JANUARY 1982
VOL. 61, NO. 1

THE BELL SYSTEM TECHNICAL JOURNAL

THE BELL SYSTEM TECHNICAL JOURNAL

THE BELL SYSTEM TECHNICAL JOURNAL is published monthly, except for the May-June and July-August combined issues, by the American Telephone and Telegraph Company, C. L. Brown, Chairman and Chief Executive Officer; W. M. Ellinghaus, President; V. A. Dwyer, Vice President and Treasurer; F. A. Hutson, Jr., Secretary. Editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room WB 1L-336, Crawfords Corner Road, Holmdel, N.J. 07733. Checks for subscriptions should be made payable to The Bell System Technical Journal and should be addressed to Bell Laboratories, Circulation Group, 101 J. F. Kennedy Parkway, Short Hills, N.J. 07078. Subscriptions $20.00 per year; single copies $2.00 each. Foreign postage $1.00 per year; 15 cents per copy. Printed in U.S.A. Second-class postage paid at New Providence, New Jersey 07974 and additional mailing offices.

Comments on the technical content of any article or brief are welcome. These and other editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room WB 1L-336, Crawfords Corner Road, Holmdel, N.J. 07733. Comments and inquiries, whether or not published, shall not be regarded as confidential or otherwise restricted in use and will become the property of the American Telephone and Telegraph Company. Comments selected for publication may be edited for brevity, subject to author approval.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 61 January 1982 Number 1

Kalman Filter Models for Network Forecasting

By C. D. PACK and B. A. WHITAKER

(Manuscript received December 31, 1980)

The Bell System has recently completed studies that are expected to result in substantially improved forecasts for use in network planning. These improved forecasts are achieved through the use of new forecasting algorithms that employ Kalman filter models. To motivate the selection of Kalman filter forecasting procedures, we describe the Bell System's special data characteristics and processing requirements in the network planning process. We also discuss the Kalman filter models, their statistical properties, the model identification process, and certain implementation considerations.

I. INTRODUCTION

Projections of message circuit usage (as measured in hundred call seconds or CCS), from which message circuit requirements (trunks) are developed, and special services* circuit demand are fundamental parts of the Bell System's network planning and provisioning process. An overview of the information flows in this process is shown in Fig. 1; more details are given in Refs. 1 to 4 and in the companion forecasting papers in this issue.

Since these projections strongly influence the allocations of several billions of construction dollars annually, it is important that they possess high quality statistical properties. For example, the projections should be unbiased; that is, the forecasts should not be consistently

* *Special services* is a generic term referring to all Bell System services other than ordinary message telephone service. Examples include foreign exchange lines, tie lines, and private lines.

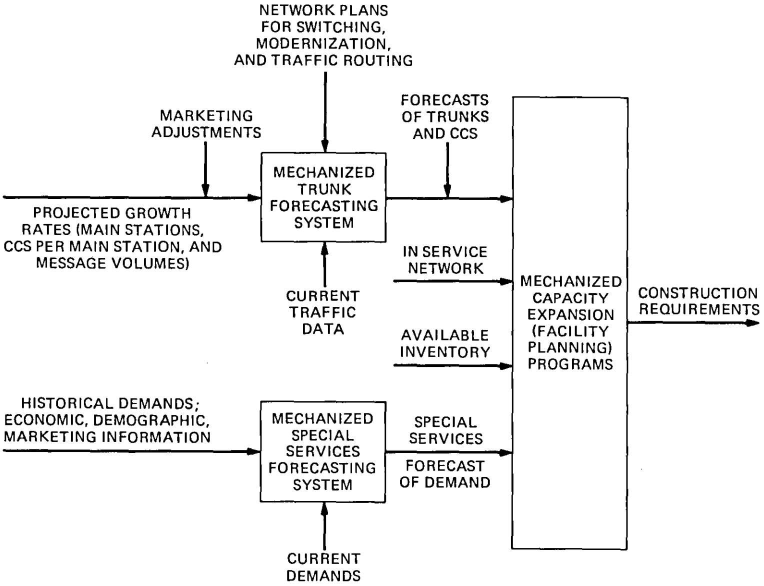
NETWORK PLANS FOR SWITCHING, MODERNIZATION, AND TRAFFIC ROUTING
MARKETING ADJUSTMENTS
FORECASTS OF TRUNKS AND CCS
MECHANIZED TRUNK FORECASTING SYSTEM
PROJECTED GROWTH RATES (MAIN STATIONS, CCS PER MAIN STATION, AND MESSAGE VOLUMES)
IN SERVICE NETWORK
CURRENT TRAFFIC DATA
AVAILABLE INVENTORY
MECHANIZED CAPACITY EXPANSION (FACILITY PLANNING) PROGRAMS
CONSTRUCTION REQUIREMENTS
HISTORICAL DEMANDS; ECONOMIC, DEMOGRAPHIC, MARKETING INFORMATION
MECHANIZED SPECIAL SERVICES FORECASTING SYSTEM
SPECIAL SERVICES
FORECAST OF DEMAND
CURRENT DEMANDS

Fig. 1—Network planning and provisioning process.

high or low. A biased forecast will result in either over expenditures or equipment shortages, depending on the sign of the bias. Also, the forecasts should be stable, or precise, never varying too much from the realized true requirements. Previous studies have shown that highly variable forecasts result in increased reserve capacity requirements necessary to meet customer demands.[5]

The service and economic motivations for high quality forecasts of message trunks and special services circuits have led the Bell System to reevaluate the existing projection algorithms used in these processes and to recommend improved methods, as necessary. It should be noted that a further motivation for this reevaluation is that most existing projection methods utilize curve fitting and extrapolation algorithms that were originally designed for manual calculations and that were available prior to the widespread use of computers and the advent of modern estimation techniques.[1,2]

The possible approaches to improved projection methods were influenced strongly by the particular characteristics of the time series and by the typical mode in which the forecasts are produced.

Two factors were crucial in the selection of forecasting models and algorithms: the need to produce a large number of forecasts in a fully mechanized system and the typical dearth of data for any individual series.

A large number of forecasts is necessary because trunk and special services circuit requirements must be forecast at least once per year for each trunk group and special services circuit group on record. Thus, there may be up to 100,000 trunk group time series and tens of thousands of special services circuit time series forecast in each run of the mechanized forecasting systems illustrated in Fig. 1. Therefore, these procedures must be fully mechanized. Moreover, for ease of comprehension and for computational efficiency, the algorithms must be simple to program and maintain.

The sparsity of data is perhaps the more restrictive of the two key considerations in selecting new projection algorithms. Typically, the individual time series have between 1 and 10 years of relevant data, with 3 to 5 years being common. The fact that a projection may have to be based on less than 5 years of data (up to 60 points if monthly data were available), make the use of many approaches, including the popular Box-Jenkins ARIMA (autoregressive integrated moving average) models[6] that require more than 100 data points, infeasible.

The linear Kalman filter, in its most general form, was derived by Kalman[7] and Kalman and Bucy[8] in the early 1960s. The motivation for their work had its roots in various control theory applications. Of particular note was the implementation of the algorithms in tracking systems for the aerospace industry. In recent years, the models have also been used in power systems, process control, and forecasting.

In Section III, we describe the general Kalman filter linear model, analyze and interpret the key matrices of the formulation, and relate the Kalman filter to other common forecasting models. In Section IV, we discuss various implementation considerations that are necessary to reduce the Kalman filter algorithm to practice. In Section V, we indicate methods for evaluating the effectiveness of the Kalman filter models and discuss the importance of certain key statistics. Finally, in Section VI, we summarize our conclusions concerning the use of Kalman filters for forecasting in the Bell System.

Three companion papers to this overview follow. These papers describe the successful use of Kalman filter models for three Bell System forecasting applications—two concern message trunk forecasting and one covers special services demand forecasting.

The first trunk forecasting paper, by J. P. Moreland, describes a simple linear, two-state model for use in projecting busy season (yearly peak) trunk group loads.[9] The second paper, by A. Ionescu-Graff, describes a linear, two-state Kalman filter with an absorbing barrier that can improve the quality of special services demand forecasts.[11] The third paper, by C. R. Szelag, derives a Kalman filter for traffic that has not only linear growth, but also a seasonal within-year pattern as well.[10] That is, Szelag illustrates how, for trunk group load patterns exhibiting seasonality, nonbusy season data can be used to update and improve estimates of imminent busy season loads.

II. KALMAN FILTER DESCRIPTION

The Kalman filter, described in more detail in Section III, has many desirable properties. Most of these properties are not unique to the Kalman filter; however, because of its generality and particular form, as well as statistical properties, computational characteristics, and robust qualities it should be considered in most estimation applications.

2.1 Models

The models used are based on state-space representations of the variables being estimated. The state-space formulation implies that, at each point in time, the process being modeled is described by a vector of state variables that summarize all relevant quantities of interest. In most instances, the state variables have physical interpretations, such as trunk quantities, growth rates, and so on. Then, a further characterization of the model specifies how this state vector evolves over time.

The Kalman filter algorithm uses this model of the time behavior of the system along with "noisy" observations or measurements of some system variables to produce optimal estimates of all state variables. These estimates are then used in the process model to determine state estimates for future time periods.

The distinction between the state-space models and the ARIMA models is mostly in the model identification process. That is, while the interpretation of the state-space models permits the user to choose a model based on physical properties (and hence when only limited data are available), the "time series" approach of Box and Jenkins attempts to have the data specify the model based on certain first- and second-order characteristics. However, after a state-space time-series model is known, one can find nearly equivalent representations of that model for either theory.

The particular state-space formulation of Kalman has some desirable features. First, it lends itself to simple, recursive estimation of the parameters. That is, no data history need be stored. As new data or observations become available, they are processed and the stored state vectors are updated accordingly. In fact, this recursive calculation suggests a Markovian property of the filter: the current state vector summarizes all relevant historical information concerning the history of the time series. Second, there is a provision for separate characterization of the two sources of significant estimation errors: the dynamics of the true process and relationship between the state variables and the measurements used to estimate these variables. Third, the model provides an analytic framework for studying relationships among the first- and second-order properties of the state variables and measurements. For example, one can derive analytic expressions for forecast variances as functions of the number of data points processed and the autocorrelation matrix of the measurement errors. Finally, the model is general enough to include as special cases the common models: exponential smoothing, weighted least squares, multiple linear regression, and Wiener filtering.

2.2 *Statistical properties*

A correctly specified linear Kalman filter produces forecasts that have minimum mean square error.* Moreover, the forecasts are unbiased and have minimum variance. When the errors are Gaussian, these properties hold without restriction to the class of linear models. The estimators can also be derived using maximum likelihood or Bayes models. When the models are only approximately correct, the generality of the formulation allows one to analyze the filter's suboptimal performance and, if desired, to adjust the filter's parameters as appropriate.[12,13]

III. DISCRETE-TIME LINEAR KALMAN FILTER MODEL

3.1 *The model*

It is assumed that the true process dynamics are described by the following linear transition equation

$$\mathbf{X}_{n+1} = \phi\mathbf{X}_n + \mathbf{U}_n + \omega_n\,, \tag{1}$$

where

$\mathbf{X}_n$ = an s-vector of state variables in period n,

ϕ = an $s \times s$ transition matrix that may, in general, depend on n,

ω_n = an s-vector of random modeling errors, i.e., random deviations of the true process from the assumed linear relation defined by ϕ, and

$\mathbf{U}_n$ = an s-vector of deterministic changes in state.

The one-step projection formula is given by

$$\hat{\mathbf{X}}_{n+1,n} = \phi\hat{\mathbf{X}}_{n,n} + \mathbf{U}_n\,, \tag{2}$$

where, in general, $\hat{\mathbf{X}}_{n+k,n}$ is an estimate of $\mathbf{X}_{n+k}$ ($k \geq 0$) given data $\mathbf{y}_1, \cdots, \mathbf{y}_n$ in periods 1 through n, where $\mathbf{y}_n$ is a d-vector of observed variables in period n.

The relations that distinguish the Kalman filter model and associated computational procedures from other linear estimation techniques are the particular model relating $\mathbf{X}_n$ to $\mathbf{y}_n$ and the algorithm for

* These terms and others are defined in Section V.

computing $\hat{\mathbf{X}}_{n,n}$. The $d \times s$ matrix H, which in general may depend on n, defines the relationship between $\mathbf{y}_n$ and $\mathbf{X}_n$ as

$$\mathbf{y}_n = H\mathbf{X}_n + \nu_n , \tag{3}$$

where ν_n is a d-vector of measurement errors. At time n, the vector $\hat{\mathbf{X}}_{n,n}$ is computed by

$$\hat{\mathbf{X}}_{n,n} = \hat{\mathbf{X}}_{n,n-1} + K_n(\mathbf{y}_n - H\hat{\mathbf{X}}_{n,n-1}). \tag{4}$$

The $s \times d$ "Kalman gain" matrix K_n can be calculated recursively by the following equations:

$$\begin{aligned} K_n &= P_nH^T(HP_nH^T + R)^{-1} \\ S_n &= (I - K_nH)P_n \\ P_{n+1} &= \phi S_n\phi^T + Q , \end{aligned} \tag{5}$$

where

(*i*) R is the covariance matrix of the measurement errors, i.e., $R \equiv E(\nu_n\nu_n^T)$,

(*ii*) Q is the covariance matrix of the modeling errors, i.e., $Q \equiv E(\omega_n\omega_n^T)$, and

(*iii*) it is assumed that $E(\nu_n) = E(\omega_n) = 0$ for all n, $E(\omega_n\nu_i^T) = 0$ for all (n, i), $E(\omega_n\omega_i^T) = 0$ for all $n \neq i$, and $E(\nu_n\nu_i^T) = 0$ for $n \neq i$, and

(*iv*) in general, Q and R may depend on n.

Thus, in summary, the forecasting procedure has the following steps:

1. When $n = 0$, the filter is initialized by a user-supplied estimate $\hat{\mathbf{X}}_{0,0}$ of the initial state vector $\mathbf{X}_0$ and S_0.
2. Using these estimates, a one-period-ahead forecast is produced using eq. (2).
3. The gain matrix, K_n, and the matrices S_n and P_{n+1} are calculated.
4. When a new observation is received, eq. (4) is used to obtain a "smoothed" estimate $\hat{\mathbf{X}}_{n,n}$ of the present state vector.
5. Then using this new estimate, a one-period-ahead forecast is produced. The period index n is incremented and Step 3 is repeated.

The algorithm continues to process new observations and produce forecasts in this manner.

Several points should be noted:

1. If the matrices H, R, ϕ, and Q are independent of n, the gain matrix, K_n, and the matrices S and P are independent of the observations. Thus, these matrices can be precalculated for use in the algorithm.
2. No past observations must be stored since all historical information is contained in the "smoothed" estimate $\hat{\mathbf{X}}_{n,n}$ or, equivalently via eq. (2), the one-period-ahead projection $\hat{\mathbf{X}}_{n+1,n}$.

3. Only the one-period-ahead forecast must be saved to be used in the next period's process.

When the assumptions listed above in (*iii*) are valid and the models accurately describe the true process dynamics and the measurement system, the Kalman filter produces unbiased estimates of $\hat{\mathbf{X}}_{n+k,n}$; that is, $E(\hat{\mathbf{X}}_{n+k,n}) = \mathbf{X}_{n+k}$ for $k \geq 0$. In addition, the estimates $\hat{\mathbf{X}}_{n+k,n}$ have minimum variance in the class of all unbiased estimators.[13] If ω_n and ν_n are Gaussian, then no restriction to the class of linear estimates is required and the estimators can also be derived from maximum likelihood and Bayesian models.[13] Conveniently, the Kalman formulation actually provides estimators for the estimation error variance matrices of interest:
(smooth)

$$S_n \equiv E(\hat{\mathbf{X}}_{n,n} - \mathbf{X}_n)(\hat{\mathbf{X}}_{n,n} - \mathbf{X}_n)^T, \tag{6}$$

and, for $k \geq 1$,
(predict)

$$P_{n+k} = E(\hat{\mathbf{X}}_{n+k,n} - \mathbf{X}_{n+k})(\hat{\mathbf{X}}_{n+k,n} - \mathbf{X}_{n+k})^T. \tag{7}$$

Therefore, the requirement that the user specify an initial estimate of S_0 is a need for an estimate of the covariance matrix of the initial state vector $\hat{\mathbf{X}}_{0,0}$.

Analyses of the sensitivity of the filter performance to model accuracy and of the modified estimators of error covariance matrices for the case of suboptimal gains K' is given in Ref. 13.

3.2 Interpretation

From eq. (4), we see that the vector $\hat{\mathbf{X}}_{n,n}$, the smoothed estimate of $\mathbf{X}_n$, is derived as the previous one-step projection $\hat{\mathbf{X}}_{n,n-1}$, plus a linear combination (weighting) of the differences between the measurement $\mathbf{y}_n$ and the previous estimate (forecast) of these measurements, $H\hat{\mathbf{X}}_{n,n-1}$. The weights assigned to the difference terms are appropriate components of the gain matrix K_n. It is also important to note that $\hat{\mathbf{X}}_{n,n}$ depends on $\hat{\mathbf{X}}_{n,n-1}$, $\mathbf{y}_n$, K_n, and H, but not explicitly on $\mathbf{y}_1, \cdots, \mathbf{y}_{n-1}$. This recursive nature of the Kalman filter eliminates the need for storage of historical data.

We can give some insight into the effect K_n has on algorithm performance, without actually describing a specific model. It can be seen from eq. (5) that K_n has terms which are directly proportional to the elements of the covariance matrix Q* and inversely proportional

* As we indicate later in Section 3.4, this "proportionality" to Q is strongest for large n.

to the elements of R. That is, K_n is, in a sense, proportional to the variability of the true process dynamics and inversely proportional to the measurement variability. Thus, it is the "Q/R" relationship which defines the responsiveness of the filter (via the K matrix) to estimated errors in state ($\mathbf{y}_n - H\hat{\mathbf{X}}_{n,n-1}$). As the elements of K_n decrease (by decreasing Q or increasing R), the forecasts become more stable. That is, if we have more confidence in the unbiasedness of the process model or less confidence in our observations, the filter will be designed to respond more slowly to apparent deviations from the predicted trend line. As K_n increases (by increasing Q or decreasing R), forecasts detect and respond to data that deviate from the assumed deterministic linear model, represented by the expected value of eq. (1).

3.3 An example

Consider a simple linear two-state model ($s = 2$), where

$$\mathbf{X}_{n+1} = \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} \omega_n \\ \dot{\omega}_n \end{bmatrix} \tag{8}$$

and ($d = 1$)

$$y_n = x_n + \nu_n . \tag{9}$$

We further assume that $R \equiv 1$ (since the gain matrix depends only on Q/R, we do this with no loss in generality). To complete the model, the 2×2 matrix Q must be specified. This simple two-state model is fundamental to the models presented in the three companion papers. In these models, x_n is referred to as the level of the process, and $\dot{x}_n$, as the growth increment.

We now show how the various components of Q influence the response of the filter to a measurement y_n. First, note that eq. (4) can be rewritten as

$$\hat{\mathbf{X}}_{n,n} = \begin{bmatrix} \hat{x}_{n,n} \\ \hat{\dot{x}}_{n,n} \end{bmatrix} = \begin{bmatrix} \hat{x}_{n,n-1} + k_{11}^{(n)}(y_n - \hat{x}_{n,n-1}) \\ \hat{\dot{x}}_{n,n-1} + k_{21}^{(n)}(y_n - \hat{x}_{n,n-1}) \end{bmatrix}. \tag{10}$$

Hence, the smoothing process is determined by the specification of two sequences of number $k_{11}^{(1)}, \cdots, k_{11}^{(n)}, \cdots$ and $k_{21}^{(1)}, \cdots, k_{21}^{(n)}, \cdots$. As we indicated above, these gains tend to be proportional to the elements of Q.

Fig. 2 shows the filter operation and the role of the gain sequence $\{k_{11}^{(n)}\}$. Initially, attention should be focused on the trend line at time $n - 1$ derived from $n - 1$ pieces of data, $y_1, \cdots, y_{n-1}$ and note that $\hat{x}_{n-1,n-1}$ lies on that line. Further, $\hat{x}_{n,n-1}$ is a straight projection of this trend one-period ahead. When the new observation, y_n, is obtained, the slope of the trend line is adjusted upward in the direction of y_n

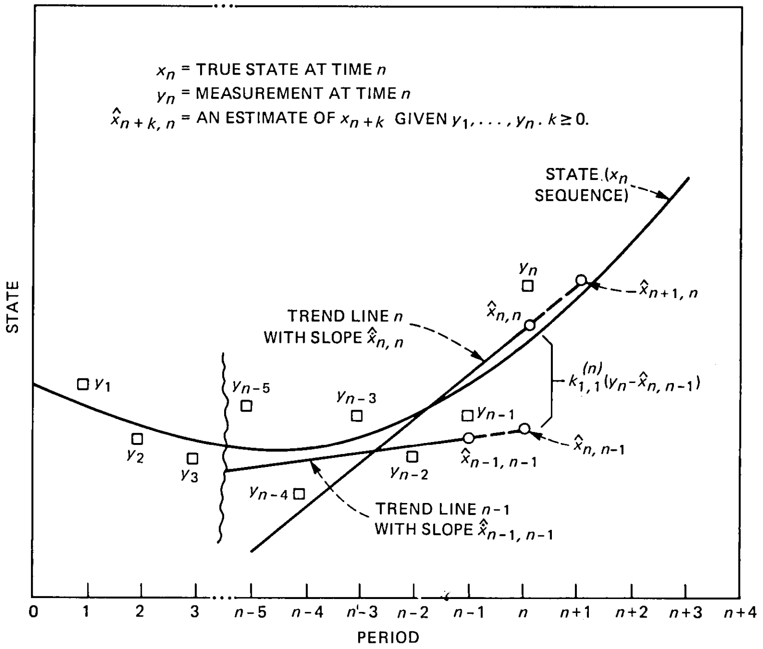
x_n = TRUE STATE AT TIME n
y_n = MEASUREMENT AT TIME n
$\hat{x}_{n+k,n}$ = AN ESTIMATE OF x_{n+k} GIVEN $y_1, \ldots, y_n$. $k \geq 0$.
STATE (x_n SEQUENCE)
y_n
$\hat{x}_{n+1,n}$
$\hat{x}_{n,n}$
TREND LINE n WITH SLOPE $\hat{x}_{n,n}$
$k_{1,1}^{(n)}(y_n - \hat{x}_{n,n-1})$
y_1
y_2
y_3
y_{n-5}
y_{n-3}
y_{n-1}
$\hat{x}_{n,n-1}$
$\hat{x}_{n-1,n-1}$
y_{n-2}
y_{n-4}
TREND LINE $n-1$ WITH SLOPE $\hat{x}_{n-1,n-1}$
STATE
0
1
2
3
$n-5$
$n-4$
$n'-3$
$n-2$
$n-1$
n
$n+1$
$n+2$
$n+3$
$n+4$
PERIOD

Fig. 2—Projection using the Kalman filter.

relative to $\hat{x}_{n,n-1}$. The smoothed estimate $\hat{x}_{n,n}$ of x_n is $\hat{x}_{n,n-1}$, plus a factor $k_{11}^{(n)}$ times the increment $y_n - \hat{x}_{n,n-1}$. The new trend line passes through $\hat{x}_{n,n}$ and its extrapolation produces $\hat{x}_{n+1,n}$.

The choice of Q (and R) determines the $k_{11}^{(n)}$ and, hence, the response of the filter to observation y_n relative to $\hat{x}_{n,n-1}$. Note that the *smoothed* estimate $\hat{x}_{n,n}$ of x_n, *not* y_n, is projected into the future. Thus, we do not project all of the noise in y_n into the future. This is a fundamental distinction between Kalman filter projections and most existing load-projection algorithms used by the Bell System.

3.4 Estimating matrices

If all Kalman filter model matrices (ϕ, H, S_0, Q, and R) are known, the algorithm is completely specified; moreover, the desirable statistical properties of the filter are assured. However, in general, the true model is not known precisely, or it is likely that the model is to be applied to many different estimation problems. Hence, in practice, we often settle for less optimal (in a bias or variance sense), but more robust properties for our forecasts. That is, we choose to achieve a reasonable balance between bias and variability (stability) over a wide range of practical interest for key parameters while relaxing some of the assumptions in Section 3.1.

Usually, it is the case that the processes being modeled suggest a reasonable choice of matrices ϕ and H. (Examples are discussed in the companion papers in this issue.) Also, typically, the data characteristics can be analyzed so that R is known either analytically or empirically. Therefore, the main concern is the selection of matrices S_0 and Q. The former strongly influences the transient characteristics of the filter; that is, K_n is strongly dependent on "S_0/R" for n small (<10). The latter affects the steady-state performance; that is, K_n approaches "Q/R" for large n.

These matrices S_0 and Q can be determined analytically or empirically to provide the desired responsiveness and statistical properties over many time series. The key to the analytical approach is the matrix P_{n+1} in eq. (5), which describes forecast variance as a function of model matrices, when the assumptions of Section 3.1 are correct. Empirical studies attempt to tune filter performance so that a robust balance of unbiasedness and stability is achieved over the test cases. The performance trade-offs are illustrated generically in Fig. 3 as a function of K_n whose dependence on S_0 and Q was stated previously. The studies described in the companion papers illustrate these two approaches to filter design.

We have ignored the extensive literature (see, for example, Ref. 14) on model identification for Kalman filters, because, as for the Box-Jenkins technique, substantial data is required for each time series.

3.5 Special cases

The Kalman filter model includes as special cases many common estimation techniques. We indicate three such examples.

3.5.1 Multiple regression

The multiple regression approach to estimation assumes that a time series $\{y_n\}$ is well-approximated by

$$y_n = H_n^T\mathbf{X} + \nu_n\,, \tag{11}$$

where H_n is an s-vector of observations of the independent regression variables, $\mathbf{X}$ is an s-vector of (constant) regression weights, and ν_n is an error term. The Kalman filter model is obtained by allowing the regression coefficients to depend on n and by adding the dynamics relation

$$\mathbf{X}_{n+1} = \mathbf{X}_n\,. \tag{12}$$

Clearly, the model includes the autoregressive case (H_n is composed of previous realizations of y_n) and can be generalized to vectors $\mathbf{y}_n$.

3.5.2 Exponential smoothing

Exponential smoothing is a process whereby current estimates of

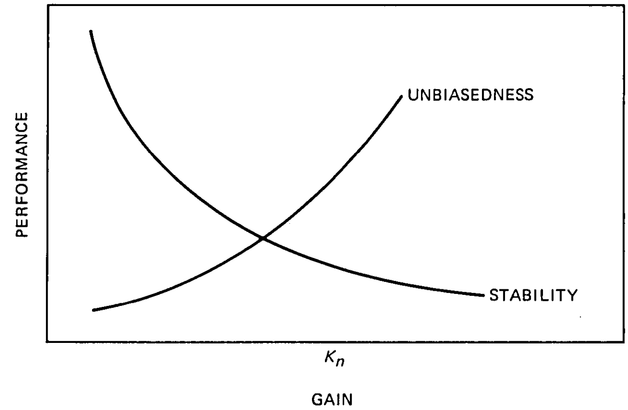

Fig. 3—Empirical performance trade-offs.

state $\hat{x}_{n,n}$ are composed of a weighted average of the previous estimate at state $(\hat{x}_{n-1})$ and the current measurement (y_n). That is,

$$\hat{x}_n = \hat{x}_{n-1} + k(y_n - \hat{x}_{n-1}). \tag{13}$$

Note that the weight k is a constant. The term exponential smoothing was chosen because the previous measurements influence current estimates by weights that decrease exponentially with lag.

Clearly, eq. (13) is of the form of eq. (4) and can be generalized to a Kalman filter by including eqs. (2) and (3), and the dependence of k on n.

3.5.3 Wiener filtering

Wiener filtering[15] corresponds to the case of constant Kalman gain matrix, i.e., when $K_n \equiv K$ for all n. If ϕ, H, R, and Q are constant, then the steady-state error covariance matrices exist and a Kalman filter with gains calculated using these matrices is identical to a Wiener filter.

IV. IMPLEMENTATION CONSIDERATIONS

Experience has shown that the Kalman algorithm, as described in Sections II and III, performs well, as long as the model is reasonable and the data $\mathbf{y}_n$ are consistent with the model assumptions. However, in practice, additional logic or considerations are necessary to improve the performance when outlier data are present or when certain types of nonstationarity are present and to improve the computational efficiency of the procedures.

An overview of a typical Kalman filter implementation is shown in Fig. 4. We will describe the components of the Kalman filter system in the following subsections.

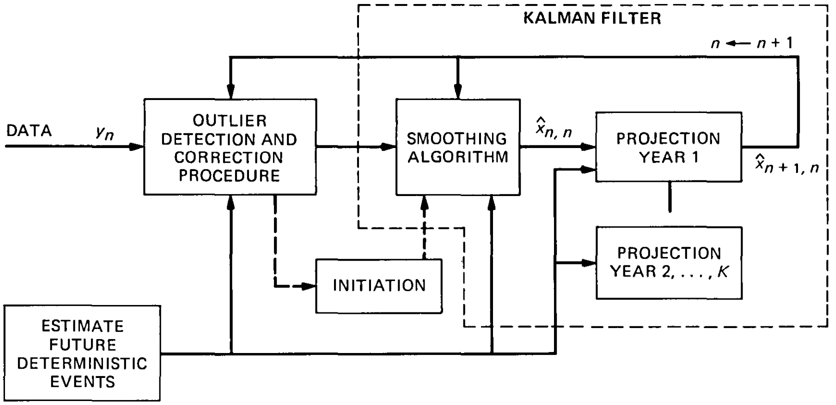

Fig. 4—Kalman filter implementation.

4.1 Data smoothing and projection

The Kalman filter algorithm, described in Section 3.1, comprises the smoothing and projection functions.

4.2 Initiation, transient response, and gain sequences

In some Kalman filter applications, the transient period is very short relative to the total time the filter is in operation. However, in the applications described in Refs. 9 to 11, a series rarely exists for more than 10 years, with 3 to 5 years being typical. Therefore, the transient response of the filter is important and $\hat{\mathbf{X}}_{0,0}$ and S_0 (and, hence, K_n for small n) must be carefully chosen to provide good performance during the transient period.

Surprisingly, good statistical performance in both the transient and equilibrium states is often achieved with fixed ($K_n = K$ for all n) or finite ($K = K_{n*}$ for $n \geq n^*$) gain sequences. Moreover, significant computational efficiencies are obtained when the precomputed finite gains are stored for on-line use. The companion papers describe the success of fixed and finite gain sequences for the respective applications.

4.3 Outlier detection, deterministic events

We define unusual data to be that representing either an unusually large (outlier) measurement error (ν_n) or an unforeseen deterministic event ($\mathbf{U}_n$). The importance of these two types of errors is that the smoothing process (4) tends to adjust the previous forecast $\hat{\mathbf{X}}_{n,n-1}$ in the direction of the new measurement $\mathbf{y}_n$, with the movement being proportional to the estimated error ($\mathbf{y}_n - H\hat{\mathbf{X}}_{n,n-1}$). Therefore, without additional logic of an adaptive nature, outliers would cause overreaction to measurements; unforeseen deterministic events would be insufficiently accounted for because the *full* (rather than smoothed) impact should have been entered in eq. (2).

However, in practice, one cannot always satisfy the two conflicting objectives—ignore bad measurements and react strongly to unforeseen deterministic events. Hence, either robust estimators must be employed or logic must be provided to identify and distinguish the two cases and to not overreact to either. The balance is again between bias and stability: over-response reduces bias but degrades stability, and vice versa.

Considerable discussion of this trade-off and the resulting filter design for one application is described in Ref. 9.

V. MODEL EVALUATION

In previous sections of this paper, we have alluded to various statistical criteria that may be used to evaluate the performance of the filter. Clearly, none is right or wrong unless it can be shown that some significant costs are directly and uniquely related to a particular criterion. We have never seen such justifications. However, it is usually the case that costs are related in some indirect fashion to various first- and second-order error statistics of the forecasts. The companion papers will each refer to some subset of the following statistics, possibly normalized to be stated as a percent:

Bias

$$A_{n+k,n} \equiv E(\hat{\mathbf{X}}_{n+k,n} - \mathbf{X}_n)$$

(In)stability (view-over-view variability)

$$S_{n+k,n} = E(\hat{\mathbf{X}}_{n+k,n} - \hat{\mathbf{X}}_{n+k,n-1})(\hat{\mathbf{X}}_{n+k,n} - \hat{\mathbf{X}}_{n+k,n-1})^T$$

(Im)precision or variance

$$P_{n+k,n} = E(\hat{\mathbf{X}}_{n+k,n} - E(\hat{\mathbf{X}}_{n+k,n}))(\hat{\mathbf{X}}_{n+k,n} - E(\hat{\mathbf{X}}_{n+k,n}))^T$$

Mean square error

$$M_{n+k,n} = E(\hat{\mathbf{X}}_{n+k,n} - \mathbf{X}_n)(\hat{\mathbf{X}}_{n+k,n} - \mathbf{X}_n)^T$$

Clearly, there exist some analytical relationships among these statistics. We will not derive any here; however, it is of interest to point out that, if $\hat{\mathbf{X}}_{n+k,n}$ and $\mathbf{X}_n$ are uncorrelated (a sufficient condition is that $Q \equiv 0$ or $k = 0$ for the linear model of Section III), then

$$M_{n+k,n} = A_{n+k,n}A^T_{n+k,n} + P_{n+k,n}$$

which, in the scalar case, reduces to

$$M_{n+k,n} = A_{n+k,n}^2 + P_{n+k,n} .$$

The rms error is commonly calculated as $(M_{n+k,n})^{1/2}$ for this last case.

VI. STATUS AND CONCLUSIONS

The Kalman filter model promises to provide forecasts, for use in Bell System network planning, that are both substantially improved in a statistical sense relative to existing methods and computationally more efficient. The former claim is based on testing by analysis, computer simulation, and field study. The latter, because of the efficient recursive calculations of the filter, has been borne out through Bell Laboratories programs and limited field studies. The companion papers in this issue will elaborate on these conclusions and will describe plans to implement the algorithms in standard, mechanized network planning tools.

REFERENCES

1. N. D. Blair, "Forecasting Telephone Traffic in the Bell System," International Teletraffic Congress 8, Melbourne, November 1976, pp. 211:1-11.
2. W. C. Johnson and W. P. McGuire, "The Trunk Forecasting System Tells What, Where, When," Bell Laboratories Record (June 1974), pp. 192–8.
3. C. Cirillo, "Controlling Circit Orders with TIRKS," Bell Laboratories Record (March 1977), pp. 73-7.
4. R. C. Chang, R. D. Pease, and G. L. Wright, "Facility Planning. . .Over the Long Haul It's IFRPS," Bell Laboratories Record (June 1980), pp. 196-203.
5. R. L. Franks et al., "A Model Relating Measurement and Forecast Errors to the Provisioning of Direct Final Trunk Groups," B.S.T.J., *58* (February 1979), pp. 351-78.
6. G. E. P. Box and G. M. Jenkins, *Time Series Analysis*, San Francisco: Holden-Day, 1970.
7. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," J. Basic Eng., *82* (1960), pp. 340-5.
8. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," Trans. ASME Ser. D, J. Basic Eng., *83* (1961), pp. 95-107.
9. J. P. Moreland, "A Robust Sequential Projection Algorithm for Traffic Load Forecasting," B.S.T.J., this issue.
10. C. R. Szelag, "A Short-Term Forecasting Algorithm for Trunk Demand Servicing," B.S.T.J., this issue.
11. A. Ionescu-Graff, "A Sequential Projection Algorithm for Special-Services Demand," B.S.T.J., this issue.
12. R. K. Mehra, "Kalman Filters and their Applications to Forecasting," *TIMS Studies in the Management Sciences*, Amsterdam: North-Holland Publishing Co., *12*, 1979, pp. 75–94.
13. A. Gelb, ed., *Applied Optimal Estimation*, Cambridge, Mass.: MIT Press, 1974.
14. R. K. Mehra and D. G. Laintootis, eds. *System Identification, Advances and Case Studies*, New York: Academic Press, 1976.
15. N. Wiener, *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series With Engineering Applications*, New York: John Wiley, 1949.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 1, January 1982
Printed in U.S.A.

