Predictor" block.

4. BPA Load Evaluation

Fundamental to developing the hourly and 5-minute predictors in Figure 1 is understanding the dynamic nature of their respective inputs. The amount of correlation, the frequency content, and the magnitude of the errors e_1 and e_2 provide rich information for developing the predictors. Because e_1 is the primary driving function for the prediction strategy, we focus our evaluation on this error for a sampled BPA data set. Evaluation of e_2 is carried out later, once specific hourly predictors are developed. Any stochastic predictor must have some knowledge of the dynamic behavior of e_1 ; the range of potential results include:

- If time samples of the error are correlated and look like colored noise, then additional predictive power is available.
- If the error looks like white uncorrelated noise, then no intrinsic predictability gain is anticipated; however, the variance of the noise can be used to provide upper and lower bounds on the forecast.

The obvious approach for examining the data is in the time domain. Figure 2 shows a 2-day sample of BPA load versus the 1-hour forecasts f_0 . The top plot shows the two data sets and the bottom the difference (corresponding to e_1 in Figure 1). Typical of the BPA load is a double-hump 24-hour cyclic peak. One measure of accuracy of a forecast is to calculate the root-mean-square (rms) value of e_1 (the definition of rms is presented in several electrical engineering text books, e.g., see Nilsson 1983). A typical rms error for the BPA load using the forecaster shown in Figure 2 is 235 MW. Evaluation of a predictor's performance through studying time-domain plots is very difficult because of the random nature of the data. As an alternative, autocorrelation and spectral analysis provide more of an "average" performance index.

If a given hourly predictor provides improvement, then f_1 will more accurately represent future samples of the load y_1 than the original forecast f_0 . A fundamental question is: how far into the future can one expect to improve the forecast? One way of addressing this question is to study the time autocorrelation of e_1 . A graphical way of representing the autocorrelation is through a "lag plot." With a lag plot, $e_1(t)$ is plotted against $e_1(t+\tau)$. If $e_1(t)$ is correlated with $e_1(t+\tau)$, then the plot will approximate a straight line with unity slope. As the two become uncorrelated, the plot takes on a random "scattered" appearance. The value of τ at which the plot becomes scattered effectively represents the maximum horizon in which a predictor can improve the forecast (i.e., after τ minutes the optimum value for $p_2(t-30+\tau/t)$ is zero — or some constant if there is a steady-state difference between f_0 and y_1). Figure 3 shows four lag plots for an 83-day sample of 1-hour average BPA load minus forecast (i.e., e_1 in Figure 1). One can see the transition from a high correlation in the $e_1(t)$ -versus- $e_1(t+60)$ figure to very little correlation in the $e_1(t)$ -versus- $e_1(t+240)$ figure. Such results are typical for BPA data recorded in 1994 and 1995. From this, one can conclude that improved forecasts over the first 2 hours is likely, but improvement past 4 hours is unlikely.

Figure 2. Sample of 1995 BPA 1-hour load averages (y_1) and 1-hour forecasts (f_0) versus time. Top plot shows load and forecast; bottom plot shows load - forecast (e_1).

Figure 3. Lag plots for a sample of 1995 BPA 1-hour load averages minus 1-hour forecasts (e_1). Top left plot shows a 1-hour lag progressing to a 4-hour lag in the bottom right plot.

Yet another approach for understanding the characteristics of e_1 is to consider its power spectral density (PSD). The PSD is the Fourier transform of the autocorrelation function and

effectively represents a spectral measure of e_1 . If the PSD of e_1 is large at a given frequency, then e_1 contains a large component at that frequency. Figure 4 contains the PSD estimate for 1 year of BPA e_1 data. Close examination of Figure 4 shows the spectral content of e_1 can be broken into two components:

1. The lower frequencies are dominated by spectral peaks at the frequencies of 1/24 and 1/12 cycles per hour. This indicates that e_1 contains significant energy at these frequencies. Such an error is expected because of the double-hump 24-hour cyclic peaking that occurs in the BPA system. The time derivatives at the peaks and valleys are changing sign; therefore, these are the most difficult times to accurately forecast the load.
2. As the frequency increases above 1/12 cycles per hour, the PSD trend decreases at a fairly constant rate of approximately 15 dB/decade.

The split in the spectral characteristics of e_1 is a primary motivation for splitting the prediction algorithm into the hourly predictor and 5-minute predictor in Figure 1.

Figure 4. Power Spectral Density (PSD) of approximately 1 year of BPA 1-hour load averages minus 1-hour forecasts (e_1) taken in years 1994 and 1995.

5. Zero-Phase Filter

The goal of the "Filter" block in Figure 1 is to have $y_2(t)$ represent a 5-minute average of the 10-second load samples $y(t)$. This is done for two reasons: 1) it is undesirable for BPA to have the AGC system reacting to higher frequency changes in the load; and 2) to avoid aliasing caused by resampling at a 5-minute rate. Standard filtering approaches can introduce significant delay between $y(t)$ and $y_2(t)$; for AGC applications, this is recognized as a serious problem in Ross et al. (1978). Because $y_2(t)$ is sampled at a lower rate, one can use a zero-phase filtering approach to minimize the phase delay between $y(t)$ and $y_2(t)$. This section discusses the filter that was developed for this purpose.

One method for removing the undesired high frequency components and resampling is to simply calculate the 5-minute moving average of the 10-second load samples and sample this every 5 minutes. Mathematically, this corresponds to filtering the 10-second data with a moving average (MA) filter of order 30 and sampling the output every 5 minutes. A drawback to this technique is that the MA filter will also produce a $2\frac{1}{2}$ -minute delay in the filtered load data with respect to the actual data. Because control is taken every 5 minutes, this delay can cause undesired effects. Thus, a filter with a magnitude characteristic similar to that of the 30th-order MA filter, but with minimal phase delay, is desirable.

Ideally, a filter with the desired magnitude characteristic but with no phase delay should be used for filtering the 10-second load data. Zero-phase filtering techniques have been developed and are discussed in Oppenheim and Schaffer (1989). One technique involves filtering the actual load data forwards and then backwards. Let $\{h(k), k = 0, 1, \dots, \infty\}$ denote the sampled impulse response sequence of a causal filter with arbitrary phase characteristic and let $\{x(k), k = 0, 1, \dots, N-1\}$ denote the sequence of samples to be filtered. The filtering operation is performed as shown in Figure 5. It can be shown that this scheme results in an overall filter transfer function of $H(z)H(z^{-1}) = |H(z)|^2$, where $H(z)$ is the transfer function of the original filter and is the Z-transform of $h(k)$. It is then easy to see that the resulting filtered sequence $r(n)$ has precisely zero phase delay.

Figure 5. Zero-phase filtering process.

Before applying this technique to filter a sequence $x(k)$, special care must be taken to minimize startup and ending transients associated with filtering through $H(z)$. Krauss et al. (1994) recommends forming an augmented sequence consisting of $x(k)$ along with several filter lengths of a reversed and reflected copy of $x(k)$ prepended to each end. This sequence is then filtered and the added points are removed from the filtered sequence. Let n be the order of the filter $h(k)$ and let m be an integer. The augmented sequence filtered is $(2x(0)-x(mn), \dots, 2x(0)-x(1), x(0), \dots, x(N-1), 2x(N-1)-x(N-2), \dots, 2x(N-1)-x(N-mn-1))$ and the first and last mn samples are removed from $r(k)$.

To apply the zero-phase filtering technique to the problem of low-pass filtering the 10-second load samples, a filter with a magnitude-squared characteristic similar to the magnitude characteristic of the 30th-order MA filter is needed. It was found that a second-order Butterworth filter designed to have a cutoff frequency of 0.0025 Hz in series with a third-order elliptic with cutoff frequency 0.0022 Hz (for $T = 10$ s) approximates the desired magnitude-squared characteristic in the passband and has a better stopband. Figure 6 shows the 30th-order MA filter frequency response along with the magnitude-squared response of a Butterworth-Elliptic filter combination. It has been found that startup and ending transients associated with using this filter can be reduced significantly by folding and reflecting the first and last 15 samples of the load sequence before filtering.

Figure 6. Magnitude frequency responses of the 30th-order MA filter and the zero-phase Butterworth-Elliptic filter.

Applying the zero-phase filtering technique directly to the problem of filtering the 10-second load samples would require knowledge of $y[k]$ at all points in the past and all points in the future with respect to the current time. We use $[k]$ to represent the time argument for y to emphasize that we are considering the k th sample. The technique is thus unsuitable for direct on-line implementation, where futuristic data is not available. It has been found that approximate zero-phase filtering can be obtained by applying the technique to overlapping segments of load data. The process used for producing the filtered 10-second load samples operates every 5 minutes and is described below:

1. Wait until 30 samples ($y[0], y[1], \dots, y[29]$) of 10-second load data have accumulated. This will represent 5 minutes. Prepend to this sequence a reversed and reflected copy of the sequence to give a new sequence ($2y[0]-y[29], 2y[0]-y[28], \dots, 2y[0]-y[1], y[0], y[1], \dots, y[29]$) of length 59 samples. Continue folding and reflecting the sequence backwards in this manner until a sequence of length 180 samples is obtained. Zero-phase filter this and retain the last 30 samples of the resulting sequence as samples $y_f[0], y_f[1], \dots, y_f[29]$ of the filtered load sequence. Retain sample $y_f[28]$ as the sampled y_2 of the decimated filtered load sequence.

