AN ADAPTIVE ALGORITHM TO EVALUATE CLOCK PERFORMANCE IN REAL TIME*

Dr. James A. Barnes
Austron
Boulder, Co.

Abstract

Kalman filters and ARIMA models provide optimum control and evaluation techniques (in a minimum squared error sense) for clocks and precision oscillators. Typically, before the models can be used, an analysis of data provides estimates of the model parameters (e.g., the phi's and theta's for an ARIMA model). These model parameters are often evaluated in a batch mode on a computer after a large amount of data is obtained.

An alternative approach is to devise an adaptive algorithm which "learns" the important parameters while the device is being used and up-dates the parameters recursively. Clearly, one must give up some amount of precision if one deviates even slightly from the truly optimum techniques, but, as this study shows, the costs in performance are not large at all. If one chooses the best sampling intervals, the loss in precision can be negligible.

The physical models used in this paper are based on the assumption of a combination of white PM, white FM, random walk FM, and linear frequency drift. In ARIMA models, this is equivalent to an ARIMA(0,2,2) with a non-zero average second difference. Using simulation techniques, this paper compares real-time estimation techniques with the conventional batch mode. The criterion for judging performance is to compare the mean square errors of prediction between the batch mode and the recursive mode of parameter estimation operating on the same data sets.

INTRODUCTION

Before working directly on the ARIMA(0,2,2) models [1], it is of value to establish a few important relations. An ARIMA(1,0,0) model is often referred to as an exponential filter since its impulse response function is an exponential. That is:

$$\begin{aligned} x_n &= \phi X_{n-1} + a_n \\ &= \phi^n \qquad \text{for } n = 1,\ 2,\ \ldots \end{aligned} \tag{1}$$

Where the input to the filter, a_n, is taken to be $a_0 = 1$ and $a_n = 0$ for $n > 0$ (the unit impulse). This filter is the digital equivalent of a simple $R-C$ low-pass filter with $\phi = \exp(-t/RC)$ where

*Work supported in part by the Defense Communications Agency, DCA 100-88-c-0072

t is the time interval between steps in the counting index, n, and R and C are the resistance and capacitance values in the analog filter. In frequency domain, the filter transfer function is unity for low frequencies and drops off at -6dB/octave from the cut–off frequency, f_c , given by:

$$f_c = 1/(2\pi RC) \tag{2}$$

Given the cut–off frequency, f_c, and the sample time interval, t, the ARIMA coefficient, ϕ, can be calculated with the equation:

$$\phi = \exp(-2\pi f_c t_0) = \exp(-t_0/\tau) \tag{3}$$

Thus, for a given physical system, the ARIMA parameter is dependent on the data sampling rate.

The inverse function for the ARIMA(1,0,0) is just an ARIMA(0,0,1) which can be seen by solving Eq. 1 for a_n. Such a filter would be constant for the low frequency end changing to an increase of gain by +6dB/octave above the same cut–off frequency derived above.

The ARIMA models considered here are models for the phase of the clock comparisons (not their instantaneous frequencies). We can begin with a physical model of a clock with pure White FM. Since the ARIMA model is for phase, this first example would be given by an ARIMA(0,1,0) model, a random walk of phase. The power spectral density (of phase) for this noise is that of a constant decrease of -6dB/octave everywhere below the Nyquist frequency, $1/(2t_0)$.

We can add white noise modulation to this model by going to an ARIMA(0,1,1) model where the moving average parameter, θ, is computed from Eqs. 2 and 3, above, for $\tau = RC$. In actuality this is not an adequate model for many real clocks. One often encounters more low frequency divergent noises than the ARIMA(0,1,1) which require an additional integration: that is, one needs either an ARIMA(0,2,1) or an ARIMA(0,2,2). Physically, the ARIMA(0,2,1) model corresponds to a superposition of white FM and random walk FM. Again the transition between the two noise regimes is accomplished by using Eqs. 2 and 3, above.

If one now adds white PM one must go to the ARIMA(0,2,2) model (see Fig. 1), which has two break points corresponding to the transitions between random walk and white FM and between white FM and white PM. The equations above allow one to calculate the two parameters separately, say, θ'_1 and θ'_2, for theta–values of two cascaded, MA filters. These two filters can be combined in one MA(2) filter whose theta–parameters must be combined as factors to realize the correct MA filter. This combination is obtained as follows:

$$(1 - \theta'_1 B)(1 - \theta'_2 B) = (1 - \theta_1 B - \theta_2 B^2) \tag{4}$$

which yields $\theta_1 = (\theta'_1 + \theta'_2)$ and $\theta_2 = -(\theta'_1 \theta'_2)$ where B is the index lowering operator [1] and θ'_1 and θ'_2 are calculated using Eqs. 2 and 3. A linear frequency drift can also be important.