A Robust Sequential Projection Algorithm for Traffic Load Forecasting

By J. P. MORELAND

(Manuscript received December 31, 1980)

Forecasts of busy season trunk group traffic loads are required for planning the Bell System's message network. Forecasting algorithms currently in use obtain estimates of future loads by multiplying the most recent measurement of busy season load by an aggregate growth factor. Because of statistical errors in measured loads and differences between individual trunk group and aggregate growth factors, the resulting forecasts can have large statistical errors. In this paper we extend earlier work to develop a new algorithm, called the sequential protection algorithm (SPA), based on a linear two-state Kalman filter, together with logic for detecting and responding to unusually large measurement errors or changes in trend. In typical applications of Kalman filtering, the statistics of system noises, measurement errors, and initial conditions are known and the filter parameters (Kalman gains) are selected accordingly. For our application, however, these statistics cannot be determined without error. Consequently, we develop a method for selecting robust filter parameters which provide improved performance, independent of system noises, measurement errors, and initial conditions. In particular, under the assumption of linear growth for 5-year intervals, the average rms 1-year forecast error of SPA is about 10 percent less than that of the existing algorithms. Field test results confirm the theoretical results presented here. Accordingly, specifications have been written for inclusion of SPA in the Bell System's standard trunk forecasting systems.

I. INTRODUCTION AND SUMMARY

Forecasts of busy season (yearly peak) trunk group traffic-loads are required for the planning of the Bell System's message network. These forecasts are used to design traffic networks which minimize the cost of the trunks required to satisfy anticipated customer demands.

The standard load forecasting algorithms currently in use in the Bell System obtain estimates of future loads by multiplying the most recent measurement of busy season load by an aggregate growth factor; for example, the average of the growth factors obtained by trending the total office loads at each end of the trunk group. Descriptions and comparisons of the various algorithms currently in use are given in Ref. 1.

As explained in Ref. 1, these algorithms have two significant sources of error: (*i*) Because of the finite amount of data upon which measurements are based, measured loads can have large statistical errors; standard deviations fall in the range of about 5 to 40 percent depending upon load size and type of measurement system.[2] (*ii*) Individual trunk group growth factors can differ from the aggregate growth factor; standard deviations of 6 percent have been observed. These forecast errors are significant since they lead to an increase in the reserve trunk capacity required to satisfy customer demands.[3]

To reduce forecast error and, hence, reserve trunking capacity, a new algorithm, called the sequential projection algorithm (SPA), has been developed to forecast busy season traffic loads within the Bell System. The SPA is based on a linear two-state Kalman filter model, whose use in traffic forecasting was studied first by David and Pack,[1] together with logic for detecting and responding to outlier measurements, i.e., unusually large measurement errors or changes in trend.

As discussed in Ref. 1, David and Pack tested several Kalman filter models—some with as many as eight state variables and four data variables. In summary, for the planning interval of interest (1 to 5 years ahead) none performed consistently or significantly better than the relatively simple two-state (traffic load and incremental growth), one-data variable (measured load) model.

The two-state Kalman filter establishes a linear trend for individual traffic loads as follows: As illustrated in Fig. 1, the level, or smoothed base load, is a weighted average of the most recent measurement, or base load, and the previous 1-year forecast. Similarly, the smoothed growth increment is a weighted average of the measured and previously forecasted increments. (The measured increment is, by definition, the measured load minus the previous year's smoothed base load.)

The performance of the Kalman filter, as measured by mean square forecast error, depends upon the filter parameters, i.e., the gains α_n and β_n in Fig. 1, the standard deviation of load measurement and growth estimation errors, and upon the assumed evolution of the true load.

Since SPA will be used under a variety of possible operating conditions, the filter parameters could be tuned to provide optimal performance, i.e., minimum mean square forecast error, for each application.

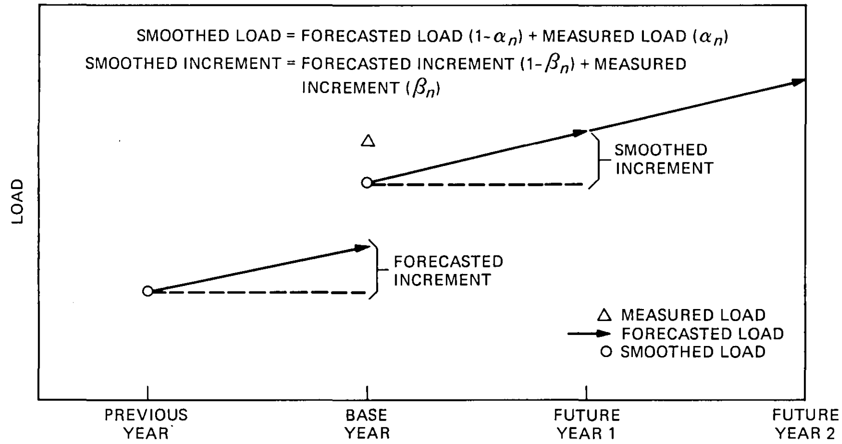
SMOOTHED LOAD = FORECASTED LOAD (1-α_n) + MEASURED LOAD (α_n)
SMOOTHED INCREMENT = FORECASTED INCREMENT (1-β_n) + MEASURED INCREMENT (β_n)
LOAD
SMOOTHED INCREMENT
FORECASTED INCREMENT
△ MEASURED LOAD
→ FORECASTED LOAD
○ SMOOTHED LOAD
PREVIOUS YEAR
BASE YEAR
FUTURE YEAR 1
FUTURE YEAR 2

Fig. 1—Basic operation of SPA.

For example, in the Bell System we employ two types of load measurement systems; Bell Operating Companies obtain load estimates from a direct measurement of trunk group usage, while Long Lines derives estimates from point-to-point, e.g., end-office to end-office data provided by the Centralized Message Data System (CMDS).[2] For trunk group data, the standard deviation of measurement error is in the range of about 5 to 10 percent, depending on load size. For point-to-point data, the range is about 10 to 40 percent. (See Fig. 2.) Accordingly, different parameters could be used for different measurement systems and for different load ranges.

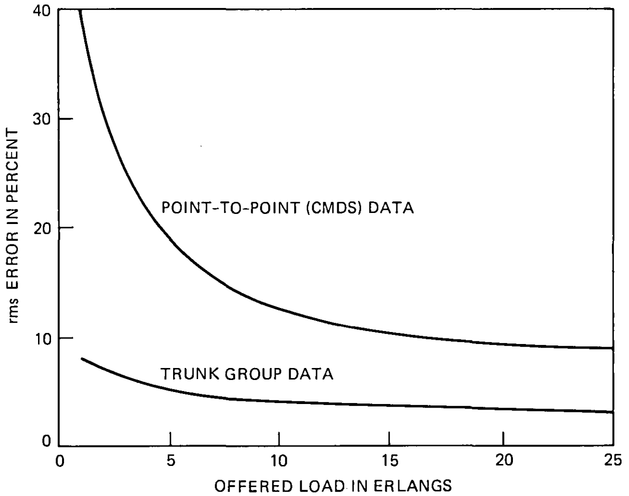

Fig. 2—Sampling error vs. offered load.

Indeed, the initial development of SPA[1] was intended for use with trunk group data only, and the objective of the current study was to extend the original work to applications using point-to-point data. As suggested above, one possible solution would be to develop multiple versions of SPA.

Instead, in this paper, we develop a single, robust SPA whose parameters are selected to provide improved performance over the entire range of operating conditions, including the use of either trunk group or point-to-point data.

The use of robust parameters is important for two reasons: (*i*) A single SPA for all applications should be simpler to implement and maintain than multiple versions. (*ii*) The actual values of the statistical parameters for each application cannot be determined without error, and our results show that erroneously assumed values can lead to a performance substantially worse than that of conventional projection methods. Of course, as with any robust technique, we pay a premium by receiving less than theoretically optimal performance for protection against the possibility of a performance worse than conventional methods.

Section II of this paper provides a qualitative overview of the functions performed by SPA that include procedures for detecting and

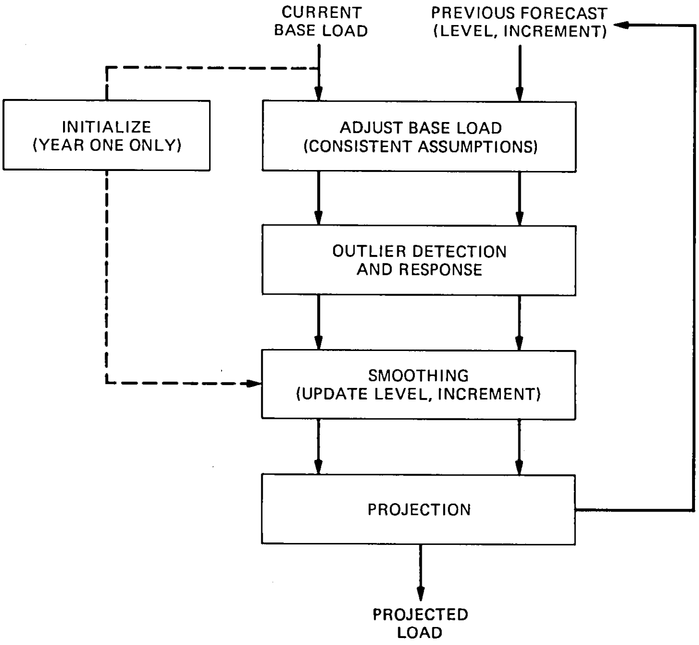
CURRENT BASE LOAD
PREVIOUS FORECAST (LEVEL, INCREMENT)
INITIALIZE (YEAR ONE ONLY)
ADJUST BASE LOAD (CONSISTENT ASSUMPTIONS)
OUTLIER DETECTION AND RESPONSE
SMOOTHING (UPDATE LEVEL, INCREMENT)
PROJECTION
PROJECTED LOAD

Fig. 3—Sequential projection algorithm.

responding to outliers, as well as the smoothing and projection functions of the Kalman filter. Section III defines the mathematical model and procedures for selecting robust filter parameters; Section IV gives numerical results; Section V develops the outlier detection thresholds; and Section VI gives the conclusions.

II. OVERVIEW OF SPA

As shown in Fig. 3, SPA is composed of five major components: algorithm initialization, base-load adjustments, outlier detection, smoothing, and projection. In the following paragraphs, we provide a qualitative description of the operations performed by each of these components.

2.1 Initialization

As indicated in Fig. 4, the smoothed base load in the first year of operation, and in certain other cases discussed in Section 2.3, is equal to the measured base load. The smoothed growth increment is calculated by multiplying this base load by a growth factor obtained, for example, by trending the total office loads at each end of the trunk group.[1]

2.2 Base load adjustments

In the second and subsequent years of operation, SPA updates both the level and incremental growth by comparing the most recent base load with the previous 1-year forecast of that same load. Since the base and forecasted loads must correspond to the same traffic routings,

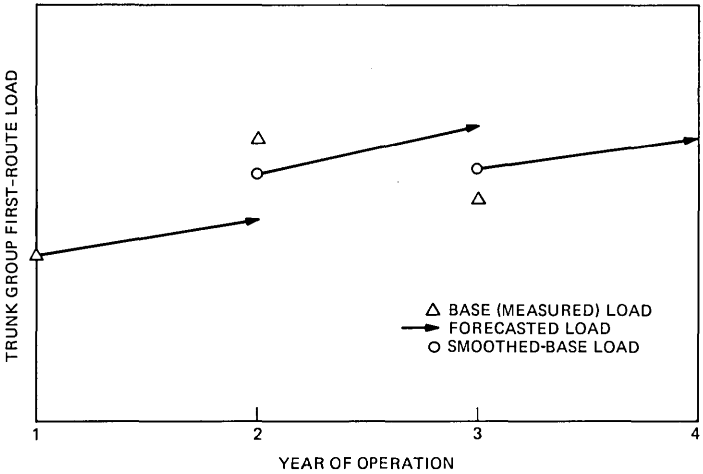

Fig. 4—Initialization of SPA.

differences between the previous and current routings are accounted for by adding an adjustment term to the base load so that the adjusted base load agrees with previous routing assumptions. After performing the outlier detection and smoothing functions described below, the routing adjustment term is subtracted from the smoothed base load so that it agrees with the current routing assumptions.

Furthermore, the previously forecasted load is adjusted to remove the impact of deterministic events, such as a proposed tariff change, that were predicted but did not occur.

2.3 Outlier detection

Under the assumption that the observed forecast error (i.e., the difference between the forecasted and measured loads) has a Gaussian distribution with zero mean, the linear Kalman filter upon which SPA is based provides the minimum attainable mean squared forecast error.[4] In practice, however, the normal statistical errors (because of the finite measurement interval, day-to-day load variation, random variations in CMDS point-to-point sample size,[2] and growth errors) are occasionally contaminated by wiring errors, recording errors, or unexpected changes in the trend of the true load. In such cases, when the observed forecast error deviates from a Gaussian distribution, the linear Kalman filter model can have a mean squared forecast error which is substantially greater than the minimum.[4]

Our approach to this problem, which is based in part on the nonlinear

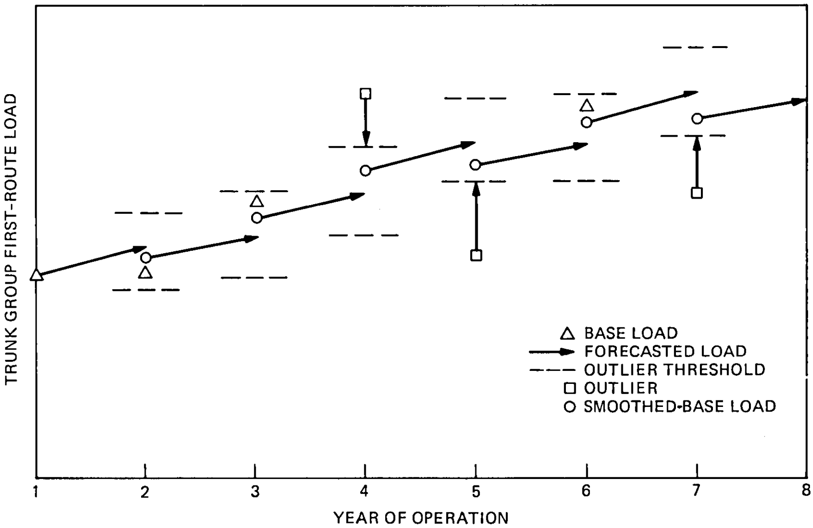

Fig. 5—Response of outliers (sign not repeated).

Kalman filter model described in Ref. 4, is the following: If the difference between the possibly adjusted base and forecasted loads exceeds present thresholds, the base load is declared to be an outlier. In response, SPA will adjust the base load or restart at the measured level, depending only upon whether an outlier of the same sign occurred in the previous year.

If an outlier of the same sign did not occur in the previous year, SPA replaces the base load by the nearest threshold value as indicated by the vertical arrows in Fig. 5. Qualitatively, the underlying assumption here is that in most cases such an outlier signals an invalid or atypical measurement caused by, for example, a recording error and not a change in trend. Formally, this modification of the Kalman filter is equivalent to that proposed by Masreliez and Martin in Ref. 4.

Alternatively, as indicated in Fig. 6, if an outlier of the same sign occurs in two consecutive years, SPA restarts at the measured level. The assumption here is that in most cases two consecutive outliers of the same sign signal a change in trend. In theory, additional improvement could be obtained by restarting at the previous outlier and then smoothing with the current measurement. However, such a procedure would be more complicated to implement, and our studies show that it would have negligible impact on performance.

As indicated by Huber's studies,[5] and as supported by our field test results,[6] adequate protection against outliers is obtained when the thresholds are set anywhere in the range of about one or two times the

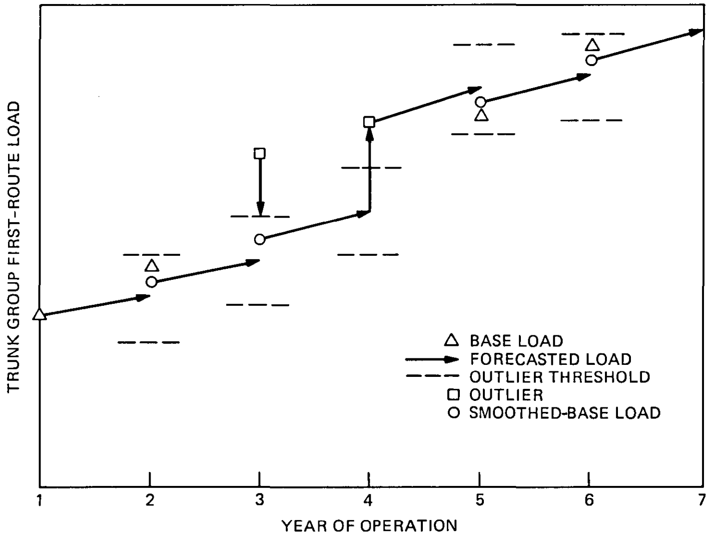

Fig. 6—Response to outliers (sign repeated).

rms observed forecast error. This result is important: we show that it allows us to use thresholds which are independent of the number of data points processed and the type of measurement system. Numerical values for these thresholds are provided in Section V.

2.4 Smoothing

The possibly adjusted base and forecasted loads are combined to produce a smoothed base load and a smoothed growth increment. As indicated in Fig. 1, the smoothed base load is a weighted average of the base and 1-year forecasted loads. Similarly, the smoothed growth increment is a weighted average of the measured and forecasted growth increments. Procedures for selecting the appropriate gains α_n and β_n are described in Section IV.

2.5 Projection

As indicated in Fig. 1, the smoothed base load and growth increment are combined to establish a linear projection. That is, the projected load for the kth future year is obtained by adding k smoothed growth increments to the smoothed base load. In practice, the resulting trunk group load forecasts can be adjusted to include user supplied estimates of the impact of deterministic events (caused by, for example, proposed tariff changes) or to agree with other aggregate load forecasts.

III. MATHEMATICAL MODEL

3.1 General

Although SPA is based on a linear two-state, i.e., trunk group load and incremental growth, Kalman filter, we considered more complex models with additional state variables, e.g., aggregations of other trunk group loads and growth factors. Therefore, for completeness, we first summarize the equations which define the general discrete-time linear Kalman filter. For a more complete discussion, see Ref. 7.

The true time-behavior of the state variables is assumed to be defined by the linear transition equation

$$\underline{X}_{n+1} = \phi \underline{X}_n + \underline{W}_n + \underline{U}_n, \tag{1}$$

where $\underline{X}_n$ is an s-vector of true state variables in period (year) n, ϕ is an $s \times s$ transition matrix which may depend on n, $\underline{W}_n$ is an s-vector of zero-mean random modelling errors, and $\underline{U}_n$ is an s-vector of deterministic changes in state.

The one-period-ahead projection formula is

$$\underline{X}_{n+1,n} = \phi \underline{X}_{n,n} + \underline{U}_n, \tag{2}$$

where $\underline{X}_{n+k,n}$ denotes an estimate of $\underline{X}_{n+k}$ based on data through period n. The "smoothed" estimate $\underline{X}_{n,n}$ is calculated by

$$\underline{X}_{n,n} = \underline{X}_{n,n-1} + K_n(\underline{y}_n - H\underline{X}_{n,n-1}), \tag{3}$$

where K_n is a $d \times s$ "Kalman gain" matrix and $\underline{y}_n$ is a d-vector of measurements in period n related to $\underline{X}_n$ by

$$\underline{y}_n = H\underline{X}_n + \underline{V}_n, \tag{4}$$

where H is a $d \times s$ matrix and $\underline{V}_n$ is a d-vector of zero-mean measurement errors.

The optimal gain matrix K_n is determined recursively by the equations

$$K_n = P_nH^T(HP_nH^T + R)^{-1} \tag{5}$$

$$S_n = (I - K_nH)P_n, \tag{6}$$

and

$$P_{n+1} = \phi S_n \phi^T + Q, \tag{7}$$

where R is the covariance matrix of measurement errors, i.e., $R = E(\underline{V}_n\underline{V}_n^T)$, Q is the covariance matrix of modelling errors, and S_0 is an estimate of the covariance matrix of the initial state estimate $\underline{X}_{0,0}$. Furthermore, it is assumed that $\underline{V}_i$ and $\underline{W}_j$ have zero mean and are pairwise uncorrelated among themselves and with each other for all i, j.

As discussed in Section I, we will be concerned with filter performance for nonoptimal gains. In this case, the covariance matrix P_{n+1} of the projection error $(\underline{X}_{n+1,n} - \underline{X}_{n+1})$ is related to the covariance matrix S_n of $\underline{X}_{n,n}$ by eq. (7), but S_n is related to P_n by

$$S_n = (I - K_nH)P_n(I - K_nH)^T + K_nRK_n^T, \tag{8}$$

which reduces to eq. (6) only when K_n is given by eq. (5).

3.2 Two-state model

As noted in Section I, we considered models with as many as $s = 8$ state variables and $d = 4$ measurement variables. However, our studies[1] showed that none performed consistently or significantly better than the simple two-state, one-data variable model defined by the equations

$$\underline{X}_{n+1} = \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} w_n \\ \dot{w}_n \end{bmatrix} \tag{9}$$

and

$$y_n = x_n + v_n, \tag{10}$$

where x_n and $\dot{x}_n$ are, respectively, the true load and true incremental growth in year n, and y_n is the measured load in year n. These equations correspond to eqs. (1) and (4) with

$$\phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \tag{11}$$

and

$$H = [1, 0]. \tag{12}$$

The more complex models were rejected since the additional variables that we considered, e.g., aggregations of other trunk group loads and growth factors, are usually not highly correlated with trunk group load. Moreover, when the correlation is high the improvement in performance is minimal and, more importantly, when high correlation is incorrectly assumed, the penalties outweigh the expected benefits.[1]

To complete the description of our two-state model, note that the smoothing eq. (3) becomes

$$\underline{X}_{n,n} = \begin{bmatrix} x_{n,n} \\ \dot{x}_{n,n} \end{bmatrix} = \begin{bmatrix} x_{n,n-1} + \alpha_n(y_n - x_{n,n-1}) \\ \dot{x}_{n,n-1} + \beta_n(y_n - x_{n,n-1}) \end{bmatrix}, \tag{13}$$

where α_n and β_n are the Kalman gains in year n. The 1-year-ahead projection formula is

$$\underline{X}_{n+1,n} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{n,n} \\ \dot{x}_{n,n} \end{bmatrix}. \tag{14}$$

Note: Our analysis of the two-state model will ignore the possibility of deterministic changes in state; that is, we assume $\underline{U}_n = 0$. Also, by combining eqs. (13) and (14) it follows that eq. (13) may be written in the form

$$\begin{bmatrix} x_{n,n} \\ \dot{x}_{n,n} \end{bmatrix} = \begin{bmatrix} (1 - \alpha_n)x_{n,n-1} + \alpha_n y_n \\ (1 - \beta_n)\dot{x}_{n,n-1} + \beta_n(y_n - x_{n-1,n-1}) \end{bmatrix}, \tag{15}$$

which emphasizes that $x_{n,n}$ is a weighted average of the forecasted and measured levels and, similarly, $\dot{x}_{n,n}$ is a weighted average of the forecasted and measured increments.

Finally, note that $R = \sigma^2$, where σ is the standard deviation of load measurement error; the dependence of σ on load size and type of measurement system is discussed in Section 4.2.2. In the following section, we define the SPA initialization procedure and in Section 3.4 we describe our procedure for selecting values for the filter gains α_n and β_n and the associated assumptions about Q.

3.3 Initialization

As explained in Section 2.1, the smoothed-base load and growth increment in the first year of operation are given by

$$x_{0,0} = y_0 \tag{16}$$

and

$$\dot{x}_{0,0} = \hat{g} y_0, \tag{17}$$

where $\hat{g}$ is an aggregate growth factor. For example, if $\hat{g} = 0.1$, the increment is 10 percent of the base load.

The normalized covariance matrix $S_0/\sigma^2 = (s_{ij}/\sigma^2)$ of the initial state estimate is given by

$$\frac{s_{11}(0)}{\sigma^2} = \frac{E(x_0 - y_0)^2}{\sigma^2} = 1, \tag{18}$$

$$\begin{aligned} \frac{s_{22}(0)}{\sigma^2} &= \frac{E(gx_0 - \hat{g}y_0)^2}{\sigma^2} \\ &= \frac{E[(g-\hat{g})x_0 + \hat{g}(x_0 - y_0)]^2}{\sigma^2} \\ &= \frac{\sigma_g^2 x_0^2}{\sigma^2} + \hat{g}^2, \end{aligned} \tag{19}$$

where g is the true growth factor and σ_g is the standard deviation of the estimate $\hat{g}$, and

$$\begin{aligned} \frac{s_{12}(0)}{\sigma^2} &= \frac{E[(x_0 - y_0)(gx_0 - \hat{g}y_0)]}{\sigma^2} \\ &= \frac{E\{(x_0 - y_0)[(g-\hat{g})x_0 + \hat{g}(x_0 - y_0)]\}}{\sigma^2} \\ &= \hat{g}. \end{aligned} \tag{20}$$

3.4 Filter parameters

As discussed in Section I, our objective is to select robust values for the gains α_n and β_n; that is, values which will provide improved performance over the range of operating conditions. Our approach to this problem starts with the following idealistic assumption, which will be relaxed in a later section:

We first assume that the true load displays constant incremental growth. That is, we first assume that Q, the covariance matrix of modelling errors, is identically zero. Under this assumption, it follows from eqs. (5) to (7) that the optimal gains depend only on the ratio S_0/σ^2. In turn, for a given value of $\hat{g}$, it follows from eqs. (18) to (20) that S_0/σ^2 is determined by the ratio

$$G^2 = \frac{\sigma_g^2}{(\sigma/x_0)^2}. \tag{21}$$

In Section 4.2, we display the optimal gains and corresponding performance (as measured by mean square forecast error) for various known values of G. More importantly, we show that an erroneously assumed value for G can lead to a performance worse than that of the conventional projection method; equivalently, since the conventional method is used to initialize SPA, the mean square forecast error can increase with the number of data points processed.

Next, in Section 4.2.3, we show that the gains corresponding to one particular value of G provide a performance that is nearly independent of the actual value of G. We call these gains the "robust gains for linear growth."

Finally, in Section 4.3, we relax the assumption of linear growth and show that certain constant gains, derived from the "robust gains for linear growth," provide improved performance in the presence of system noise, as well as the ideal case where the trend remains constant.

IV. NUMERICAL RESULTS

4.1 Examples

To help explain our results, we first consider several examples which show how the gains and the mean-square forecast error depend upon the ratio G. In these examples, we assume that the true load has constant incremental growth, and we assume two-point distributions for measurement and growth estimation errors. That is, the measured load is either high or low by equal amounts and with equal probability, and the initial estimate of incremental growth is either high or low by equal amounts and with equal probability. Moreover, in each example we assume that the aggregate growth factor $\hat{g}$ is zero.

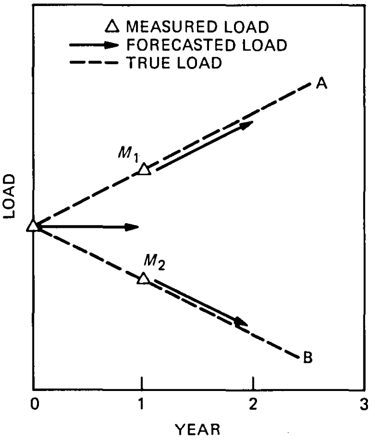

Fig. 7—Optimal filter: no measurement error ($G = \infty$).

4.1.1 No measurement error ($G = \infty$)

In the example illustrated in Fig. 7, we assume no measurement error, i.e., $\sigma^2 = 0$, but an error of plus or minus one unit in the initial increment so that $G = \infty$. Thus, with this information, the true load lies along the dashed line marked A with probability 0.5 or along the dashed line B with probability 0.5. Accordingly, the initial mean square forecast error is $0.5(1)^2 + 0.5(-1)^2 = 1$.

As shown in Fig. 7, the possible measurements in year 1 are M_1 or M_2 with equal probability. However, since there is no measurement error and since two points determine a straight line, it is clear that in either case the forecast error can be reduced to zero after processing the second measurement; examination of eq. (13) shows that the appropriate gains are $\alpha_1 = \beta_1 = 1$.

4.1.2 No growth error ($G = 0$)

At the other extreme, suppose there is no error in the initial growth estimate, i.e., $\sigma_g^2 = 0$, but an error of plus or minus one unit in the measured load so that $G = 0$. As shown in Fig. 8, the trend line is either A or B with equal probability; hence, the initial mean square forecast error is 1. The possible measurements in year 1 are M_1 with probability 0.25 [since A and a high measurement occur with probability (0.5)(0.5)], M_2 with probability 0.5, and M_3 with probability 0.25.

Since M_1 and M_3 correspond uniquely to A and B, respectively, the forecast error can be reduced to zero after processing either of these measurements; eq. (13) and Fig. 8 show that the appropriate gains are $\alpha_1 = 0.5$ and $\beta_1 = 0$. However, if M_2 occurs, no new information is obtained. Consequently, after processing the second data point, the

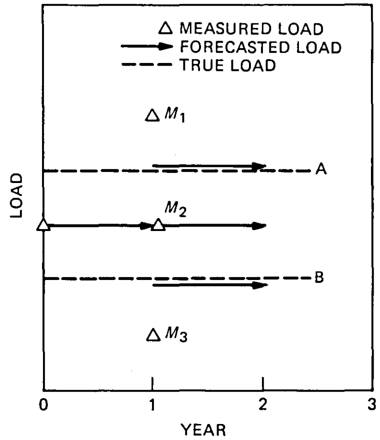

Fig. 8—Optimal filter: no growth error ($G = 0$).

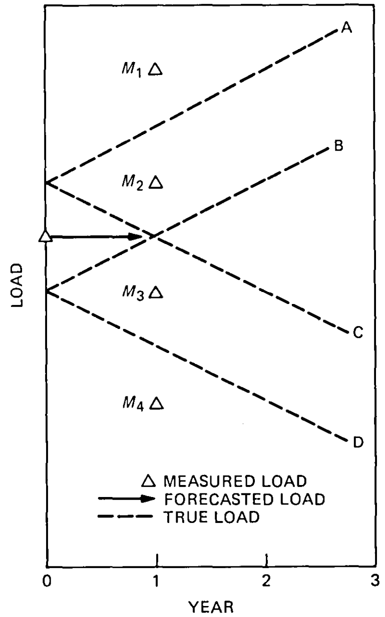

Fig. 9—Optimal filter: equal measurement and growth errors ($G = 1$).

mean square forecast error is $0.5(0)^2 + 0.5(1)^2 = 0.5$. Thus, the rms forecast error is reduced by a factor of $1/\sqrt{2}$.

4.1.3 Equal measurement and growth errors (G = 1)

For values of G between the above extremes, i.e., $0 < G < \infty$, the reductions in forecast error might be expected to fall between those corresponding to the extreme values of G. But, this is not generally true.

For example, consider the case illustrated in Fig. 9 in which $G = 1.0$. We assume uncorrelated errors of plus or minus one unit in the measured load and in the initial increment. Thus, as shown in Fig. 9, the possible trend lines are A, B, C, or D, each with equal probability. Consequently, the initial mean square forecast error is $0.25(2)^2 + 0.5(0)^2 + 0.25(-2)^2 = 2$.

In year 1 the possible measurements are M_1 with probability $[0.25(0.5) = 0.125]$, M_2 with probability 0.375 (since M_2 may correspond to A, B, or C), M_3 with probability 0.375, and M_4 with probability 0.125.

If M_1 occurs, the trend line must be A and the forecast error could be reduced to zero by setting $\alpha_1 = 2/3$ and $\beta_1 = 1/3$. Similarly, if M_4 occurs the trend line must be D; again $\alpha_1 = 2/3$ and $\beta_1 = 1/3$ are appropriate.

If M_2 occurs, the trend line is either A, B, or C with equal probability. Since B is halfway between A and C in year 2, it follows that the mean square forecast error is minimized by forecasting the level of B in year 2; this result is obtained with $\alpha_1 = 2/3$ and $\beta_1 = 1/3$. Similarly, if M_3 occurs, the forecast with these gains is on C in year 2, and the forecast error is minimized.

Thus, the optimal gains are $\alpha_1 = 2/3$, $\beta_1 = 1/3$ and, after processing the second data point, the mean square forecast error is $0.5(0)^2 + 0.25(2)^2 + 0.25(-2)^2 = 2$, which is identical to the initial value. Thus, for this example there is no reduction in forecast error after processing the second data point.

4.1.4 G unknown

The above examples assumed that G was known exactly. Suppose now, however, that the gains are chosen under the assumption that $G = \infty$, i.e., $\alpha_1 = \beta_1 = 1$, but in fact $G = 0$. In this case, which is illustrated in Fig. 10, the mean square forecast error in year 2 is $0.25(3)^2 + 0.25(1)^2 + 0.25(-1)^2 + 0.25(-3)^2 = 5$, which is five times the initial value.

This result, although exaggerated, is important because it shows that SPA could actually perform worse than the conventional forecasting procedure. Indeed, this fact, combined with the practical impossibility of estimating G without error, is the major consideration in our decision to use the robust version of SPA described in Sections 4.2.3 and 4.3.2.

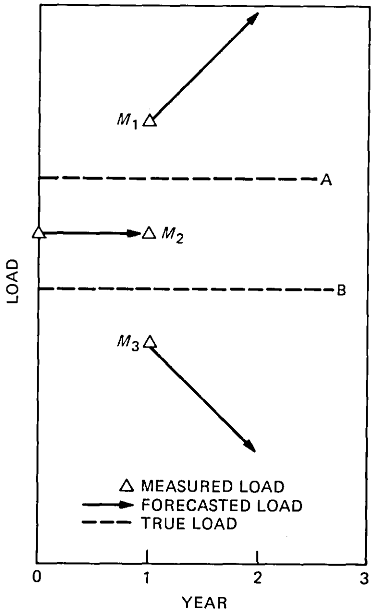

Fig. 10—Nonoptimal filter: assumed $G = \infty$, actual $G = 0$.

4.2 Linear growth

4.2.1 G known

The results of this section apply when the true load has constant incremental growth and when the ratio G, eq. (21), is known exactly.

Figure 11 displays the normalized rms 1-year forecast error as a function of the number of data points processed after initialization. The normalization is with respect to the rms forecast error of the conventional projection method. Since this method is used to initialize SPA, the initial normalized rms error equals 1.0. Each curve corresponds to a different value of G and, accordingly, to a different set of gains α_n and β_n. For example, the gains corresponding to three different values of G are shown in Fig. 12.

The results shown in Fig. 11 assume that the true load displays constant incremental growth; under this assumption, the forecast error approaches zero as n increases. In practice, however, unexpected changes in trend will occur and, consequently, as described in Section 2.3, SPA will occasionally be reinitialized. Accordingly, average forecast error over the interval between reinitializations is a more meaningful figure of merit. In the following paragraphs, we assume an average interval of five years. We emphasize, however, that this assumption will underestimate the benefits of SPA under more stable conditions and overestimate the benefits under less stable conditions.

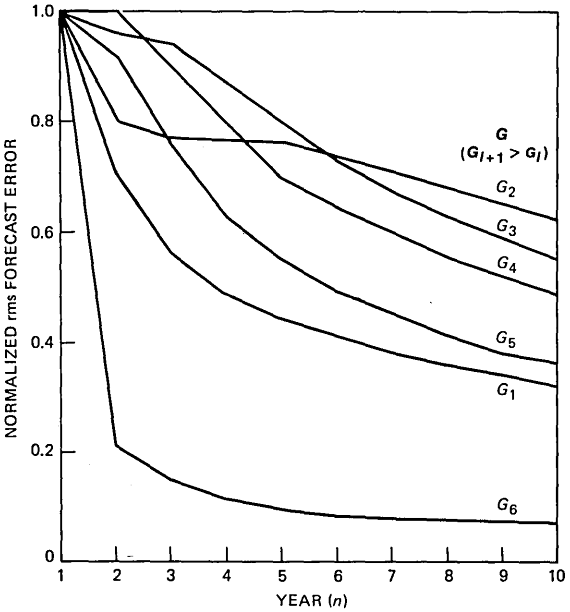

Fig. 11—Optimal filter: forecast error vs. year.