2. Wait 5 minutes until 60 samples of load data have accumulated. Fold and reflect this sequence enough times to obtain a sequence of 180 samples. Zero-phase filter this and retain the last 30 samples of the resulting sequence as samples $y_f[30]$, $y_f[31]$, ..., $y_f[59]$ of the filtered load sequence. Retain sample $y_f[58]$ as the next sample y_2 of the decimated filtered load sequence.
3. Repeat this procedure each time $30k$ ($k = 3, 4, 5, 6$) load samples have accumulated.
4. At each 5-minute interval, zero-phase filter the previous 30 minutes of data and retain the second-to-last sample in the filtered sequence as the decimated filtered load sample y_2 .

Figure 7 shows the results of filtering a 1000 MW/hour ramp using this process along with filtering using the 30th-order MA filter. It is easy to see that the delay associated with filtering using the zero-phase filtering process is less than one-half of that associated with the MA filter. Figure 8 shows typical results obtained when using both the zero-phase and MA filtering algorithms on actual power system load data. Plotted are the 10-second load data $y(t)$ and the decimated 5-minute filtered data $y_2(t)$. As can be seen, the zero-phase filtered data follows the actual load with less delay.

Figure 7. Filter results from a 1000 MW/hr ramp.

Figure 8. Typical results obtained from filtering actual load data.

The FORTRAN routines used to implement the zero-phase include: subroutine DFILTERSEG1, subroutine DZERFILT1, subroutine DFILTER2, and include file FILTER.CMN.

6. Kalman Prediction

Referring to Figure 1, the strategy formulated in Section 3 contains two predictors: an hourly predictor that estimates future values of error e_1 ; and a 5-minute predictor that estimates future values of error e_2 . If a relatively accurate hourly forecast is used (i.e., f_0), it is not unreasonable to expect that e_1 and e_2 are primarily random. Therefore, the predictor objective is to optimally estimate the future values of a random process. A well proven and established approach for fulfilling this objective is Kalman filtering. There are thousands of published papers and several text books devoted to Kalman filtering theory and applications (e.g., see Brown and Hwang 1992). Because the literature is very extensive and mature, only a summary overview of the method as it pertains to the prediction problem is discussed here. The reader is referred to Brown and Hwang (1992) for more detail.

The Kalman filter assumes that a general signal to be predicted consists of the superposition of two components. The first is modeled as the response of a linear time-invariant system to a random Gaussian white-noise input resulting in a colored noise output. The second component is a Gaussian white-noise signal. The prediction is optimal in a mean-squared error sense. In discrete-time format, the signal model is represented in state-space form as

$$x(k+1) = Ax(k) + w(k) \quad (2a)$$

$$z(k) = Hx(k) + v(k) \quad (2b)$$

where x is the state vector of order $nx1$, A is the nxn state transition matrix, H is a $1xn$ observation vector, w is an $nx1$ vector of white random noise with covariance matrix Q , v is a white random scalar with variance R uncorrelated with w , z is the observed output, and the argument k represents the k th time sample. All parameters except k are assumed real; k is an integer. By knowing model parameters A , H , Q , and R , and measuring the system output z , the Kalman predictor provides an iterative solution to form optimal estimates of future values of x and z .

A Kalman predictor is implemented using the following recursive equations

$$K(k) = \frac{P(k)H^T}{HP(k)H^T + R} \quad (3a)$$

$$P(k+1) = A(P(k) - K(k)HP(k))A^T + Q \quad (3b)$$

$$x_e(k+1|k) = A(x_e(k|k-1) + K(k)(z(k) - Hx_e(k|k-1))) \quad (3c)$$

$$z_e(k+1|k) = Hx_e(k+1|k) \quad (3d)$$

where $K(k)$ (order $nx1$) is the Kalman gain vector at sample k , $x_e(k+1|k)$ (order $nx1$) is the optimal estimate of $x(k+1)$ given observations to sample k , $P(k)$ is a positive-definite symmetric matrix of order nxn and represents the error covariance matrix for the state estimate at sample k , $z_e(k+1|k)$ is the optimal estimate of $z(k+1)$ given observations to sample k . Superscript T represents the transpose operation.

Predictions past $k+1$ are formed using the equations

$$x_e(k + mlk) = Ax_e(k + m - 1lk) \quad (4a)$$

$$z_e(k + mlk) = Hx_e(k + mlk) \quad (4b)$$

for $m = 2, 3, \dots, n_{hor}$, where $x_e(k+mlk)$ is the optimal estimate of $x(k+m)$ given observations to k , $z_e(k+mlk)$ is the optimal estimate of $z(k+m)$ given observations to k , and n_{hor} is the prediction horizon.

The critical step in developing a Kalman predictor is to correctly identify the signal model parameters A , H , Q , and R . One approach to doing this is to formulate the parameters so that the model's spectral response closely matches the spectral content of the signal (Brown and Hwang 1992). This is the approach used in developing the parameters for hourly and 5-minute predictors. Also, the predictor must be initialized by providing estimates for $P(0)$ and $x_e(0)$.

6.1. Adaptive Kalman Prediction

There are various choices to be made in constructing a state-space model to forecast the residual loads e_1 . Adaptive strategies that allow some of the model parameters to vary in time, with the data being used to drive how the components vary, are described in Kitagawa (1981) and in Pole, West and Harrison (1994). Kitagawa presents a strategy that allows the components of H and Q to vary for an auto-regressive process of specific order. Pole et al. presents a strategy that allows the observation variance R to vary. We use this latter strategy to forecast the short-term loads.

There is little change in the fundamental model using the adaptive strategy. The changes are that, in addition to requiring an initial state estimate and covariance for the Kalman filter, initial information is required for the observation variance. Also, a parameter, known as a "discount" parameter, is needed to determine the degree to which data far in the past contributes to the current estimate of the observation error.

This change in the Kalman filter results in the following:

- The forecast confidence intervals from the Kalman filter are more likely to contain the future values even if the parameters for the Kalman filter are chosen poorly.
- One less variance needs to be chosen in this adaptive filter than would otherwise be needed (at the cost of specifying additional prior information).

The form of this adaptive Kalman filter is similar to Equations (3a-3d); the modified equations are

$$K(k) = \frac{P(k)H^T}{HP(k)H^T + S(k)} \quad (5a)$$

$$d(k+1) = \delta d(k) + \frac{S(k)(z(k) - Hq(k))^2}{HP(k)H^T + S(k)} \quad (5b)$$

$$g(k+1) = \delta g(k) + 1 \quad (5c)$$

$$S(k+1) = \frac{d(k)}{g(k)} \quad (5d)$$

$$P(k+1) = A \frac{S(k+1)}{S(k)} (P(k) - K(k)HP(k))A^T + Q \quad (5e)$$

$$x_e(k+1|k) = A(x_e(k|k-1) + K(k)(z(k) - Hx_e(k|k-1))) \quad (5f)$$

$$z_e(k+1|k) = Hx_e(k+1|k) \quad (5g)$$

where $0 < \delta \leq 1$ is the discount factor.

The subroutine KALMANA (see Appendix A) implements Equations (3) or (5), depending on the option selected. The prediction equations, Equations (4), are implemented as part of subroutine PREDICTA.

7. Hourly Predictors

Two basic approaches were taken in developing an hourly predictor. With the first, a sample set of the error e_1 was analyzed in great detail to obtain a high-order model for a standard Kalman predictor that accurately represented the data. With the second, a simple low-order adaptive Kalman predictor was employed. The adaptive model parameters were selected using a maximum likelihood optimization algorithm. Both approaches provide improved forecasts for the data set analyzed.

One would expect that the standard (first) approach would provide a more accurate forecast for the e_1 data set used to select the Kalman model parameters; but if the characteristics of e_1 change, the performance may not be as good. Similarly, one would expect the adaptive algorithm to not perform as well for the specific training data, but to do better than the fixed parameter algorithm as the characteristics of the data change.

7.1. 5th-Order Standard Hourly Predictor

If one drives a linear system with random white noise over a long period, then the average spectral content of the system output will simply be the frequency response of the system. This provides a basis for selecting the model parameters. Using the information discussed in Section 4 with considerable spectral analysis and trial-and-error, a 5th-order model was developed. The model consists of two oscillators at frequencies of 1/24 cycles per hour and 1/12 cycles per hour, and a low-pass component.

Each oscillator and the low-pass element are assumed driven by white noise. Figure 9 shows a representation of the system in Laplace domain where the assumed time-base is hours. A_1 is the amplitude of the low-pass component. A_2 is the amplitude of the 1/12 cycles per hour oscillator. A_3 is the amplitude of the 1/24 cycles per hour oscillator. Signals w_1 , w_2 , w_3 , and v are unity-variance uncorrelated white noise signals. Finally, G_N is the gain for v . The open switches in Figure 9 represent samplers at a rate of 1 sample per hour, and ZOH stands for a zero-order hold; these represent a standard approach for discretizing a continuous system (Franklin et al. 1990). Equations required to convert the model to discrete-time form are contained in the FORTRAN include file PREHR5_3.CMN shown in Appendix A.

Using over 7000 samples of BPA data collected in 1995, extensive trial-and-error and spectral analysis was used to select parameters A_1 , A_2 , A_3 , and G_N . To evaluate a predictor, we compare the power spectrum densities (PSD) of the input to the predictor to that of the corrected forecast error (using Figure 1 notation, we compare the PSD of e_1 with $y_1 - f_1$). The first parameters selected are $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. The PSD's are shown in Figure 10. As can be seen, the predictor provides considerable improvement in the forecast for a 1-hour ahead prediction ($y_1(t+30) - f_1(t+30)$ in Figure 1) and the improvement progressively decreases out to the 4-hour ahead prediction. The improvement is especially strong at the lower frequencies. The reader should note the notches at the 1/12 and 1/24 cycles/hour frequencies. Above frequencies of 0.01 cycles/hour the predictor has slightly larger errors than e_1 . The rms value of e_1 is 235 MW, the rms value of $y_1(t+30) - f_1(t+30)$ is 123 MW, the rms value of $y_1(t+90) - f_1(t+90)$ is 159 MW, $y_1(t+150) - f_1(t+150)$ is 185 MW, and the rms value of $y_1(t+210) - f_1(t+210)$ is 202 MW.