ADAPTIVE APPROACH TO TESTING ARIMA PARAMETERS

There are many methods of estimating parameters — for example, just a guess is one means. Of importance are issues such as bias, confidence intervals, efficiency, and likelihood. While it is easy to present various estimating procedures that "work" it is often difficult to evaluate how well they perform. This section develops theoretically the effects of errors in the ARIMA parameters.

Using simulation techniques, these errors are evaluated in "real–time" and compared to the conventional batch method, after the fact. The theoretical consequences of parameter errors are surprisingly mild. That is, many models are robust in regard to fairly poor estimates and they can give surprisingly good results. Still there are regions of operation where problems can arise. For example, taking data very frequently does not improve one's knowledge of basically long–term performance. Knowledge of the long–term performance simply requires long–term data.

The adaptive (real–time) estimation is based on the fact that an ARIMA(0,0,1) has an autocorrelation function given by:

$$R(n) = \begin{cases} 1 & \text{for } n = 0 \\ -\theta/(1+\theta^2) & \text{for } n = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Figure 2 presents a block diagram of an adaptive algorithm which effectively servos the theta–estimate (denoted "ϕ") to the "true" theta–value and also provides a current estimate of the variance of the residuals and a drift estimate. Basically the algorithm computes contributions to the first autocorrelation coefficient which in turn adjusts the estimated theta–value, effectively driving the first autocorrelation coefficient of the residuals to zero. The first autocorrelation coefficient of the residuals (given by Eq. 8, below) is proportional to the difference — (see Ref. [1]) giving both direction and value for the servo. Figure 3 depicts a simulation of the servo performance. Although the servo "works" we need to compare its performance with a conventional batch approach for estimation.

THE ARIMA(0,2,1) MODEL

Figure 4 is a diagram of the forecast system for an ARIMA(0,2,1) noise model. In this case, theta is the "real" parameter which is unknown, but is estimated with the value of phi. If phi were to equal theta, then the system would provide an optimal forecast of X_n. Since there will always be some error in the estimate of theta, it is of value to explore the consequences of such an error. The following is a detailed description of the estimation process and the evaluation of the errors caused by an error in phi.

Following Fig. 4, the data, X_n, are the only observables from the clock comparisons and the model is white noise FM and random walk FM: That is, an ARIMA(0,2,1) model. I explicitly assume that the model is a good model, but theta is unknown. Theta can be estimated by the methods given in Ref. [1], or by recursive filters developed here.

The output data, X_n, are filtered with an "inverse" filter and the residuals, W_n, are obtained. In the Box and Jenkins method, phi would be adjusted to give minimum variance to W_n. This, however, is accomplished in a batch mode after the fact and not in real–time. Regardless of how theta is estimated, phi is used in the forecaster as shown in Fig. 4: Indeed, there is no other value to use. With W_n as input to the forecaster (the switch in Fig. 4 in the "up" position), the system is allowed to run for a time to let all transients die out. At this point the estimated output, $\hat{X}_n = X_n$ exactly since the inverse filter is the exact inverse of the forecast filter.

At some point in time, $t = n + 1$, W_n is no longer available as input to the forecast filter and its input is set to zero (switch in the "down" position). For the first forecast value $\hat{X}_n + 1$, the two previous values, X_n and X_{n-1} are available to use in the forecast as shown. The error of the forecast (after the fact), can be found by subtraction of the two equations:

$$\begin{aligned} X_{n+1} &= 2X_n - X_{n-1} + a_{n+1} - \theta a_n \\ \hat{X}_{n+1} &= 2X_n - X_{n-1} - \theta W_n \\ \hline \delta_1 &= X_{n+1} - \hat{X}_{n+1} \qquad = \quad a_{n+1} - \theta a_n - \theta W_n \end{aligned} \tag{6}$$

where δ_1 is the error in the first forecast.

Following Box and Jenkins [1], W_n can be expressed as an infinite series which incorporates the psi–weights and the uncorrelated innovations, a_n:

$$W_n = \sum_{0}^{\infty} \Psi_i a_{n-i} \tag{7}$$

The psi–weights can be expressed in terms of phi, theta, and the innovations, a_n, as shown below by requiring W_n to satisfy the equation (see Fig. 4):

$$W_n - \phi W_{n-1} = a_n - \theta a_{n-1} \tag{8}$$

for all n, the result is given by:

$$\Psi = \begin{cases} 1 & \text{for } i = 0 \\ (\phi - \theta)\phi^{i-1} & \text{for } i = 1,\ 2,\ 3,\ \cdots \end{cases} \tag{9}$$

We can now evaluate the expected square of the first forecast error as:

$$E[\delta^2] = \sigma^2[1 + \frac{(\phi - \theta)^2}{1 - \phi^2}] \tag{10}$$

For phi equal to theta we obtain the classical result that the variance of the first forecast error is just the variance of the innovations.