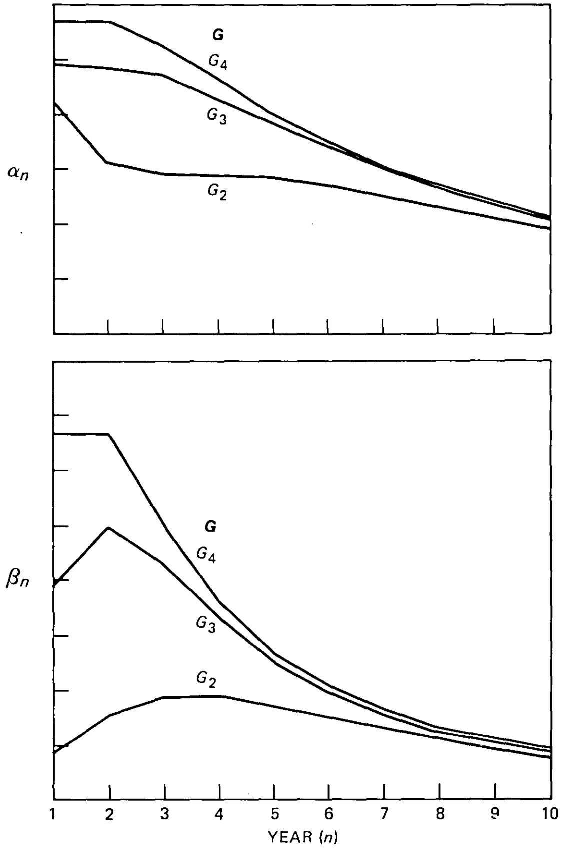

Fig. 12—Optimal gains vs. year.

Thus, the curve labeled optimal in Fig. 13 displays the normalized 5-year average forecast error—the average of the first five values in Fig. 11—as a function of the parameter G. If, for example, $G = G_2$ and we choose the gains accordingly, the average rms error would be about 20 percent less than that for the conventional forecasting method. Alternatively, if $G = G_4$, we would choose a different set of gains and the average reduction would be approximately 15 percent. Note as suggested by the examples discussed above, that the reduction in forecast error is a convex function of G.

4.2.2 G unknown

Suppose now that the actual value of G differs from the assumed value. For example, suppose we use the gains corresponding to $G = G_2$, but in fact $G = G_4$. In this case, the curve labeled $G = G_2$ in Fig. 13 shows that the 5-year average rms error would be about 20 percent

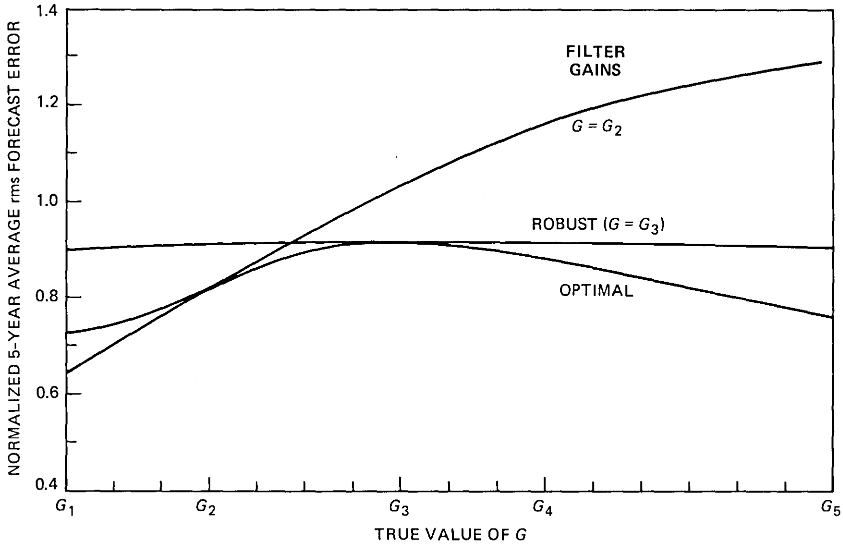

Fig. 13—Average rms forecast error for several filters.

larger than its initial value, in agreement with the example discussed in Section 4.1.4.

The results of Fig. 13 can also be interpreted as follows: Suppose that $G = G_4$ is appropriate when trunk group loads are derived from trunk group data. Then $G = G_2 = G_4/3$ would be appropriate when loads are derived from point-to-point (CMDS) data since, from Fig. 2, the rms measurement error for CMDS data is approximately three times that for trunk group data. Figure 13 then shows that an SPA tuned for point-to-point data would perform poorly with trunk group data.

The results shown in Fig. 13 might suggest that we should use different gains for different applications. We considered this approach but found it to have two practical problems.

First, as shown in Fig. 2, measurement error depends not only upon the type of data but also upon load size. And, although we could allow the gains to be a function of load size and data type, multiple versions of SPA would be relatively more difficult to implement and maintain.

Second, and more important, in practice it will not be possible to determine the exact value of G. For, in general, actual loads will not display constant incremental growth, but may exhibit random fluctuations about a trend line. In this case, the model described in Section III is still appropriate if x_n denotes the trend level, instead of the true load. However, σ^2 now consists of two components: one because of the difference between the measured and true loads and the other because of the difference between the true load and the trend line. Although

Ref. 2 quantifies the first component, we have no way a priori to assess the contribution of the second component. Thus, our assumed value for σ^2 may differ from the actual value.

Similarly, since we cannot determine a priori the difference between our estimate of the aggregate and individual trunk group load growth rates, our estimate of σ_g^2, eq. (19), will, in general, differ from the actual value.

Consequently, the assumed value of G will in general also differ from the actual value. Accordingly, to guard against the consequences of an error in our estimate of G, we would like to employ an algorithm which performs well under a variety of possible operating conditions.

4.2.3 Robust gains for linear growth

Fortunately, there exists a robust set of gains which provides the same average performance independent of the true value of G. That is, as shown by the curve labeled "robust" in Fig. 13, if we use the gains corresponding to $G = G_3$ (see Fig. 12) then the 5-year average rms error will be about 10 percent less than that of the conventional projection method—independent of the actual value of G.

Of course, as with any robust technique, Fig. 13 shows that we pay a premium by receiving less than the theoretically optimal performance for protection against the possibility of a performance substantially worse than that of the conventional projection method. However, we used the data from our field test[5] to estimate, a posteriori, the value

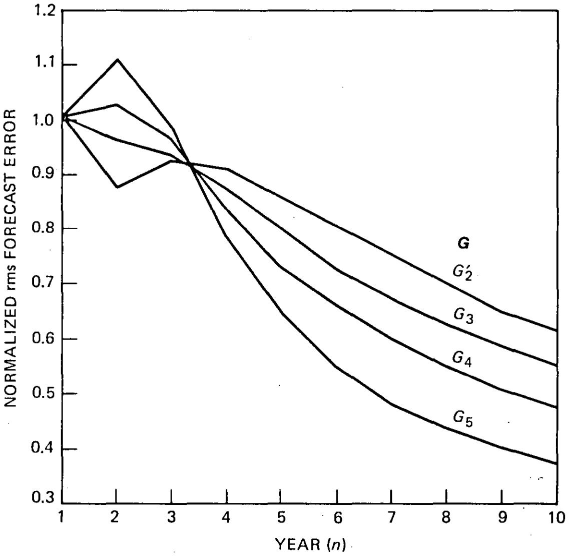

Fig. 14—Robust filter (variable gains): rms forecast error vs. year.

of G. Remarkably, the observed value was $G \simeq G_3$. Thus, at least for this one case, our robust filter was in fact nearly optimal.

Although the robust gains yield an average performance which is independent of G, Fig. 14 shows that the actual performance in each year is not independent of G. For example, if $G = G_5$ the rms error of SPA increases above that of the conventional method by about 10 percent in year 2, but is decreased by about 35 percent in year 5. Thus, during the first couple of years after initialization, SPA may perform slightly worse than the conventional method for some trunk groups. Thereafter, SPA will perform better, provided that the trend remains constant.

4.3 Robust, constant gains for SPA

4.3.1 Response to unexpected changes in trend

With the "robust gains for linear growth," SPA is robust to measurement and initial growth estimation errors. For practical applications, however, it is equally important that SPA also be made robust to deviations from linear growth.

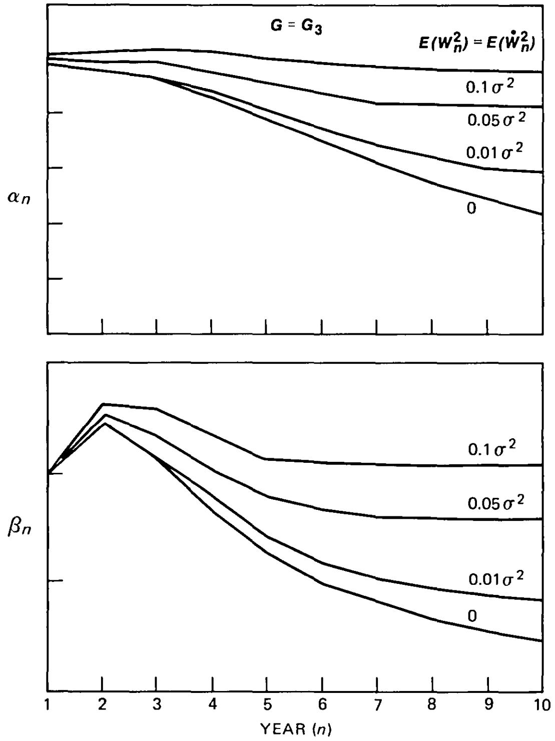

Fig. 15—Optimal gains with modelling error.

That is, the results of Section 4.2.3 assume that the trend line displays constant incremental growth. Under this assumption, since the forecast error approaches zero, the gains α_n and β_n approach zero as n increases. Accordingly, as discussed in Section 2.3, SPA would eventually respond to unexpected changes in the level or slope of the trend only when their cumulative effect produced two consecutive outliers of the same sign.

4.3.2 Constant gains

At the expense of receiving less than the theoretically optimal performance in the ideal case where the trend remains constant, we can decrease SPA's response time to unexpected changes in trend by not allowing the gains α_n and β_n to approach zero. As discussed in Ref. 7, one approach to selecting limiting values for α_n and β_n is to add zero-mean, uncorrelated random variables (w_n and $\dot{w}_n$) to the description of the true load in eq. (9). That is, we assume that the covariance matrix Q of modelling errors is nonzero. These modelling error terms lead to gains α_n and β_n which approach nonzero constants that depend upon the mean square value of these error terms. For example, Fig. 15 shows the optimal gains corresponding to $G = G_3$ for several values of $E(w_n^2)$ and $E(\dot{w}_n^2)$.

Although theoretically appealing, the above approach leaves the practical problem of determining appropriate values for $E(w_n^2)$ and $E(\dot{w}_n^2)$. Values could be gleaned from a large amount of historical data,

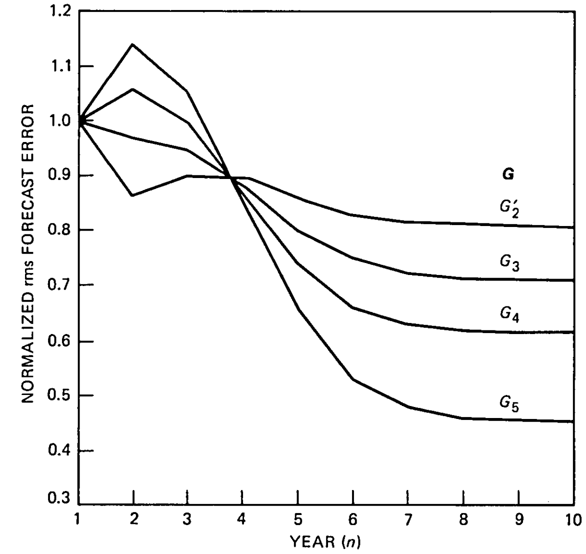

Fig. 16—Robust filter (constant gains): rms forecast error vs. year.

which we do not have, but there is no guarantee that they would be appropriate for the future.

Therefore, instead of pursuing the above approach, we simply replaced the variable gains corresponding to $G = G_3$ in Fig. 12 by the average of the first few terms and made the following observations: First, as shown in Fig. 16, the performance in the ideal case where the trend remains constant is nearly identical for the first six years after initialization to that for the robust gains for linear growth. Thereafter, the performance is somewhat poorer, but we anticipate that only a small fraction of groups will remain on a constant linear trend beyond 6 years. Second, as illustrated in Fig. 17, when the true load displays random deviations from a linear trend, i.e., when w_n, $\dot{w}_n$ are nonzero, the constant gains perform better than those designed for linear growth. Thus, although the performance is somewhat degraded in the ideal case, the use of constant gains provides protection against the very real possibility of unexpected changes in trend. For comparison, Fig. 17 also displays the performance with gains obtained by truncating the robust gains for linear growth in year 6. As indicated, performance is somewhat better in the ideal case, but worse when modelling errors are present. Since we do not have sufficient historical data to estimate the level of modelling errors, we are, therefore, recommending the use of the constant gains.

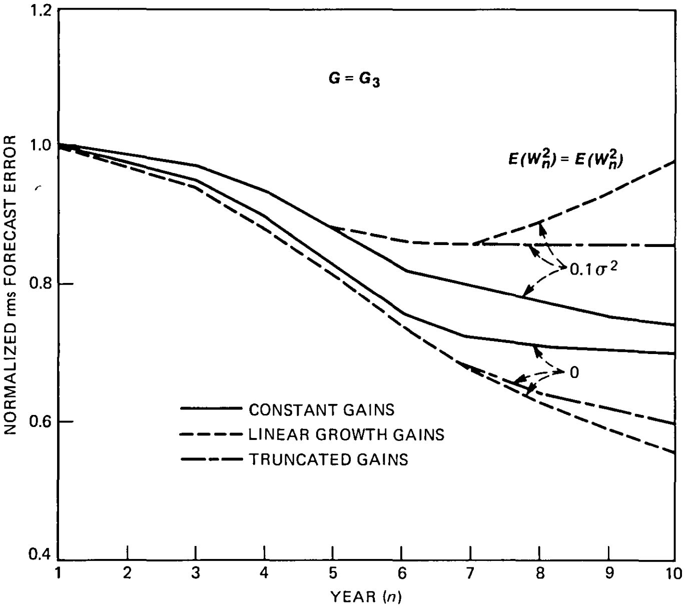

Fig. 17—Forecast error with modelling error.

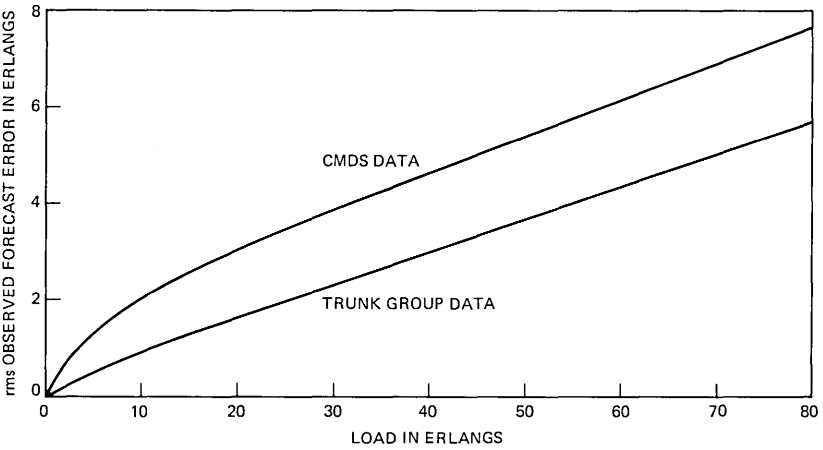

Fig. 18—Root-mean-square observed forecast error vs. load: outlier thresholds.

V. OUTLIER THRESHOLDS

As discussed in Section 2.3, Huber's studies[5] show that the outlier thresholds may be set anywhere in the range of one to two times the rms observed forecast error. In theory, the observed forecast error depends upon the number of data points processed after initialization; however, since the average rms forecast error decreases by only about 10 percent and since performance is not sensitive to the precise setting of the thresholds, we will set them based upon the initial value of the rms observed forecast error; that is, by

$$\rho^2 = E(x_{1,0} - y_1)^2. \tag{22}$$

Since $x_{1,0} = (1 + \hat{g})y_0$ and $x_1 = (1 + g)x_0$, it follows that

$$\begin{aligned} \rho^2 &= E[(y_0 - x_0) + (y_0 - x_0)\hat{g} + x_0(\hat{g} - g) - (y_1 - x_1)]^2 \\ &= 2\sigma^2(1 + \hat{g} + \hat{g}^2/2) + s_0^2\sigma_g^2, \end{aligned} \tag{23}$$

or, since $|\hat{g}| \ll 1$,

$$\rho^2 \approx x_0^2\sigma_g^2 + 2\sigma^2. \tag{24}$$

To establish numerical values for ρ, we use the theoretical expressions for mean square measurement error given in Ref. 2:

$$\sigma^2 = \frac{1}{20}\left(\frac{2x_0 h}{p} + V_d\right), \tag{25}$$

where h is the mean call-holding-time in hours, p is the sampling rate (for trunk group data, $p = 1.0$; for CMDS data, $p = 0.05$), and

$$V_d = \max(0, 0.13x_0^{\phi} - 2x_0 h) \tag{26}$$

is the variance of the daily source loads. For first-routed loads, $\phi = 1.5$ is usually appropriate.

Figure 18 displays ρ as a function of the load x_0 for both trunk group data ($p = 1.0$) and CMDS data ($p = 0.05$). In each case, we assume $\sigma_g = 0.06$, which is the value observed during our field test,[6] and $h = 1/12$.

Based upon the results of Fig. 18 and Huber's studies,[5] it follows that outlier thresholds set at two times the value of ρ for trunk group data should provide adequate performance for both trunk group and CMDS data. That is, this setting places the thresholds within the allowed range of about one to two times the actual value of ρ for both trunk group and CMDS data.

VI. CONCLUSIONS

The SPA employs a linear two-state Kalman filter, together with logic for detecting and responding to outlier measurements. The parameters of SPA have been selected to provide improved performance over the range of operating conditions, including the use of either trunk group or point-to-point traffic measurements.

Under the assumption of linear growth for 5-year intervals, the average rms 1-year forecast of SPA is about 10 percent less than that of the forecasting methods currently in use in the Bell System. Moreover, the filter parameters and outlier procedures have been designed so that SPA will respond to changes in trend.

Field test results confirm the theoretical results presented here. Accordingly, specifications have been written for the inclusion of SPA in the Bell System's standard mechanized trunk forecasting systems.

REFERENCES

1. A. J. David and C. D. Pack, "The Sequential Projection Algorithm: A New and Improved Traffic Forecasting Procedure," International Teletraffic Congress 9, Malaga, Spain, 1979.
2. J. P. Moreland, "Estimation of Point-to-Point Telephone Traffic," B.S.T.J., *57,* No. 8 (October 1978), pp. 2857–63.
3. R. L. Franks et al., "A Model Relating Measurement and Forecast Errors to the Provisioning of Direct Final Trunk Groups," B.S.T.J., *58,* No. 2 (February 1979), pp. 351–78.
4. C. J. Masreliez and R. D. Martin, "Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter," IEEE Trans. Automatic Control (June 1977) pp. 361–71.
5. P. J. Huber, "Robust Estimation of a Location Parameter," Annals of Mathematical Statistics, *35* (1964), pp. 73–110.
6. R. H. Harris and R. E. Sharkey, "Implementing the Sequential Projection Algorithm in an Operating Telephone Company," International Teletraffic Congress 9, Malaga, Spain, 1979.
7. C. D. Pack and B. A. Whitaker, "Kalman Filter Models for Network Forecasting," B.S.T.J., this issue.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 1, January 1982
Printed in U.S.A.

A Sequential Projection Algorithm for Special-Services Demand

By A. IONESCU-GRAFF

(Manuscript received December 31, 1980)

Performance analysis of the four time-independent regression models presently used by Bell operating companies to forecast special-services circuit requirements, and the characteristics of actual special-services demand history observed from three operating companies, indicate a need for a new method to forecast these difficult time series. A new special-services demand sequential projection algorithm (SSD-SPA) is developed based on a linear Kalman filter model. It includes methods to detect previous deterministic events, to accept and process exogenous information affecting the demand, and to recognize and adapt to a "no-growth" situation. Compared to the present algorithm, SSD-SPA generates significantly better forecasts: approximately 30 percent improvement in forecast accuracy and stability, 25 percent reduction in rms error, and 22 percent reduction in circuit misplacements.

I. INTRODUCTION AND SUMMARY

In recent years, the demand for special-services circuits has grown at more than twice the annual rate of the demand for message telephone service (9 percent versus 4 percent). This rapid growth, the development of new technologies, and the problems in the existing special-service provisioning process have led to a reexamination of the overall process of special-services planning and provisioning.

Key inputs to this process are special-services demand forecasts; they are required for the marketing, budgeting, and engineering functions. Presently, in most Bell operating companies (BOCs), the short-range forecast (1 to 5 years) of point-to-point demands for interoffice special-service circuits is provided by forecasting systems based on time-independent trending models or by applying a user specified growth factor to the most recent demand.

Previous studies of the special-service circuits life-time distribution found no single, common distribution that reasonably fit the observed data. The time series consist mainly of small integers with demand levels ranging from very volatile to perfectly constant, and displaying numerous "jumps," probably the effects of deterministic events. These data characteristics explain the inadequacy of the present time-independent (unweighted) regression models used to fit the past data: linear, exponential, and first- and second-order autoregressive.

Consequently, a new algorithm—the special services demand sequential projection algorithm (SSD-SPA)—is proposed, based on a dynamic time-series model with deterministic event input, the Kalman filtering technique for state-vector estimation and prediction, and an additional procedure to process outliers. The attributes and specific parameters of this model are derived from the demand history for special services from three BOCs.

Section II gives background information on the study. It describes the data available for analyses, summarizes the main characteristics of the demand time series to be forecasted, and presents the measures to be used in the empirical investigation of the algorithms' performances. A brief overview of the existing forecasting models, and results of the forecasting algorithm performance analysis follow. A list of the desirable features of a new special-services demand projection algorithm are derived from these results and the characteristics of the actual demand history mentioned in Section 2.1.

In Section III, a linear Kalman filter model is formulated, and the choice of specific parameters is studied. Implementation considerations include initialization, outlier detection, deterministic event (level or growth) detection and processing, as well as filter gain sèlection. Special-services demand sequential projection algorithm forecasts are then tested and compared to the present forecasting algorithm. Results include the comparative forecast qualities for the case of small integer projection and an estimated economic impact of the new algorithm. Finally, conclusions and recommendations are summarized in Section IV.

II. BACKGROUND

Evaluation and comparison of various demand projection algorithms require a description of the characteristics of the time series to be projected, so that the appropriateness of the model can be determined; also, a definition of the performance statistics used for algorithm comparison is needed, so that the best feasible model can be selected.

2.1 Special-services demand

2.1.1 Data for analyses

The term *special services* refers to all Bell System services other than ordinary message telephone service. Examples of special services are foreign exchange, tie lines, off-premise extensions, and private lines. The classification of special-service circuits varies from one BOC to another, and for a single BOC over time. For example, one BOC recognizes about 500 different circuit types, while another recognizes only 150.

Two types of special-services history files were available from three major BOCs to support our studies: detailed-demand history files (DDHF) and grouped-demand history files (GDHF). The maximum number of available months varied among the three BOCs: 60 for Company A, 67 for Company B, and 71 for Company C. The maximum could not exceed 71 months since this is the maximum that can be stored and processed by the present forecasting system.

The DDHF contains *individual* records of the number of special-service circuits of a given BOC class of service between a pair of central offices (COs). The large number of possible point-to-point individual special-service type circuit records on the DDHF (for example, 500 types for each pair of COs times all possible combinations of CO pairs) and the small size of these groups (more than 90 percent have only one circuit) makes any attempt to forecast each time series impractical. Consequently, in the design of the present forecasting system (the special-services forecasting system, or SSFS), the decision was made to group the individual records before projection.

This grouping of DDHF records, according to a user specified grouping strategy, results in a GDHF. The resultant grouped special-service time series are the basic input to the forecasting routines and represent the numbers of circuits of one or more types between a given pair of offices.

For the special-services demand analysis, both types of files were used. For the present forecasting algorithm performance study, only the GDHFs from Companies B and C could be used since only they had the format required by the input routine. These two files were also used for the SSD-SPA performance tests.

Both tapes were created using grouping strategies specified by the facility and equipment planners: 14 grouping types in Company B and 19 types in Company C. The file from Company B covers the time period between January, 1973 and July, 1978 and contains 20,036 such grouped records. The file from Company C extends over the period January, 1973 to November, 1978 and contains 41,073 records.

2.1.2 Demand characteristics

The special-services demand analysis identified the following significant characteristics:

(*i*) Very skewed circuit group size distribution, regardless of the grouping strategy. More than 80 percent of the point-to-point groups consist of less than 10 circuits. Fig. 1 plots the maximum number of circuits in service over the history for each circuit group against the frequency of that particular size. The histograms for the three BOCs are remarkably similar, even though the grouping strategies used were different. The skewness of the size distribution would be accentuated if we had plotted the group sizes at a given point in time, instead of the maximum size over the whole history. In the same time, the long tail of the distribution shows that, although most of the point-to-points are very small in size, the majority of the special-service circuits are placed in a few very large groups. For example, in Company C only 6.5 percent of the groups consist of more than 50 circuits, but these groups are extremely large and account for more than 75 percent of all special circuits in service.

(*ii*) No seasonal pattern.

(*iii*) High volatility of the time series, even at high levels of aggregation.

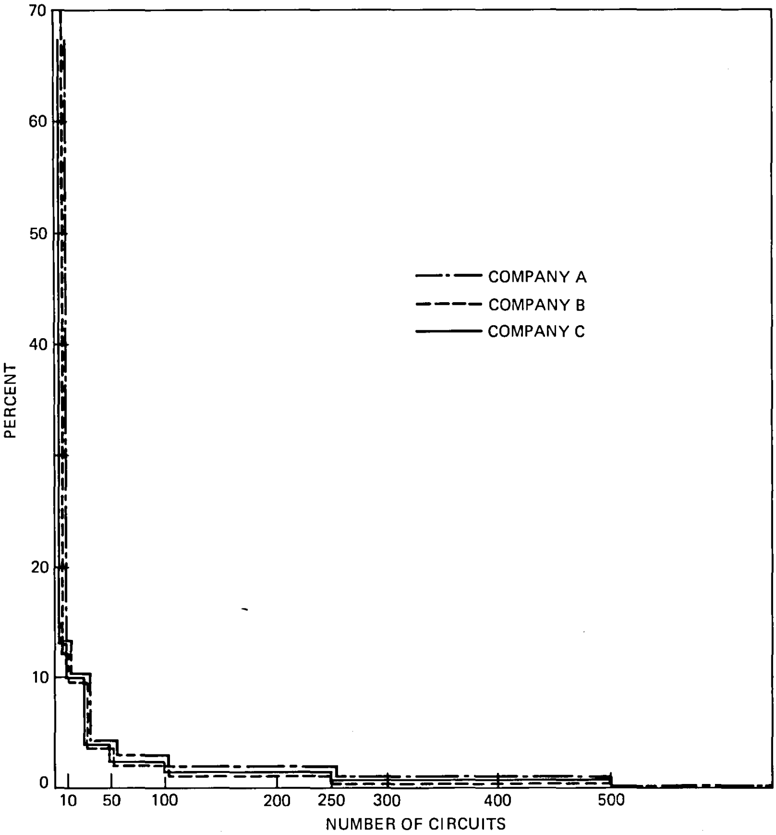

Fig. 1—Maximum demand per circuit forecasting group.

(*iv*) Jumps in the demand level. This is a frequent phenomenon; circuit groups remain at one size for an extended period of time, then jump to another value, and remain at this second level for some time. This stepwise change in the demand level is probably caused by deterministic events, such as large customers moving in or out, routing changes, tariff changes, or market stimulation.

(*v*) Constant level circuit groups. Approximately 40 percent of each company's grouped point-to-point records showed no change in the demand level over the period of time that data were available.

(*vi*) Vanishing circuit groups. About 30 percent of the groups have all their circuits eventually disconnected (i.e., the demand for these groups goes to zero) with practically no regeneration during the period.

2.2 Algorithm performance measures

Previous studies[1,2] indicated that good performance measures for algorithm comparisons are accuracy (average forecast error), rms error (square root of the mean squared error), and stability (the variability of consecutive views of the same future period). For the present analysis, a fourth forecast attribute, misplacement, is defined (total positive or negative forecast error).

To quantitatively measure and compare the performance of both SSFS and SSD-SPA forecasting procedures, we used both algorithms to generate demand forecasts for each circuit group (from the same data base), and then compared the results using relative forecast error statistics.

Let:

$y_{n+k} \equiv$ the recorded number of special-service circuits at time $n + k$
$\hat{x}_{n+k,n} \equiv$ the forecasted demand at time $n + k$, given data through time n; i.e., a k-period forecast.

Then the relative accuracy, $\hat{A}_{n+k,n}$, of the k-period forecast from period n is defined to be

$$\hat{A}_{n+k,n} = \left(\frac{\hat{x}_{n+k,n} - y_{n+k}}{y_{n+k}}\right). \tag{1}$$

The relative rms error, $\hat{R}_{n+k,n}$, is defined to be the square root of

$$\hat{R}^2_{n+k,n} = \left[\frac{\hat{x}_{n+k,n} - y_{n+k}}{y_{n+k}}\right]^2. \tag{2}$$

The relative stability, $\hat{S}_{n+k,n}$, of consecutive forecasts from periods $n - 1$ and n for a fixed target date $n + k$ is defined by:

$$\hat{S}^2_{n+k,n} = \left[\frac{\hat{x}_{n+k,n} - \hat{x}_{n+k,n-1}}{y_{n+k}}\right]^2. \tag{3}$$

Treatment of $y_{n+k} = 0$ is discussed later. Accuracy and stability are actually measuring inaccuracy and instability. Consequently, a decrease in either measure is equivalent to an improvement. All three performance measurements are empirical estimates of the normalized statistics (accuracy, rms error, and stability) as described in Ref. 2. Relative statistics, as opposed to absolute statistics, were used so that a small absolute error on a large group would not obscure large absolute errors on many other small groups. Network estimates of accuracy, stability, and rms error are produced by averaging individual estimates over all groups.

We define total error ($\hat{\text{TE}}$) to be

$$\hat{\text{TE}} = \frac{\text{Total forecast} - \text{total demand}}{\text{Total demand}}.$$

Note that $\hat{\text{TE}}$ can be almost zero as a result of error cancellations; therefore, misplacement is a better measure of the total number of circuits erroneously forecasted. Misplacements translate directly into inefficient capital expenditures.

The positive misplacement, $\hat{M}^{+}_{n+k,n}$, of the total number of circuits forecasted from period n for the target period $n + k$ is defined by:

$$\hat{M}^{+}_{n+k,n} = \left(\frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} y^{(i)}_{n+k}} \right), \tag{4}$$

where $i = 1, 2, \cdots, N$ is the index over all circuit groups in the network and

$$\begin{aligned} d_i &= \hat{x}^{(i)}_{n+k,n} - y^{(i)}_{n+k} \quad \text{if} \quad \hat{x}^{(i)}_{n+k,n} \geq y^{(i)}_{n+k} \\ &= 0 \qquad\qquad\qquad\qquad \text{otherwise.} \end{aligned}$$

Similarly, the negative misplacement, $\hat{M}^{-}_{n+k,n}$, of the total number of circuits forecasted from period n for the target period $n + k$ is defined by:

$$\hat{M}^{-}_{n+k,n} = \left(\frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} y^{(i)}_{n+k}} \right),$$

where $i = 1, 2, \cdots, N$ is the index over all circuit groups in the network and

$$\begin{aligned} d_i &= y^{(i)}_{n+k} - \hat{x}^{(i)}_{n+k,n} \quad \text{if} \quad \hat{x}^{(i)}_{n+k,n} < y^{(i)}_{n+k} \\ &= 0 \qquad\qquad\qquad\qquad \text{otherwise.} \end{aligned}$$

We call $M^{+}_{n+k,n}$ a measure of total network overprovisioning and the negative misplacement, $M^{-}_{n+k,n}$, a measure of total network underprovisioning. Positive misplacement may translate into underutilization, while negative misplacement may translate into orders lost or held, or misroutings.

Note that TE and misplacements are related as

$$\text{TE} = M^{+}_{n+k,n} - M^{-}_{n+k,n} .$$

2.3 Present forecasting algorithm

This section presents a brief overview of the projection algorithm presently used in SSFS, the performance testing procedure, and its results.

2.3.1 Overview

The present forecasting algorithm produces point-to-point demand forecasts of interoffice special-services circuits for the current year and for each of the next 5 years.

The forecast is generated in two major steps—the preliminary forecast and the final forecast. The preliminary forecast employs one of four statistical models or user-stated growth factors to predict future circuit requirements. The four regression models are linear, exponential, and first- and second-order autoregressive. They are used only when the group has sufficient demand history; at least 12 months of history are always required, and the default value is 24 months. Before forecasting, the available history is smoothed using a 3-month moving average.

The parameters for each model are determined by minimizing an unweighted sum of squared errors over the smoothed data. The model with the smallest sum of squared errors or, equivalently, the model with the highest R^2 statistic (the coefficient of determination of "goodness of fit"), is selected. However, the exponential model is rejected if any of the history is zero or if it would lead to a prediction of explosive growth, and the autoregressive models are rejected, unless the demand time series is sufficiently stationary.

Finally, if the model chosen was linear or exponential and the current demand has shifted significantly from the historical growth trend, then the forecast is also shifted to coincide with the current demand. A significant shift is defined relative to the estimated standard error of the unsmoothed demand history (excluding the current demand) from the trend line. Since at each forecast view all history is reprocessed to recalculate the regression parameters, treatment of such discontinuities may be inconsistent from one forecast view to another. The sensitivity of this test may be adjusted by the user; the default value is two standard deviations. No adjustment of this kind is considered for the autoregressive models.

When the forecast groups do not have the required number of months of history, forecasts are produced applying default growth factors to the forecast group's current demand.

The forecaster reviews the preliminary forecast and makes manual adjustments when appropriate. An example is when advance knowledge is available on new businesses moving into an area or new services are being offered.

The following section describes the results of our study to quantify the present forecasting algorithm performance. This analysis only covers the preliminary forecast. The impact of manual adjustments was not studied, since no records were available. The main deficiencies of the existing forecasting technique are summarized and explained in view of the demand time series characteristics.

2.3.2 Performance analysis

The algorithm performance is specified in terms of statistical attributes (accuracy, rms error, stability, misplacements); the analysis sought to verify if there is indeed a benefit in having four different models to choose from, to identify the main forecasting problem, and, based on the demand time series characteristics, to derive requirements for a new forecasting algorithm.

A modified version of SSFS was used to produce up to three consecutive forecasts for each point-to-point demand, depending on the length of each demand history available. To ensure compatibility with other planning tools, SSFS is required to produce quarterly average forecasts of the future demand for special services. The data files available extend up to 71 months, and since SSFS requires at least 12 months of history for the forecast initialization, the longest forecast that can be produced and checked against actuals is 18 to 19 quarters, i.e., about 4 years. For simplicity, instead of estimating 18 to 19 values of $\hat{A}$, $\hat{R}$, $\hat{S}$, $\widehat{\text{TE}}$, and $\hat{M}$, we only looked at one quarter in each year (the same quarter each year, right justified by the last quarter of available data). Consequently, for those records with at least 60 months of history, three forecasts were provided, as shown in Fig. 2 (1 year initialization plus 4-year-span forecast, then 2 years initialization and 3-year-span forecast, and 3 years initialization and 2-year-span forecast). Only two forecasts were produced for records with 48 to 59 months of history (3- and 2-year-span forecasts), and only one forecast for records with 36 to 47 months of history.

Since each forecast is made after at least 12 months of data are processed, only a steady-state analysis is necessary. Thus, the subscripts n for $\hat{A}_{n+k,n}$, $\hat{R}_{n+k,n}$, and $\hat{S}_{n+k,n}$ are dropped. For each circuit

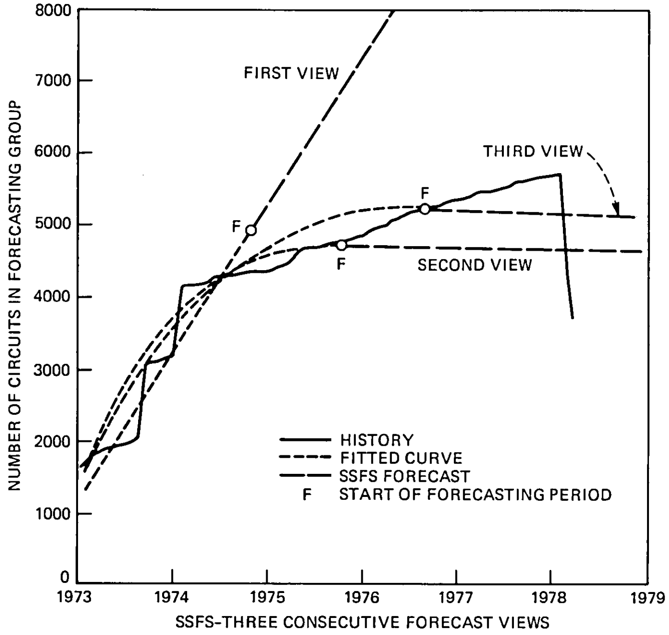

Fig. 2—Algorithms performance analysis test plan.

group, accuracy, rms error, and stability are estimated using all forecasts produced. For example, the accuracy of a 3-year-ahead forecast for a circuit group with 60 months of history available is estimated by:

$$\hat{A}_3 \triangleq \frac{1}{2}\left(\frac{\hat{x}_{77,74} - y_{77}}{y_{77}} + \frac{\hat{x}_{78,75} - y_{78}}{y_{78}}\right),$$

where, if q is the last quarter of available data in the last year of history, then

$\hat{x}_{i,j}$ = forecast of the average demand in quarter q; year i, made from quarter q; year j.