Figure 9. 5th-order standard hourly Kalman predictor model. ZOH = zero-order hold. Open switches represent samplers sampling at a rate of 1 sample per hour. Time base is hours.

Figure 10. Power Spectral Densities (PSD) of 5th-order hourly predictor input and output for $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30) - f_1(t+30|t)\}$, 2-hour = $\text{PSD}\{y_1(t+90) - f_1(t+90|t)\}$, 3-hour = $\text{PSD}\{y_1(t+150) - f_1(t+150|t)\}$, and 4-hour = $\text{PSD}\{y_1(t+210) - f_1(t+210|t)\}$.

The high frequency error can be slightly decreased by using the parameters $A_1 = 100$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$; but, these cause the degradation of the lower frequency results. Figure 11 shows the PSD's for these parameters. For these parameters, the rms errors are: 143 MW for $y_1(t+30) - f_1(t+30|t)$, 178 MW for $y_1(t+90) - f_1(t+90|t)$, 206 MW for $y_1(t+150) - f_1(t+150|t)$, and 226 MW for $y_1(t+210) - f_1(t+210|t)$. Figures 12 and 13 compare the $A_1 = 500$ and $A_1 = 100$ results in the time-domain.

Figure 11. Power Spectral Densities (PSD) of 5th-order hourly predictor input and output for $A_1 = 100$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30)-f_1(t+30|t)\}$, 2-hour = $\text{PSD}\{y_1(t+90)-f_1(t+90|t)\}$, 3-hour = $\text{PSD}\{y_1(t+150)-f_1(t+150|t)\}$, and 4-hour = $\text{PSD}\{y_1(t+210)-f_1(t+210|t)\}$.

Figure 12. Time-domain of 5th-order hourly predictor input and output for $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30)-f_1(t+30|t)$.

Figure 13. Time-domain of 5th-order hourly predictor input and output for $A_1 = 100$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30)-f_1(t+30|t)$.

An important issue is the sensitivity of the results to changes in the characteristics in the data. Because no additional data sets were available from BPA, the sensitivity of the results to model parameter variations was studied. Although such a study does not provide a direct evaluation of the predictor to data changes, it does provide valuable insight into the general sensitivity. In general, the results are quite insensitive to changes in the model parameters. Variations in model parameters by a factor of two still provide consistent results. Comparison of the $A_1 = 100$ and $A_1 = 500$ results provides a good example of this. Although the results are different, both provide improved prediction.

7.2. 1st-Order Adaptive Hourly Predictor

The choice of model parameters for a Kalman filter can also be guided using diagnostics including how well the model forecasts future observations and how well the model represents the assumed dynamics. An additional diagnostic is the likelihood function. In stochastic models, like the model behind the Kalman filter, the likelihood function has long been used to contribute to evaluating the choice of parameters in the model or selecting among different models (Akaike 1974, Ljung 1987). The likelihood function is the probability density of the observations viewed as function of the model parameters. By maximizing the likelihood function, optimal model parameters can be selected.

A 1st-order model with $A = 1$ and $H = 1$ was assumed for Equation (2) and a maximum likelihood analysis was used to select the discount factor and Q . A subset of e_1 data was initially examined. For this subset, various discount factors δ were considered; for each discount, the Q parameter was estimated using maximum likelihood. The analysis indicated that a good choice of parameters are $\delta = 0.96$ and $Q = 11000$. The discount parameter $\delta = 0.96$ was not chosen by maximum likelihood, although the choice has good likelihood properties. This choice corresponds to

using about 1 day worth of observations to estimate R , so this filter is highly adaptive with respect to R .

Using the same data as used with the standard hourly predictor, we compare the power spectral densities of e_1 and $y_1 - f_1$ in Figure 14. The adaptive algorithm improves the 1-hour ahead prediction considerably, and the improvement decreases as the horizon increases. Unlike the standard hourly predictor, the adaptive predictor does not remove the peaked errors at the 1/12 and 1/24 cycle/hour frequencies. For the adaptive predictor, the rms errors are: 130 MW for $y_1(t+30) - f_1(t+30)$, 169 MW for $y_1(t+90) - f_1(t+90)$, 202 MW for $y_1(t+150) - f_1(t+150)$, and 226 MW for $y_1(t+210) - f_1(t+210)$. Figure 15 shows the time-domain performance of the predictor; this can be compared with the hourly predictor results in Figures 12 and 13.

Figure 14. Power Spectral Densities (PSD) of 1st-order adaptive hourly predictor input and output. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; 1-hour = $\text{PSD}\{y_1(t+30) - f_1(t+30)\}$, 2-hour = $\text{PSD}\{y_1(t+90) - f_1(t+90)\}$, 3-hour = $\text{PSD}\{y_1(t+150) - f_1(t+150)\}$, and 4-hour = $\text{PSD}\{y_1(t+210) - f_1(t+210)\}$.

In comparing the hourly and adaptive results, the hourly predictor with $A_1 = 500$ provides the best results. The advantage of the adaptive predictor is in its ability to adjust for varying characteristics in e_1 . These advantages could not be fully evaluated because of limited data availability.

Equations required to implement the adaptive algorithm in the subroutine PREDICTA are in the include file PREHR1_4.CMN. These code segments are reprinted in Appendix A.

Figure 15. Time-domain of 1st-order adaptive hourly predictor input and output. Using Figure 1 notation, original = $e_1(t+30)$ and 1-hour = $y_1(t+30) - f_1(t+30|t)$.

8. 5-Minute Predictors

Two 5-minute standard Kalman predictors are compared in this section. The first is a low-pass 3rd-order model, and the second is a simple 1st-order random walk model. The parameters of both models were developed using a similar approach as the standard hourly predictor using BPA collected data with extensive trial-and-error and spectral analysis. The BPA data was collected from August 24, 1995 to December 24, 1995. The 3rd-order model provides slightly better performance, but because of the simplicity of the 1st-order model, it may be a better choice for implementation.

Figure 16 shows a representation of the 5-minute predictor model in the Laplace domain, where the assumed time-base is seconds. Signals w and v are unity-variance uncorrelated white noise signals; and G_N is the gain for v . The open switches in Figure 16 represent samplers sampling at a rate of 5 minutes per sample, and ZOH stands for a zero-order hold. Equations required to convert the model to discrete-time form are contained in the FORTRAN include files PREM3_2.CMN for the 3rd-order model and PREM1_1.CMN for the 1st-order model. For the 3rd-order model $H(s)$ is

$$H(s) = \frac{A_1(s + \pi 0.0006)(s + \pi 0.0001)}{(s + \pi 0.0014)(s + \pi 0.0003)(s + \pi 0.00004)} \quad (6)$$

where $A_1 = 0.4$, and $G_N = 1.0$. For the 1st-order model, $H(s)$ is

$$H(s) = \frac{A_1}{s} \quad (7)$$

where $A_1 = 0.03$, and $G_N = 10.0$.

Figure 16. 5-minute standard Kalman predictor model.

Figure 17 compares the power spectral densities of the input to the predictor (e_2) to the output ($y_2 - p_3$) for the 3rd-order model. PSD's for selected prediction horizons from 5 minutes to 120 minutes are shown. The hourly predictor described in Section 7.1 with the parameters $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$ was used to form f_2 . The comparison with the e_1 was done by using spine interpolation on e_1 . As can be seen, the accuracy of p_3 is considerably better than f_0 or f_2 ; this is especially true at the lower frequencies, while the higher frequencies start to slightly degrade after a prediction horizon of 15 minutes. For the data set shown in Figure 17, the rms value for e_1 is 583 MW; the rms value for $y_2(t+5) - p_3(t+5|t)$ is 73 MW; the rms value for $y_2(t+15) - p_3(t+15|t)$ is 105 MW; the rms value for $y_2(t+30) - p_3(t+30|t)$ is 144 MW; and the rms value for $y_2(t+120) - p_3(t+120|t)$ is 245 MW.

Figure 17. Power spectral densities (FSD) of 3rd-order 5-minute predictor input and output. Using Figure 1 notation: original = $\text{PSD}\{e_1(t)\}$; hourly = $\text{PSD}\{e_2(t)\}$; 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$.

The 3rd-order model is compared with the 1st-order in Figure 18. The 3rd-order results are the same as used for Figure 17 while the model in (7) was used to obtain the 1st-order results. This shows that the 3rd-order model provides only slightly better prediction than the 1st-order. For the 1st-order predictor results in Figure 18, the rms value for $y_2(t+5)-p_3(t+5|t)$ is 84 MW; the rms value for $y_2(t+15)-p_3(t+15|t)$ is 114 MW; the rms value for $y_2(t+30)-p_3(t+30|t)$ is 154 MW; and the rms value for $y_2(t+120)-p_3(t+120|t)$ is 264 MW. Figure 19 shows a time-domain sample of the performance of the 1st-order predictor used for Figure 18 versus the original hourly forecast.

The effect of the using the adaptive hourly predictor in Section 7.2 is shown in Figure 20. The "standard" curves used the hourly predictor described in Section 7.1 with the parameters $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$ was used to form f_2 . As can be seen, the standard hourly predictor performs considerably better.

As with the hourly standard predictor, the sensitivity of the 5-minute predictor to changes in data characteristics can be investigated by studying the performance for variations in the Kalman model parameters. Figure 21 shows the predictor performance for changing the gain of the 1st-order predictor in (7) from $A_1 = 0.03$ to 0.01, and Figure 22 shows the performance for changing the gain of the 3rd-order predictor in (6) from $A_1 = 0.4$ to 0.04. In both cases, the reduced gain results in poorer performance, but still provides improved prediction. In general, consistent performance is obtained for factor of 2 variations in A_1 and/or G_N for both predictors.

Figure 18. Power spectral densities (PSD) of 3rd-order 5-minute predictor versus 1st-order predictor. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$.

Figure 19. Time-domain sample 1st-order 5-minute predictor used in Figure 18 ($p_3(t+5|t)$) versus the original forecast ($f_0(t)$) and the actual load ($y_2(t)$).

Figure 20. Power spectral densities (PSD) of 1st-order 5-minute predictor using the 5th-order standard hourly predictor versus one using the 1st-order adaptive hourly predictor. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$.

Figure 21. Power spectral densities (PSD) of 1st-order 5-minute predictor with $A_1 = 0.03$ versus $A_1 = 0.01$. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$.

Figure 22. Power spectral densities (PSD) of 3rd-order 5-minute predictor with $A_1 = 0.4$ versus $A_1 = 0.04$. Using Figure 1 notation: 5 min. = $\text{PSD}\{y_2(t+5)-p_3(t+5|t)\}$; 15 min. = $\text{PSD}\{y_2(t+15)-p_3(t+15|t)\}$; 30 min. = $\text{PSD}\{y_2(t+30)-p_3(t+30|t)\}$; and 120 min. = $\text{PSD}\{y_2(t+120)-p_3(t+120|t)\}$.

9. Conclusions

This report describes a methodology for estimating a power system load in 5-minute steps out to 2 hours. The methodology is applied to data supplied by the Bonneville Power Administration (BPA) and specific algorithms (and FORTRAN subroutines) are presented for filtering system load samples and forming short-term future estimates. Presented results include a description of an overall prediction strategy consisting of: 1) an hourly predictor; 2) a zero-phase filter; and 3) a 5-minute predictor. In testing the strategy and algorithms on BPA supplied data, future loads are predicted with considerable accuracy. Two general hourly predictors are presented: a 5th-order hourly Kalman predictor; and a 1st-order adaptive Kalman predictor. The 5th-order hourly predictor with parameters $A_1 = 500$, $A_2 = 10$, $A_3 = 10$, and $G_N = 200$ provides the most accurate predictions. Two 5-minute predictors are compared; a 3rd-order one and a 1st-order one. Both provide similar results, although the 3rd-order is slightly more accurate.

10. References

- Akaike, H. 1974. "A New Look at Statistical Model Identification." *IEEE Transactions on Automatic Control* AC-19(6):716-723.
- Bakirtzis, A. G., V. Petridis, S. J. Klartzis, and M. C. Alexiadis. 1996. "A Neural Network Short Term Load Forecasting Model for the Greek Power System." *IEEE Transactions on Power Systems* 11(2-May):858-863.
- Brown, R. C., P. Y. C. Hwang. 1992. *Introduction to Random Signals and Applied Kalman Filtering*. 2nd ed. John Wiley & Sons, New York.
- Franklin, G. F., J. D. Powell, and M. L. Workman. 1990. *Digital Control of Dynamic Systems*. 2nd ed. Addison-Wesley, New York.
- Gross, G., and F. D. Galiana. 1987. "Short Term Load Forecasting." *Proceedings of the IEEE* 75(12):1558-1573.
- Highley, D., and T. Hilmes. 1993. "Load Forecasting by ANN." *IEEE Computer Applications in Power* 6(3):10-15.
- Kahaner, D., C. Mohler, and S. Nash. 1988. *Numerical Methods and Software*. Prentice Hall, New York.
- Kitagawa, G. 1981. "State Space Modeling of Nonstationary Time Series and Smoothing of Unequally Spaced Data." *Applied Time Series Analysis II*, Academic Press.
- Krauss, T. P., L. Shure, and J. N. Little. 1994. *Signal Processing Toolbox for Matlab*. The Mathworks, Inc., Natick, Massachusetts.
- Krauss, T. P., Shure, L., and Little, J. N. 1994. *Signal Processing Toolbox User's Guide*. The MathWorks, Inc., Natick, Massachusetts.
- Ljung, L. 1987. *System Identification: Theory for the User*. Prentice-Hall, New York.
- Liu, K., S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. Lewis, and J. Naccarino. 1996. "Comparison of Very Short-Term Load Forecasting Techniques." *IEEE Transactions on Power Systems* 11(2-May): 877-882.
- Nilsson, J. W. 1983. *Electric Circuits*. Addison-Wesley, New York.
- Oppenheim, A. V., and R. W. Schaffer, 1989. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Papalexopoulos, A. D., and T. C. Hesterberg. 1990. "A Regression-Based Approach to Short Term System Load Forecasting." *IEEE Transactions on Power Systems* 5(4):1535-1547.
- Pole, A., M. West, and J. Harrison. 1994. *Applied Bayesian Forecasting and Time Series Analysis*. Chapman & Hall, New York.

Ross, D., T. Athay, S. Kim, G. Ackerman, R. Podmore, and S. Virmani. 1978. "Development and Implementation of Advanced Automatic Generation Control: Final Report 2, Definition of Performance Criteria and Development of Prototype Algorithms." EC-77-01-2118, U.S. Department of Energy, Washington, D.C.

Schulte, R. P. 1996. "An Automatic Generation Control Modification for Present Demands on Interconnected Power Systems." *IEEE Transactions on Power Systems* 11(3): 1286-1294.

A. Appendix: FORTRAN Codes

A.1. KALMANA.F

```

C This is a compiling option used by BPA
C   OPTIONS/check=bounds/F77/EXTEND_SOURCE/I4

C
C*****
C KALMANA: Iterates adaptive kalman filter equations. The adaptive
C           may be implemented if desired.
C
C Contributors: D. Trudnowski, Montana Tech
C Date: Mar. 1995., Aug. 1996
C Reference: 1. R. Brown and P. Hwang, "Introduction to Random Signals
C               and Applied Kalman Filtering," John Wiley & Sons, 1992.
C               ?
C
C Performs one iteration of an adaptive Kalman filter for state-space model
C
C        $XV(t+1) = PHIM(t) * XV(t) + W(t)$ 
C        $ZV(t) = HM(t) * XV(t) + V(t)$ 
C
C For the nonadaptive case, the form of the filter being used is similar
C to the one on page 235 of reference 1. For the adaptive case, the
C update equations are from ? and have been converted to the variables
C used in Brown.
C
C Input:
C   PHIM    = State transition matrix at t.  NSxNS.
C   HM      = Observation matrix at t.  1xNS.
C   QM      = Covariance matrix for white noise W at t.  NSxNS.
C   RM      = Nonadaptive case = Covariance matrix for white noise V at t.  1x1.
C             Adaptive case = Discount factor (less than 1).  1x1.
C   PM      = Kalman error covariance at t.  NSxNS.
C   XV(1:NS) = Kalman state estimate at t.  NSx1.
C   XV(NS+1) = If greater than 0, assumed adaptive;
C             else nonadaptive.
C   XV(NS+2) = If adaptive, 1st update variable for adaptive variance at t.
C   XV(NS+3) = If adaptive, 2nd update variable for adaptive variance at t.
C   ZV      = Measured output at t.  1x1.
C   NS      = State dimension.
C
C Output:
C   PM      = Kalman error covariance at t+1.  NSxNS
C   XV(1:NS) = Kalman state estimate at t+1.  NSx1.
C   XV(NS+2) = If adaptive, 1st update variable for adaptive variance at t+1.
C   XV(NS+3) = If adaptive, 2nd update variable for adaptive variance at t+1.
C
C Significant internal variables:
C   KM      = Kalman gain matrix at time t.  NSx1.
C   EPS     = Working zero.
C   VAR     = Observation error variance.
C   ERROR   = Observation error.
C   RATIO   = Multiplier if adaptive case.
C   WORK... = Working variables.
C
C*****
C   SUBROUTINE KALMANA(PHIM,HM,QM,RM,PM,XV,ZV,NS)
C   IMPLICIT NONE
C   Include file contains dimension parameters
C   NSDIM
C   BPA's include
C   INCLUDE 'RODS_INCLUDE:[AGC]PREDICT_INCLUDE.FOR'
C   Tech's include
C   INCLUDE 'PREDICTA.CMN'
C
C   INTEGER*4 NS
C   INTEGER*4 I,J,K
C   REAL*8 PHIM(NSDIM,*),HM(*),QM(NSDIM,*),
C   REAL*8 PM(NSDIM,*),XV(*),ZV,RM