$\hat{y}_i$ = average measurements of the demand in quarter q; year i.

Or, stability of 3- versus 4-year-ahead forecasts for the same group is estimated by:

$$\hat{S}_3 \triangleq \left|\frac{\hat{x}_{78,74} - \hat{x}_{78,75}}{y_{78}}\right|.$$

For groups where $\hat{y}_{n+k} = 0$, a normalization factor of 1 is used. This may bias the statistics (to look worse than they actually are), but since the objective is to *compare* the performance of different algorithms, and they all use this rule, we may expect this normalization to affect them equally. In fact, the performance results measured using this normalization were similar to those obtained with nonvanishing groups only. Three consecutive forecasts were produced for about 56 percent of the groups (those groups with 60 to 67 months of history*), two forecasts for about 10 percent (48 to 59 months of history), and only one forecast for 9 percent (36 to 47 months of history). The remaining 25 percent of the groups were not used, since their recorded histories were shorter than 36 months.

2.3.2.1 Statistical performance. The results showed that the demand forecasts are often inaccurate and unstable. The numerical results can be deduced from the values presented in Section 3.3 on the SSD-SPA performance, and relative improvement versus the present SSFS.

The accuracy histogram showed about 40 percent of the groups had 1-year forecasts with no error. This was to be expected since about 40 percent of the point-to-point groups have constant demand. Additionally, the existing forecasting algorithms more often overforecasted than underforecasted.

The significant instability observed for consecutive forecasts was due not only to the intrinsic volatility of the demand time series, but also to the change in forecasting models used each year.

Small total errors resulted from cancellations of up to 50 percent misplaced circuits (large total misplacement). It was interesting to observe that although accuracy was always positive, many times (especially for Company C) the total error was negative. This means that even if on the average most of the circuit groups are overforecasted, some of the very large point-to-point groups are underforecasted so that the total forecast over the whole company is less than the realized demand.

2.3.2.2 Correlation between forecasting model fit (R^2) and projection error (accuracy and rms). As previously described, the existing algorithm chooses from the four regression models the one with the highest coefficient of determination (R^2, or "goodness of fit"). The intuitive reason for this is that the curve that best fits the past data should extrapolate most accurately into the future. If indeed, there is a benefit in having four different models from which to choose, then we would expect to find some correlation between how well the chosen models fit the data (R^2) and the forecast quality. Subsequent testing, designed to consider all combinations of models and forecasting spans, showed that the choice of four projection models appeared unjustified since the correlation between the goodness of the model fit to the history

* The number of months of history refers to how long ago the first circuits were installed on that group, not to the actual length of time the demand was nonzero. About 30 percent of the groups with more than 36 months of recorded history vanished during that period (demand had zero value eventually).

data (R^2) and the forecast errors (accuracy or rms) was statistically insignificant. In other words, even perfect knowledge of the past does not necessarily imply good knowledge of the future.

2.3.2.3 Outlier detection procedure. Many of the demand time series display a stepwise, highly volatile growth pattern, with the jumps probably generated by deterministic events. The existing shift option reacts to a significant difference between the actual demand and the forecasted value only if it happens in the last month of history. Any other jumps in previous months are treated as normal trend. Moreover, the error monitoring capability, which can detect large forecast deviations from the actual demand, is exterior to the main forecasting process. Consequently, the next projection cannot be improved based on the detected past errors. Figs. 3a and 3b give examples when the wrong model or parameters were selected for projection because of improper treatment of past special events.

Another deficiency is the rather slow response to changes in trend; the equal weight assigned to each history point prevents the system from adjusting itself quickly to recent changes.

2.3.2.4 Manual adjustments. The present forecasting algorithm cannot accept and process exogenous information. The forecaster has to review manually the forecasts and supply any modifications based on up-to-date knowledge. For example, about 70 percent of the forecasts in Company C are manually adjusted.

2.3.2.5 History requirements. The special-services forecasting system needs at least 12 months (usually 24 months) of history to produce a forecast based on one of the four statistical models. If less data are available, growth factors are used (default or manually input values);

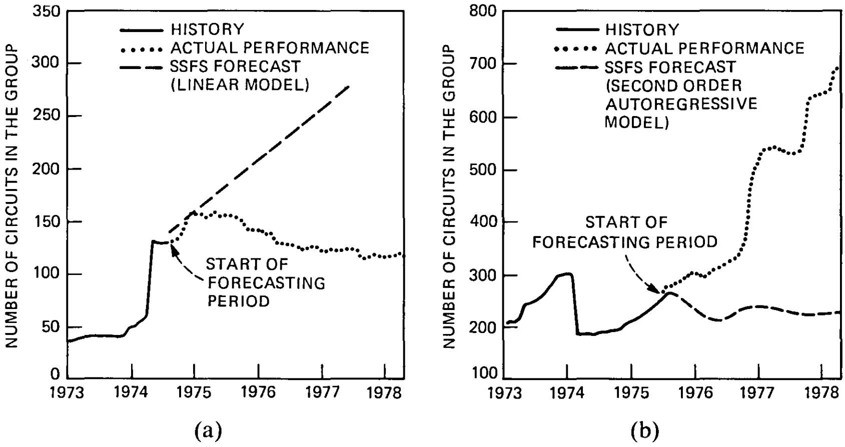

Fig. 3—Circuit groups with deterministic "jumps." SSFS forecasting problem: special event treated as normal growth. (a) Example 1; (b) Example 2.

15.2 percent of the Company B data base and 17 percent of the Company C data base consist of circuit group records with less than 24 months of history.

2.3.3 New algorithm requirements

The demand time series characteristics and the results from the present algorithm's performance suggest some desirable properties of a new algorithm:

(*i*) Unequal weighting of data. Weigh the most recent data more heavily to allow the forecasting algorithm to adapt to dynamic changes in the demand pattern.

(*ii*) Acceptance of exogenous information. Point-to-point demand levels are significantly affected by special events, such as large customers moving in or out, tariff changes, or by market stimulation. Many of these events are known in advance and their impact on the individual time series can be estimated. The forecasting system should accept those estimates and use them in projecting future demand levels.

(*iii*) Shorter initialization period. The special-services segment of the total Bell System network is constantly changing and growing. New technologies, services, and rates are changing the customer demand patterns. Many special-service circuit groups are eventually disconnected (on an individual basis, 50 percent of the special-services circuits had a lifetime of less than 36 months; and 30 percent of the total number of groups "died" over a 5-year period); other groups come into service. Thus, a forecasting system must produce accurate forecasts based on small amounts of historical data, e.g., less than 12 months.

(*iv*) Recognition of past deterministic events (step changes and constant levels). The system should be able to recognize and react to "significant" changes in the demand level. Significant has to be defined as a function of the observed demand time series characteristics.

(*v*) Forecast of small integers. About 80 percent of the special-services circuit groups have less than 10 circuits in service. Whatever the model selected for projection, it should produce stable and accurate forecasts of integers from 1 to 10.

(*vi*) Computational efficiency. Users find it useful to run SSFS on a monthly basis.

III. SPECIAL-SERVICES DEMAND SEQUENTIAL PROJECTION ALGORITHM

A linear dynamic system with linear growth and deterministic input is shown to be a reasonable and robust approximation for the special-services demand time series, and a simple (two-dimensional) linear

Kalman filter is selected as a method to estimate the state variables of this system. Filter parameter selection is examined and procedures to detect and respond appropriately to outliers are added to capture the stepwise growth pattern of the demand time series. Using data from Companies B and C, we test the performance of this new algorithm and compare it with the present algorithm.

3.1 Linear Kalman filter model

3.1.1 Model formulation

In a linear dynamic model, as discussed in Reference 2, the behavior of the discrete time series is determined by an s-dimensional state-vector process $\{\mathbf{X}_n\}$. The following two equations describe the time evolution of the process $\{\mathbf{X}_n\}$ and the relation between $\mathbf{X}_n$ and the corresponding observation $\mathbf{y}_i$:

$$\text{System equation: } \mathbf{X}_{n+1} = \phi_n \mathbf{X}_n + \mathbf{U}_{n+1} + \omega_{n+1} \tag{5}$$

$$\text{Observation equation: } \quad \mathbf{y}_n = H_n \mathbf{X}_n + \nu_n\,, \tag{6}$$

where ϕ_n is an $s \times s$ state transition matrix, ω_n is an s-dimensional modeling error vector, $\mathbf{U}_n$ is an s-dimensional deterministic input, H_n is a $d \times s$ observation matrix, and ν_n is a s-dimensional measurement noise vector. Furthermore,

$$E(\nu_n) = E(\omega_n) = 0$$

$$E(\omega_n \omega_j^T) \equiv \begin{cases} 0 & \text{if } n \neq j \\ Q_n & \text{if } n = j \end{cases} \qquad Q_n \text{ an } s \times s \text{ matrix}$$

$$E(\nu_n \nu_j^T) \equiv \begin{cases} 0 & \text{if } n \neq j \\ R_n & \text{if } n = j \end{cases} \qquad R_n \text{ an } d \times d \text{ matrix}$$

$$E(\nu_n \omega_j^T) \equiv 0 \qquad \text{for all } n, j.$$

The $s \times s$ matrix, Q_n, is known as the modeling error covariance matrix and R_n is the measurment error covariance matrix.

In our demand analysis, it was demonstrated that no single common demand pattern exists for special services, but that for the majority of groups a linear model fit the historical data best. Furthermore, earlier work using Kalman filters for forecasting message trunk group loads,[1,3] showed that for short-term forecasting applications a linear Kalman filter model performs well even for nonlinear processes such as an exponential.* Consequently, we chose to develop a linear model that accounts for the special-demand characteristics discussed earlier.

* Reference 1, for example, analyzed the performance of different Kalman filter models (linear, log-linear, etc.) to forecast busy season trunk group loads. It showed that, given measurement and modeling errors and errors in the initial state estimates, the linear Kalman filter would produce short-term (1 to 5 years) forecasts as good as any other model with respect to accuracy, rms, and stability measures.

Given the univariate measurement time series (i.e., $d = 1$) with linear growth, the special-services demand model can be represented by a two-dimensional linear model with the following parameters:

$$\phi_n \equiv \phi = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}; \qquad H_n = (1, 0); \tag{7}$$

$$\mathbf{X}_n = \begin{pmatrix} x_n^1 \\ x_n^2 \end{pmatrix}, \tag{8}$$

where x_n^1 represents the demand level at time n, and x_n^2 the incremental growth.

3.1.2 Kalman filter (filtering and prediction)

The Kalman filter is a recursive method that produces a minimum variance unbiased estimate of the state vector $\{\mathbf{X}_n\}$ of a dynamic linear system from noisy observations $\mathbf{y}_1, \cdots, \mathbf{y}_n$ and uses these estimates to predict future state values.

Let $\hat{\mathbf{X}}_{n,n-i}$ be the estimate of the state vector $\mathbf{X}_n$ based on information available through time $n - i$. Let

$$P_n = E[(\mathbf{X}_n - \hat{\mathbf{X}}_{n,n-1})(\mathbf{X}_n - \hat{\mathbf{X}}_{n,n-1})^T]$$

be the one-step prediction error covariance matrix and

$$S_n = E[(\mathbf{X}_n - \hat{\mathbf{X}}_{n,n})(\mathbf{X}_n - \hat{\mathbf{X}}_{n,n})^T]$$

be the estimation error covariance matrix. Then, given a prior estimate of the system state $\hat{\mathbf{X}}_{n,n-1}$, the filtering problem is to find an updated estimated $\hat{\mathbf{X}}_{n,n}$, based on the measurement $\mathbf{y}_n$.

The unbiased estimate is given by the linear recursive form

$$\hat{\mathbf{X}}_{n,n} = \hat{\mathbf{X}}_{n,n-1} + K_n(\mathbf{y}_n - H_n\hat{\mathbf{X}}_{n,n-1}), \tag{9}$$

where K_n is a time-varying weighting matrix known as the Kalman filter gain matrix. The optimal* K_n is given by

$$K_n = P_n H_n^T (H_n P_n H_n^T + R_n)^{-1}. \tag{10}$$

The error covariance matrices are found to be:

$$S_n = (I - K_n H_n) P_n, \tag{11}$$

where I is the $s \times s$ identity matrix, and

$$P_{n+1} = \phi_n S_n \phi_n^T + Q_n. \tag{12}$$

* The criterion of optimality is to minimize the mean square estimation error. When ω_n and ν_n are normally distributed, then the same result is obtained by a Bayesian method or the method of maximum likelihood.

The estimates of the future state vectors are obtained by extrapolation using eq. (5)

$$\begin{aligned}\hat{\mathbf{X}}_{n+k,n} &= \phi_{n+k-1}\hat{\mathbf{X}}_{n+k-1,n} + \mathbf{U}_{n+k} \\ &= \left(\prod_{l=0}^{k-1} \phi_{n+l}\right)\hat{\mathbf{X}}_{n,n} + \sum_{l=1}^{k-1} \mathbf{U}_{n+l}\left(\prod_{m=l}^{k-1} \phi_{n+m}\right) + \mathbf{U}_{n+k}\,. \qquad (13)\end{aligned}$$

It should be mentioned that if ω_n and ν_n are Gaussian, the Kalman filter estimate is at least as good as any other estimate (either linear or nonlinear). If the noise terms cannot be assumed normal, then the Kalman filter yields the optimal linear unbiased minimum variance estimate, but there may be a nonlinear estimate that is superior in mean square error.[4,5]

To implement the above described algorithm, we note that:

(*i*) An initial state estimate and error covariance are necessary to start the recursion. This problem will be considered in Section 3.2.1.

(*ii*) Since the estimation error covariance matrix S_n and prediction error covariance matrix P_n do not rely on observed data, for given sequences $\{Q_n\}$, $\{R_n\}$, and initial P_{n_0},* the gain sequence $\{K_n\}$ can be precalculated. Specification of Q_n and R_n will be discussed in Section 3.2.2. The choice of the gain sequence will be examined in Section 3.2.3.

(*iii*) It is not necessary to store the history $\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$ since all relevant information concerning the series is included in the state vector estimate $\hat{\mathbf{X}}_{n,n}$.

(*iv*) The algorithm assumes the knowledge of the future deterministic events $\{\mathbf{U}_n\}$. If estimates of the impact of these events are not available (user input) or are in error, the system needs a recovery procedure. However, when a significant change in the demand is observed, the algorithm has to differentiate between outliers (because of measurement errors, or demand volatility) and deterministic events. The problems of outlier detection and response to special events are considered in Sections 3.2.4 and 3.2.5.

3.2 Implementation considerations and parameter selection

In most applications, the exact statistical structure of the individual time series is unknown. Consequently, implementation of the Kalman filter model requires selection of estimated values for the algorithm parameters, usually through experimentation. Three methods to obtain initial estimates for $\hat{\mathbf{X}}_{n_0,n_0-1}$ and $\hat{P}_{n_0}$ will be analyzed next. Then, the specification of R and Q and the choice of gains and outlier thresholds will be considered.

* Time n_0 is the assumed "present time" for filter initialization, given the available data history $\{y_1, \cdots y_{n_0}, \cdots y_n\}$.

3.2.1 Filter initialization

Although the special-services segment of the total network is rapidly growing (at a rate of more than 9 percent per year) there are very few circuit groups or point-to-point disaggregated groups constantly growing. Most of them vary around a constant value, many of the groups have all their circuits disconnected (about 30 percent of the groups "die" over a 5-year period), and more new groups appear.

This frequent in-and-out activity rules out delaying the forecast until sufficient data is available to make accurate estimates of $\mathbf{X}_{n_0,n_0-1}$ and P_{n_0}. It is important to have initial state-vector estimates as soon as observations are available.

We considered three filter initialization methods for implementation in SSFS. Subsequent testing on actual data files (described in Section 3.3) was used to decide on the most appropriate one. Each method assumed that the length of every circuit group history is somewhere between 2 and 71 months.* As mentioned in Section 2.2, we look at quarterly average values of the demand for special services. The three methods are the following:

(*i*) Estimate $\hat{\mathbf{X}}_{n_0,n_0-1}$ and $\hat{P}_{n_0}$ by unweighted least squares. Assume a linear first-order model of the form $y = \beta_0 + \beta_1 z + \epsilon$, where z is the time variable (in our case, it is just the index of the observations, since the seasonal analysis can be assumed equally spaced in time), and ϵ, an error variable with mean zero and unknown variance σ^2. Given the observations $y = (y_0, y_1, \cdots, y_{n-1})$ taken at times $z = (0, 1, \cdots, n-1)$, y_n is estimated (by least squares) by $\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 z_n$ and $\hat{y}_0 = \hat{\beta}_0$. Therefore,

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 z_n = \hat{\beta}_0 + \hat{\beta}_1 n = \hat{y}_{n-1} + \hat{\beta}_1,$$

$$\hat{x}^1_{n_0,n_0-1} = \hat{y}_n,\ \hat{x}^2_{n_0,n_0-1} = \hat{\beta}_1,\ \hat{p}^{11}_{n_0} = \text{var}\ \hat{y}_n,$$

$$\hat{p}^{22}_{n_0} = \text{var}\ \hat{\beta}_1, \quad \text{and} \quad \hat{p}^{12}_{n_0} = \hat{p}^{21}_{n_0} = \text{cov}(\hat{y}_n, \hat{\beta}_1)$$

$$= \tfrac{1}{2}(\text{var}\ \hat{y}_n + \text{var}\ \hat{\beta}_1 - \text{var}\ \hat{y}_{n+1}).$$

The estimates $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{y}_n$, and $\hat{\sigma}^2$ are obtained with the usual regression formulas (as in Ref. 6).

(*ii*) Use the present SSFS model prediction for $\hat{\mathbf{X}}_{n_0,n_0-1}$ and estimate $\hat{P}_{n_0}$ from method (*i*).

(*iii*) Use the first quarter of data (2 or 3 months) to obtain a crude estimate $\hat{\mathbf{X}}_{1,1}(\hat{x}^1_{1,1}$ = quarterly average, $\hat{x}^2_{1,1}$ = the slope of a line best fitting the data). Then use the Kalman filter algorithm sequentially on each quarterly average demand up to the current date. Estimate, as in

* When only one month is available, an arbitrary growth factor has to be used, and 71 months is the maximum history length stored by the SSFS history files.

(*i*), the initial error covariance matrix as the average of the errors in extrapolating for the 3 months in the second quarter.

Intuitively, this last method was expected to perform best, since, in most of the cases, enough history was available for the filter to achieve steady-state performance. Experimental testing of these initialization procedures indicated that indeed method (*iii*), the use of the filter algorithms as early in the history as possible, resulted in the best initial state-vector estimates $\hat{\mathbf{X}}_{n_0,n_{0-1}}$ and $\hat{P}_{n_0}$.

Weighted least squares, using minimum variance initialization estimates,[7] was not considered because special-services demand data may have many deterministic jumps. A distinction between these jumps and possible outliers could not be made since there were no records to indicate when such significant events occurred. This lack of information is equivalent to changing the characteristics of the vector $\mathbf{U}_n$ in eq. (5) into a random variable with unknown distribution.

3.2.2 Specification of R and Q

Various procedures exist for the estimation of the $\{R_n\}$ and $\{Q_n\}$ parameters. The methods vary in their relative complexity and the number of assumptions needed for the underlying statistical properties of the system. In most applications with relatively short time series, little improvement in performance is expected from a highly sophisticated specification procedure. A simpler method is used: instead of trying to identify $\{R_n\}$ and $\{Q_n\}$ for each individual time series, a scalar R and a matrix Q are determined that approximate the general nature of all series considered. Consequently, only one gain sequence $\{K_n\}$ and initialization matrix P_{n_0} has to be precomputed and applied to all circuit histories.

In our case, estimation of R and Q is obscured by the occurrence of deterministic events. For example, to estimate R, the series first has to be cleansed of special events, but any special events recognition is based on a measure of R. Nevertheless, upper bounds for the measurement error variance can be estimated assuming no deterministic events. The estimate measurement error, R, was found to be approximately 5 for Company B and 19 for Company C.

It can be shown[4,8] that the calculation of the gain sequence depends only on the relative magnitude of Q compared to R. Hence, if R is normalized to unity, only Q needs to be specified. We discuss the influence of R and Q on the gain sequence, filter responsiveness, and the selection of specific values for Q, based on experimental testing, in the next section.

3.2.3 The gain sequence

Accurate specification of the elements of Q is important, especially as they affect the gain sequence $\{K_n\}$ values for large n. Fig. 4 shows

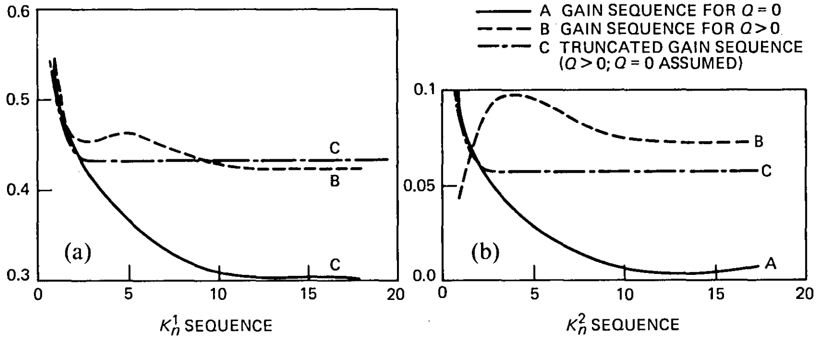

Fig. 4—Kalman filter gain sequences. (a) K_n^1 sequence; (b) K_n^2 sequence.

how the gain sequence is influenced by different assumptions made on the elements of Q. For $Q \equiv 0$ the gain sequence converges to zero, since $Q = 0$ is equivalent to a process $\{\mathbf{X}_n\}$ evolving deterministically relative to the initial set of parameters (see Fig. 4, curve A). A nonzero Q will force the sequence $\{K_n\}$ to give enough weight to new observations $\mathbf{y}_n$ so that a nonstationary process is correctly tracked by the filter (Fig. 4, curve B).

The choice of $\{K_n\}$ is based on the desire to be responsive to changes in demand, while maintaining relatively stable forecasts. To obtain this result even when the true statistical nature is not known and Q estimates are difficult to obtain, a truncated gain sequence can be used.[1,7] A truncated gain sequence K'_n (Fig. 4, curve C) is defined as

$$K'_n = \begin{cases} K_n & \text{if} \quad n \le n^* \\ \\ K_{n^*} & \text{if} \quad n > n^* \end{cases} \qquad \text{and} \quad n^* \ge 1,$$

where K_n is calculated under the false assumption that $Q \equiv 0$, and n^* is empirically determined to ensure sufficient responsiveness and near-optimal filter performance. Another advantage in using the K'_n sequence over the optimal sequence is that a finite vector $\{K_1, \cdots, K_{n^*}\}$ can be computed and stored. [Fig. 4 is derived from eqs. (10), (11), and (12), and estimates of R, Q, and P_{n_0}.]

Given the demand data characteristics in our case, the matrix Q had to be nonzero to ensure filter responsiveness to random variation in the model parameters. For the normalized R_0 value of 1, different matrices Q were tested and corresponding steady-state gains calculated.

Figure 5 shows the theoretical performance of different gain sequences for the SSD-SPA algorithm when the true modeling error covariance matrix is constant and not zero ($Q \neq 0$): Curve A is the theoretical performance when the gain sequence is calculated under the false assumption that $Q_n \equiv 0$, curve B is obtained when the true Q

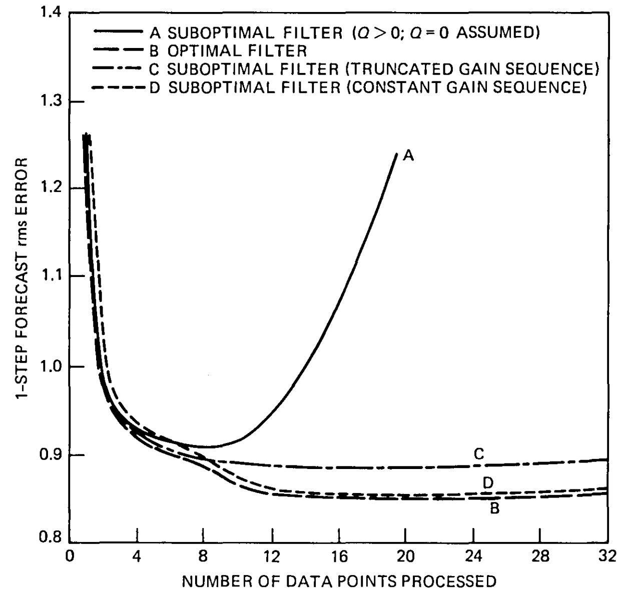

Fig. 5—Theoretical algorithm performance.

is used in obtaining the gain sequence, while curve C shows the theoretical performance of the truncated gain sequence ($n^* = 10$), also computed under the assumption that $Q_n \equiv 0$. This figure is derived from the generalized formula for S_n

$$S_n = (I - K_nH_n)P_n(I - K_nH_n)^T + K_nR_nK_n^T,$$

which is true for an arbitrary gain matrix K_n.[4,7]

The initial values of the gain sequence depend on the $\hat{P}_{n_0}$ values. The absolute values of the $\hat{P}_{n_0}$ were varied to obtain the transient gains (K_n, $n \leq 14$) that gave the best algorithm performance. This $\hat{P}_{n_0}$ and Q produced a near constant (over time) gain-vector sequence. Further testing on many time series indicated that a single algorithm using constant gains performs as well as any other. This empirical result is substantiated by theoretical near-optimal performance of a constant gain sequence, shown in Fig. 5, curve D. The constant gain was selected to be the best approximation obtained to the optimal gain. Consequently, constant gains were selected for the SSD-SPA implementation.

3.2.4 Outlier detection and data validation

An important characteristic of the special-service demand time series is the high volatility. This volatility affects the expected quality of the forecasting procedures and the determination of outlier detection screens and responses. Usually, outliers are defined as those measurements that are significantly different from the trend because of data volatility, measurement errors, or deterministic changes in the level. Significance is determined based on the observed measurement variance about the assumed trend line (usually a band of 2 to 3 standard deviations around the expected line). Once a measurement falls beyond these boundaries, the outlier detection routine determines whether the measurement indicates a change in the level of the trend or whether it is an outlier (data volatility or measurement error). The former case is decided based on subsequent measurements, i.e., if the following data conform to the change. In the latter case, the measurement has to be partially or complete ignored, based on the probability of being a true measurement of volatile demand or an erroneous one.

Previous studies indicated that the majority of the grouped demand time series is truly of a highly volatile nature with sudden changes; large and frequent changes (up to 1000 circuits added or subtracted in a single month), many times in opposite directions (big rise in level followed shortly by a big drop), were evident.

Consequently, it is impossible in the case of the special-services demand data to decide if an outlier was produced by a data-base error; therefore, no data validation decisions are recommended for the outlier detection routines.

Instead, such errors if present will be handled by the deterministic event detection and response procedure described in the next section.

3.2.5 Deterministic events detection and response

As mentioned in Section 2.1, two other important demand patterns must be considered in designing a projection algorithm:

(*i*) Significant changes in the demand level when subsequent observations confirm the supposition that a special event had taken place.

(*ii*) Zero growth when the time series remain for long periods of time at constant levels.

3.2.5.1 Step changes in demand levels. Since data is available monthly, detection of significant level changes should be made monthly even though the forecast is made quarterly. In this way, a quarterly response will be the result of at least three, and up to a maximum of six monthly movements. Let

$$d_i = y_{\text{month}_i} - y_{\text{month}_{i-1}},$$

$$\bar{d}_i = (y_{\text{month}_i} + y_{\text{month}_{i-1}})/2, \text{ and}$$

d_i^* = maximum value $|d_i|$ can have before it is considered significant.

Two functional forms are usually used[1,3] to calculate d_i^*: the linear function $d_i^* = a + b\bar{d}_i$ or the mixed linear-exponential function $d_i^* =$

$\bar{d}_i(a + be^{c\bar{d}_i})$. In general, the latter form has the advantage that it increases the boundaries, percentagewise for small values of $\bar{d}_i$. For the special-services demand time series, the integral nature of the d_i's made this advantage insignificant. Optimum values for a, b, and c parameters were experimentally tested for both functional forms, but no improvement was found in the algorithm performance when the mixed linear-exponential boundaries were used. Since the simple linear form reduced the total computational time, we recommended the following linear deterministic event boundaries for SSD-SPA:

$$d_i^* = 0.7 + 0.11\,(y_{\text{month}_i} + y_{\text{month}_{i-1}}) \quad \text{for all} \quad i \geq 2.$$

We present next a brief description of the detection and response to past deterministic changes in the demand.

(*i*) *Detection step*

This procedure first determines if the given d_{i-1} is significant, and second if the subsequent d_i confirms this event. This confirmation means that either d_i has the same sign as d_{i-1}, or the net difference between $|d_i|$ and $|d_{i-1}|$ is large enough to be a deterministic event by itself. There are four possible cases: $d_{i-1} > 0$ and $d_i > 0$; $d_{i-1} < 0$ and $d_i < 0$; $d_{i-1} > 0$ and $d_i < 0$; and $d_{i-1} < 0$ and $d_i > 0$. In the first two cases, d_{i-1} is confirmed, since the next movement has the same sign. In the last two cases, the movements have opposite signs. To distinguish between volatility and special events, we subtract from both what can be attributed to volatility, i.e., minimum $\{|d_i|, |d_{i-1}|\}$. There are, then, two cases: $|d_{i-1}| > |d_i|$, and $|d_{i-1}| < |d_i|$.

Case 1: $|d_{i-1}| > |d_i|$. Then new $d'_{i-1} = d_{i-1} + d_i$ and new $d'_i = 0$. If d'_{i-1} compared to d^*_{i-1} is significant, then d'_{i-1} is a special event and $d'_i = 0$. If not, both d'_i and d'_{i-1} are zero.

Case 2: $|d_{i-1}| < |d_i|$. Then new $d'_i = d_i + d_{i-1}$ and new $d'_{i-1} = 0$. If $|d'_i| \geq d^*_i$, then d'_i is a special event and $d'_{i-1} = 0$. Otherwise both are zero.

(*ii*) *Response step*

From these monthly detected level changes, the quarterly events have to be calculated. A quarterly value Y_j is an average of three months: y_i, y_{i+1}, and y_{i+2}. (The Kalman filter model uses this Y_j as data, as described in paragraphs 3.2.1 and 3.2.2.) The effect of a monthly change on the quarterly average depends on the position of the month in that quarter. If the change happens in the first month of the quarter (d_i), then all y's in Y_j are moved to this second level, and the change in quarterly averages is $D_j = Y_j - Y_{j-1} = d_i$. If the change happens in the second month (d_{i+1}), then only y_{i+1} and y_{i+2} reflect the change, and $Y_j - Y_{j-1} = \frac{2}{3}d_{i+1}$. The remaining $\frac{1}{3}d_{i+1}$ will appear as a difference between Y_j and Y_{j+1}. Finally, if the change appears at y_{i+2},(d_{i+2}), then $Y_j - Y_{j-1} = \frac{1}{3}d_{i+2}$ and $Y_{j+1} - Y_j = \frac{2}{3}d_{i+2}$.

Consequently, the changes observed on quarterly values are determined by five possible monthly events:

$$D_j = \tfrac{1}{3}d_{i-2} + \tfrac{2}{3}d_{i-1} + d_i + \tfrac{2}{3}d_{i+1} + \tfrac{1}{3}d_{i+2}\,.$$

Since we do not know how much of any D_j is actually normal growth, we recommend that when D_j fully explains the difference between Y_j and Y_{j-1}, to consider that it already included the growth. Certainly D_j is not available before Y_j, and therefore, the state estimate $\hat{x}^1_{j,j,-1}$ is calculated first using the estimates $\hat{u}^1_{j,j-1}$ of future deterministic events which can be input to SSD-SPA, or from eq. 13:

$$\hat{\mathbf{X}}_{j,j-1} = \Phi\hat{\mathbf{X}}_{j-1,j-1} + \hat{\mathbf{U}}_{j,j-1}.$$

After D_j can be calculated ($D_j = \hat{u}^1_{j,j}$ = estimate of u^1_j after Y_j has been observed), then the state estimate $\hat{x}^1_{j,j-1}$ can be updated by:

$$\begin{aligned} \hat{x}^1_{j,j-1} &\leftarrow \hat{x}^1_{j,j-1} - \hat{u}^1_{j,j-1} + D_j \quad \text{if} \quad D_j \neq Y_j - Y_{j-1} \quad \text{or} \quad D_j = 0 \\ &\leftarrow \hat{x}^1_{j,j-1} - \hat{u}^1_{j,j-1} + D_j - \hat{x}^2_{j-1,j-1} \quad \text{if} \quad D_j = Y_j - Y_{j-1} \quad \text{and} \quad D_j \neq 0. \end{aligned}$$

Then, eq. 9 follows to calculate $\hat{\mathbf{x}}_{j,j}$.

Since events often do not occur as planned, this procedure also ensures algorithm recovery when erroneous estimates of future deterministic events are input to SSD-SPA.

3.2.5.2 Zero growth. Two quarters (or 6 months) with constant level of demand are regarded as sufficient evidence that the main tendency of that particular circuit group is to stay at that level for a longer period of time. However, if the filter estimate of the growth is not very close to zero, it may take many quarters to finally converge to zero since the filter has to be robust enough to perform on other higher volatile series. An appropriate procedure to force the growth estimate to converge to zero faster is to reduce the growth estimate ($\hat{x}^2_{n,n}$) by a factor γ (i.e., $x^2_{n,n}$ becomes $x^2_{n,n}/\gamma$) whenever zero quarterly growth is observed. Subsequent testing found $\gamma = 2$ to be a good value and concluded that this test is very robust for small variations of the γ parameter.

3.3 Performance analysis

Three objectives were identified for the SSD-SPA performance analysis: First, to determine and quantify the improvement in forecast accuracy, rms error, stability, and misplacements relative to the existing forecasting algorithm (described in Section 2.3). Second, to determine if the proposed algorithm has the desired properties (listed in Section 2.3.3) derived from the special-service demand data characteristics. Third, to assess the potential economic benefits resulting from incorporating SSD-SPA into SSFS versus its implementation costs.

The SSD-SPA was evaluated quantitatively using the accuracy, rms error, stability, and misplacement relative error statistics. To ensure relative algorithm performance analysis consistency, the same data was used as in the SSFS study (Sections 2.1 and 2.3.2). For these time series, equivalent consecutive forecasts were produced using the new sequential projection algorithm, and forecast performance measures were calculated.

Network aggregated error statistics were used in the selection of algorithm parameters, as well as in comparing the new SSD-SPA performance to the present algorithm.

The resulting forecasting algorithm was found to be robust over small variations of all parameters around their optimum values.

3.3.1 Results: Accuracy, rms error, stability, misplacements

Figure 6 displays graphically the performance of the SSD-SPA using both companies' history data. Special-services demand sequential projection algorithm generates forecasts that are significantly more accurate and stable. Tables I and II give the SSD-SPA versus present algorithm relative improvements in forecast accuracy, rms error, stability, and total misplacement.

Figures 7a and 7b present two examples of the SSD-SPA versus present algorithm total error (TE) and misplacement (M) relative improvements for the forecasts generated in 1974.

Figure 8 gives histograms of relative improvement for the 1-year-ahead forecast accuracy, rms error, total misplacements, and stability

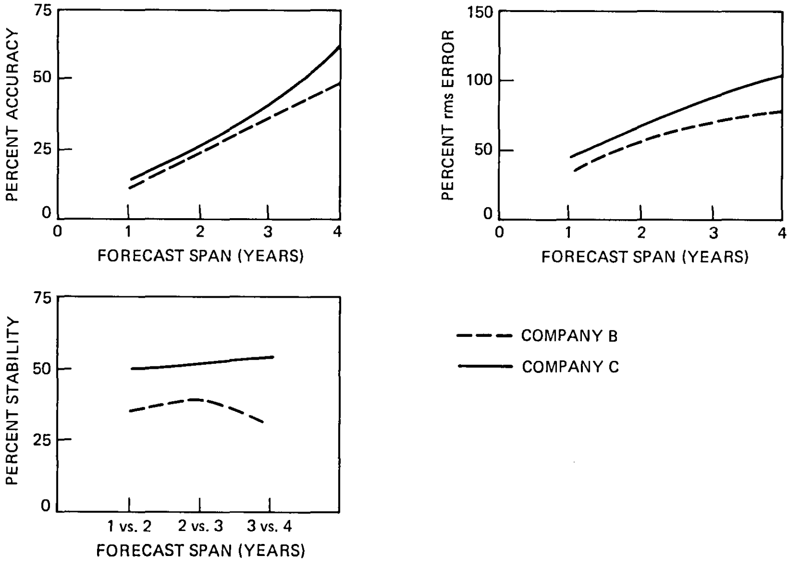

Fig. 6—SSD-SPA network average forecasting performance.

Table I—SSD-SPA Percent relative forecast improvement over present algorithm

Forecast Attribute	Company	1-Year Span	2-Year Span	3-Year Span	4-Year Span
Accuracy	C	30.0	29.6	29.6	31.9
	B	25.0	24.0	25.0	29.0
Root-mean-square	C	17.8	18.3	20.7	23.0
	B	19.7	19.1	22.2	24.0
Stability	C	15.0	27.0	38.8	
	B	20.0	33.0	51.0	

Table II—SSD-SPA Percent relative improvement in total misplacements over present algorithm

Company	Base Year Forecast	Forecasted Year			
		1975	1976	1977	1978
C	1974	17.4	19.5	20.2	21.6
	1975		18.3	15.7	13.7
	1976			18.3	13.1
B	1974	21.9	19.2	19.0	23.3
	1975		17.8	19.0	22.4
	1976			22.9	22.3

(1 versus 2-years-ahead for stability). Much of the observed improved performance is because the new algorithm can detect and properly respond to step changes in the demand level. Figs. 9a and 9b show how SSD-SPA processes the data shown previously in Figs. 3a and 3b as examples of SSFS poor performance.

It should be noted that both examples only show how past deterministic events (before the start of the forecasting period, i.e., before July, 1974, in Fig. 9a, and July, 1975, in Fig. 9b) are treated. No knowledge was assumed about future special events, such as the one on October, 1976 (Fig. 9b). Once the data up to these events are available, even if no, incomplete, or wrong information would be input into SSD-SPA, the algorithm could recognize them and properly adjust the forecast, as was shown for the events on May, 1974 (Fig. 9a) and February, 1974 (Fig. 9b). The present SSFS algorithms treated these events as part of the normal growth, as shown in Figs. 3a and 3b.

3.3.2 Small integer forecast

In Section 2.3.3, we stated six desirable properties for the new forecasting algorithm based on the demand time series characteristics.

The unequal weighting of data, acceptance of exogenous information, and a short initialization period are shown to be part of the proposed mathematical model itself (Section 3.1). The recursive filter model adds computational efficiency since it does not explicitly use past data. Recognition of the past deterministic events and algorithm

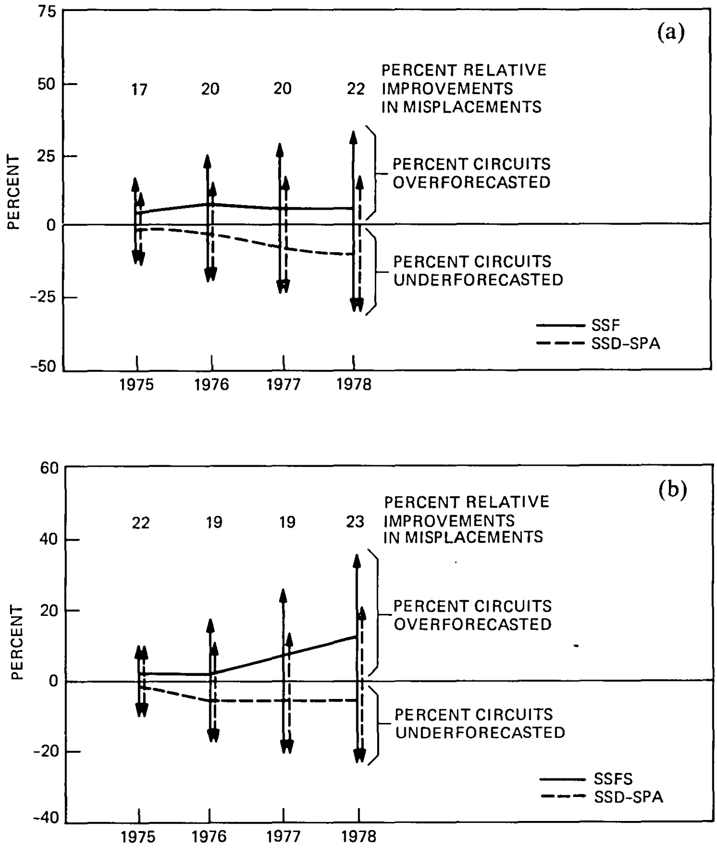

Fig. 7—SSD-SPA versus SSFS: total error and misplacement (forecasts generated in 1974). (a) Company C; (b) Company B.

recovery are ensured by the two procedures described in Section 3.2.5. It only remains to see if SSD-SPA performs adequately when small integers are to be forecasted.

To quantify this, the tests were repeated using only those point-to-point time series consisting of integers less than 10 (approximately 80 percent of all point-to-point demand time series).

Results of these tests on both companies' data bases showed that for small integers the relative forecast improvement of SSD-SPA is about 50 percent in accuracy, 30 percent in rms error, 30 to 66 percent in stability, and 50 percent in total misplacement. Moreover, total forecast error was found to range between 1 to 3 percent for SSD-SPA versus 3 to 28 percent for the present method. These last results excluded the "vanishing" time series in order to obtain unbiased attribute estimates.

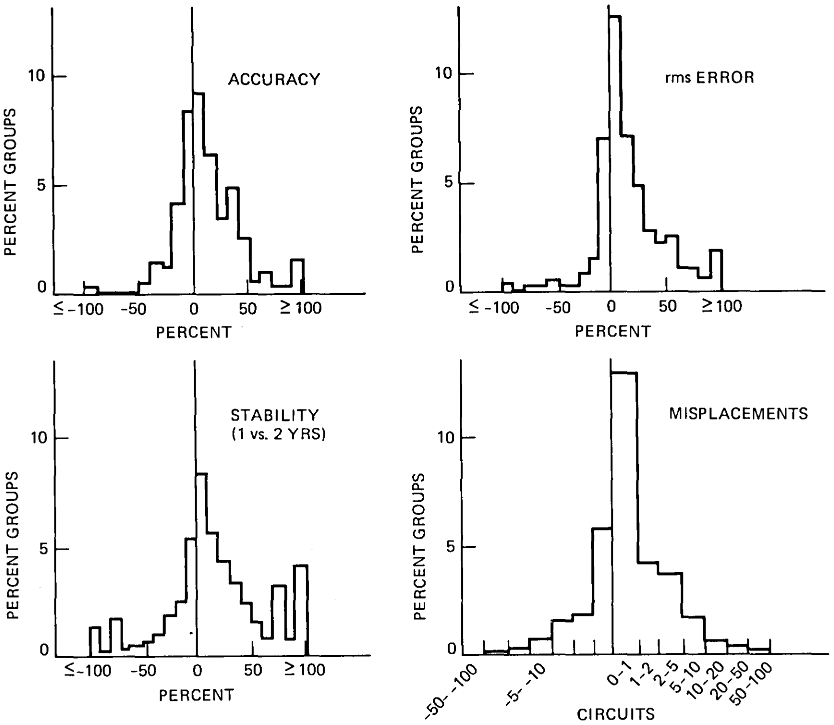

Fig. 8—SSD-SPA versus SSFS: percent relative improvements in 1-year forecast attributes for Company C.

3.3.3 Economic benefits and implementation costs

The comparative study of the present SSFS forecasting algorithm and the SSD-SPA showed that the new algorithm generates forecasts that are significantly more accurate and stable. Implementation of SSD-SPA in SSFS would, therefore, translate into important economic benefits in three areas: capital expenditures, forecasters' time, and electronic data processing costs.

(*i*) The major impact is expected to be on capital savings. The following analysis is based on the SSFS preliminary forecast before any manual adjustments are made. (There are no records available with the final adjusted forecasts made at different times in the past, nor with the exogenous information available to the forecaster.) The results showed the 1-year SSFS forecast positive misplacement of circuits to be 12 percent, on the average. That is, 12 percent of the total special-services circuits in the 1-year forecast could be in the wrong groups resulting in an overprovisioning. One-year results are used to be conservative; additionally, for a 1-year error there is less chance to reuse misplaced facilities.

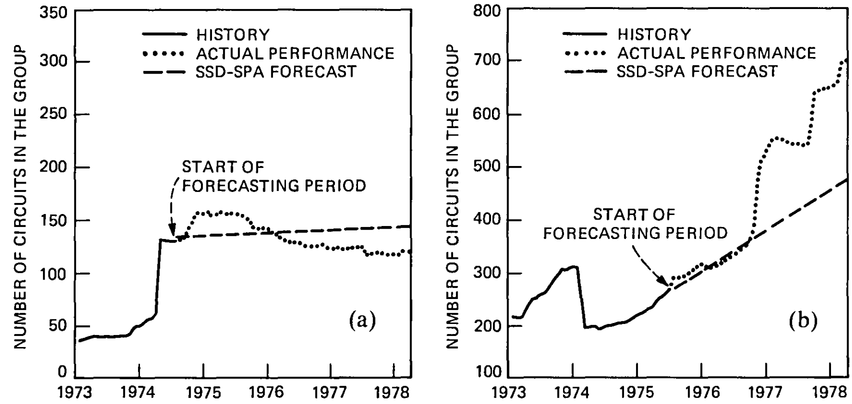

Fig. 9—Circuit groups with deterministic jumps. SSD-SPA treatment. (a) Example 1; (b) Example 2.

The SSD-SPA reduces the misplacement to 7 percent; a reduction of 5 percent. Theoretically, 5 percent of the total special-services circuit network could be removed without a change in service. Underprovisioning is approximately the same for both algorithms.*

(*ii*) The improved forecast accuracy, the recognition of past deterministic events, and the shorter forecast initialization requirements are SSD-SPA features that translate into fewer manual forecast adjustments. Fewer adjustments would permit the forecasters to concentrate more of their efforts to follow the economic conditions and estimate their impact on the future demand for special services.

(*iii*) The SSD-SPA is based on one forecasting model only and makes no explicit use of all the data history. Consequently, run times and core usage would be reduced. Although the absolute savings are not large, they would make SSFS very suitable for an on-line use.

IV. CONCLUSIONS

The goal of our work was to design an algorithm able to forecast future demands for special services: highly volatile time series mainly consisting of small integers, and with numerous deterministic jumps. We have shown that a linear, dynamic time-series model with linear growth and deterministic input, together with the Kalman filtering technique for state vector estimation and prediction, can produce demand forecasts which are significantly more accurate and stable

* This apparent positive bias is due to two types of groups. The first is those groups which "vanish" during the period. The second is those in which large deterministic events occurred. In the new algorithm, these events can be handled by input of marketing information.

than the forecasts produced by the best (highest R^2) choice of four unweighted regression models: the linear, exponential, and first- and second-order autoregressive. The new model, its attributes, and specific parameters were selected based on the characteristics of actual special-service demand history from three BOCs.

The improvement in accuracy is due to the capability of the system to track nonstationary processes, and also to recognize and react properly to deterministic changes in the demand, even when no, or wrong exogenous input was available. The use of a single model is responsible for much of the stability improvement. Additionally, SSD-SPA can produce many views of future demand using different assumptions on future events, it requires a short initialization period, and it results in the need for fewer manual adjustments. Therefore, we propose to replace the existing algorithm in the SSFS by this simple and more efficient algorithm.

V. ACKNOWLEDGMENTS

The author wishes to thank J. P. Moreland, S. R. Neal, and B. A. Whitaker for their helpful comments and suggestions. Special thanks go to R. Troester for the time and affort spent to modify SSFS, and to D. E. Sohn for help in obtaining and preparing the data bases. The curves of Fig. 5 were obtained from a modified version of C. R. Szelag's program.

REFERENCES

1. A. J. David and C. D. Pack, "The Sequential Projection Algorithm: A New and Improved Traffic Forecasting Procedure," Proc. Ninth Int'l. Teletraffic Congress, Torremolinos, Spain, 1979.
2. C. D. Pack and B. A. Whitaker, "Kalman Filter Models for Network Forecasting," B.S.T.J., this issue.
3. J. P. Moreland, "A Robust Sequential Projection Algorithm for Traffic Load Forecasting," B.S.T.J., this issue.
4. A. Gelb, ed., *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1974.
5. A. M. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
6. N. R. Draper and H. Smith, *Applied Regression Analysis*, John Wiley and Sons, N.Y., 1966.
7. C. R. Szelag, "A Short-Term Forecasting Algorithm for Trunk Demand Servicing," B.S.T.J., this issue.
8. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," J. Basic Eng., *82*, pp. 34–45, March, 1960.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 1, January 1982
Printed in U.S.A.

A Short-Term Forecasting Algorithm for Trunk Demand Servicing

By C. R. SZELAG

(Manuscript received December 31, 1980)

Trunk servicing is the continual process of collecting trunk group traffic measurements, monitoring network service, and augmenting the network when necessary. This study addresses the possibility of using a short-term forecast to determine the adequacy of trunk quantities planned for the imminent busy season. When seasonal patterns of demand exist, it may be possible to use observed, pre-busy-season traffic levels to predict accurately that busy-season demand will exceed the planned trunk group capacity and to determine appropriate corrective action. Toward this end, we develop a seasonal load forecasting algorithm based on Kalman filter estimation techniques and analyze the effectiveness of this approach using Bell operating company data. For trunk groups exhibiting seasonal demand, the short-term (1½ months ahead), seasonal forecast error is 50 percent less than that of the sequential projection algorithm (SPA), which linearly trends the yearly busy-season loads. Much of this improvement is attributed to the ability of the seasonal algorithm to utilize recent observations; the one-year ahead seasonal forecast error is only 20 percent less than that of SPA. We conclude that the greater generality and simplicity of SPA makes that algorithm the appropriate choice for the annual busy-season trunk forecast used in medium-range network planning. However, the seasonal algorithm demonstrated the ability to use recent data to respond quickly and accurately to various situations that result in inaccurate SPA forecasts. For this reason, the short-term forecasting algorithm developed herein is a potentially valuable tool for network administration.

I. INTRODUCTION

1.1 Motivation

Demand servicing is the process responsible for detecting and correcting overload conditions in the trunk network. Such conditions inevitably occur when unanticipated traffic levels exceed the planned capacity, which must be maintained at a reasonably low level to provide good service at low cost.

The planned capacity for each trunk group is determined primarily by the annual forecast of busy-season trunk requirements. Trunk servicing, which includes demand servicing, is the continual process of collecting trunk group traffic data, monitoring network service, and augmenting the network when needed. This paper addresses the possibility of supplementing the annual forecast and weekly monitoring process with a short-term forecast of imminent busy-season requirements. Specifically, when seasonal patterns of demand exist, it may be possible to use observed, pre-busy-season traffic levels to predict accurately that busy-season demand will exceed the planned trunk group capacity. Thus, service problems may be predicted and possibly avoided by "anticipative" demand servicing action.

When the need for demand servicing arises, the trunk servicer must decide on the locations and magnitudes of trunk group augments required to restore service. If current traffic levels already exceed those forecast, the servicer would like to know whether the peak load level has already been reached, or whether even higher levels are imminent. For each trunk group that must be augmented, the servicer should know the minimum amount of additional capacity required to both relieve the existing problem and to provide adequate service through the remainder of the busy season.

1.2 Short-term trunk forecast

Both of the trunk servicing functions described above, namely, the anticipation of imminent service problems and the determination of appropriate demand servicing augments, could be performed with the use of a short-term trunk forecasting system. Such a system would recognize within-the-year demand patterns and use this information to make accurate, short-term predictions of busy-season load. As such, the short-term forecast would serve as a "back-up" to the yearly busy-season forecast, recommending remedial action in those cases where the latter is significantly in error.

The purpose of this study is to investigate a short-term forecast to provide the trunk servicer with accurate, useful information concerning near-term traffic levels. With such a tool, service problems could be avoided by anticipative demand servicing and useful reserve capacity could be identified. This would allow the servicer to make more efficient use of existing facilities and equipment, thus, reducing the amount of reserve capacity required to maintain good service.

1.3 Overview

Section II is a discussion of the general requirements of a short-term trunk forecasting system. Motivated by these requirements, we consider in Section III the class of linear dynamic models and show how these models can be used to represent trunk group load histories exhibiting seasonal variation. In Section IV, we discuss a recursive estimation procedure, known as the Kalman filter, that is appropriate for this class of models. Using trunk group data obtained from a Bell operating company (BOC), we test the performance of a specific forecasting algorithm in Section V, and compare its performance with the year-to-year forecast produced by the recently developed sequential projection algorithm (SPA).[1,2] Section VI summarizes our findings.

II. SYSTEM REQUIREMENTS

The selection of an appropriate class of time-series models and forecasting procedures for consideration in this study depends heavily on the intended mode of operation and operating environment. The general requirements of a short-term trunk forecasting system are discussed in this section. These requirements will motivate the class of time-series forecasting algorithms considered in Section III.

2.1 Time series model

Underlying any time-series forecasting procedure is a mathematical model describing the structure of the series being forecast. For short-term trunk forecasting, the time-series model used must be sufficiently flexible to model a wide range of trunk group growth patterns accurately and to track changes in these growth patterns closely. This requirement is shown in Fig. 1. Figure 1a shows the load history of a BOC only-route group exhibiting a highly regular seasonal pattern modulating a nearly constant, linear trend. The behavior of such demand can be accurately predicted by an reasonably appropriate procedure. In particular, the annual 1-year SPA forecasts of busy-season demand are quite accurate.

In contrast, the trunk group load history shown in Fig. 1b requires a more sophisticated treatment. The distinctly nonconstant trend leads to gross errors in the SPA forecasts of busy-season loads, since the SPA algorithm design assumes approximately linear growth or decline.

Although the growth pattern shown in this figure is clearly nonconstant, it is not unpredictable. The short-term trunk forecasting algorithm should have sufficient built-in flexibility to accommodate such behavior.

2.2 Data requirements

To provide good service at low cost, the Bell System network is constantly evolving. This evolution accounts for the introduction of new and improved technology with the accompanying changes in

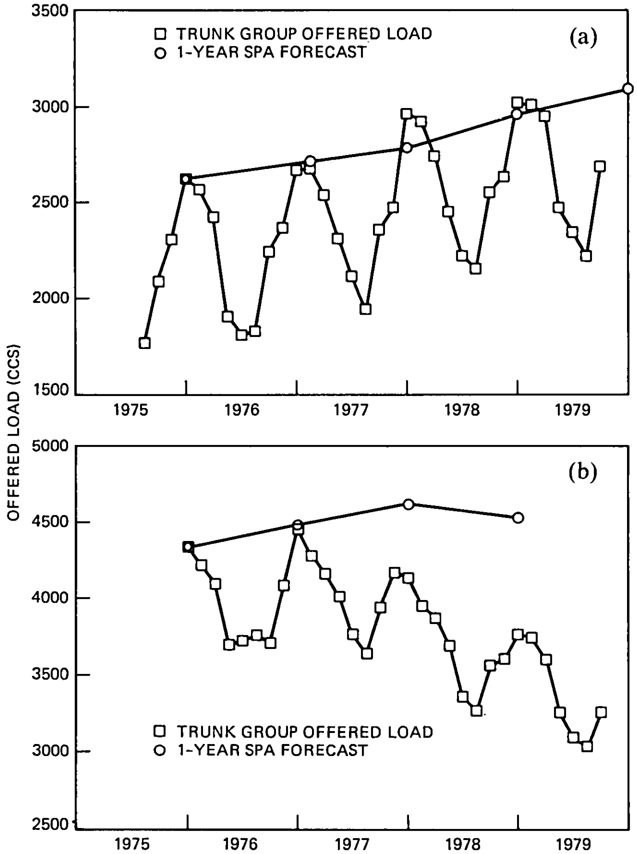

Fig. 1—(a) Trunk group AA001444—annual busy-season forecast. (b) Trunk group AA024225—annual busy-season forecast.

network configuration, and for change and growth in customer demand patterns. As the network evolves, new trunk groups, connecting new switching systems in a more economical network configuration, come into service and serve traffic previously carried by existing groups, which may be phased out of service. Also, in the past, trunk group histories were not maintained and data collection schedules were less comprehensive. For these reasons, long, uninterrupted trunk group load histories are, and will continue to be, atypical.

If it is to be useful, the short-term trunk forecasting system must be capable of operating in this type of environment. It should be capable of producing accurate forecasts based on relatively small amounts of historical data (e.g., 2 years). Also, it should be able to process and respond to information concerning semideterministic events (e.g., main station transfers and traffic reroutes) that affect the trunk group load pattern.

2.3 Computational requirements

In most BOCs, the trunk servicer relies on the Trunk Servicing System (TSS) to process recent trunk group traffic measurements and to obtain estimates of current load levels, traffic characteristics, and trunk requirements.[3] On a weekly basis, the system must process measurements and produce reports on several thousand trunk groups. Therefore, when considering candidate short-term forecasting algorithms for implementation in such a system, computational efficiency is of great importance. Equally important is the need for mechanized procedures requiring minimal intervention by the servicer, who typically must administer several hundred, and possibly thousands, of groups. Thus, computational efficiency and automation requirements rule out those procedures commonly used to perform detailed statistical analyses of individual time series. (See for example, the methods described in Ref. 4.)

2.4 Summary

In summary, the short-term trunk forecasting algorithms considered in this study should be able to track the trunk group load series accurately and adapt to dynamic changes in the demand pattern. In addition, they should perform adequately after a minimal initialization period and be computationally efficient.

In the next section, we describe a class of algorithms satisfying these requirements.

III. LINEAR DYNAMIC MODELS OF SEASONAL TIME SERIES

In this section, we consider the representation of trunk group load histories exhibiting seasonal variation by a linear, dynamic time-series model. The motivation for considering such a model for the application considered in this paper is discussed in Ref. 5. To summarize, this formulation is sufficiently general to describe a large number of time series of practical interest and is compatible with certain computationally efficient estimation techniques that make optimal use of limited amounts of data. The estimation and forecasting of these models will be considered in Section IV.

3.1 Mathematical model

A discrete time series is a sequence of observations of some quantity of interest. We think of such a series as being a realization of some stochastic process $\{\mathbf{y}_t\}$, which serves as a mathematical model explaining the observations, and which allows us to make inferences concerning future values of the series.

In the linear dynamic model, the behavior of the series is determined by an n-dimensional state-vector process $\{\mathbf{X}_t\}$ and the following two equations that describe the time evolution of $\{\mathbf{X}_t\}$ and the relation of $\mathbf{X}_t$ to the observation $\mathbf{y}_t$.

The first equation, called the system equation, is

$$\mathbf{X}_{t+1} = \phi_t \mathbf{X}_t + \mathbf{W}_t, \tag{1}$$

where ϕ_t if an $n \times n$ "transition" matrix and $\{\mathbf{W}_t\}$ is an n-dimensional, zero-mean white noise process with

$$E(\mathbf{W}_t \mathbf{W}_s') = \begin{cases} \mathbf{0} & \text{if } s \neq t \\ \mathbf{Q}_t & \text{if } s = t. \end{cases} \tag{2}$$

More generally, we can consider the system equation

$$\mathbf{X}_{t+1} = \phi_t \mathbf{X}_t + \mathbf{U}_t + \mathbf{W}_t, \tag{3}$$

where $\mathbf{U}_t$ is an n-dimensional deterministic or stochastic input to the system at time t.

The matrix ϕ_t in eqs. (1) and (3) describes the deterministic movement of the state variables comprising $\mathbf{X}_t$. The white noise process $\{\mathbf{W}_t\}$ explicitly allows for random variations in these state variables and, therefore, significantly enhances the flexibility of this formulation.

The second equation, called the observation equation, is

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{X}_t + \epsilon_t, \tag{4}$$

where $\mathbf{H}_t$ is a $k \times n$ "observation" matrix and ϵ_t is a k-dimensional, zero-mean, white-noise vector sequence, independent of $\{\mathbf{X}_t\}$, with

$$E[\epsilon_t \epsilon_s'\} = \begin{cases} \mathbf{0} & \text{if } s \neq t \\ \mathbf{R}_t & \text{if } s = t. \end{cases} \tag{5}$$

Thus, the observation $\mathbf{y}_t$ is a linear function of the state variables corrupted by an additive disturbance ϵ_t.

Since we will only consider univariate time series $\{y_t\}$ in this paper, we will assume henceforth that $k = 1$.

3.2 Examples

Next, we present a few simple, but relevant, examples to demonstrate that this formulation describes many kinds of time series. Additional examples can be found in Ref. 6.

3.2.1 Linear growth model

In many applications in which a quantity is to be forecast over a relatively short time span, it is reasonable to assume that the quantity is growing approximately linearly with time. In particular, the SPA[2] design assumes that busy-season trunk group load varies from year to year along a nearly linear trend.

Linear growth can be described by the two-dimensional linear model with the following parameters:

$$\phi_t \equiv \phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; \qquad \mathbf{H}_t \equiv \mathbf{H} = [1, 0];$$

$$\mathbf{X}_t = \begin{bmatrix} x_t^{(1)} \\ x_t^{(2)} \end{bmatrix}. \tag{6}$$

In this model, $x_t^{(1)}$ represents the level of the series at time t, and $x_t^{(2)}$ the instantaneous rate of growth. The inclusion of the white noise process described by $\{\mathbf{Q}_t\}$ allows the trajectory to deviate randomly from a straight line.

3.2.2 Seasonal models

A time series $\{y_t\}$ is said to exhibit seasonality if observations separated in time by some fixed interval (usually one year) exhibit similar behavior. The simplest kind of seasonality is periodicity; that is $y_t \approx y_{t+nL}$ for some positive integer L and every integer n. Two linear models of periodic behavior are described below.

A useful model of seasonal variation is available via the trigonometric representation for periodic sequences.[7] Let $\{y_t\}$ be periodic, with period L. Then $\{y_t\}$ can be represented as follows:

$$y_t = \alpha_1 + \sum_{j=1}^{\left[\frac{L-1}{2}\right]} [\alpha_{2j}\cos(j\omega t) + \alpha_{2j+1}\sin(j\omega t)] + \alpha_L(-1)^t, \tag{7}$$

where $\omega = 2\pi/L$ and the last term is omitted if L is odd. Hereafter, we will assume that L is even. The coefficients $\{\alpha_k\}$ in the trigonometric expansion are determined by

$$\alpha_1 = \frac{1}{L}\sum_{i=1}^{L} y_i$$

$$\alpha_{2j} = \frac{2}{L}\sum_{i=1}^{L} y_i\cos(ij\omega)$$

$$\alpha_{2j+1} = \frac{2}{L}\sum_{i=1}^{L} y_i\sin(ij\omega)$$

$$\alpha_L = \frac{1}{L}\sum_{i=1}^{L} y_i(-1)^i. \tag{8}$$

Note that the first coefficient α_1 represents the average level of the series $\{y_t\}$. It can be omitted from the expansion (7) if we are representing a series having zero mean.

The existence of the representation (7) follows from the fact that the set of time functions

$$\mathbf{F}_0 = [1, \cos \omega t, \sin \omega t, \cdots, (-1)^t]$$

forms an orthogonal basis for the linear space of all real, periodic sequences with period L. Also note that for every integer τ, the set of τ-translates of $\mathbf{F}_0$, defined by

$$\mathbf{F}_\tau = [1, \cos \omega(t-\tau), \sin \omega(t-\tau), \cdots, (-1)^{t-\tau}]$$

also has this property. Therefore, for each τ there exists a representation of $\{y_t\}$ as a linear combination of the elements of $\mathbf{F}_\tau$;

$$y_t = \alpha_1^{(\tau)} + \sum_j \{\alpha_{2j}^{(\tau)} \cos[j\omega(t-\tau)] + \alpha_{2j+1}^{(\tau)} \sin[j\omega(t-\tau)]\}$$
$$+ \alpha_L^{(\tau)}(-1)^{t-\tau} \tag{9}$$

with $\alpha_k^{(0)} = \alpha_k$ as in eq. (7).

The expansion of the series $\{y_t\}$ relative to the set of functions $\mathbf{F}_\tau$ can be viewed as a representation in which $t = \tau$ serves as the new time origin. In the linear dynamic model described below, at each new time epoch $t = \tau + 1$ we will "rearrange" the representation (9) in terms of the functions $\mathbf{F}_{\tau+1}$ via a simple, linear transformation of the coefficients $\{\alpha_k^{(\tau)}\}$.

$$\text{Let } y_t = \alpha_1^{(\tau)} + \sum_j [\alpha_{2j}^{(\tau)} \cos j\omega(t-\tau) + \alpha_{2j+1}^{(\tau)} \sin j\omega(t-\tau)]$$
$$+ \alpha_L^{(\tau)}(-1)^{t-\tau}$$
$$= \alpha_1^{(\tau+1)} + \sum_j \{\alpha_{2j}^{(\tau+1)} \cos j\omega[t-(\tau+1)] \tag{10}$$
$$+ \alpha_{2j+1}^{(\tau+1)} \sin j\omega[t-(\tau+1)]\} + \alpha_L^{(\tau+1)}(-1)^{t-(\tau+1)}. \tag{11}$$

By expanding each term in eq. (11) and equating coefficients of similar terms with the coefficients in eq. (10) yields

$$\alpha_1^{(\tau)} = \alpha_1^{(\tau+1)}$$
$$\begin{bmatrix} \alpha_{2j}^{(\tau)} \\ \alpha_{2j+1}^{(\tau)} \end{bmatrix} = \phi_j^{-1} \begin{bmatrix} \alpha_{2j}^{(\tau+1)} \\ \alpha_{2j}^{(\tau+1)} \end{bmatrix}$$
$$\alpha_L^{(\tau)} = (-1)\alpha_L^{(\tau+1)}, \tag{12}$$

where

$$\phi_j^{-1} = \begin{bmatrix} \cos j\omega & -\sin j\omega \\ \sin j\omega & \cos j\omega \end{bmatrix}. \tag{13}$$

$$\text{Let } \mathbf{X}_\tau = [\alpha_1^{(\tau)}, \cdots, \alpha_L^{(\tau)}]'. \tag{14}$$

Then

$$\mathbf{X}_\tau = \phi^{-1}\mathbf{X}_{\tau+1}, \tag{15}$$

where ϕ^{-1} is the block diagonal matrix defined by

$$\phi^{-1} = \begin{bmatrix} 1 & & & & & \\ & \phi_1^{-1} & & & & \mathbf{0} \\ & & \phi_2^{-1} & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & \mathbf{0} & & & \cdot \\ & & & & & -1 \end{bmatrix} \tag{16}$$

Equivalently,

$$\mathbf{X}_{\tau+1} = \phi \mathbf{X}_\tau. \tag{17}$$

To summarize, the trigonometric expansion (7) of $\{y_t\}$ can be represented by a linear dynamic system with state vector $\mathbf{X}_\tau = [\alpha_1^{(\tau)}, \cdots, \alpha_L^{(\tau)}]'$ comprising the coefficients in the expansion relative to the current time τ. The transition matrix ϕ, defined by eq. (16), provides the mechanism by which the coefficients at time $\tau + 1$ are obtained from those at time τ. Finally, at time τ, we observe

$$\begin{aligned} y_\tau &= \alpha_1^{(\tau)} + \sum_j \alpha_{2j}^{(\tau)} + \alpha_L^{(\tau)} + \epsilon_\tau \\ &= \mathbf{H}\mathbf{X}_\tau + \epsilon_\tau, \end{aligned} \tag{18}$$

where the L-dimensional vector $\mathbf{H}$ is given by

$$\mathbf{H} = \mathbf{H}_\tau = [1101\cdots01]. \tag{19}$$

By including a nonzero white-noise input in the dynamics (16), i.e.,

$$\mathbf{X}_{\tau+1} = \phi \mathbf{X}_\tau + \mathbf{W}_\tau, \tag{20}$$

we can allow for random variations in the trigonometric coefficients, and therefore, in the seasonal pattern.

In addition to the trigonometric model described above, another commonly used seasonal representation, the seasonal index model (Ref. 6, page 217) was considered. However, empirical performance results, analogous to those to be presented in Section V, indicated that the trigonometric model gave somewhat better results. For the sake of brevity, this model will not be discussed further.

3.3 Seasonal trunk group model

In Section 3.2, we showed how the general linear model eqs. (1) to (4) can be used to represent either linear-growth or periodic time series. We now demonstrate that these models can be superimposed to form a model capable of representing a wide variety of trunk group load series.

Consider again the trunk group load histories shown in Fig. 1. These series may be characterized by an underlying linear trend with a seasonal pattern superimposed. (This observation is consistent with the assumptions used in the design of SPA.[2] See Section 3.2.1.) The former component can be represented by the linear growth model described in Section 3.2.1; the latter, by the periodic model described in Section 3.2.2. If we assume that these two effects combine in an additive fashion, we can represent the observed behavior by the linear model with state vector

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{X}_t^{(l)} \\ \cdots\cdot \\ \mathbf{X}_t^{(s)} \end{bmatrix}, \tag{21}$$

where $\mathbf{X}_t^{(l)}$ is the state vector defined for the linear growth model (6) and $\mathbf{X}_t^{(s)}$ is that of the periodic model (14). (In the seasonal component $\mathbf{X}_t^{(s)}$, the "level" term $\alpha_1^{(t)}$ included in eq. (7) is omitted, since a similar term appears in the linear growth component $\mathbf{X}_t^{(l)}$.)

The dynamics of this system are described by the transition matrix

$$\phi = \begin{bmatrix} \phi^{(l)} & \mathbf{0} \\ \mathbf{0} & \phi^{(s)} \end{bmatrix}, \tag{22}$$

where $\phi^{(l)}$ and $\phi^{(s)}$ are the transition matrices of the linear and seasonal models, respectively.

Similarly, the observation matrix, $\mathbf{H}$, is given by

$$\mathbf{H} = [\mathbf{H}^{(l)} \vdots \mathbf{H}^{(s)}], \tag{23}$$

which decomposes the observation into the sum of a trend component $\mathbf{H}^{(l)}\mathbf{X}_t^{(l)}$ and a seasonal component, $\mathbf{H}^{(s)}\mathbf{X}_t^{(s)}$.

Finally, random variation in the components of $\mathbf{X}_t$ is modeled as a white-noise input, described by the sequence $\{\mathbf{Q}_t\}$. In particular, if the disturbances to the trend and seasonal components are uncorrelated, then $\mathbf{Q}_t$ may be decomposed as

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{Q}_t^{(l)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_t^{(s)} \end{bmatrix}.$$

3.4 Summary

In this section, we discussed the class of linear dynamic time-series models and showed how such models can be used to represent time series exhibiting both linear growth and seasonal variation. In particular, a specific model was proposed in Section 3.3 for the representation of trunk group load histories exhibiting seasonality.

In the next section, we consider the problem of estimating the parameters of such a model from the observed time series and the use of these estimates in forecasting.

IV. PARAMETER ESTIMATION AND PREDICTION

Complementing the class of linear dynamic time-series models is a sequential parameter estimation algorithm known as the Kalman filter.[8,9] With this technique, we estimate the model parameters comprising the state vector $\mathbf{X}_t$ from the observations $y_1, \cdots, y_t$ and use this estimate to make inferences concerning future values of the series (prediction).

The general procedures by which this is accomplished are discussed in this section. We begin by describing a recursive algorithm for sequentially updating the state vector estimate as new data becomes available. The recursion is initiated by a weighted least-squares estimation procedure derived in Section 4.2. The application of these procedures to the seasonal trunk group model derived in Section III yields a forecasting algorithm capable of accurately tracking and predicting the values of the seasonal time series considered in this study.

The performance of this algorithm on actual trunk group data will be examined in Section V.