```

```

      REAL*8 KM(NSDIM),WORKRM1(NSDIM,NSDIM)
      REAL*8 WORKRM2(NSDIM,NSDIM),EPS,VAR,ERROR,RATIO
C
C *** CONSTANTS
C   EPS=WORKING ZERO
C   EPS=1.0D-12
C
C *** CHECK TO MAKE SURE XV(NS+3) NOT EQUAL TO 0 IF ADAPTIVE.
C   IF (XV(NS+1).GT.0.0D0) THEN
C     IF (DABS(XV(NS+3)).LT.EPS) XV(NS+3)=EPS
C   END IF
C
C *** COMPUTE KALMAN GAIN KM WHERE KM*VAR=PM*HM' AND OBSERVATION
C   ERROR VARIANCE IS VAR = HM*PM*HM'+RM FOR NONADAPTIVE AND
C   VAR = HM*PM*HM'+(XV(NS+2)/XV(NS+3)) FOR ADAPTIVE.
C   DO I=1,NS
C     WORKRM1(I,1)=0.0D0
C     DO K=1,NS
C       WORKRM1(I,1)=WORKRM1(I,1)+PM(I,K)*HM(K)
C     END DO
C     IF (XV(NS+1).GT.0.0D0) THEN
C       VAR=XV(NS+2)/XV(NS+3)
C     ELSE
C       VAR=RM
C     END IF
C     DO K=1,NS
C       VAR=VAR+HM(K)*WORKRM1(K,1)
C     END DO
C *** SOLVE LINEAR PROBLEM: KM*WORKRM2=WORKRM1 FOR MATRIX KM.
C   NOTE: FOR THIS VERSION, NOUT=1; THEREFORE VAR IS A
C   1x1 MATRIX. FUTURE VERSIONS REQUIRE A LINEAR SOLVER SUBROUTINE CALL.
C   IF (DABS(VAR).LT.EPS) VAR=EPS
C   DO I=1,NS
C     KM(I)=WORKRM1(I,1)/VAR
C   END DO
C *** END OF KM AND VAR CALCULATIONS.
C
C *** FORM OBSERVATION ERROR, ERROR = ZV-HM*XV(1:NS)
C   ERROR = ZV
C   DO J=1,NS
C     ERROR=ERROR-HM(J)*XV(J)
C   END DO
C *** END OF ERROR CALCULATION.
C
C *** UPDATE STATE ESTIMATE: XV=PHIM*(XV+KM*ERROR)
C   DO I=1,NS
C     WORKRM1(I,1)=XV(I)+KM(I)*ERROR
C   END DO
C   DO I=1,NS
C     XV(I)=0.0D0
C     DO J=1,NS
C       XV(I)=XV(I)+PHIM(I,J)*WORKRM1(J,1)
C     END DO
C   END DO
C *** END OF XV UPDATE
C
C *** IF ADAPTIVE, UPDATE XV(NS+2) AND XV(NS+3), AND CALCULATE RATIO.
C   IN THE NONADAPTIVE CASE, RATIO = 1
C   RATIO = 1.0D0
C   IF (XV(NS+1).GT.0.0D0) THEN
C     WORKRM1(1,1)=XV(NS+2)/XV(NS+3)
C     XV(NS+3)=RM*XV(NS+3)+ 1.0D0
C     XV(NS+2)=RM*XV(NS+2)+WORKRM1(1,1)*ERROR*ERROR/VAR
C     IF (DABS(WORKRM1(1,1)).LT.EPS) WORKRM1(1,1)=EPS
C     RATIO=XV(NS+2)/(XV(NS+3)*WORKRM1(1,1))
C   END IF
C
C *** UPDATE ERROR COVARIANCE MATRIX: PM=RATIO*PHIM*(I-KM*HM)*PM*PHIM'+QM
C   DO I=1,NS
C     DO J=1,NS
C       WORKRM1(I,J)=0.0D0
C       IF (I.EQ.J) WORKRM1(I,J)=1.0D0
C       WORKRM1(I,J)=WORKRM1(I,J)-KM(I)*HM(J)
C     END DO

```



```

      END DO
      DO I=1,NS
        DO J=1,NS
          WORKRM2(I,J)=0.0D0
          DO K=1,NS
            WORKRM2(I,J)=WORKRM2(I,J)+WORKRM1(I,K)*PM(K,J)
          END DO
          WORKRM2(I,J)=WORKRM2(I,J)*RATIO
        END DO
      END DO
      DO I=1,NS
        DO J=1,NS
          WORKRM1(I,J)=0.0D0
          DO K=1,NS
            WORKRM1(I,J)=WORKRM1(I,J)+WORKRM2(I,K)*PHIM(J,K)
          END DO
        END DO
      END DO
      DO I=1,NS
        DO J=1,NS
          PM(I,J)=QM(I,J)
          DO K=1,NS
            PM(I,J)=PM(I,J)+PHIM(I,K)*WORKRM1(K,J)
          END DO
        END DO
      END DO
C *** FORCE SYMETRIC PM
      IF (NS.GT.1) THEN
        DO I=2,NS
          DO J=1,I-1
            PM(I,J)=(PM(I,J)+PM(J,I))/2.0D0
            PM(J,I)=PM(I,J)
          END DO
        END DO
      END IF
C *** END OF ERROR COVARIANCE MATRIX UPDATE.
C
C *** RETURN
      RETURN
      END

```

A.2 PREDICTA.F

```

C This is a compiling option used by BPA
C   OPTIONS/check=bounds/F77/EXTEND_SOURCE/I4
C
C*****
C Subroutine PREDICTA: Using inputs of current system load, current and
C                     future forecasts, this subroutine predicts
C                     (or re-forecasts) the future loads using a Kalman
C                     predictor.
C
C The Kalman predictor operates on system load - forecast. It is either
C adaptive or nonadaptive. The model is in the form
C
C        $x(k+1) = PHIM*x(k) + w(k)$ 
C        $z(k) = HM*x(k) + v(k)$ 
C       QM = covariance matrix of w(k)
C       RM = variance of white noise v(k) if predictor is nonadaptive
C           or is the discount factor if adaptive.
C
C y(k) is the system load - forecast.
C
C Contributors: D. Trudnowski
C Dates: Mar. 1995, Mar. 1996., Aug. 1996
C
C Inputs:
C       TIME      = Current time (currently not used).
C       FORSP     = Vector of forecasts. NHOR+1 elements.
C                   FORSP(1) = Forecast at time TIME.
C                   FORSP(2) = Forecast at time TIME + 1 sample
C                   FORSP(3) = Forecast at time TIME + 2 samples
C
C

```

```

C          FORSP(NHOR+1) = Forecast at time TIME + NHOR
C                      samples
C  LOADF = System load at time TIME.
C  RESTART= Flag for restart.
C          If = 1, Reset Kalman error covariance matrix
C          and state vector.
C  TYPE = Flag for changing between two different predictors.
C          Use this to change between hourly and 5-minute
C          predictors.
C          If = 1, use 1st one (hourly).
C          If = 2, use 2nd one (5-minute).
C  PCOV = Kalman error covariance at time TIME.
C          NS x NS real symmetric matrix.
C          Dimensioned (NSDIM,*)
C  XSTATE = Kalman state at time TIME concatenated with a
C          3x1 vector of adaptive prediction parameters.
C          Dimensioned (NSDIM+3)
C          XSTATE(1:NS) = State vector.
C          XSTATE(NS+1) = If greater than 0, filter is adaptive
C          XSTATE(NS+2) = If adaptive, 1st update variable for
C          adaptive variance at TIME.
C          XSTATE(NS+3) = If adaptive, 2nd update variable for
C          adaptive variance at TIME.
C
C  Outputs:
C          PCOV = Kalman error-covariance at time TIME + 1 sample
C          XSTATE = Kalman state at time TIME+1 concatenated with a
C          3x1 vector of updated adaptive prediction parameters.
C          LOADPRED = Load predictions. NHOR x 1 vector.
C          LOADPRED(1) = Prediction for time TIME + 1 sample.
C          LOADPRED(2) = Prediction for time TIME + 2 samples
C
C          .
C          LOADPRED(NHOR) = Prediction for time TIME + NHOR samples
C
C  Significant internal variables (defined using INCLUDE statement):
C          NHOR = Prediction horizon.
C          NS = Kalman predictor model order.
C          DELTAT = Sample Period.
C          PHIM = Kalman state transition matrix.
C          HM = Observation matrix.
C          QM = Covariance matrix for Kalman driving noise.
C          RM = If nonadaptive, covariance matrix for Kalman
C          If adaptive, discount factor.
C          ZETA1,ZETA2,A1,A2,A3,PI,GN = Misc. model parameters
C
C  Other significant internal variables:
C          ZV = Observation error.
C          PI = 3.14...
C
C  Subroutines called:
C          KALMANA: Iterates kalman filter equations.
C
C *****
C          SUBROUTINE PREDICTA (TIME,FORSP,LOADF,LOADPRED,PCOV,XSTATE,RESTART,
C          & TYPE)
C          IMPLICIT NONE
C          *** INCLUDE FILE CONTAINS DIMENSION PARAMETERS NSDIM
C          *** FOR BPA
C          INCLUDE 'RODS_INCLUDE:[AGC]PREDICT_INCLUDE.FOR'
C          INCLUDE 'PREDICTA.CMN'
C          REAL*4 LOADPRED(*)
C          REAL*8 LOADF,FORSP(*),PCOV(NSDIM,*),XSTATE(*)
C          REAL*8 PHIM(NSDIM,NSDIM),HM(NSDIM),QM(NSDIM,NSDIM),
C          & RM,ZV,DELTAT
C          REAL*8 WORKRM1(NSDIM,NSDIM)
C          INTEGER * 4 RESTART,TIME,NHOR,TYPE
C          INTEGER * 4 NS
C          INTEGER * 4 I,J,K
C          *** PARAMETERS USED IN INCLUDE FILES
C          REAL*8 ZETA1,ZETA2,A1,A2,A3,PI,GN
C
C          *** CONSTANTS
C          PI=4.0D0*DATAN(1.0D0)
C

```

```

C *** INCLUDE CONTAINS THE FOLLOWING: NHOR,NS,DELTAT,PHIM,HM,QM,RM,
C *** AND XSTATE(NS+1).
C *** BPA FOLKS MAY WANT TO USE THE TYPE IF LOOP TO CHOOSE THE DESIRED
C *** ONE
      IF (TYPE.EQ.1) THEN
C *** USE ONE OF THE FOLLOWING HOURLY PREDICTORS:
C       PREHR5_3.CMN (5th-order standard)
C       PREHR1_4.CMN (1st-order adaptive)
C       INCLUDE 'PREHR1_4.CMN'
C       INCLUDE 'PREHR5_3.CMN'
C     ELSE
C *** USE ONE OF THE FOLLOWING HOURLY PREDICTORS:
C       PREMN3_2.CMN (3rd-order standard)
C       PREMN1_1.CMN (1st-order standard)
C       INCLUDE 'PREMN3_2.CMN'
C       INCLUDE 'PREMN1_1.CMN'
C     END IF
C
C *** RESTART
      IF (RESTART.EQ.1) THEN
        PCOV(1,1)=1.0D3
        XSTATE(1)=0.0D0
        DO I=2,NS
          XSTATE(I)=0.0D0
          PCOV(I,I)=1.0D3
          DO J=1,I-1
            PCOV(I,J)=0.0D0
            PCOV(J,I)=0.0D0
          END DO
        END DO
        IF (XSTATE(NS+1).GT.0.0D0) THEN
          XSTATE(NS+2) = 1.0D0
          XSTATE(NS+3) = 200.0D0**2
        END IF
      END IF
C *** END RESTART
C
C *** FORM OBSERVATION
      ZV = LOADF - FORSP(1)
C *** END OBSERVATION
C
C *** UPDATE KALMAN STATE (SEE SUBROUTINE KALMAN FOR STRUCTURE)
C *** ITERATE FILTER
      CALL KALMANA(PHIM,HM,QM,RM,PCOV,XSTATE,ZV,NS)
C *** END KALMAN STATE UPDATE
C
C *** PREDICT LOADPRED(I)=HM*(PHIM^(i-1))*XSTATE
      LOADPRED(1)=FORSP(2)
      DO J=1,NS
        LOADPRED(1)=LOADPRED(1)+HM(J)*XSTATE(J)
        WORKRM1(J,1)=XSTATE(J)
      END DO
      DO I=2,NHOR
        DO J=1,NS
          WORKRM1(J,2)=0.0D0
          DO K=1,NS
            WORKRM1(J,2)=WORKRM1(J,2)+PHIM(J,K)*WORKRM1(K,1)
          END DO
        END DO
        LOADPRED(I)=FORSP(I+1)
        DO J=1,NS
          LOADPRED(I)=LOADPRED(I)+HM(J)*WORKRM1(J,2)
          WORKRM1(J,1)=WORKRM1(J,2)
        END DO
      END DO
C *** END OF PREDICTION
C
C *** DONE
      RETURN
      END

```

A.3. PREDICTA.CMN

C NSDIM is the max. dimension of the kalman filter.

C Remeber the Kalman state should be dimenstioned by NSDIM+3
 C INTEGER NSDIM
 C PARAMETER (NSDIM=6)

A.4. PREHR1_4.CMN

C Include PREHR1_4: Used in PREDICTA.F. Performs 1 hour load predictions
 C over 4 hours using a 1st-order adaptive kalman filter. Version 4
 C is in the file PREHR1_4.CMN (the 4 at the end of PREHR1_4
 C references the version #).
 C
 C The input is the current system load and the corresponding forecast. The
 C output are improved estimates of future forecasts over the next 4 hours.
 C
 C The discrete-time Kalman model is:
 C
 C $x(k) = PHIM * x(k) + w(k)$
 C $z(k) = HM * x(k) + v(k)$
 C QM = covariance matrix for w(k)
 C z(k) is the load-forecast.
 C
 C Parameters selected by Paul Whitney (8/96) are
 C PHIM = 1.0
 C HM = 1.0
 C QM = 11000
 C RM = 0.96 (Discount factor).
 C
 C Author: D. Trudnowski
 C Dates: Apr. 1996, Aug. 1996
 C References: D. Trudnowski, J. Johnson, P. Whitney, and M. Donnelly,
 C "Power System Very Short-Term Load Prediction," PNNL report
 C to be published, 1996.
 C D. Trudnowski, Engineering notebook: BPA Task Order PNL13, notebook 1.
 C R. Brown, P. Hwang, "Introduction to Random Signals and
 C Applied Kalman Filtering," Chap. 5, John Wiley & Sons, 1992.
 C
 C *** SAMPLE PERIOD (1 hour)
 C DELTAT=1
 C *** PREDICTION HORIZON
 C NHOR=4
 C *** KALMAN MODEL ORDER
 C NS=1
 C *** STATE TRANSITION MATRIX (CONSTANT)
 C PHIM(1,1)=1.0D0
 C *** OBSERVATION MATRIX (CONSTANT)
 C HM(1)=1.0D0
 C *** DRIVING NOISE COVARIANCE (CONSTANT)
 C QM(1,1)=11000.0
 C *** DISCOUNT FACTOR (CONSTANT)
 C RM=0.96
 C *** FLAG TO MAKE IT ADAPTIVE
 C XSTATE(NS+1)=1.0D0

A.5. PREHR5_3.CMN

C Include PREHR5_3: Used in PREDICTA.F. Performs 1 hour load predictions
 C over 4 hours using a 5th-order kalman filter. Version 3
 C is in the file PREHR5_3.CMN (the 3 at the end of PREHR5_3
 C references the version #).
 C
 C The input is the current system load and the corresponding forecast. The
 C output are improved estimates of future forecasts over the next 4 hours.
 C
 C The 5th-order Kalman filter model is
 C
 C
$$Z(s) = \frac{A1}{s+2\pi*0.03} W1(s)$$

$$+ \frac{A2}{s^2 + 2\zeta\omega_n s + \omega_n^2} W2(s)$$

```
C
C                                     A3
C      + ----- W3(s) + GN*v(s)
C      s^2 + 2*zeta2*f2*s + (f2)^2
C
C where f1 = (2*pi/12)/sqrt(1-zeta1^2), f2 = (2*pi/24)/sqrt(1-zeta2^2),
C w1(t), ..., w4(t), v(t) are normal uncorrelated unity-variance white noise.
C The first term is a low-pass model, the second is a 12 hour
C oscillator, and the third is a 24 hour oscillator.
C Note: The time base for this model is hours (NOT seconds).
C It is assumed this model is driven by white noise sampled at T=1hr.
C thru a Zero-Order-Hold (ZOH). The discrete-time model is:
C
C      x(k) = PHIM*x(k) + B* [w1(k)|
C                             |w2(k)|
C                             |w3(k)|
C
C      z(k) = HM*x(k) + GN*v(k)
C      QM = B*B(transpose)
C      RM = GN*GN
C z(k) is the load-forecast.
C
C Author: D. Trudnowski
C Dates: Mar. 1996.
C References: D. Trudnowski, J. Johnson, P. Whitney, and M. Donnelly,
C "Power System Very Short-Term Load Prediction," PNNL report
C to be published, 1996.
C D. Trudnowski, Engineering notebook: BPA Task Order PNL13, notebook 1.
C R. Brown, P. Hwang, "Introduction to Random Signals and
C Applied Kalman Filtering," Chap. 5, John Wiley & Sons, 1992.
C
C *** MODEL PARAMETERS
C ZETA1=0.0D0
C ZETA2=0.0D0
C A1=500.0D0
C A2=10.0D0
C A3=10.0D0
C GN=200.0D0
C *** SAMPLE PERIOD (1 hour)
C DELTAT=1
C *** PREDICTION HORIZON
C NHOR=4
C *** KALMAN MODEL ORDER
C NS=5
C
C *** STATE TRANSITION MATRIX (CONSTANT)
C DO I=1,5
C   DO J=1,5
C     PHIM(I,J)=0.0D0
C   ENDDO
C ENDDO
C
C s+0.03*2*pi TERM
C PHIM(1,1)=0.82820418130686D0
C
C 12 HR. OSCILLATOR
C PHIM(2,2)= 0.0D0
C PHIM(2,3)=-DEXP(-4.0D0*ZETA1*PI/(12.0D0*DSQRT(1.0D0-ZETA1**2)))
C PHIM(3,2)=1.0D0
C PHIM(3,3)=2.0D0*
C &    DEXP(-2.0D0*ZETA1*PI/(12.0D0*DSQRT(1.0D0-ZETA1**2)))*
C &    DCOS(2.0D0*PI/12.0D0)
C
C 24 HR. OSCILLATOR
C PHIM(4,4)= 0.0D0
C PHIM(4,5)=-DEXP(-4.0D0*ZETA2*PI/(24.0D0*DSQRT(1.0D0-ZETA2**2)))
C PHIM(5,4)=1.0D0
C PHIM(5,5)=2.0D0*
C &    DEXP(-2.0D0*ZETA2*PI/(24.0D0*DSQRT(1.0D0-ZETA2**2)))*
C &    DCOS(2.0D0*PI/24.0D0)
C
C *** OBSERVATION MATRIX (CONSTANT)
C s TERM
C HM(1)=A1
```



```

C      x(k) = PHIM*x(k-1) + B*w(k)
C      z(k) = HM*x(k) + GN*v(k)
C      QM = B*B
C      RM = GN*GN
C PHIM=1.0, B=300, HM=A1, z(k) is the load-forecast.
C
C Author: D. Trudnowski
C Dates: May 1996.
C References: D. Trudnowski, J. Johnson, P. Whitney, and M. Donnelly,
C             "Power System Very Short-Term Load Prediction," PNNL report
C             to be published, 1996.
C             D. Trudnowski, Engineering notebook: BPA Task Order PNL13, notebook 1.
C             R. Brown, P. Hwang, "Introduction to Random Signals and
C             Applied Kalman Filtering," Chap. 5, John Wiley & Sons, 1992.
C
C *** MODEL PARAMETERS
C      A1=0.03D0
C      GN=10.0D0
C *** SAMPLE PERIOD (300 sec)
C      DELTAT=300
C *** PREDICTION HORIZON
C      NHOR=24
C *** KALMAN MODEL ORDER
C      NS=1
C
C *** STATE TRANSITION MATRIX (CONSTANT)
C      PHIM(1,1)=1.0D0
C
C *** OBSERVATION MATRIX (CONSTANT)
C      HM(1)=A1
C
C *** DRIVING NOISE COVARIANCE (CONSTANT)
C      QM(1,1)=9000.0D0
C
C *** MEASUREMENT NOISE COVARIANCE (CONSTANT)
C      RM=GN**2
C
C *** NONADAPTIVE
C      XSTATE(NS+1)=-1.0D0