4.1 The Kalman filter

Consider the linear dynamic model described by the equations

$$\mathbf{X}_{t+1} = \phi_t \mathbf{X}_t + \mathbf{W}_t \tag{24}$$

and

$$y_t = \mathbf{H}_t \mathbf{X}_t + \epsilon_t. \tag{25}$$

Assume that

$$\begin{aligned} E(\mathbf{W}_t) &= \mathbf{0}, \qquad E(\epsilon_t) = 0, \\ E[\mathbf{W}_t \mathbf{W}_s'] &= \begin{cases} \mathbf{0} & \text{if } s \neq t \\ \mathbf{Q}_t & \text{if } s = t, \end{cases} \\ E[\mathbf{W}_t \epsilon_s] &= \mathbf{0}, \end{aligned} \tag{26}$$

and

$$E[\epsilon_t \epsilon_s] = \begin{cases} 0 & \text{if } s \neq t \\ R_t & \text{if } s = t. \end{cases}$$

Suppose that at time t, prior to observing y_t, we have available an initial estimate of the state vector $\mathbf{X}_t$. Calling this prior estimate $\hat{\mathbf{X}}_{t,t-1}$,* let us also postulate that the estimation error $\tilde{\mathbf{X}}_t = \hat{\mathbf{X}}_{t,t-1} - \mathbf{X}_t$ has zero mean and is uncorrelated with ϵ_t, the observation error in y_t. Let us also assume that the error covariance matrix of the estimate,

* In general, the notation $\hat{\mathbf{X}}_{t,s}$ will be used to denote an estimate of $\mathbf{X}_t$ based on information available at time s.

$$\mathbf{P}_{t,t-1} = E(\tilde{\mathbf{X}}_t\tilde{\mathbf{X}}_t'), \tag{27}$$

is known.

When the current observation y_t becomes available, we want to combine the prior estimate with the new observation in an optimal manner. Specifically, we seek unbiased linear estimates $\hat{\mathbf{X}}_{t+k,t}$, $k \geq 0$, of the current and future states, which minimize the quantities trace $(\mathbf{P}_{t+k,t})$, where $\mathbf{P}_{t+k,t}$ is the error covariance matrix of $\hat{\mathbf{X}}_{t+k,t}$. Thus, we are seeking minimum variance, unbiased linear state-vector estimates.

The following procedure, which yields such estimates in a recursive manner, is known as the Kalman filter.[9]

4.1.1 Filtering

The problem of determining the optimal estimate $\hat{\mathbf{X}}_{t,t}$ from data available at time t is known as the filtering problem. Its solution is given by

$$\hat{\mathbf{X}}_{t,t} = \hat{\mathbf{X}}_{t,t-1} + \mathbf{K}_t[y_t - \mathbf{H}_t\hat{\mathbf{X}}_{t,t-1}], \tag{28}$$

where the optimal gain $\mathbf{K}_t$, is determined as

$$\mathbf{K}_t = \mathbf{P}_{t,t-1}\mathbf{H}_t'[\mathbf{H}_t\mathbf{P}_{t,t-1}\mathbf{H}_t' + R_t]^{-1}. \tag{29}$$

The error covariance matrix of this estimate is given by

$$\mathbf{P}_{t,t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t,t-1}(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)' + \mathbf{K}_tR_t\mathbf{K}_t' \tag{30}$$

$$= [\mathbf{I} - \mathbf{K}_t\mathbf{H}_t]\mathbf{P}_{t,t-1} \tag{31}$$

The more complex expression (30) is included since this relationship is valid for an arbitrary gain matrix $\mathbf{K}_t$, whereas (31) is valid only when $\mathbf{K}_t$ is the optimal gain (29).

4.1.2 Prediction

Having obtained the optimal linear estimate $\hat{\mathbf{X}}_{t,t}$ of $\mathbf{X}_t$ from the data available at time t, we now want to predict future values of the state vector and the series. These are obtained by extrapolation, using the linear dynamics relationship (24).

That is, the optimal linear estimates of $\mathbf{X}_{t+k}$ and y_{t+k}, based on data available at time t, are given, for $k > 0$, by

$$\hat{\mathbf{X}}_{t+k,t} = \left[\prod_{i=1}^{k} \phi_{t+i-1}\right]\hat{\mathbf{X}}_{t,t} \tag{32}$$

and

$$\hat{y}_{t+k,t} = \mathbf{H}_{t+k}\hat{\mathbf{X}}_{t+k,t}. \tag{33}$$

In particular, the optimal 1-step predictors are

$$\hat{\mathbf{X}}_{t+1,t} = \phi_t\hat{\mathbf{X}}_{t,t}, \tag{34}$$

and

$$\hat{y}_{t+1,t} = \mathbf{H}_{t+1}\hat{\mathbf{X}}_{t+1,t}.$$

The error covariance matrix of $\hat{\mathbf{X}}_{t+1,t}$ is

$$\mathbf{P}_{t+1,t} = \phi_t \mathbf{P}_{t,t} \phi'_t + \mathbf{Q}_t; \tag{35}$$

the one-step forecast error variance is

$$\text{var}(\hat{y}_{t+1,t}) = \mathbf{H}_{t+1}\mathbf{P}_{t+1,t}\mathbf{H}'_{t+1}. \tag{36}$$

4.1.3 Sequential estimation

We now show how the results of Sections 4.1.1 and 4.1.2 can be used in an efficient procedure for processing the observations $\{y_t\}$ to obtain estimates of the parameters of the underlying linear model.

First note that the 1-step prediction $\hat{\mathbf{X}}_{t+1,t}$ given by eq. (34) satisfies the requirements of the initial estimate of the state vector at time $t + 1$. Thus, we can use the filtering procedure described in Section 4.1.1 to process the next observation, y_{t+1}, when it becomes available.

In general, starting with an initial state estimate $\mathbf{X}_{t_0,t_0-1}$ at some time t_0, we can alternately apply the procedures described in Section 4.1.1 (filtering) and Section 4.1.2 (prediction) to process subsequent observations y_{t_0}, y_{t_0+1}, $\cdots$ in an efficient, recursive manner. The procedure is summarized below.

Inputs: Initial estimate $\hat{\mathbf{X}}_{t_0,t_0-1}$ with (known) error covariance matrix $\mathbf{P}_{t_0,t_0-1}$. Set $t = t_0$.

1. Compute the Kalman gain matrix $\mathbf{K}_t$, using $\mathbf{P}_{t,t-1}$ and eq. (29).
2. Use the current observation, y_t, and eq. (28) to compute the updated state-vector estimate, $\hat{\mathbf{X}}_{t,t}$. The error covariance matrix of this estimate, $\mathbf{P}_{t,t}$, is given by eq. (31).
3. Obtain $\hat{\mathbf{X}}_{t+1,t}$ and $\mathbf{P}_{t+1,t}$ using eqs. 34 and 35, respectively.
4. Set $t = t + 1$. Go to 1.

We make the following observations regarding implementation of the algorithm described above.

First, at each time t, all relevant information concerning the series is embodied in the state-vector estimate $\hat{\mathbf{X}}_{t,t}$. It is not necessary to store the history $\{y_1, \cdots, y_t\}$.

Second, the optimal gain sequence $\{\mathbf{K}_t\}$ is determined by the second-order statistics $\{\mathbf{Q}_t\}$ and $\{R_t\}$ describing the variability of the process and the measurements, respectively. Thus, if these quantities are known in advance, the gain sequence $\{\mathbf{K}_t\}$ can be precomputed.

Third, to start the recursion, an initial state estimate with known error covariance is necessary. The problem of obtaining such an estimate is considered below.

4.2 Initialization

In many Kalman filter applications, it is important to have an initial estimate of the model state vector as soon as the observations $y_1, y_2, \cdots$ begin. Since this estimate, $\hat{\mathbf{X}}_{1,0}$, is made prior to observing the series $\{y_t\}$, either a judicious guess must be made or information from an external source must be provided. In addition, the accuracy of this estimate must be quantified via the covariance matrix $\mathbf{P}_{1,0}$.

For complex models, the specification of good initialization parameters can be difficult. Also, improper specification can result in poor initial performance, because of improper weighting of the first observations. Both of these problems can be avoided completely if we can afford to wait until sufficient observations have been made that an initial state-vector estimate can be based on the data alone.

Since the short-term forecasting application considered in this paper is for use as a supplement to the yearly busy-season trunk forecast, it is neither necessary nor desirable to consider the use of the short-term forecast until sufficient data is available to make accurate predictions. For this reason, we recommend that the series $\{y_t\}$ be observed over an initialization period, say from $t = 1$ to $t = T$, so that the initial state estimate can be based on the data alone. The method by which this is accomplished is described below.

4.2.1 Linear model

Consider the linear dynamic system described by the equations

$$\mathbf{X}_{t+1} = \phi \mathbf{X}_t + \mathbf{W}_t \tag{37}$$

and

$$y_t = \mathbf{H}\mathbf{X}_t + \epsilon_t. \tag{38}$$

We assume that the matrices H and ϕ do not vary with time,* and that the latter is invertible.

We will show that each observation y_{T-k} within the initialization period can be expressed as a linear function of the state vector $\mathbf{X}_T$ at the end of the period, of future disturbances $\mathbf{W}_{T-k+j}$, and of ϵ_{T-k}, the observation error at time $t = T - k$.

We first note that, from eq. (37)

$$\mathbf{X}_{t-1} = \phi^{-1}(\mathbf{X}_t - \mathbf{W}_{t-1}), \tag{39}$$

so that

* This assumption is not necessary for the results of this section; however, it allows sufficient generality for the model considered and will simplify notation.

$$y_T = \mathbf{H}\mathbf{X}_t + \epsilon_T$$

$$y_{T-1} = \mathbf{H}[\phi^{-1}\mathbf{X}_T - \phi^{-1}\mathbf{W}_{T-1}] + \epsilon_{T-1}$$

$$y_{T-2} = \mathbf{H}[\phi^{-2}\mathbf{X}_T - \phi^{-2}\mathbf{W}_{T-1} - \phi^{-1}\mathbf{W}_{T-2}] + \epsilon_{T-2}.$$

In general, for $1 \leq k < T$,

$$y_{T-k} = \mathbf{H}\phi^{-k}\mathbf{X}_T - \mathbf{H}\left[\sum_{i=1}^{k} \phi^{-k+i-1}\mathbf{W}_{T-i}\right] + \epsilon_{T-k}. \tag{40}$$

In matrix notation,

$$\mathbf{Y}_T \equiv (y_T, \cdots, y_1)' = \mathbf{\Phi}_T\mathbf{X}_T + \mathbf{\Theta}_T, \tag{41}$$

where

$$\mathbf{\Phi}_T = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\phi^{-1} \\ \mathbf{H}\phi^{-T+1} \end{bmatrix}, \tag{42}$$

$$\mathbf{\Theta}_T = (\theta_T, \cdots, \theta_1)', \quad \text{and for} \quad 1 \leq k < T,$$

$$\theta_{T-k} = -\sum_{i=1}^{k} \mathbf{H}\phi^{-k+i-1}\mathbf{W}_{T-i} + \epsilon_{T-k}. \tag{43}$$

Equation (41) expresses the first T observations as a linear function of $\mathbf{X}_T$, plus an additive vector $\mathbf{\Theta}_T$ of zero-mean, correlated noise terms. Correlation between the components of $\mathbf{\Theta}_T$ is given by the covariance matrix

$$\mathbf{V} = (v_{jk}) = E[\mathbf{\Theta}_T\mathbf{\Theta}_T'].$$

Under the set of assumptions (26) the covariances v_{jk}, for $j \leq k$, are given by

$$\begin{aligned} v_{jk} &= \text{Cov}(\theta_{T-j+1}, \theta_{T{\not-}k+1}) \\ &= \sum_{i=1}^{j-1} E[\mathbf{H}\phi^{-j+i}\mathbf{W}_{T-i}][\mathbf{H}\phi^{-k+i}\mathbf{W}_{T-i}]' \\ &\quad + E(\epsilon_{T-j+1}\epsilon_{T-k+1}) \qquad (44) \\ &= \sum_{i=1}^{j-1} \mathbf{H}\phi^{i-j}\mathbf{Q}_{T-i}(\mathbf{H}\phi^{i-k})' + \delta_{jk}R_k. \qquad (45) \end{aligned}$$

4.2.2 Minimum variance initialization estimate

Using the linear model (41) relating the first T observations to the value $\mathbf{X}_T$ of the state vector at the end of the initialization period, we can estimate $\mathbf{X}_T$ by weighted least-squares. That is, if the matrix $\mathbf{\Phi}_T$ defined in eq. (42) has rank n,* the minimum variance, unbiased linear

* For the linear growth, seasonal demand model defined in Section 3.3, the matrix (42) has rank n if $T > n$, the dimension of the state vector. That is, the matrices ϕ and $\mathbf{H}$ define an observable linear system.[9]

estimate of $\mathbf{X}_T$ is given[10] by

$$\hat{\mathbf{X}}_{T,T} = [(\mathbf{\Phi}'_T\mathbf{V}^{-1}\mathbf{\Phi}_T)^{-1}\mathbf{\Phi}'_T\mathbf{V}^{-1}]\mathbf{Y}_T. \tag{46}$$

The error covariance matrix, $\mathbf{P}_{T,T}$, is given by

$$\mathbf{P}_{T,T} = [\mathbf{\Phi}'_T\mathbf{V}^{-1}\mathbf{\Phi}_T]^{-1}. \tag{47}$$

Using eqs. (34) and (35), we can extrapolate these quantities to time $t = T + 1$. That is,

$$\hat{\mathbf{X}}_{T+1,T} = \phi\hat{\mathbf{X}}_{T,T} \tag{48}$$

and

$$\mathbf{P}_{T+1,T} = \phi\mathbf{P}_{T,T}\phi' + \mathbf{Q}_T. \tag{49}$$

Using eq. (48) as the prior state-vector estimate at time $T + 1$, we can process subsequent observations y_{T+1}, y_{T+2}, $\cdots$ sequentially, as described in Section 4.1.1.

To summarize, we have shown that an initial state-vector estimate, satisfying the conditions of Section 4.1.1, can be obtained from the linear model (41) using weighted least-squares. This estimate can then be used to start the recursive algorithm summarized in Section 4.1.3.

We will now discuss some general considerations regarding the application of these procedures to time-series forecasting.

4.3 Implementation considerations

So far in this section, we have shown how the weighted least-squares method can be used to obtain an initial state-vector estimate at time T, which is sequentially updated by the Kalman filter algorithm as new data becomes available. The application of each of these procedures is predicated on the knowledge of the second moments $\{\mathbf{Q}_t\}$ and $\{R_t\}$, which describe the variability of the state-vector process and the observations, respectively. In practice, however, these quantities are rarely known. Therefore, before we can apply these methods to an observed time series, the problem of determining appropriate values for these parameters must be considered.

4.3.1 Specification of Q_t and R_t

Various procedures have been proposed for the on-line identification of the parameters $\{\mathbf{Q}_t\}$ and $\{R_t\}$. (For example, see Ref. 11.) However, such an approach would add considerable complexity to the estimation procedures described in Sections 4.2 and 4.3 and, for the relatively short time series associated with the application considered in this paper, would probably yield little improvement in performance compared to the much simpler alternative, discussed below.

Instead of trying to estimate the parameters $\{R_t\}$ and $\{\mathbf{Q}_t\}$ describing the stochastic structure of each individual series, it may be possible to determine a single set of parameters, say $[R_t^*]$ and $[\mathbf{Q}_t^*]$[†], which adequately approximates the true stochastic nature of the ensemble of time series being considered. That is, if within the domain of application, the performance of the procedures described in this section is relatively insensitive to deviations for the assumed values $[\mathbf{Q}_t^*]$ and $[R_t^*]$, then the use of a single set of parameters can be justified. Also, using this approach, a single gain sequence $\{\mathbf{K}_t\}$ and initialization matrix (46) can be precomputed and applied to all series. Thus, implementation is greatly simplified.

4.3.2 Truncated gain sequence

Another simplifying approximation is available for implementing the Kalman filter algorithm, in which the optimal gain sequence $\{\mathbf{K}_t\}$ is replaced by a simpler sequence, $\{\mathbf{K}_t^*\}$. For example, the truncated gain sequence, defined as

$$\mathbf{K}_{T+t}^* = \begin{cases} \mathbf{K}_{T+t} & \text{if} \quad t \leq \tau \\ \mathbf{K}_{T+\tau} & \text{if} \quad t \geq \tau \end{cases} \qquad (\tau \geq 1) \tag{50}$$

can often be used with good results. Two advantages of the truncated sequence over the full, optimal sequence are discussed below.

First, if τ is relatively small, only a few gain vectors $\{\mathbf{K}_{T+1}, \cdots, \mathbf{K}_{T+\tau}\}$ need to be computed and stored.

Second, and more important, is the use of the truncated gain sequence to avoid poor filter performance resulting from inaccurate specification of the parameters $\{\mathbf{Q}_t\}$. This is demonstrated in Fig. 2, which shows the theoretical performance of three seasonal trunk group algorithms based on the model described in Section 3.3.

In this example, the gain sequence has been computed under the false assumption that $\mathbf{Q}_t \equiv \mathbf{0}$[‡], when in fact, $\mathbf{Q}_t > \mathbf{0}$. Under this assumption, the full gain sequence $\{\mathbf{K}_t\}$ converges to zero. Thus, new observations y_t are given insufficient weight in eq. (28) to allow the filter to track the randomly varying process.

Also shown in Fig. 2 is the theoretical performance of the truncated gain sequence ($\tau = 1$), again computed under the false assumption that $\mathbf{Q}_t = \mathbf{0}$. In this case, however, the gain is held at a constant, nonzero

[†] Actually, it is sufficient to determine appropriate values for $\{\mathbf{Q}_t\}$, since the gain $\mathbf{K}_t$ and initialization matrix (46) depend only on the relative magnitude of $\mathbf{Q}_t$ compared to the scalar R_t. For this reason, we will hereafter assume that $R_t \equiv 1$.

[‡] The assumption that $\mathbf{Q}_t = \mathbf{0}$ implies that the process $\{\mathbf{X}_t\}$ evolves in a deterministic manner relative to a constant set of parameters, say $\mathbf{X}_0$. In this case, the minimum-variance estimates coincide with the usual (fixed-parameter) least-squares estimates. Thus, Fig. 2 illustrates the pitfalls of using such methods, when in fact, the underlying process is changing with time.

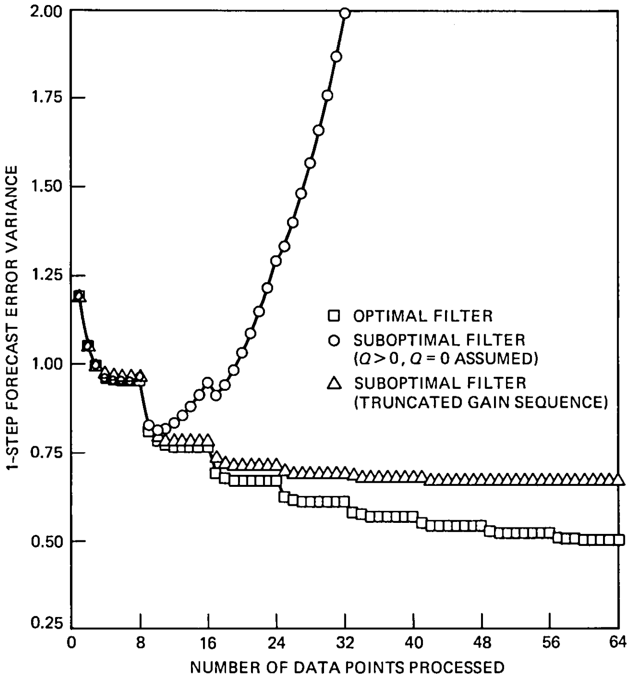

Fig. 2—Theoretical algorithm performance.

value, and the filter is able to track the series in a nearly optimal manner.

The main point being illustrated here is that, while correct specification of the parameters $\{\mathbf{Q}_t\}$ in the gain computation gives the filter sufficient responsiveness to track the process in an optimal manner, approximately the same effect can be achieved by truncating the gain sequence at some nonzero value. Thus, the truncated gain sequence is a useful implementation technique when a statistical description of the variability of the process is unavailable.

4.4 Summary

This section described the use of minimum variance, linear estimation procedures to estimate the parameters of a linear dynamic time-series model. By observing the process over a sufficiently long initialization period, an initial state-vector estimate can be obtained by the weighted least-squares method. Using the Kalman filter algorithm, this initial estimate can then be updated in an efficient, recursive manner as new observations become available.

In addition, we discussed the problem of obtaining good, suboptimal estimates when the exact statistical structure of each individual series is unknown. The application of these techniques will be illustrated in the next section, where we examine the performance of the proposed algorithm on actual trunk group data.

V. ALGORITHM PERFORMANCE-EMPIRICAL RESULTS

In Section III, we proposed a seasonal time-series model for trunk group load·histories. We then described, in Section IV, a general procedure for estimating the parameters of such a model and for predicting future values of the time series. The performance of the resulting forecasting procedure on actual trunk group data is studied in this section.

We begin this section by describing the objectives that motivated us to undertake the empirical study described herein.

5.1 Study objectives

We wanted to use BOC trunk group data to determine the effectiveness of the proposed forecasting procedure in the three areas discussed below.

5.1.1 Forecast accuracy

The primary goal of the short-term trunk forecast is to obtain an accurate view of the approaching busy-season based on observed pre-busy-season traffic levels. In contrast, the yearly busy-season forecast (e.g., SPA) considers only the quantities of direct interest, namely, the time-series of yearly busy-season (peak) loads. While the short-term forecast has the advantage of operating with a shorter lead time and uses more frequent observations, the potential improvement, if any, in forecast accuracy over the busy-season forecast should be quantified.

5.1.2 Predictability

In addition to forecast accuracy, we would also like to investigate the degree with which service problems reveal themselves prior to the busy season. That is, how effective is a short-term forecast in predicting such events?

5.1.3 Error characterization

The intended application of the short-term forecast is to provide information regarding the adequacy of the planned trunk level for each trunk group. As measurements are collected and short-term forecasts are generated, the servicer may decide to

(*i*) take no action,

(*ii*) augment the group immeditately ("demand servicing"), or

(*iii*) augment the group in the near future ("anticipative demand servicing").

The removal of excess trunks is the responsibility of the yearly planned-servicing activity recommended by the yearly trunk forecast. No action should be taken unless the current or anticipated level of demand exceeds the in-service capacity.

The effective solution of the "anticipative demand servicing" decision problem requires that the uncertainty regarding the short-term forecast be understood and quantified. That is, servicing action should be taken only when, with a fair degree of certainty, action is required.

5.2 Methodology

In principal, the quantities of interest defined above can be determined (either analytically or by simulation) from the statistical structure of the time series being considered. In practice, however, the true underlying structure and dynamics are unknown. Therefore, to obtain meaningful answers to the questions posed above, we must test the performance of our forecasting procedures on real data.

5.2.1 The data

Historical trunk group data is retained in computer accessible form by most BOCs in the extended administrative history files of the Trunk Servicing System (TSS). Unfortunately, the longest histories available today consist of only about four and one-half years of data, and are available in those companies that made early use of this feature.

In the TSS system, trunk group traffic measurements are averaged and reported over "study periods" consisting of up to 20 business days. Roughly speaking, each such study period represents a rolling average of the four most recent weeks of valid data. For grade-of-service trunk groups, the busy-season corresponds to the study period (within the year) that has the highest offered load. This is the quantity that we wish to forecast.

Although we could attempt to model, track, and predict the complete time series of study period loads, this level of detail is neither necessary nor desirable. Instead, we can partition the year into some number L of "forecast periods," select the largest study period load within each forecast period, and work with the resulting time series of forecast period loads. Since the study period load series and the forecast period load series have the same maximum value in each year, it is sufficient to predict the latter.

The choice of the parameter L, the number of forecast periods per year, determines a trade-off between model complexity and response time. For large values of L, the seasonal pattern and time-series model required to accurately represent it are complex; also, data must be collected frequently. However, since the observations occur more frequently, it is possible to react more quickly to significant changes in the series $\{y_t\}$.

For the intended application, it was decided that by partitioning the year into eight short-term forecast periods, a reasonable balance between system complexity and response time would be achieved.

5.2.2 Selection of groups

To test the performance of the proposed algorithm in a controlled operating environment, it was necessary to select from the BOC data a subset of those trunk groups having reasonably "clean" data. That is, we omitted all groups whose histories exhibited one or more of the following characteristics.

(i) No discernible pattern: The load histories of certain trunk groups appear to have neither a within-the-year pattern nor a general growth trend. (This frequency occurs on groups serving very small volumes of traffic.) For these groups, our model is inappropriate; trending the yearly peaks (as in SPA) appears to be the most reasonable approach.

(ii) Very short histories: Because new groups begin and old groups leave service, the amount of available data varies among groups. We considered only those groups having at least four years of valid data.

(iii) Missing data: A group was not considered if, within any of the 32 forecast periods comprising the first four years, no valid study period load measurement was available.

(iv) Deterministic changes in the series: For certain groups, it was apparent that a major, deterministic change (e.g., a main station transfer) had drastically affected the load history. In principal, such changes can be accommodated by the model (3); however, information concerning such occurrences was not available in our data base, so these groups were not considered. (As we mentioned in Section 2.2, the ability to react appropriately to such occurrences is highly desirable for the intended application and will be considered in future work.)

A sample of approximately 300 trunk groups whose load histories satisfied conditions *(i)* to *(iv)* was selected for use in our study.

5.2.3 The forecasts

For each trunk group, the seasonal algorithm was initialized using the procedure developed in Section 4.2 and the first two years of data (16 data points). As we mentioned earlier, the application of both the minimum variance initialization and the Kalman filter algorithm requires specification of the matrix $\mathbf{Q}_t$ describing the variability of the state-vector process. This quantity was left as an adjustable parameter in the study.

The error covariance matrix $\mathbf{P}_{T+1,T}$ of the prior estimate $\hat{\mathbf{X}}_{T+1,T}$ was computed from eq. (49), using the error covariance matrix (47) of the initialization estimate $\hat{\mathbf{X}}_{T,T}$. Then, eq. (29) was used to compute the gain sequence $\{\mathbf{K}_{T+1}, \mathbf{K}_{T+2}, \cdots\}$. Since we decided to use a single value of $\{\mathbf{Q}_t\}$ for all trunk groups, the same gain sequence and initialization matrix could be computed once and applied to each time series.

Although the seasonal algorithm was designed to track the entire time series of forecast period loads, the principal quantity of interest for grade-of-service trunk groups is the yearly peak (busy season) load. To determine the accuracy with which the seasonal algorithm predicts this quantity based on data available k-periods prior to the busy season, we compared the busy-season load with the maximum value of the series predicted by our algorithm.

To compare the performance of the seasonal forecasting algorithm with SPA, both algorithms were run in parallel on the same set of data. However, since the information to be used in initializing the SPA (aggregate growth rate estimates for the offices on which the trunk group terminates[2]) was not available in our data, an alternate procedure had to be used.

To simulate the SPA initial estimates based on aggregate growth rate information, the busy-season loads were summed over all groups in each of the first two years of data. The ratio of the second-year sum to the first-year sum was used as the aggregate growth ratio R, with $R = 1.0325$ for the ensemble of trunk groups considered in the study. For each group, the busy-season load in the first year was used as its initial level estimate, $\hat{x}_0^{(1)}$, and the initial growth increment estimate was

$$\hat{x}_0^{(2)} = (R - 1)\hat{x}_0^{(1)}.$$

5.3 Examples

Before discussing the selection of algorithm parameters and the corresponding aggregate accuracy statistics, let us observe the performance of the two forecasting procedures on a few of the trunk groups considered in this study. Along with the aggregate statistics, which quantify the average improvement in accuracy provided by seasonal forecasting, these examples identify a number of general cases in which seasonal forecasting offers dramatic improvements over the yearly trending method of SPA.

Let us again consider the two example trunk group load histories discussed in Section 2.1. Figure 3 shows the load history of trunk group AA001444, except that now the results of the seasonal forecast are also shown. Recall that the first two years of data are used to obtain the initial state-vector estimate by a weighted least-squares procedure. The initialization is illustrated by the trajectory of circles, which represents an extrapolation backward in time of the state estimate at

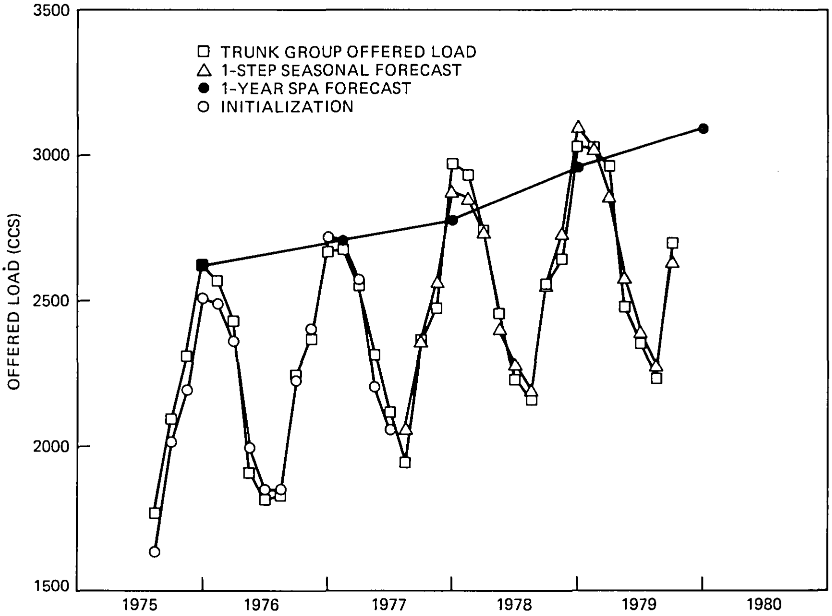

Fig. 3—Trunk group AA001444—seasonal forecast.

the end of the initialization period. Close examination reveals that the trajectory more closely fits the data at the end of the period than at the beginning; this is consistent with the fact that we are trying to estimate the current state.

The sequence of triangles beginning where the circles leave off is the sequence of one-step ahead seasonal forecasts obtained using the seasonal Kalman filter. It is evident that the seasonal algorithm is closely tracking the seasonal variation and underlying linear trend of this group and that SPA is predicting well the busy-season peaks. In fact, both procedures appear to perform equally well, on average, in predicting the yearly peak load.

A more interesting example is given in Fig. 4, which shows the performance of both algorithms on trunk group AA024225. As we mentioned earlier, the underlying growth pattern for this group exhibits a significant change in trend, which causes both algorithms to overforecast the peak load in the third year. However, by observing subsequent off-busy-season data points, the seasonal algorithm is able to detect this change, quickly home-in on the "signal," and give a reasonably good forecast of the busy-season load in the fourth year. In contrast, the yearly SPA forecast continues to be far off, since at the time the year-four forecast is made, only a single data point (the year-three busy-season load) off the previous trend line has been observed.

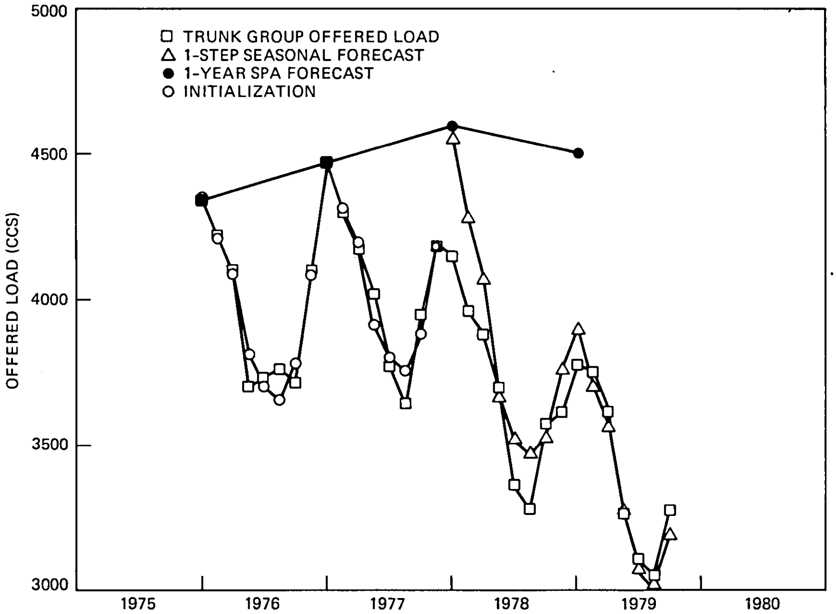

Fig. 4—Trunk group AA024225—seasonal forecast.

This example dramatically illustrates the potential value of the inter-busy-season information exploited by the seasonal algorithm.

A third interesting example, more in line with the intended anticipative demand servicing application discussed in Section 1.1, is given in Fig. 5. In this example, it appears that the trunk group load has undergone a moderate change in level and trend in the third year. The change occurs suddenly,* causing both algorithms to forecast significantly low in the third year. However, after processing subsequent data, the seasonal algorithm is able to forecast the next busy season with remarkable accuracy. In contrast, the SPA forecast again falls significantly low.

Finally, consider the trunk group shown in Fig. 6. In the second year, the busy-season load lies significantly above the trend for the other three years. This "outlier," which falls within the interval used by the SPA algorithm to screen for outliers, is accepted as a valid data point and processed accordingly. This causes SPA to "overshoot" the busy seasons in the next two years. In contrast, the seasonal algorithm, by processing the inter-busy-season data, recovers quickly and provides accurate forecasts in the next two years.

* Such a change, if known in advance, can be input by the forecaster, and the forecast can be changed accordingly. However, such "deterministic events" may be overlooked by the trunk forecaster, who typically has responsibility for a large number of groups.

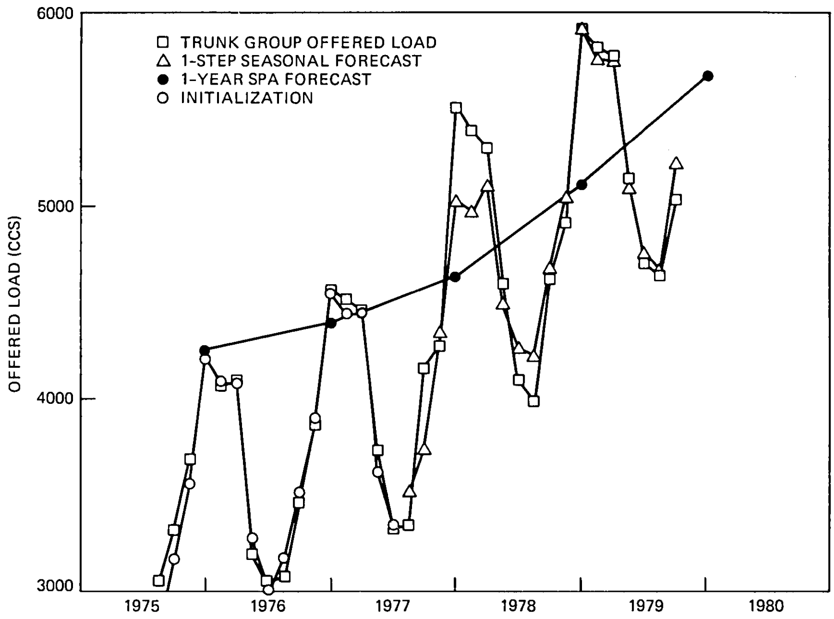

Fig. 5—Trunk group AA017330—seasonal forecast.

The significance of this fourth example lies in the additional protection against outliers provided by the seasonal forecast. If, in this example, the second SPA data point had been unusually low, it is likely that the SPA algorithm would have forecast significantly low in the next two years, possibly resulting in demand servicing activity. In such a case, the seasonal algorithm could be used to predict the service problem prior to its realization, or to estimate the additional capacity needed to relieve the problem when it occurred.

These four examples, while not constituting a meaningful statistical sample, dramatically illustrate a number of situations likely to occur in practice, where seasonal forecasting offers significant advantages over the yearly forecast of busy-season demand.

Before comparing the average accuracy results obtained for the SPA and seasonal forecasts, we will discuss the selection of the specific algorithm parameters used in the study.

5.4 Parameter selection

Recall from Section 4.3 that the ability of the algorithm to track random variations in the components of the state vector can be enhanced by either

(*i*) including a nonzero matrix $\mathbf{Q}_t$ that explicitly describes the variability of the parameters (Section 4.3.1),

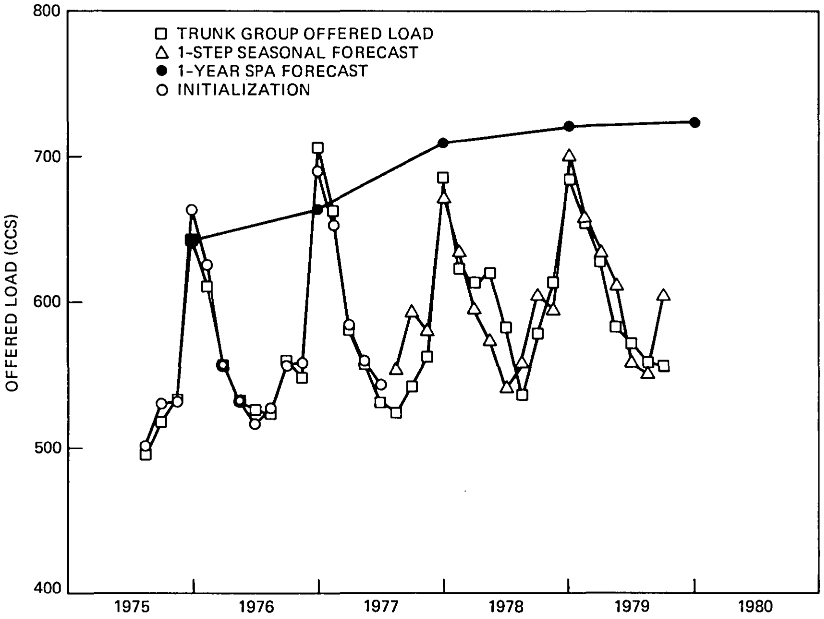

Fig. 6—Trunk group AA043161—seasonal forecast.

(*ii*) the use of certain heuristics, such as the truncated gain sequence (Section 4.3.2), or

(*iii*) some combination of (*i*) and (*ii*).

The effectiveness of these techniques was determined through experimentation. Specifically, for each version of the algorithm that we tested, the steady-state forecast error variance was estimated for each group and summed over all groups tested. With $\mathbf{Q}_t \equiv 0$, good results were obtained using the truncated gain sequence (50) with $\tau = 1$ (i.e., the first term $\mathbf{K}_{T+1}$ of the Kalman gain sequence was used to process each data point after initialization). Consistent with the hypothesis that some allowance must be made for random variation in the model parameters, this constant gain-vector sequence out-performed the full gain sequence computed under the assumption that $\mathbf{Q}_t \equiv 0$. Since the constant gain sequence also offers certain simplifications in algorithm implementation, this approach seemed very attractive.

By considering the behavior of various trunk group time series (as we discussed in Section 5.3.1), it also became clear that the seasonal forecasting algorithm should be able to follow variations in the underlying linear trend (see Fig. 2b). In addition to the responsiveness obtained through the use of the truncated gain sequence, additional responsiveness in the trend parameter is obtained by including a positive entry q_{22} in the matrix $\mathbf{Q}_t = (q_{ij})$. The optimal value of this parameter was determined empirically.

Using the average forecast error statistics, we then compared the performance of the resulting seasonal algorithm with that of SPA. The results are discussed below.

5.5 Accuracy

To quantitatively measure the performance of both the seasonal and the SPA forecasting procedures, both algorithms were used to forecast the busy-season loads on each of the trunk groups considered in the study. The results were compared using the relative error statistic, defined below.

Let a_g be the measured busy-season trunk group load in a given year, and $\hat{a}_g$ a forecast (obtained by either method) of this quantity. Then the relative forecast error is defined to be the quantity

$$e_g = \frac{\hat{a}_g - a_g}{a_g}.$$

To compare the approximately steady-state performance of the algorithms, this statistic was computed for each busy-season forecast in the fourth year. The results are shown in Fig. 7, which shows the distributions of the SPA and seasonal forecast errors, and in Table I. The latter compares the accuracy statistics of the seasonal forecasts,

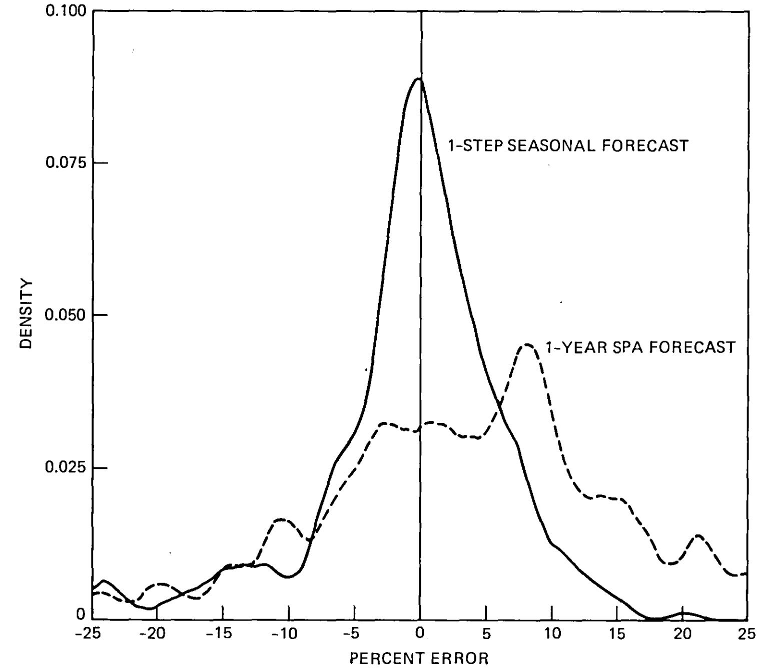

Fig. 7—Year 4 forecast error distributions.

Table I—Year 4 busy-season forecast results

	Percent		
Forecast	Bias	Average absolute Error	rms Error
1-Year SPA	3.5	10.3	13.5
1-Step seasonal	−0.7	5.1	7.5
2-Step seasonal	−0.8	6.1	8.5
3-Step seasonal	−0.5	6.5	9.0
4-Step seasonal	0.1	7.0	9.4
5-Step seasonal	0.2	7.6	10.4
6-Step seasonal	0.4	7.9	10.9
7-Step seasonal	0.5	8.0	10.8
8-Step seasonal	1.8	8.4	11.6

made one forecast period prior to the busy-season, with those of the SPA one-year-ahead forecast. The results show a significant improvement in accuracy, using either the average absolute relative error or the rms relative error. In either case, the observed error in the seasonal forecast is approximately half as large as that of the SPA forecast. Also, recall that these statistics, obtained by comparing the forecast with the measured load, also reflect load measurement error. Thus, on average, the seasonal forecast is off by less than 5 percent.

Part of this improvement must be attributed to the fact that the short-term busy-season forecast is made approximately 1½ months prior to the busy season, compared with one year for SPA. Additional improvement can be attributed to the exploitation of the additional structure of the seasonal time series. The relative importance of these two factors is illustrated in Fig. 8, which shows the relationship between short-term busy-season forecast error and lead time. Begin-

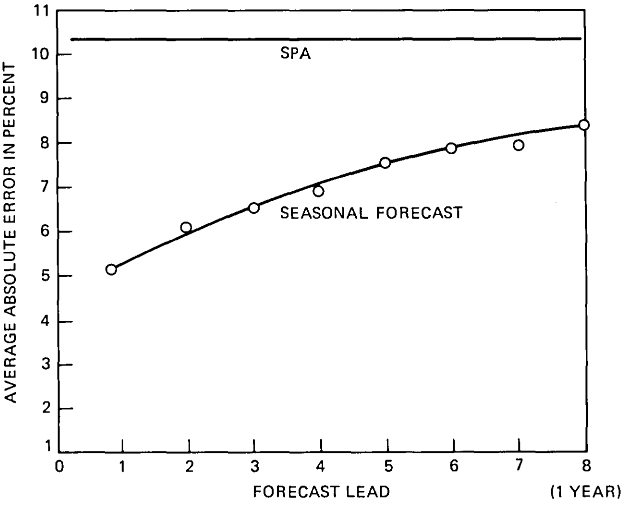

Fig. 8—Busy-season forecast error.

ning at 5.1 percent for the one-step forecast, the average absolute error statistic is a concave function of lead time, increasing to 8.4 percent for the one-year-ahead seasonal forecast. This suggests that most of the improvement in forecast accuracy can be attributed to the seasonal algorithm's ability to utilize recent data.

5.6 Summary

In this section, we described an empirical investigation of the performance of the seasonal trunk forecasting algorithm developed in Section IV. We showed that, on average, a short-term (one forecast period ahead) forecast of the approaching busy season is approximately twice as accurate as the yearly SPA forecast. This improvement in accuracy deteriorates with forecast lead time, being only 20 percent more accurate than SPA when projecting a full year ahead.

More important, we were able to identify a number of situations, likely to occur in practice, where the seasonal forecast significantly outperforms SPA. These situations tend to occur where unusual data, such as outliers or sudden, unanticipated changes in the growth pattern occur. By using the information provided by the inter-busy-season traffic levels, the seasonal algorithm is able to recover rapidly from such disturbances, and provide accurate forecasts in subsequent years. In contrast, the SPA algorithm, which processes a single data point each year, takes considerably more time to recover.

VI. SUMMARY AND CONCLUSIONS

This paper has described a comprehensive study of the use of seasonal forecasting algorithms in the trunk servicing process.

We began by discussing the basic requirements of a short-term trunk forecasting system and then described a general class of time-series models well suited for such applications. After developing a model for linear growth and seasonal demand, we considered various minimum variance procedures for estimating the parameters of such a model from a given time series. By observing the series over a two-year initialization period, an initial estimate of the model parameters is obtained by weighted least-squares. Subsequent observations are processed by an efficient recursive procedure known as the Kalman filter. Because of the optimality of these algorithms, a minimum amount of data is required before accurate forecasts can be made.

To verify the appropriateness of these procedures on actual trunk group data, an empirical investigation of the performance of the proposed algorithm was undertaken. The main conclusions of the study are discussed below:

6.1 Accuracy

On the average, the one-step (≈1½ months) ahead seasonal forecast of the busy-season load is approximately twice as accurate as the year-to-year forecast provided by the SPA. Most of this improvement can be attributed to the ability of the seasonal algorithm to utilize effectively recent observations. The one-year ahead, seasonal forecast error is only 20 percent less than that of SPA.

6.2 Anticipative demand servicing

The seasonal algorithm demonstrates the ability to respond quickly and accurately to various situations that tend to result in inaccurate SPA forecasts. Thus, the seasonal algorithm represents a potentially valuable tool for trunk network administration. In many cases, it can accurately predict that the currently planned trunk level is inadequate for the approaching busy-season demand level, and the appropriate trunk group augment. Conversely, it can also be used to identify potentially useful reserve capacity.

6.3 Relation to SPA

Since current trunk provisioning methods generally require that planned-servicing decisions be made nearly a year in advance, these results lead us to the conclusion that the greater simplicity and generality of the linear trending method of SPA makes that algorithm the appropriate choice for the annual busy-season trunk forecast. In the future, however, new technologies may make it both possible and economical to maintain the network in a near-optimal configuration through more-frequent planned-servicing adjustments. In that case, an accurate, seasonal forecast may be a better alternative for planned servicing.

REFERENCES

1. A. J. David and C. D. Pack, "The Sequential Projection Algorithm: A New and Improved Traffic Forecasting Procedure," Proc. Ninth Int'l Teletraffic Congress, Torremolinos, Spain, 1979.
2. J. P. Moreland, "A Robust Sequential Projection Algorithm for Traffic Load Forecasting," B.S.T.J., this issue.
3. R. J. Armstrong, R. Gottdenker, and R. L. Kornegay, "Servicing Trunks by Computer," Bell Laboratories Record, *54,* No. 2 (February, 1976).
4. G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control,* San Francisco: Holden-Day, 1976.
5. C. D. Pack and B. A. Whitaker, "Kalman Filter Models For Network Forecasting," B.S.T.J., this issue.
6. P. J. Harrison and C. F. Stevens, "Bayesian Forecasting," J. Royal Stat. Soc. (Series B), *38,* No. 3, (1976), pp. 205–47.
7. T. W. Anderson, *The Statistical Analysis of Time Series,* New York: John Wiley, 1971.
8. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," J. Basic Eng., *82* (March 1960), pp. 34–45.
9. A. Gelb, ed., *Applied Optimal Estimation,* Cambridge, Mass.: MIT Press, 1974.
10. M. Kendall and A. Stuart, *The Advanced Theory of Statistics, 2,* New York: MacMillan, 1979.
11. R. K. Mehra, "On the Identification of Variances and Adaptive Kalman Filtering," IEEE Trans. Automatic Control, *AC-15* (April 1970), pp. 175–84.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 1, January 1982
Printed in U.S.A.

Fast Recursive Estimation Using the Lattice Structure

By E. SHICHOR

(Manuscript received April 10, 1981)

This paper presents the theory for a rapidly converging adaptive linear digital filter. The filter weights are updated for every new input sample. This way the filter is optimal (in the minimum mean square error sense) for all past data up to the present, at all instants of time. This adaptive filter has thus the fastest possible rate of convergence. Such an adaptive filter, which is highly desirable for use in dynamical systems, e.g., digital equalizers, used to require on the order of N^2 multiplications for an N-tap filter at each instant of time. Recent "fast" algorithms have reduced this number to like 10 N. One of these algorithms has the lattice form, and is shown here to have some interesting properties: It decorrelates the input data to a new set of orthogonal components using an adaptive, Gram-Schmidt like, transformation. Unlike other fast algorithms of the Kalman form, the filter length can be changed at any time with no need to restart or modify previous results. It is conjectured that these properties will make it less sensitive to digital quantization errors in finite word-length implementation.

I. INTRODUCTION

Gradient algorithms are widely used in adaptive tapped delay-line filters, such as equalizers, to derive a set of tap coefficients that gives the desired output with a minimum mean square error (mmse). It is widely recognized[1] that when the input samples presented to the adaptive system are highly correlated, convergence to the optimum filter coefficients is slow. An important contribution to solving this problem of slow convergence was made by Godard[2] who obtained an adaptive algorithm that minimizes the total mse at all instants of time. Consequently, the Godard algorithm has the fastest possible rate of convergence in an mmse sense, and is usually referred to as the optimal mean-square adaptive estimator. This algorithm has the structure of a Kalman filter, and its complexity is on the order of N^2 multiplications and additions per iteration, where N is the number of filter coefficients being adjusted. Fast convergence results when successive corrections to the coefficients' vector are adaptively decorrelated. Based on this observation, other practical, less complex, schemes of orthogonalizing the corrections were proposed, see for example Ref. 1. Recently, an efficient (or "fast") computing procedure, called the fast Kalman algorithm, was obtained which provides a fast-converging estimator identical to that of Godard, but which requires only on the order of 10 N multiplications.[3,4,5]

Another approach to accelerated convergence is to transform the input data to obtain uncorrelated inputs to the estimator.[6] When the characteristics of the channel are fixed and known, the transformation can be found from the data autocorrelation matrix. When this matrix is unknown, the transformation has to be adaptive. Since the lattice structure, whose computational complexity grows only like N, is known to generate "white" uncorrelated outputs by a process called inverse filtering that keeps removing correlated components from the input signal,[7-11] it has been proposed for this application. However, the outputs of the lattice structure are uncorrelated only after it has converged to its steady state; therefore, it may not converge as fast as the Godard algorithm. Recently, Morf was able to formulate the lattice algorithm in a special form such that its outputs are uncorrelated in the mean square sense for all instants of time.[12,13] Our purpose is to extend Morf's works to compute an adaptive estimator which is equivalent in performance to Godard's. Moreover, we will demonstrate that the computational complexity of the adaptive lattice algorithm compares well with the fast Kalman algorithm of Falconer and Ljung.[5] The advantage of the lattice structure is the ease of changing the number of coefficients. It is also conjectured that the lattice algorithm will be less sensitive than the Kalman algorithm to finite-precision digital implementation. Recently, this was observed in Ref. 14. It is also discussed in Ref. 15, where the development of an equalizer based on the adaptive lattice algorithm is presented in a form similar to the one given here. One case to illustrate this property of the lattice will be given at the end of this paper.

In the next section, the optimal least mean square estimator and predictor are precisely defined, and the minimal error that results is given. In Section III, several properties of the optimal predictor are explored and are related to the estimation problem. In Section IV, an efficient (in the sense of small number of computations) lattice form is derived, using the relations developed in Section III, that maintains the optimal convergence. In Section V, the properties of this lattice form are compared to the steady-state lattice structure. Suggestions for further work are also included.

II. OPTIMAL MEAN SQUARE ESTIMATION

2.1 Notation and definitions

Given a discrete time input data sequence $\{y_i\}$ $i = 0, 1, \cdots$, it is desired to find the set of weights for a transversal tapped delay-line filter such that the output of this filter be a good estimate of another sequence $\{d_i\}$. An adaptive equalizer, for example, has the received signal as its input, while its output should provide an estimate of the transmitted data. In a transversal filter, a vector of filter coefficients, of length $p + 1$, operates on vectors of data that are shifted versions of the input data for time $0 \leq t \leq T$ defined by

$$y^t_{p,T} = (y_T, y_{T-1}, \cdots, y_{T-p}), \tag{1}$$

with y^t being the transpose of y and it is assumed that $y_i = 0$ for $i < 0$. As we are concerned with an adaptive, i.e., time varying filter, its weight vector of order $p + 1$ will be denoted

$$w^t_{p,T} = [w_{p,T}(0), w_{p,T}(1), \cdots, w_{p,T}(p)]. \tag{2}$$

Using these definitions, the output of the $p + 1$ long linear estimator at time T is $\hat{d}_{p,T}$ given by

$$\hat{d}_{p,T} = w^t_{p,T}\, y_{p,T}. \tag{3}$$

Now suppose that $w_{p,T}$ is the best predictor for time T, then the total, or accumulated, mean square estimation error up to time T, when using this predictor, is given by

$$E(w_{p,T}) = \sum_{i=0}^{T} (d_i - w^t_{p,T}\, y_{p,i})^2. \tag{4}$$

The sequence of weight vectors that minimizes eq. (4) at every instant of time T, is the most rapidly converging sequence, and is called the optimal adaptive filter.

Making use of the following time-domain definitions of the cross correlation vector and the autocorrelation matrix,

$$\sum_{i=0}^{T} d_i y_{p,i} \equiv g_{p,T} \tag{5}$$

$$\sum_{i=0}^{T} y_{p,i} y^t_{p,i} \equiv R_{p,T}, \tag{6}$$

one obtains

$$E(w_{p,T}) = \sum_{i=0}^{T} d_i^2 - 2w^t_{p,T} g_{p,T} + w^t_{p,T} R_{p,T} w_{p,T}. \tag{7}$$

2.2 The optimal estimator

Equating the gradient of eq. (7) with respect to $w_{p,T}$ to zero gives

$$R_{p,T} w_{p,T} = g_{p,T}. \tag{8}$$

It is seen that solving for the optimal estimator is equivalent to inverting a matrix at every new sample point:

$$w_{p,T} = R_{p,T}^{-1} g_{p,T}. \tag{9}$$

Godard showed that $w_{p,T}$ can be updated with on the order of p^2 calculations,[1,2] which is an improvement over the simple matrix inversion, requiring on the order of p^3 calculations. Algorithms that require only on the order of $10p$ operations for obtaining the optimal estimator appeared[3,4] subsequently, and are called "fast" algorithms.

The essence of this paper is to derive the fast algorithm in a special form, called the lattice form. This form was proposed to speed the convergence of the weight vector of an adaptive predictor to its optimal value (see Refs. 7 to 11). As will be seen in the next paragraph, the estimator problem is closely related to the prediction problem.

From eqs. (7) and (8) it is seen that the minimal total mse that results using the optimal estimator is simply

$$E_{\text{opt}}(w_{p,T}) = \sum_{i=0}^{T} d_i^2 - g_{p,T}^t R_{p,T}^{-1} g_{p,T}. \tag{10}$$

This is in constrast to the adaptive gradient algorithm whose performance is more difficult to analyze.

2.3 Optimal prediction

The problem of prediction is more basic, but similar to the problem of estimation. Solving this problem will be shown to simplify the solution of the estimation problem. For linear prediction, a set of weights is used to linearly estimate the present input point from past values of the input data. Let the set of p weights at time T be $\{-a_{p,T}(1), -a_{p,T}(2), \cdots, -a_{p,T}(p)\}$, so that the error when predicting the input point y_i is given by

$$e_{p,i} = A_{p,T}^t y_{p,i}, \tag{11}$$

with

$$A_{p,T}^t = [1, a_{p,T}(1), \cdots, a_{p,T}(p)]. \tag{12}$$

The error generated in predicting the input is that part of the input which is uncorrelated to past values of the input. This is a desired feature for fast convergence of adaptive filters.

The total square error up to time T is, thus, given by

$$\sum_{i=0}^{T} e_{p,i}^2 = A_{p,T}^t R_{p,T} A_{p,T}. \tag{13}$$

Taking derivatives with respect to $a_{p,T}(1)$ to $a_{p,T}(p)$, it is found that the predictor weight vector that will minimize the total mse up to time

T is the solution of the last p equations of the expression

$$R_{p,T}A_{p,T} = \begin{pmatrix} R^e_{p,T} \\ 0^p \end{pmatrix}, \tag{14}$$

with $R^e_{p,T}$ yet unknown and $0^p = (0, \cdots, 0)^t$ vector of order p. Using this optimal predictor in eq. (13) gives

$$\sum_{i=0}^{T} e^2_{p,i} = A^t_{p,T}R_{p,T}A_{p,T} = A^t_{p,T}\begin{pmatrix} R^e_{p,T} \\ 0^p \end{pmatrix} = R^e_{p,T}. \tag{15}$$

Therefore, $R^e_{p,T}$ is the minimal total mse that will result.

As before, obtaining the optimal predictor $A_{p,T}$ for all T is equivalent to inverting the matrix $R_{p,T}$ for all T. An efficient algorithm for doing this will be described. It should be noted, from comparing eqs. (8) and (14), that the latter is a simpler "homogenous" set of equations, except for the end term $R^e_{p,T}$; therefore, its solutions can serve as a basis for the solution of eq. (8).

III. DERIVATION OF THE ORDER AND TIME UPDATE RELATIONS

3.1 Time shift properties of $R_{p,T}$

The vectors $y_{p,T}$ for successive values of T are shifts of each other. As these vectors build up the matrix $R_{p,T}$ in eq. (6), it is expected that shifted versions of the solutions to the predictor and estimator equations will serve in updating these solutions. For doing this, the shift properties of $R_{p,T}$ are explored. For the (i, j) term in eq. (6), we have

$$R_{p,T}(i, j) = \sum_{k=0}^{T} y_{k+1-i}y_{k+1-j} = R_{p-1,T}(i, j) \quad \text{for all } p - 1 \geq i, j \geq 1 \tag{16}$$

$$\begin{aligned} R_{p,T}(i+1, j+1) &= \sum_{k=0}^{T} y_{k-i}y_{k-j} = \sum_{k=-1}^{T-1} y_{k+1-i}y_{k+1-j} \\ &= \sum_{k=0}^{T-1} y_{k+1-i}y_{k+1-j} = R_{p-1,T-1}(i, j) \quad \text{for all } p-1 \geq i, j \geq 1, \end{aligned} \tag{17}$$

where the fact that $y_i \equiv 0$ for $i < 0$ was used. Using Morf's notation, these relations can conveniently be written as

$$R_{p,T} = \begin{pmatrix} R_{p-1,T} & X \\ X & X \end{pmatrix} = \begin{pmatrix} X & X \\ X & R_{p-1,T-1} \end{pmatrix}, \tag{18}$$

with X being any other term in the matrix. It is clear from eq. (18) that $R_{p,T}$ is symmetric, but not Toeplitz, if steady state is not reached. Therefore, the properties of Toeplitz matrices cannot be used, as is done for example in claiming fast convergence in Ref. 7.

3.2 Order update relations

It will be useful to define, similar to eq. (12), a backward prediction vector

$$B^t_{p,T} = (b_{p,T}(p),\, b_{p,T}(p-1),\, \cdots,\, b_{p,T}(1),\, 1), \tag{19}$$

with the backward error given by

$$r_{p,T} = B^t_{p,T} y_{p,T}, \tag{20}$$

i.e., it is the error in predicting y_{T-p} from y_T to y_{T-P+1}. To minimize the total mean square backward prediction error up to time T, $B_{p,T}$ should be the solution of

$$R_{p,T}B_{p,T} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix}. \tag{21}$$

It is seen that this is another set of homogenous equations, except for the lower one. Again, the optimal error $r_{p,T}$ will be orthogonal to $y_{T-p+1}, \cdots, y_T$.

A recursive procedure will be derived in the Appendix for generating solutions to eqs. (14) and (21) for increasing order p.

It is shown to be

$$A_{p+1,T} = \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} - k_{p,T}R^{-r}_{p,T-1} \begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix} \tag{22}$$

for $k_{p,T}$ as defined in the Appendix, and the higher order total error is

$$R^e_{p+1,T} = R^e_{p,T} - k^2_{p,T}R^{-r}_{p,T-1} = R^e_{p,T}(1 - k^2_{p,T}R^{-e}_{p,T}R^{-r}_{p,T-1}). \tag{23}$$

As increasing the predictor order would not increase, and typically will decrease the error, it should be that

$$1 \geq k^2_{p,T}R^{-e}_{p,T}R^{-r}_{p,T-1} \geq 0. \tag{24}$$

Similarly,

$$B_{p+1,T} = \begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix} - k_{p,T}R^{-e}_{p,T} \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \tag{25}$$

and

$$R^r_{p+1,T} = R^r_{p,T-1} - k^2_{p,T}R^{-e}_{p,T}. \tag{26}$$

Similar relations hold for the prediction error, when multiplying eqs. (22) and (25) by $y_{p+1,T}$

$$e_{p+1,T} = e_{p,T} - k_{p,T}R^{-r}_{p,T-1}r_{p,T-1} \tag{27}$$

$$r_{p+1,T} = r_{p,T-1} - k_{p,T}R^{-e}_{p,T}e_{p,T}. \tag{28}$$

The following auxiliary quantities are needed

$$C_{p,T} = R^{-1}_{p,T}y_{p,T} \tag{29}$$

$$\gamma_{p,T} = C^t_{p,T} y_{p,T} = y^t_{p,T} R^{-1}_{p,T} y_{p,T}. \tag{30}$$

The order update of these quantities will also be derived in the Appendix. It is shown to be

$$\begin{aligned} C_{p+1,T} &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + r_{p+1,T} R^{-r}_{p+1,T} B_{p+1,T} \\ &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + \mu_{p+1,T} B_{p+1,T} = \begin{pmatrix} X \\ \mu_{p+1,T} \end{pmatrix}, \end{aligned} \tag{31}$$

with $\mu_{p+1,T}$ defined by

$$\mu_{p+1,T} = r_{p+1,T} R^{-r}_{p+1,T} \tag{32}$$

and, as seen, is the last term of $C_{p+1,T}$.

3.3 Time update relations

To obtain the time update of $A_{p,T}$, use is made of the following:

$$R_{p,T+1} = R_{p,T} + y_{p,T+1} y^t_{p,T+1}. \tag{33}$$

This relation is shown to give

$$A_{p,T+1} = A_{p,T} - e^0_{p,T+1} \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}. \tag{34}$$

Here, the definition

$$e^0_{p,T+1} = A^t_{p,T} y_{p,T+1} \tag{35}$$

is used for the tentative prediction error before updating the prediction coefficients. As for the minimal total mse, it is updated according to

$$\begin{aligned} R^e_{p,T+1} &= A^t_{p,T} \begin{pmatrix} R^e_{p,T+1} \\ 0^p \end{pmatrix} = A^t_{p,T} R_{p,T+1} A_{p,T+1} \\ &= A^t_{p,T} (R_{p,T} + y_{p,T+1} y^t_{p,T+1}) A_{p,T+1} = R^e_{p,T} + e^0_{p,T+1} e_{p,T+1}. \end{aligned} \tag{36}$$

It should be mentioned here that only in the stationary case $e_{p,T+1} = e^0_{p,T+1}$ and, thus, $R^e_{p,T+1} = R^e_{p,T} + e^2_{p,T+1}$. As for updating $B_{p,T}$, two different possibilities are derived in the Appendix:

$$B_{p,T+1} = B_{p,T} - r^0_{p,T+1} \begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} \tag{37}$$

or

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1} C_{p,T+1}) \times \frac{1}{1 - r^0_{p,T+1} \mu_{p,T+1}}. \tag{38}$$

From eq. (37), a relation like eq. (36) can be obtained

$$R^r_{p,T+1} = R^r_{p,T} + r_{p,T+1} r^0_{p,T+1}. \tag{39}$$

Again, $r^0_{p,T+1}$ is the tentative backward prediction error. The time update of $C_{p,T}$ is also obtained in the Appendix.

From these results, a simple update for $k_{p,T}$ is found to be

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1} r_{p,T}. \tag{40}$$

or alternatively, from eqs. (79) and (83)

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1} r^0_{p,T}(1 - \gamma_{p-1,T}) = k_{p,T} + e_{p,T+1} r^0_{p,T}. \tag{41}$$

All these relations are derived in the Appendix. They form the basis for the lattice network which update these quantities both in order and time.

IV. EFFICIENT CALCULATION OF THE OPTIMAL ESTIMATOR

4.1 Tapped delay line estimator

The optimal estimator for any order p and for each time T is given in eq. (9). Using eqs. (5) and (6), we get

$$\begin{aligned} R_{p,T+1} w_{p,T+1} &= g_{p,T+1} = g_{p,T} + d_{T+1} y_{T+1} \\ &= R_{p,T} w_{p,T} + d_{T+1} y_{T+1} \\ &= R_{p,T+1} w_{p,T} + (d_{T+1} - w^t_{p,T} y_{p,T+1}) y_{p,T+1}. \end{aligned} \tag{42}$$

Therefore,

$$\begin{aligned} w_{p,T+1} &= w_{p,T} + (d_{T+1} - w^t_{p,T} y_{p,T+1}) R^{-1}_{p,T+1} y_{p,T+1} \\ &= w_{p,T} + (d_{T+1} - \hat{d}^0_{p,T+1}) C_{p,T+1}. \end{aligned} \tag{43}$$

Note that updating $w_{p,T}$ involves the tentative estimate $\hat{d}^0_{p,T+1} = w_{p,T} y_{p,T+1}$ using the new data and present estimator weights. This makes it possible to implement this scheme in decision-directed equalizers, where the decision on which d_{T+1} was transmitted is based on $\hat{d}^0_{p,T+1}$. Also note that the correction to $w_{p,T}$ is in the direction of $C_{p,T+1} = R^{-1}_{p,T+1} y_{p,T+1}$ rather than $y_{p,T+1}$, as in the gradient algorithm. These vectors are parallel only if $R_{p,T+1}$ is a unit matrix times a scalar; thus, all its eigenvalues are equal. When this is not the case, and $y_{p,T+1}$ contains eigenvectors corresponding to different eigenvalues, $R^{-1}_{p,T+1}$ equalizes the gains for these vectors. Also note the similarity in the updating equations (34), (37), and (43) which is to be expected, since prediction is a special case of estimation.

The fast Kalman algorithm is an efficient recursive procedure to obtain $C_{p,T+1}$. This is given in Ref. 4 as follows:

1. Assume that all vectors are available up to and including time T.
2. Use eq. (35) to obtain $e^0_{p,T+1} = A^t_{p,T} y_{p,T+1}$.
3. Use eq. (34) to calculate $A_{p,T+1} = A_{p,T} - e^0_{p,T+1} \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}$.

4. Use eq. (11) to calculate $e_{p,T+1} = A^t_{p,T+1} y_{p,T+1}$.
5. Use eq. (36) to calculate $R^e_{p,T+1} = R^e_{p,T} + e^0_{p,T+1} e_{p,T+1}$.
6. Calculate $e_{p,T+1} R^{-e}_{p,T+1}$.
7. Use eq. (89) to calculate

$$C_{p,T+1} = \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} + e_{p,T+1} R^{-e}_{p,T+1} A_{p,T+1}.$$

8. From eq. (31), $\mu_{p,T+1}$ is found.
9. Find $r^0_{p,T+1} = B^t_{p,T} y_{p,T+1}$.
10. Use eq. (38) to calculate

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1} C_{p,T+1}) \frac{1}{1 - r^0_{p,T+1} \mu_{p,T+1}}.$$

11. Use eq. (73) to calculate

$$\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} = C_{p,T+1} - \mu_{p,T+1} B_{p,T+1}.$$

12. Calculate the tentative estimate $\hat{d}^0_{p,T+1} = w^t_{p,T} y_{p,T+1}$.
13. Use eq. (43) to update the estimator weights

$$w_{p,T+1} = w_{p,T} + (d_{T+1} - \hat{d}^0_{T+1}) C_{p,T+1}.$$

The initial conditions, when there is not enough input data so that $R_{p,T}$ in eq. (6) does not have an inverse, are discussed in Ref. 4. There are $10p + 5$ multiplications, $9p + 4$ additions, and 2 divisions for one complete updating cycle. Note that there are no matrix operations, only additions and products of scalars and vectors is involved. By comparison, the simple fixed-step gradient algorithm requires $2p + 1$ multiplications and $2p + 1$ additions per cycle.

4.2 Lattice structure

Here an equivalent algorithm that also gives the optimal estimator with about the same number of computations is derived. It is assumed that the input data are transformed by the lower triangular transformation matrix

$$L^t_{p,T} = \begin{pmatrix} B_{0,T} & B_{1,T} & \cdots & B_{p,T} \\ 0^p & 0^{p-1} & & \end{pmatrix}. \tag{44}$$

From eq. (20), the transformed data are

$$L_{p,T} y_{p,T} = (r_{0,T} r_{1,T} \cdots r_{p,T})^t \equiv \bar{r}_{p,T}. \tag{45}$$

Note that as the dimension p increases, new terms are included in $\bar{r}$, but the previous ones are unchanged. This is an important property of the lattice algorithm that enables us to change the estimation order without the need to recalculate all previous values, as for example in the fast Kalman algorithm. As before, we let the tentative transformed vector be

$$\bar{r}^{0}_{p,T+1} = L_{p,T}y_{p,T+1}. \tag{46}$$

Define a new vector of weights

$$H^{t}_{p,T} \equiv (h_{0,T}h_{1,T} \cdots h_{p,T}), \tag{47}$$

that now operates on $\bar{r}_{p,T}$ to give the estimate

$$\hat{d}_{p,T} = H^{t}_{p,T}\bar{r}_{p,T} = H^{t}_{p,T}L_{p,T}y_{p,T}. \tag{48}$$

It is seen that for this estimate to be equivalent to the optimal estimator, $w_{p,T}$ is the transform of $H_{p,T}$, and from eq. (8)

$$L_{p,T}g_{p,T} = L_{p,T}R_{p,T}w_{p,T} = L_{p,T}R_{p,T}L^{t}_{p,T}H_{p,T}. \tag{49}$$

Using eq. (21), it can be shown that

$$R_{p,T}L^{t}_{p,T} = \begin{pmatrix} R^{r}_{0,T} & 0^{1} & & 0^{p} \\ & R^{r}_{1,T} & \cdots & \\ X & X & & R^{r}_{p,T} \end{pmatrix}, \tag{50}$$

which is a lower triangular matrix. The product $L_{p,T}R_{p,T}L^{t}_{p,T}$ is, thus, a symmetric product of two lower triangular matrices; therefore, it should be symmetric, lower triangular, and diagonal, i.e.,

$$L_{p,T}R_{p,T}L^{t}_{p,T} \equiv D_{p,T}. \tag{51}$$

The diagonal terms are easily found using eq. (50)

$$D_{p,T}(i,\, i) = [B^{t}_{i,T}(0^{P-i})^{t}]\begin{pmatrix} 0^{i} \\ R^{r}_{i,T} \\ X \end{pmatrix} = R^{r}_{i,T} \tag{52}$$

and, again, they are independent of p.

At this point, a closer inspection of $L_{p,T}$ is of interest. It is a lower triangular matrix with 1 on the main diagonal. Therefore, $L^{-1}_{p,T}$ has the same structure. Therefore, eq. (45) can be rewritten as

$$\bar{r}_{p,T} = y_{p,T} - (L^{-1}_{p,T} - I_{p})\bar{r}_{p,T}. \tag{53}$$

This can be looked upon as a Gram-Schmidt procedure to calculate new orthogonal components of $\bar{r}_{p,T}$ from the components of $y_{p,T}$, minus their projections on the previous $\bar{r}_{p,T}$ components. Thus, eq. (51) represents the fact that the autocorrelation matrix of the transformed data is indeed diagonal.