```

A.7. PREM3_2.CMN

```

C Include PREM3_2: Used in PREDICTA.F. Performs 5 min load predictions
C                   over 2 hours using a 3rd-order kalman filter. Version 2
C                   is in the file PREM3_2.CMN (the 2 at the end of PREM3_2
C                   references the version #).
C
C The input is the current system load and the corresponding forecast. The
C output are improved estimates of future forecasts over the next 2 hours.
C
C The 3rd-order Kalman filter model is
C
C
C      A1*(s+2pi3e-4)(s+2pi5e-5)
C      Y(s) = ----- W(s) + GN*V(s)
C      (s+2*pi*7e-4)(s+2pi1.5e-5)(s+2pi2e-5)
C
C where w(t) and v(t) are normal uncorrelated unity-variance white noise.
C Note: The time base for this model is seconds.
C It is assumed this model is driven by white noise sampled at T=300sec.
C thru a Zero-Order-Hold (ZOH). The discrete-time model is:
C
C      x(k) = PHIM*x(k-1) + B*w(k)
C      z(k) = HM*x(k) + GN*v(k)
C      QM = B*B
C      RM = GN*GN
C z(k) is the load-forecast.
C
C
C Author: D. Trudnowski
C Dates: May 1996.
C References: D. Trudnowski, J. Johnson, P. Whitney, and M. Donnelly,
C             "Power System Very Short-Term Load Prediction," PNNL report
C             to be published, 1996.

```

```

C          D. Trudnowski, Engineering notebook: BPA Task Order PNL13, notebook 1.
C          R. Brown, P. Hwang, "Introduction to Random Signals and
C          Applied Kalman Filtering," Chap. 5, John Wiley & Sons, 1992.
C
C *** MODEL PARAMETERS
C          A1=0.4D0
C          GN=1.0D0
C *** SAMPLE PERIOD (300 sec)
C          DELTAT=300
C *** PREDICTION HORIZON
C          NHOR=24
C *** KALMAN MODEL ORDER
C          NS=3
C
C *** STATE TRANSITION MATRIX (CONSTANT)
C          DO I=1,3
C              DO J=1,3
C                  PHIM(I,J)=0.0D0
C              ENDDO
C          ENDDO
C          PHIM(1,1)=2.20240150903853D0
C          PHIM(2,1)=-1.45337034876435D0
C          PHIM(3,1)=0.25021309098570D0
C          PHIM(1,2)=1.0D0
C          PHIM(2,3)=1.0D0
C
C *** OBSERVATION MATRIX (CONSTANT)
C          HM(1)=A1
C          HM(2)=0.0D0
C          HM(3)=0.0D0
C
C *** DRIVING NOISE COVARIANCE (CONSTANT)
C          QM(1,1)=2.49411832429058D0
C          QM(1,2)=-3.71545510448621D0
C          QM(1,3)=1.31569418054748D0
C          QM(2,1)=QM(1,2)
C          QM(2,2)=5.53486436429560D0
C          QM(2,3)=-1.95997223205053D0
C          QM(3,1)=QM(1,3)
C          QM(3,2)=QM(2,3)
C          QM(3,3)=0.69405334938104D0
C
C *** MEASUREMENT NOISE COVARIANCE (CONSTANT)
C          RM=GN**2
C
C *** NONADAPTIVE
C          XSTATE(NS+1)=-1.0D0

```

A.8. DFIL2.F

```

C Compiler options for BPA
C          OPTIONS/check=bounds/F77/EXTEND_SOURCE/I4
C
C Subroutine DFILTER2 : ARMA filters data. Allows access to
C                      initial and final filter states. Version 2
C                      is in the file DFIL2.F (the 2 at the end
C                      of DFIL2 references the version #).
C This is a revised version of the following RCS version
C $Id: dfilter2.f,v 1.1 1994/09/01 23:02:05 ng622 Exp $
C This is VERSION 2.
C
C Version 1.1: Original
C Version 2 : Changed indexing on X and Y from (0:*) to (*) --
C             this was done by BPA. Added comments.
C
C Date : August 16, 1994 (Version 1.1)
C       Feb. 1996 (Version 2)
C
C This subroutine implements the ARMA filter B/A as a transposed
C direct form II structure.
C See A. V. Oppenheim and R. W. Schaffer, DISCRETE TIME SIGNAL
C PROCESSING, p. 311. Englewood Cliffs, NJ: Prentice-Hall, 1989.
C
C Inputs:

```



```

C   A   : AR (denominator) coefficients, in descending powers of z.
C   B   : MA (numerator) coefficients, in descending powers of z.
C   N   : Length of A and B. Filter order is N-1.
C   Z   : Initial state of filter. Replaced with final state upon
C         return from subroutine. Length is N-1.
C   X   : Data sequence to be filtered.
C   M   : Length of X. Number of data points.

```

```

C Outputs:

```

```

C   Y   : Filtered data sequence. Also has length M.
C   Z   : Final state of filter. Has length N-1.

```

```

C By Jeff M. Johnson, Battelle PNL (Version 1.1)
C   D. Trudnowski, Mt. Tech (Version 2)

```

```

      SUBROUTINE DFILTER2(B,A,N,Z,X,M,Y)

```

```

      IMPLICIT NONE

```

```

      INTEGER * 4 N,M,I,J

```

```

      REAL*8 B(*),A(*),Z(*),X(*),Y(*)

```

```

      CHARACTER*80 RCSID

```

```

C

```

```

C *** RCS IDENTIFICATION MARKER

```

```

      RCSID='SRCsfile: dfilter2.f,v $ $Revision: 1.1 $'

```

```

      DO I=1,M

```

```

        Y(I)=B(1)*X(I)+Z(1)

```

```

        DO J=2,N-1

```

```

          Z(J-1)=B(J)*X(I)+Z(J)-A(J)*Y(I)

```

```

        END DO

```

```

        Z(N-1)=B(N)*X(I)-A(N)*Y(I)

```

```

      END DO

```

```

      RETURN

```

```

      END

```

A.9. DFLSEG2.F

```

C This is a compiling option used by BPA

```

```

C   OPTIONS/check=bounds/F77/EXTEND_SOURCE/I4

```

```

C Subroutine DFILTERSEG : Zero-phase filtering. Version 2

```

```

C   is in the file DFLSEG2.F (the 2 at the

```

```

C   end of DFLSEG2 references the version #).

```

```

C This is a revised version of the following RCS version

```

```

C $Id: dfilterseg1.f,v 1.1 1994/09/01 22:56:58 ng622 Exp $

```

```

C This is VERSION 2.

```

```

C Version 1.1: Original

```

```

C Version 2 : Changed indexing from on XDAT and YDAT from (0:*) to
C   (*) -- this was done by BPA. Added comments.

```

```

C Dates: August 24, 1994 (Version 1.1)

```

```

C   Feb. 1996 (Version 2)

```

```

C This subroutine zero-phase filters NWIN points from a data
C vector and returns the last NSEG points from the result.

```

```

C It the input vector length is greater than NWIN, an error

```

```

C is returned. If the input vector length is less than NWIN,

```

```

C the first NWIN-(vector length) data points are extrapolated

```

```

C to give a vector with length NWIN. The ratio of NWIN to NSEG

```

```

C is always a positive integer.

```

```

C Inputs :

```

```

C NSEG : Number of points to return.

```

```

C N : Ratio of NWIN to NSEG.

```

```

C XDAT : Input data vector. XDAT(1) corresponds to time t,

```

```

C   XDAT(2) corresponds to time t+1, ..., XDAT(NDAT)

```

```

C   corresponds to time t+NDAT-1

```

```

C NDAT : Length of XDAT.