Using eqs. (49) and (51), $H_{p,T}$ can be found from the transformed $g_{p,T}$ by

$$H_{p,T} = D^{-1}_{p,T}L_{p,T}g_{p,T}. \tag{54}$$

It should be noted that only scalar divisions rather than matrix inversion is needed here and increasing the order of the lattice *estimator does not change previously calculated values of H.* This is why double indices are used in eq. (47) as compared to triple indices in eq. (2). Equation (54) can be broken to p scalar equations

$$h_{p,T} = R^{-r}_{p,T} B^t_{p,T} g_{p,T}. \tag{55}$$

To see how the right-hand side develops in time, define

$$\rho_{p,T} = B^t_{p,T} g_{p,T}. \tag{56}$$

Then from eqs. (37), (5), (29), (9), and (3) in that order

$$\begin{aligned} \rho_{p,T+1} &= B^t_{p,T+1} g_{p,T+1} \\ &= [B^t_{p,T} - r^0_{p,T+1}(C_{p-1,T+1} 0)](g_{p,T} + d_{T+1} y_{p,T+1}) \\ &= \rho_{p,T} + (d_{T+1} - \hat{d}_{p-1,T+1}) r^0_{p,T+1} \hat{=} \rho_{p,T} + V_{p-1,T+1} r^0_{p,T+1}, \end{aligned} \tag{57}$$

where $V_{p,T}$ is the estimation error after the pth order estimator. For $p = 0$

$$\rho_{0,T+1} = g_{0,T+1} = \rho_{0,T} + d_{T+1} y_{T+1}, \tag{58}$$

i.e., $\hat{d}_{-1,T+1} \equiv 0$. Obviously,

$$V_{p,T} = d_T - \hat{d}_{p,T} = d_T - H^t_{p,T} \bar{r}_{p,T} = V_{p-1,T} - h_{p,T} r_{p,T}. \tag{59}$$

The recursive solution to eq. (55) that corresponds to eq. (43) is:

$$\begin{aligned} h_{p,T+1} &= R^{-r}_{p,T+1}(\rho_{p,T} + V_{p-1,T+1} r^0_{p,T+1}) \\ &= R^{-r}_{p,T+1}[(R^r_{p,T+1} - r_{p,T+1} r^0_{p,T+1}) h_{p,T} + V_{p-1,T+1} r^0_{p,T+1}] \\ &= h_{p,T} + R^{-r}_{p,T+1}(V_{p-1,T+1} - h_{p,T} r_{p,T+1}) r^0_{p,T+1}. \end{aligned} \tag{60}$$

This is equivalent to the first tap of the conventional tapped delay line equalizer for $p = 0$ only. The tentative estimate as in eq. (43) is now

$$\hat{d}^0_{p,T+1} = w^t_{p,T} y_{p,T+1} = H^t_{p,T} L_{p,T} y_{p,T+1} = H^t_{p,T} \bar{r}^0_{p,T+1}. \tag{61}$$

The minimal total squared error is from eq. (51)

$$E_{p,T} = \sum_{i=0}^{T} d_i^2 - g^t_{p,T} R^{-1}_{p,T} g_{p,T} = \sum_{i=0}^{T} d_i^2 - (L_{p,T} g_{p,T})^t D^{-1}_{p,T} (L_{p,T} g_{p,T}). \tag{62}$$

From the structure of D and L it follows that

$$E_{p+1,T} = E_{p,T} - (B^t_{p+1,T} g_{p+1,T})^2 R^{-r}_{p+1,T} = E_{p,T} - \rho^2_{p+1,T} R^{-r}_{p+1,T}. \tag{63}$$

Using eq. (57), the residual error can be found for all instants of time. It is then simple to decide whether p should be increased, decreased, or unchanged to meet the desired performance. As mentioned before, when adding or deleting sections no recalculation of the coefficients is needed.

The procedure for recursively obtaining the estimator $H_{p,T}$ and the estimate $\hat{d}_{p,T+1}$ is as follows:

1. Assume that all quantities are known up to and including time T.
2. Start with $e_{p,T+1} = e^{0}_{p,T+1} = r_{p,T+1} = r^{0}_{p,T+1} = y_{T+1}$ for $p = 0$.
3. Use eqs. (22) and (25) to compute

$$e^{0}_{1,T+1} = e^{0}_{0,T+1} - (k_{0,T}R^{-r}_{0,T-1})r^{0}_{0,T}$$

$$r^{0}_{1,T+1} = r^{0}_{0,T} - (k_{0,T}R^{-e}_{0,T})e^{0}_{0,T+1}.$$

4. Use eq. (26) to compute $k_{0,T+1} = k_{0,T} + e^{0}_{0,T+1}r_{0,T}$.
5. Use eqs. (36) and (39) to compute

$$R^{e}_{0,T+1} = R^{e}_{0,T} + e_{0,T+1}e^{0}_{0,T+1}$$

$$R^{r}_{0,T+1} = R^{r}_{0,T} + r_{0,T+1}r^{0}_{0,T+1}.$$

6. Compute the gain terms $k_{0,T+1}R^{-r}_{0,T}$ $k_{0,T+1}R^{-e}_{0,T+1}$ to obtain from eqs. (27) and (28)

$$e_{1,T+1} = e_{0,T+1} - (k_{0,T+1}R^{-r}_{0,T})r_{0,T}$$

$$r_{1,T+1} = r_{0,T} - (k_{0,T+1}R^{-e}_{0,T+1})e_{0,T+1}.$$

These gain terms can be saved for the next recursion.

7. Repeat steps 3 to 6 for $p = 1, 2, \cdots$
8. Use eq. (61) to compute the tentative estimate

$$\hat{d}^{0}_{p,T+1} = H^{t}_{p,T}\bar{r}^{0}_{p,T+1}.$$

9. To update $H_{p,T}$ start with $V_{-1,T+1} = d_{T+1}$ from eqs. (58) and (59) and use eq. (60) to compute

$$h_{0,T+1} = h_{0,T} + R^{-r}_{0,T+1}(V_{-1,T+1} - h_{0,T}r_{0,T+1})r^{0}_{0,T+1}.$$

10. Use eq. (59) to compute

$$V_{0,T+1} = V_{-1,T+1} - h_{0,T+1}r_{0,T+1}.$$

11. Repeat steps 9 and 10 for $p = 1, 2, \cdots$.
12. Use eq. (48) to compute $d_{p,T+1} = H^{t}_{p,T+1}\overline{r_{p,T+1}}$.

Steps 3, 6, 8, and 10 can be drawn in a block diagram like in Fig. 1. The variable gain terms are $k_{p,T}R^{-r}_{p,T-1}$, $k_{p,T}R^{e}_{p,T}$, and $h_{p,T}$, and they are updated in steps 4, 6, and 9. When the system reaches a steady state, it can be illustrated in a simpler form as shown in Fig. 2.

4.2.1 Starting the algorithm

The problem of finding the optimal predictor/estimator of order p is not well defined if there are less than p input points. Therefore, when starting the algorithm, p should be 0 in the first recursion, 1 in the second, and p should grow linearly in time until it reaches the desired number of sections of the lattice. This is in contrast with the

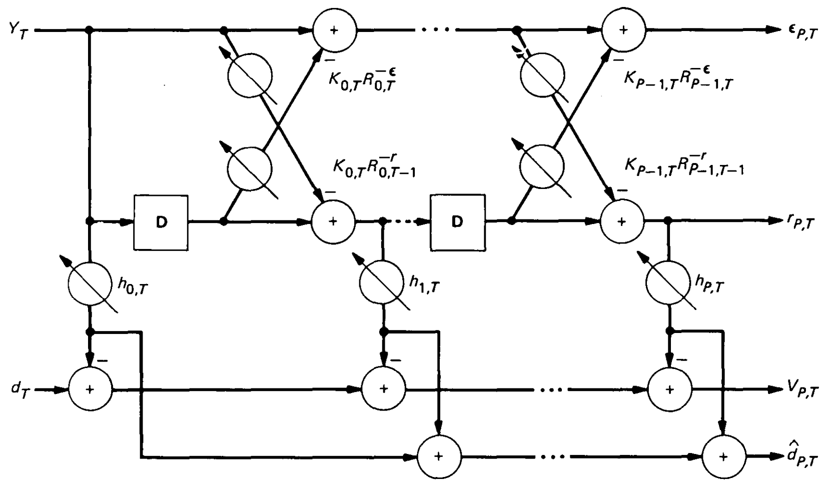

Fig. 1—The basic form of the lattice estimator.

fast Kalman algorithm, where a small diagonal matrix is assumed for $R_{p,0}$.

4.2.2 Number of operations

The number of operations required for the three algorithms, the simple gradient, the fast Kalman, and the lattice, are given below where p is the number of adaptive parameters:

Algorithm	Multiplications	Additions	Divisions
Gradient	$2p$	$2p$	—
Fast Kalman	$10p$	$9p$	2
Lattice	$12p$	$11p$	$3p$

V. DISCUSSION

It was shown in eq. (9) that the optimal linear estimator that yields the least total mse is obtained by matrix inversion. A recursive algorithm to update the optimal estimator also involves an inversion of the correlation matrix of the data as in eq. (43). If the input data are uncorrelated (i.e., low signal embedded in flat noise, or data signal with Nyquist spectral shape), then multiplying by $R^{-1}_{p,t}$ is equivalent to scalar division, which is the simple gradient algorithm. However, if the data are highly correlated and $R_{p,T}$ has its eigenvalues spread out ($\lambda_{\max}/\lambda_{\min} \gg 1$), then the optimal recursive algorithm for the fastest convergence of the estimator is more complex: The estimator can still have the form of a tapped delay line, but now the shift properties of

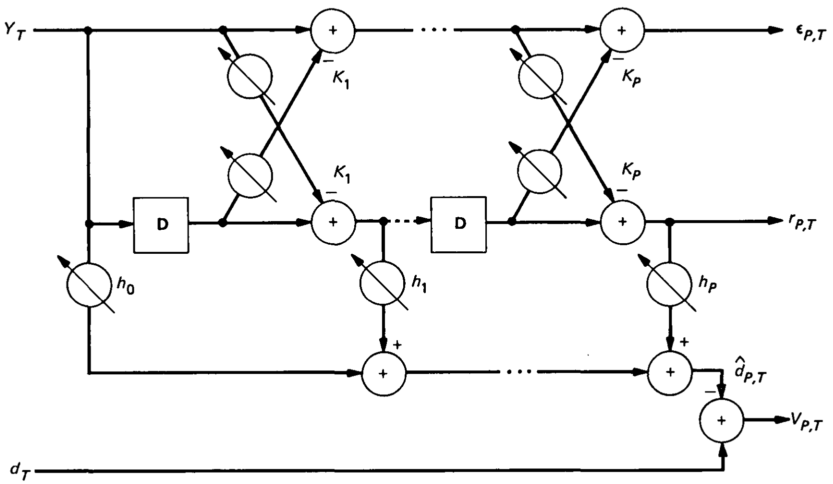

Fig. 2—The basic form of the steady state lattice estimator.

$R_{p,T}$ are used to update $w_{p,T}$ in $0(p)$ multiplications. A different approach demonstrated in this paper is to transform the input data using the lattice network to get uncorrelated inputs to the estimator. The estimator weights are now different but related to the original set through the same transformation eq. (48). This is similar to Ref. 6, except that the transformation matrix is time variant. The performance of the nonstationary lattice and the fast Kalman are the same—both give the minimal error—and in Ref. 2 it is demonstrated that convergence time can be reduced by a factor of 15, compared to the simple gradient algorithm.

The differences between the lattice and the fast Kalman algorithms in practical, finite precision digital implementation should be fully discussed elsewhere. However, an example can be given here. Examining step 5 for the Kalman algorithm, eq. (36) may occasionally render $R^e_{p,T+1}$ which is nonpositive because of the accumulation of arithmetic errors. The algorithm is useless from that time on, and has to be restarted. On the other hand, for the lattice algorithm, step 5, if either $R^e_{p,T+1}$ or $R^r_{p,T+1}$ become nonpositive, force them to be some small but positive number, and make all $k_{i,T+1}$ equal zero for $i \geqslant p$. The updating algorithm then falls back to the gradient algorithm from tap p and on, or the filter length can be shortened to length p, as desired.

Future work should try and make use of the above recurrence update relations for the exponentially weighted errors, the "fading memory" case under time-varying situations. Also simpler, suboptimal algorithms can be derived and should be compared to the exact algorithm in terms of performance and complexity.

VI. CONCLUSIONS

(*i*) The lattice algorithm gives identical results to the fast Kalman algorithm for adapting filter coefficients when both have the same number of coefficients.

(*ii*) The number of multiplications for the two algorithms is about the same, but the lattice requires more divisions for normalization by the residual error energy at each stage.

(*iii*) Changing the number of taps is easier under the lattice algorithm.

(*iv*) In limited-precision implementation under severe amplitude distorting channels, the last property of the lattice algorithm may be valuable in providing better performance.

APPENDIX

A.1 Derivation of the order update of $A_{p,T}$

From eqs. (14), (18), and (20), it is found that

$$R_{p+1,T}\begin{pmatrix}A_{p,T}\\ 0\end{pmatrix} = \begin{pmatrix}R_{p,T} & X\\ X & X\end{pmatrix}\begin{pmatrix}A_{p,T}\\ 0\end{pmatrix} = \begin{pmatrix}R^e_{p,T}\\ 0^p\\ k_{p,T}\end{pmatrix}, \tag{64}$$

and similarly

$$R_{p+1,T}\begin{pmatrix}0\\ B_{p,T-1}\end{pmatrix} = \begin{pmatrix}k'_{p,T}\\ 0^p\\ R^r_{p,T-1}\end{pmatrix} \tag{65}$$

for some $k_{p,T}, k'_{p,T}$. From the fact that $A_{p,T}(0) = B_{p,T-1}(p) \equiv 1$ it can be shown that

$$\begin{aligned} k_{p,T} &= (0\ B^t_{p,T-1})\begin{pmatrix}R^e_{p,T}\\ 0^p\\ k_{p,T}\end{pmatrix} = (0\ B^t_{p,T-1})R_{p+1,}T\begin{pmatrix}A_{p,T}\\ 0\end{pmatrix}\\ &= (k'_{p,T}(0^p)^t R^r_{p,T-1})\begin{pmatrix}A_{p,T}\\ 0\end{pmatrix} = k'_{p,T}. \end{aligned} \tag{66}$$

Combining eqs. (64) and (65) in a proper way, it is found that

$$R_{p+1,T}\left[\begin{pmatrix}A_{p,T}\\ 0\end{pmatrix} - k_{p,T}R^{-r}_{p,T-1}\begin{pmatrix}0\\ B_{p,T-1}\end{pmatrix}\right] = \begin{pmatrix}R^e_{p+1,T}\\ 0^{p+1}\end{pmatrix}, \tag{67}$$

with

$$R^{-r}_{p,T-1} \equiv (R^r_{p,T-1})^{-1}. \tag{68}$$

Therefore,

$$A_{p+1,T} = \begin{pmatrix}A_{p,T}\\ 0\end{pmatrix} - k_{p,T}R^{-r}_{p,T-1}\begin{pmatrix}0\\ B_{p,T-1}\end{pmatrix}. \tag{69}$$

A.2 The order update of $C_{p,T}$

For the order update of $C_{p,T}$, note that

$$R_{p+1,T}\begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} = \begin{pmatrix} y_{p,T} \\ X \end{pmatrix} \tag{70}$$

and

$$R_{p+1,T}B_{p+1,T} = \begin{pmatrix} 0^{p+1} \\ R^{r}_{p+1,T} \end{pmatrix}. \tag{71}$$

From this it can be seen that $C_{p+1,T}$ is a linear combination of $\begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix}$ and $B_{p+1,T}$.
From the relation

$$\begin{aligned}((0^{p+1})^{t}R^{r}_{p+1,T})C_{p+1,T} &= B^{t}_{p+1,T}R_{p+1,T}C_{p+1,T} \\ &= B^{t}_{p+1,T}y_{p+1,T} = r_{p+1.T},\end{aligned} \tag{72}$$

it then must be that

$$\begin{aligned}C_{p+1,T} &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + r_{p+1,T}R^{-r}_{p+1,T}B_{p+1,T} \\ &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + \mu_{p+1,T}B_{p+1,T} = \begin{pmatrix} X \\ \mu_{p+1,T} \end{pmatrix},\end{aligned} \tag{73}$$

with the definition

$$\mu_{p+1,T} = r_{p+1,T}R^{-r}_{p+1,T}, \tag{74}$$

i.e., $\mu_{p+1,T}$ is the last term of $C_{p+1,T}$.
Therefore,

$$\gamma_{p+1,T} = C^{t}_{p+1,T}y_{p+1,T} = \gamma_{P,T} + r^{2}_{p+1,T}R^{-r}_{p+1,T}, \tag{75}$$

and, thus,

$$\gamma_{p,T} = \sum_{i=0}^{p} r^{2}_{i,T}R^{-r}_{i,T}. \tag{76}$$

A.3 The time update relations

From eq. (33) the time update of $A_{p,T}$ is obtained as follows:

$$\begin{aligned}R_{p,T+1}A_{p,T} &= (R_{p,T} + y_{p,T+1}y^{t}_{p,T+1})A_{p,T} \\ &= \begin{pmatrix} R^{e}_{p,T} \\ 0^{p} \end{pmatrix} + y_{p,T+1}e^{0}_{p,T+1} \\ &= \begin{pmatrix} R^{e}_{p,T+1} \\ 0^{p} \end{pmatrix} + R_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} e^{0}_{p,T+1}\end{aligned} \tag{77}$$

for some $R^e_{p,T+1}$, with the definition (35) for $e^0_{p,T+1}$. From eq. (77), it can be seen that

$$A_{p,T+1} = A_{p,T} - e^0_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}, \tag{78}$$

and multiplying both sides by $y_{p,T+1}$ gives the relation

$$e_{p,T+1} = e^0_{p,T+1}(1 - \gamma_{p-1,T}) \quad \text{for} \quad p = 1, 2, \cdots \tag{79}$$

As for $p = 0$, we get from the definition that

$$e_{0,T+1} = e^0_{0,T+1} \quad \text{or} \quad \gamma_{-1,T} \equiv 0.$$

The time update of $B_{p,T}$ is obtained similarly:

$$R_{p,T+1}B_{p,T} = (R_{p,T} + y_{p,T+1}y^t_{p,T+1})B_{p,T} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + y_{p,T+1}r^0_{p,T+1}$$

$$= \begin{pmatrix} 0^p \\ R^r_{p,T+1} \end{pmatrix} + R_{p,T+1}\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} r^0_{p,T+1}. \tag{80}$$

Therefore,

$$B_{p,T+1} = B_{p,T} - r^0_{p,T+1}\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix}. \tag{81}$$

As in eqs. (36), (78), and (79),

$$R^r_{p,T+1} = R^r_{p,T} + r_{p,T+1}r^0_{p,T+1} \tag{82}$$

and

$$r_{p,T+1} = r^0_{p,T+1}(1 - \gamma_{p-1,T+1}) \quad \text{for} \quad p = 1, 2, \cdots \tag{83}$$

and

$$r_{0,T+1} = r^0_{0,T+1} = y_{t+1}.$$

The time update of $B_{p,T}$ can also be obtained as follows:

$$R_{p,T+1}B_{p,T} = (R_{p,T} + y_{p,T+1}y^t_{p,T+1})B_{p,T}$$

$$= \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + y_{p,T+1}r^0_{p,T+1} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + R_{p,T+1}C_{p,T+1}r^0_{p,T+1}. \tag{84}$$

Thus,

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1}C_{p,T+1}) \times \frac{1}{1 - r^0_{p,T+1}\mu_{p,T+1}}, \tag{85}$$

where the denominator is chosen to make $b_{p,T+1}(0) \equiv 1$ using the definition (32). The time update of $C_{p,T}$ is obtained as in eqs. (70) to (73).

$$R_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} = \begin{pmatrix} X \\ y_{p-1,T} \end{pmatrix} \tag{86}$$

and

$$R_{p,T+1}A_{p,T+1} = \begin{pmatrix} R^e_{p,T+1} \\ 0^p \end{pmatrix}. \tag{87}$$

Therefore, $C_{p,T+1}$ is a linear combination of

$$\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}$$

and $A_{p,T+1}$. From

$$[R^e_{p,T+1}(0^p)^t]C_{p,T+1} = A^t_{p,T+1}R_{p,T+1}C_{p,T+1} = A^t_{p,T+1}y_{p,T+1} = e_{p,T+1}, \tag{88}$$

it is found that

$$C_{p,T+1} = \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} + e_{p,T+1}R^{-e}_{p,T+1}A_{p,T+1}. \tag{89}$$

Multiplying by $y_{p,T+1}$ gives

$$\gamma_{p,T+1} = \gamma_{p-1,T} + e^2_{p,T+1}R^{-e}_{p,T+1} \tag{90}$$

and

$$\gamma_{p,T+1} = \sum_{i=0}^{p} e_{i,T+i+1-p}R^{-e}_{i,T+i+1-p}. \tag{91}$$

For the time update of $K_{p,T}$, use the definitions (64) and (65):

$$\begin{aligned} k_{p,T+1} &= [k_{p,T+1}(0^p)^t R^r_{p,T}] \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \\ &= (0\ B^t_{p,T})(R_{p+1,T} + y_{p+1,T+1}y^t_{p+1,T+1}) \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \\ &= (0\ B^t_{p,T}) \begin{pmatrix} R^e_{p,T} \\ 0^p \\ k_{p,T} \end{pmatrix} + r_{p,T}e^0_{p,T+1} = k_{p,T} + e^0_{p,T+1}r_{p,T} \end{aligned} \tag{92}$$

Using eqs. (79) and (83) an alternative form is

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1}r^0_{p,T}(1 - \gamma_{p-1,T}) = k_{p,T} + e_{p,T+1}r^0_{p,T}. \tag{93}$$

REFERENCES

1. R. D. Gitlin and F. R. Magee, Jr., "Self Orthogonalizing Adaptive Equalization Algorithms," IEEE Trans. Commun. *COM-25,* No. 7 (July 1977), pp. 666–72.
2. D. Godard, "Channel Equalization Using Kalman Filter for Fast Data Transmission," IBM J. Res. and Dev., *18,* No. 3, (May 1974), pp. 267–73.
3. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," Report LiTH-ISY-I-0158, Dept. of Electrical Engineering, Linkoping University, S-58183 Linkoping, Sweden, 1977.
4. L. Ljung, M. Morf, and D. Falconer, "Fast Calculation of Gain Matrices for Recursive Estimation Schemes," Int. J. Control, *27,* No. 1 (January 1978), pp. 1–19.
5. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," IEEE Trans. Commun., *COM-26,* No. 10 (October 1978), pp. 1439–46.

6. R. W. Chang, "A New Equalizer Structure for Fast Start-Up Digital Communication," B.S.T.J., *50,* No. 6 (July-August 1971), pp. 1969–2014.
7. J. Makhoul, "A Class of All-Zero Lattice Digital Filters: Properties and Applications," IEEE Trans. Acoust., Speech and Sig. Proc., *ASSP-26,* No. 4 (August 1978), pp. 304–14.
8. E. Shichor and J. Reiss, "Sequential Algorithm for Estimation of Parameters of an Autoregressive Process," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, Hartford, Connecticut (May 1977), pp. 744–6.
9. L. J. Griffith, "A Continuously Adaptive Filter Implemented as a Lattice Structure," ibid., pp. 683–6.
10. L. J. Griffiths, "An Adaptive Lattice Structure for Noise Canceling Applications," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, Tulsa, Okla., pp. 87–90, 1978.
11. E. H. Satorious and S. T. Alexander, "Rapid Equalization of Highly Dispersive Channels Using Lattice Algorithms," Naval Ocean Systems Center Tech. Report 249, San Diego, California, April, 1978.
12. M. Morf, "Ladder Forms in Estimation and System Identification," IEEE 11th Annual Asilomar Conf. Circuits, Systems, and Computers, Pacific Grove, California, November 1977, pp. 424–9.
13. M. Morf, A. Vieira, and D. T. Lee, "Ladder Forms for Identification and Speech Processing," Proc. 1977 IEEE Conf. Decision and Control, New Orleans, Louisiana (December 1977), pp. 1074–8.
14. T. L. Lim and M. S. Mueller, "Rapid Equalizer Start-up Using Least Square Algorithms," Int. Conf. Comm., Seattle, Washington, June 1980.
15. E. H. Satorius and J. D. Pack, "Application of Least Squares Lattice Algorithms to Adaptive Equalization," IEEE Trans. Comm., *COM-29,* No. 2 (February 1981), pp. 136–42.

CONTRIBUTORS TO THIS ISSUE

A. Ionescu-Graff, B.S. (Operations Research), 1977; M.S., 1978, Columbia University; Bell Laboratories, 1977—. Mrs. Ionescu-Graff has been concerned with various aspects of traffic network engineering and administration. She is currently involved in special-services network architecture studies. Member, Tau Beta Pi.

James P. Moreland, B.S.E.E., 1964; M.S.E.E., 1964; Ph.D. (E.E.), 1967, Ohio State University; Research Associate, Electroscience Laboratory, 1964–1968, Instructor, Electrical Engineering, 1967–1968, Ohio State University; Bell Laboratories, 1968—. At Ohio State, Mr. Moreland worked on studies of scattering theory and optical heterodyne detection. At Bell Laboratories, he has been concerned with clock-synchronization schemes for digital communications networks, optical-fiber transmission studies, and traffic and facility network planning. He is presently a supervisor in the Trunk Traffic Engineering Department. Member, IEEE, Eta Kappa Nu, Tau Beta Pi, Sigma Xi.

Charles D. Pack, B.S. (E.E.), 1965, University of Delaware; M.S. (Operations Research), 1967, Johns Hopkins University; D. Eng. Sci. (Operations Research), 1972, Columbia University; Bell Laboratories, 1965—. Mr. Pack is now Supervisor of the Special Services Network Topology Group, responsible for robust network designs and performance planning for special services. Member, ORSA, Tau Beta Pi, Omicron Delta Kappa, Phi Kappa Phi.

Eliahu Shichor, B.Sc. (Electrical Engineering), 1962, Technion, Israel Institute of Technology, Haifa, Israel; M.Sc., 1968, Ph.D., 1969, Brown University, Providence, R.I.; Raphael, Armament Development Authority, 1969—; Bell Laboratories, 1978–1979; Adjunct Faculty (Electrical Engineering), Technion, Israel Institute of Technology, 1969—. Mr. Shichor is working on digital signal processing problems related to speech digitization, adaptive filtering, and digital communications. Member, IEEE.

C. Russell Szelag, B.S. (Mathematics), 1972, Stevens Institute of Technology; M.S., 1974, M.S. (Computer Science), 1977, University of Wisconsin-Milwaukee; Bell Laboratories, 1977—. As a member of the Trunk Traffic Engineering Department, Mr. Szelag has worked on various aspects of trunk network administration for the public telephone network. Member, IEEE.

Bruce A. Whitaker, B.S. (Applied Mathematics), 1967, North Carolina State University; M.S. (Operations Research), 1968, Johns Hopkins University; Ph.D. (Operations Research), 1974, New York University; Bell Laboratories, 1967—. Mr. Whitaker is Supervisor of the Network Forecasting Group, responsible for the formulation of trunk forecasting methods. Member, ORSA, Pi Mu Epsilon, Phi Kappa Phi, Phi Eta Sigma.

PAPERS BY BELL LABORATORIES AUTHORS

COMPUTING/MATHEMATICS

Algorithms for Multiple-Criterion—Design of Microprogrammed Control Hardware. A. W. Nagle and A. C. Parker, Proc 18th Design Automation Conf (June 29, 1981), pp 486-93.

Construction and Display of Three-Dimensional Polygonal-Histograms. R. Hilbert, Computer Graphics, *15,* No. 2 (July 1981), pp 230–41.

Heavy-Traffic Limits for Queues With Many Exponential Servers. S. Halfin and W. Whitt, Oper Res, *29* (May–June 1981), pp 567–88.

IEEE 488 Error Handling Techniques: Pros and Cons. L. F. Santora, Jr., Computer Design, *20,* No. 6 (June 1981), pp 143–7.

An Information Theoretic Approach to Digital Fault Testing. V. D. Agrawal, IEEE Trans Computers, *C-30,* No. 8 (August 1981), pp 582–7.

The Nonexistence of a Certain Steiner System S(3,12,112). N. J. A. Sloane and J. G. Thompson, J Combinatorial Theory, *30A* (1981), pp 209–36.

Online Information Retrieval Bibliography—Fourth Update. D. T. Hawkins, Online Review, *5,* No. 2 (1981), pp 139–82.

On Stochastic Bounds for the Delay Distribution in the GI/G/A Queue. W. Whitt, Oper Res, *29* (May-June 1981), pp 604–8.

Tables of Sphere Packings and Spherical Codes. N. J. A. Sloane, IEEE Trans Inform Theory, *IT-27* (1981), pp 327–38.

Uniform Syntax for Type Expressions and Declarators. R. Sethi, Software—Practice and Experience, *11,* No. 6 (June 1981), pp 623–8.

Uniqueness of Certain Spherical Codes. E. Bannai and N. J. A. Sloane, Can J Math, *33* (1981), pp 437–49.

Vector Coding Methods for Compaction of Scan-Digitized Line Drawing Images. W. Pferd and J. P. White, Proc 2nd Scandinavian Conf Image Analysis (June 15, 1981), pp 297–302.

ENGINEERING

Bending, Damping, and Fatigue of Metals. G. F. Weissmann, Experimental Mech, *21* (July 1981), pp 255–60.

Experimental Results of 20-Mb/s FSK Digital Transmission on 4-GHz (TD) Radio. Y. Y. Wang, ICC Conf Record, *1* (June 14, 1981), pp 13.4.1–5.

High-Order Intermodulation Effects in Digital Satellite Channels. K. Y. Eng and O.-C. Yue, IEEE Trans Aerospace Elec Syst, *AES-17,* No. 3 (May 1981), pp 438–45.

Laser-Patterned Ta_2N Resistors for Thin Film Circuits. L. J. Kiszka and P. L. Scarff III, Proc 1981 31st Elec Components Conf (May 11–13, 1981), pp 449–55.

Microprocessors in Power Control, the Time has Come. P. F. Gensinger and E. A. Rosin, INTELEC 1981 Conf Proc (May 19-21, 1981), pp 1-6.

Progress in Multimode and Single-Mode Lightguides Prepared by the MC&D Process. M. I. Cohen, Tech Digest, IEEE/OSA Conf Lasers and Electro-Optics (June 10-12, 1981), pp 12–4.

PHYSICAL SCIENCES

Carbon-13 Nuclear Magnetic Resonance Studies of Solid Segmented Copolymers. 1. Mobile Domains of a Polyester Thermoplastic Elastomer. L. W. Jelinski, F. C. Schilling, and F. A. Bovey, Macromolecules, *14,* No. 3 (1981), pp 581–6.

Crystal Structure and Pyroelectric Coefficient of $Co(IO_3)_2$ and Structural Relationships Among the Anhydrous Noncentrosymmetric 3-*d*-Transition Metal Iodates. C. Svensson, S. C. Abrahams, and J. L. Bernstein, J Solid State Chem, *36* (February 1981), pp 195–204.

Development and Evaluation of a Pre-Encapsulation Cleaning Process to Improve Reliability of HIC's with Aluminum Metallized Chips. M. Iannuzzi, Proc. IEEE 31st Elect Comp Conf (May 1981), pp 228–37.

Electrochromic Cells With Iridium Oxide Display Electrodes. W. C. Dautremont-Smith, L. M. Schiavone, S. Hackwood, G. Beni, and J. L. Shay, Solid State Ionics, *2,* No. 1 (April 1981), pp 13–8.

Electrolyte-Oxide-Semi-Conductor Junction at the p-Indium and Phosphide/Vanadium (2+)-Vanadium (3+) Interface. S. Menezes, H. J. Lewerenz, F. A. Thiel, and K. J. Bachmann, Appl Phys Lett, *38,* No. 9 (May 1981), pp 710.

Enhanced Raman Scattering from Adsorbates on Metal Films in Ultra-High Vacuum. T. H. Wood, M. V. Klein, and D. A. Zwemer, Surface Science, *107* (1981), pp 625–35.

Improved Electrochromic Behavior of Reactively Sputtered Iridium Oxide Films. L. M. Schiavone, W. C. Dautremont-Smith, G. Beni, and J. L. Shay, J Electrochem Soc, *128,* No. 6 (June 1981), pp 1339–42.

Impurity States in Amorphous As_2Te_3. J. J. Hauser, R. S. Hutton, and A. Staudinger, Phil Magazine, *44,* No. 1 (July 1981), pp 109–25.

Liquid Phase Sintering of a CrCoFe Permanent Magnet Alloy. M. L. Green and C. C. Wong, Mod Dev in Powder Metallurgy, *12* (July 1981), pp 453–72.

Multiphoton Ionization Mass Spectroscopy of Deuterated Analogs of Acetaldehyde: Evidence for Deuterium Scrambling. G. J. Fisanick and T. S. Eichelberger, IV, J Chem Phys, *74,* No. 12 (June 15, 1981), pp 6692–9.

Multiphoton Ionization Spectroscopy of the 3_s Rydberg State in the Deuterated Acetaldehydes. T. S. Eichelberger, IV and G. J. Fisanick, J Chem Phys, *74,* No. 11 (June 1, 1981), pp 5962-70.

Paraelectric-Paraelastic $Rb_2KM_oO_3F_3$ Structure at 343 and 473K. S. C. Abrahams, J. L. Bernstein, and J. Ravez, Acta Crystallographica, *B37,* No. 7 (July 15, 1981), pp 1332–6.

Phase Separation Behavior of Rubber-Modified Epoxies. T. T. Wang and H. M. Zupko, J Appl Polym Sci, *26* (1981), pp 2391–401.

Phase Transitions at 462 and 495K in Barium Nitrite. S. C. Abrahams, P. K. Gallagher, H. M. O'Bryan, and R. Liminga, J Appl Phys, *52,* No. 4 (April 1981), pp 2837–40.

Plasma Etching of III-V Compound Semiconductor Materials and Their Oxides. G. Smolinsky, R. P. Chang, and T. M. Mayer, J Vac Sci Tech, *18,* No. 1 (January/February 1981), pp 12–6.

Response of Piezoelectric Transducers Used in Pulsed Optoacoustic Spectroscopy. E. T. Nelson and C. K. N. Patel, Opt Lett, *6,* No. 7 (July 1981), pp 354–6.

Surface Second-Harmonic Generation From Metal Island Films and Microlithographic Structures. A. Wokaun, J. G. Bergman, J. P. Heritage, A. M. Glass, P. F. Liao, and D. H. Olson, Phys Rev B, *24,* No. 2 (July 15, 1981), pp 849–56.

Synthesis and Characterization of $(nBu_4N)_3Re(NCS)_6$, a Rhenium (III) Thiocyanate Complex. H. S. Trop, A. Davison, and A. G. Jones, Inorganica Chimica Acta, *54* (August 1981), pp L61.

X-Ray Microbeam System With a Discrete-Spot Target. O. Fujimura and M. E. Haskin, Proc SPIE—The Int Soc Opt Eng, *273* (1981), pp 244–53.

SPEECH AND ACOUSTICS

Body-Cover Theory of the Vocal Fold and its Phonetic Implications. O. Fujimura, Vocal Fold Physiology, (1981), pp 271–88.

Temporal Organization of Articulatory Movements as a Multidimensional Phrasal Structure. O. Fujimura, Phonetica, *38* (1981), pp 66–83.

CONTENTS, FEBRUARY 1982

THE BELL SYSTEM TECHNICAL JOURNAL is abstracted or indexed by *Abstract Journal in Earthquake Engineering, Applied Mechanics Review, Applied Science & Technology Index, Chemical Abstracts, Computer Abstracts, Current Contents/ Engineering, Technology & Applied Sciences, Current Index to Statistics, Current Papers in Electrical & Electronic Engineering, Current Papers on Computers & Control, Electronics & Communications Abstracts Journal, The Engineering Index, International Aerospace Abstracts, Journal of Current Laser Abstracts, Language and Language Behavioral Abstracts, Mathematical Reviews, Science Abstracts (Series A, Physics Abstracts; Series B, Electrical and Electronic Abstracts; and Series C, Computer & Control Abstracts), Science Citation Index, Sociological Abstracts, Social Welfare, Social Planning and Social Development,* and *Solid State Abstracts Journal.* Reproductions of the Journal by years are available in microform from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Bell System