```

```

C Outputs:

```

```

C YDAT : Length NSEG filtered sequence. YDAT(1) corresponds

```

```

      A.11

```

```

C      to time t+NDAT-NSEG, YDAT(2) corresponds to time
C      t+NWIN-NSEG+1, ..., YDAT(NSEG) corresponds to time
C      t+NDAT-1.
C MRET  : Fatal error flag.
C      If == 0, No errors occurred.
C      If >= 1, Error message is described in RERR.
C RERR  : Character array with fatal error message.
C
C Subroutines called:
C DZERFILT1 : Zero phase filtering routine.
C
C Authors :
C   Dan Trudnowski, Battelle PNL (until Aug. 1995), Mt. Tech.
C   Jeff M. Johnson, Battelle PNL.
C
      SUBROUTINE DFILTERSEG1(NSEG,N,XDAT,NDAT,YDAT,MRET,RERR)

      IMPLICIT NONE
C INCLUDE file contains dimension parameters NSIZE and NTAILLE
C BPA's include
C   INCLUDE 'RODS_INCLUDE:[AGC]FILTER_INCLUDE.FOR'
C Tech's include
C   INCLUDE 'FILTER.CMN'

      INTEGER * 4 NSEG,N,NDAT,MRET
      INTEGER * 4 NWIN,NFILT,M1,NF,MR,MF,I,J
      REAL*8 XDAT(*),YDAT(*)
      REAL*8 B(NSIZE),A(NSIZE),WORKRV1(NTAILLE)
      CHARACTER*200 RERR

C *** INITIALIZE THE FILTER COEFFICIENTS. THIS IS A THIRD ORDER
C ELLIPTIC FILTER IN SERIES WITH A SECOND ORDER BUTTERWORTH.
C THIS HAS A MAGNITUDE CHARACTERISTIC SIMILAR TO THAT OF A
C 36TH ORDER MOVING AVERAGE FILTER. THESE ARE THE TRANSFER
C FUNCTION COEFFICIENTS IN DESCENDING POWERS OF Z. TRANSFER
C FUNCTION IS B/A.
      DATA NFILT / 6 /
      DATA (B(I), I=1,6) / 1.322785871309235e-04,
& 1.388195594782153e-04,
& -2.449342572199881e-04,
& -2.449342572199773e-04,
& 1.388195594782215e-04,
& 1.322785871309190e-04 /
      DATA (A(I), I=1,6) / 1.000000000000000e+00,
& -4.591751152282864e+00,
& 8.461339032736070e+00,
& -7.820189834181015e+00,
& 3.624572385963726e+00,
& -6.739181044571383e-01 /

C *** CHECK INPUT ARGUMENTS FOR ERRORS
      NWIN=N*NSEG
      MRET=0
      IF (NDAT.GT.NWIN.OR.NDAT.GT.NTAILLE) THEN
        MRET=1
C      WRITE(RERR,*) 'DFILTERSEG ERROR, INPUT DATA VECTOR TOO LONG.'
        GOTO 1000
      END IF

      IF (NSEG.LT.1) THEN
        MRET=2
C      WRITE(RERR,*) 'DFILTERSEG ERROR, SEGMENT LENGTH TOO SMALL.'
      END IF

      IF (NDAT.LT.NSEG) THEN
        MRET=3
C      WRITE(RERR,*) 'DFILTERSEG ERROR, INPUT DATA VECTOR TOO SHORT.'
        GOTO 1000
      END IF

C *** COPY DATA SEQUENCE INTO WORKING VECTOR
      J=NWIN
      DO I=NDAT,1,-1
        WORKRV1(J)=XDAT(I)
        J=J-1

```

```

      END DO
C
C *** IF NDAT IS LESS THAN NWIN, EXTRAPOLATE TO OBTAIN
C A VECTOR WITH LENGTH EQUAL TO NWIN.
      IF (NDAT.LT.NWIN) THEN
        M1=NWIN-NDAT
        NF=M1/(NDAT-1)
        MR=M1+NF-NDAT*NF
        DO I=NF,1,-1
          MF=MR+I*(NDAT-1)+1
          DO J=1,NDAT-1
            WORKRV1(MF-J)=2.D0*WORKRV1(MF)-WORKRV1(MF+J)
          END DO
        END DO

        DO J=1,MR
          WORKRV1(MR-J+1)=2.D0*WORKRV1(MR+1)-WORKRV1(MR+J+1)
        END DO
      END IF
C
C *** FILTER SEQUENCE
      CALL DZERFILT1(B,A,NFILT,WORKRV1,NWIN,YDAT,MRET,RERR)
      IF (MRET.NE.0) GOTO 1000
C
C *** COPY LAST NSEG POINTS TO BEGINNING OF YDAT
      DO I=1,NSEG
        YDAT(I)=YDAT(NWIN-NSEG+I)
      END DO

1000 RETURN
      END

```

A.10. DZFL2.F

```

C Subroutine DZERFILT1 : Zero-phase digital filtering. Version 2
C                       is in the file DZFL2.F. (the 2 at the end
C                       of DZFL2 references the version #).
C
C This is a revised version of the following RCS version
C $Id: dzerfilt1.f,v 1.1 1994/09/01 22:59:49 ng622 Exp $
C This is VERSION 2.
C
C Version 1.1: Original
C Version 2 : Changed indexing on X and Y from (0:*) to (*). Added
C             comments.
C
C Date : August 17, 1994 (Version 1.1).
C       Feb. 1996, (Version 2).
C
C This subroutine zero-phase filters data with the ARMA filter B/A.
C After filtering once in the forward direction, the filtered sequence
C is reversed and sent through the filter again. The resulting sequence
C has zero-phase distortion. Care is taken to minimize startup transients.
C
C An error is returned if the number of data points is not at least three
C times the filter order.
C
C See A. V. Oppenheim and R. W. Schaffer, DISCRETE TIME SIGNAL
C PROCESSING, p. 284. Englewood Cliffs, NJ: Prentice-Hall, 1989).
C Also see the Matlab Signal Processing Toolbox users guide.
C
C Inputs :
C   A : AR (denominator) filter coefficients, in descending powers
C       of z.
C   B : MA (numerator) filter coefficients, in descending powers of
C       z.
C   N : Length of A and B. Filter order is N-1.
C   X : Data vector to be filtered.
C   M : Length of X. Number of data points.
C
C Outputs:
C   Y : Filtered data sequence. Has length M.
C   MRET : Error flag.
C         If = 0, No errors occurred.

```

```

C      If = 1, Error message described in RERR
C      RERR : CHARACTER*200 array with error message if MRET=1.
C
C Subroutines called:
C      DFILTER2 : ARMA filters data.
C
C By Jeff M. Johnson, Battelle PNL.
C
      SUBROUTINE DZERFILT1(B,A,N,X,M,Y,MRET,RERR)

      IMPLICIT NONE
      INCLUDE 'FILTER.CMN'
      INTEGER I,N,M,MRET,NFACT,MM
      REAL*8 B(1:*),A(1:*),X(*),Y(*)
      REAL*8 Z(NSIZE),ZI(NSIZE)
      REAL*8 WORKRV1(NTAILLE+2*NSIZE),WORKRV2(NTAILLE+2*NSIZE)
      CHARACTER*200 RERR
      CHARACTER*80 RCSID

C
C *** RCS IDENTIFICATION MARKER
C      RCSID='$RCSfile: dzerfilt1.f,v $ $Revision: 1.1 $'

      NFACT=3*(N-1)
      MRET=0
      IF (M.LE.NFACT) THEN
        MRET=1
        WRITE(RERR,*) 'DZERFILT1 ERROR. DATA SEQUENCE MUST ',
& 'HAVE LENGTH AT LEAST THREE TIMES FILTER ',
& 'ORDER.'
        GOTO 1000
      END IF

C *** Set up filter's initial state to remove DC offset problems
C      at beginning and end of data sequence. See notes in journal
C      for August 19, 1994.
      WORKRV1(N-1)=A(N)
      WORKRV2(N-1)=B(N)
      DO I=N-2,1,-1
        WORKRV1(I)=WORKRV1(I+1)+A(I+1)
        WORKRV2(I)=WORKRV2(I+1)+B(I+1)
      END DO
      Z(1)=(WORKRV2(1)-B(1)*WORKRV1(1))/(1.D0+WORKRV1(1))
      DO I=2,N-1
        Z(I)=WORKRV2(I)-(B(1)+Z(1))*WORKRV1(I)
      END DO

C *** Extrapolate beginning and end of data sequences using reflection
C      method. Slopes of original and extrapolated sequences match at
C      end points. This reduces end effects.
      DO I=1,NFACT
        WORKRV1(I)=2*X(1)-X(NFACT-I+2)
        WORKRV1(NFACT+M+I)=2*X(M)-X(M-I)
      END DO
      DO I=1,M
        WORKRV1(NFACT+I)=X(I)
      END DO

C *** Filter once, reverse data, filter again.
100  MM=M+2*NFACT
      DO I=1,N-1
        ZI(I)=WORKRV1(1)*Z(I)
      END DO
      CALL DFILTER2(B,A,N,ZI,WORKRV1,MM,WORKRV2)
      DO I=1,MM
        WORKRV1(I)=WORKRV2(MM-I+1)
      END DO
      DO I=1,N-1
        ZI(I)=WORKRV1(1)*Z(I)
      END DO
      CALL DFILTER2(B,A,N,ZI,WORKRV1,MM,WORKRV2)

C *** Remove extrapolated data and reverse data.
      DO I=0,M-1
        Y(I+1)=WORKRV2(NFACT+M-I)
      END DO

```

1000 RETURN
END

A.11. FILTER.CMN

INTEGER * 4
1 NSIZE,
1 NTAILLE

*

PARAMETER (NSIZE=128, NTAILLE=1024)

Distribution

No. of
Copies

No. of
Copies

OFFSITE

ONSITE

5 Mr. Warren McReynolds
 Bonneville Power Adm.
 PO Box 3621, MS TOON
 Portland, OR 97208-3621

 Mr. William Middelstadt
 Transmission Planning - TEO
 Bonneville Power Adm.
 PO Box 3621
 Portland, OR 97208-3621

 Dr. Daniel Trudnowski
 Trudnowski Consulting
 PO Box 4651
 Butte, MT 59702

2 U.S. DOE
 Office of Scientific &
 Technical Information
 PO Box 62
 Oak Ridge, TN 37831

11 Pacific Northwest National
 Laboratory

 M. Donnelly (K5-20)
 J. Hauer (K5-20)
 J.M. Johnson (K5-20)
 P. Whitney (K5-12)
 Information Release
 Office (5) (K1-11)
 Project Files (2)