We can repeat this calculation for $t = n + 2$ by using the forecast value X_{n+1} in place of the (unknown) value, X_{n+1}. Similarly, for $t = n + 3$ and so forth. The result can be summarized in the following formula for the mean square time interval error for M lags in the future:

$$E|\delta_M^2| = M\sigma_a^2 \left\{1 + (1-\theta)(M-1)[1 + (1-\theta)(2M-1)/6] + M(\phi - \theta)^2/(1-\phi^2)\right\} \tag{11}$$

This formula has been verified using simulation techniques.

There are two points to make in regard to this relation: (1) for phi equal to theta the result is identical to the classical results as it should be, and (2) as M gets larger, the variance grows as M–cubed but the term proportional to $(\phi - \theta)$ grows as M–squared. That is, for sufficiently large M, the errors of parameter estimation become unimportant. Figure 5 shows the regions of forecast errors (1) primarily due to conventional analyses and (2) those due primarily to an error in the estimate of theta, i.e., phi.

Figure 5 shows clearly that problems develop near theta=1 and near phi=1 with phi and theta not near each other in value. The problems near theta equal to 1 can be reduced by having longer term data. Having more frequent data doesn't help.

COMPARISONS BETWEEN ADAPTIVE AND CONVENTIONAL PARAMETER ESTIMATION

Given an ARIMA model corresponding to a physical model we can simulate a noise sample and treat the data as if it were real–time data and estimate the parameters recursively. The same data can be treated in batch mode and find those estimates which minimize the sum of the squares of the residuals. Table 1 summarizes the results of the estimation process. The program generated 100 noise samples each of 200 data points in duration. Each sample noise was processed through the adaptive estimation procedure developed here and the conventional Box and Jenkins [1] treatment. Of course, the "true" values of the parameters are also known since this is only a simulation.

TABLE 1 ARIMA(0,2,1) Theta = .9049 Sig-A = 1.1051			
Quantity		Conventional Method	Adaptive Method
Phi	(Actual=.9049)	.8983	.8577
	Std. dev. of mean	.0042	.0074
	Bias rel. Actual	-.0066	-.0471
	T-ratio	-1.58	-6.38
Sig-A	(actual=1.1051)	1.1161	1.1726
	Std. dev. of mean	.0119	.0146
	Bias rel. Actual	.0111	-.0674
	T-ratio	.93	-4.63

As expected, the conventional estimates are more accurate and more precise than the adaptive methods developed here. Still, with reference to Fig. 5, even fairly large errors in phi relative to theta are soon covered by the conventional errors. Table 1 shows that for theta = .9049, the adaptive method produces statistically significant biases (T–ratios of -4.63 and -6.38).

ARIMA(0,2,2) NOISE ANALYSIS

To estimate the theta–value the servo was based on the fact that an ARIMA(0,0,1) model has only the zeroth and first autocorrelation coefficients non–zero. An ARIMA(0,0,2) model has an additional non–zero coefficient at lag 2, which is strongly dependent on the second MA coefficient. The new servo has a separate loop which takes samples of the (lag n) (lag n-2) product and adjusts ϕ_2 to null the average similarly to the loop shown in Fig. 2, for ϕ_1.

Similarly to Fig. 3, Figure 6 depicts the transient response of the ARIMA(0,2,2) adaptive estimation. As noted above, the important performance is that of forecast errors not the intermediate values of θ_1 and θ_2 shown in Fig. 6. Still it does show how the parameters stabilize.

As noted above, the means of comparing algorithms is to compare the forecast errors for similar data situations. The Box and Jenkins method can be used to estimate the forecast errors using the "psi–weights", as in equations 8 and 9, above. The model for consideration now is as ARIMA(0,2,2). The theoretical forecast errors for the model can be computed similarly to Eq. 11:

$$E[\delta_M^2] = \sigma_\alpha^2\{1 + (M-1)[(1+\theta_2)^2 + M(1-\theta_1-\theta_2)[1+\theta_2+(1-\theta_1-\theta_2)(2M-1)/6]]\} \qquad (12)$$

For the complete model, a linear frequency drift term can be added. That is, we assume that one has an initial data set (chosen to be 100 time differences between a pair of clocks). The number of data points are the same for both the adaptive servo and the batch processing. The introduction of a non–zero frequency drift significantly affects both the adaptive servo and the conventional Box and Jenkins analysis. Equation 13, below, provides the additional, independent error to the forecast errors of Eq. 12 due to frequency drift:

$$D_M^2 = \sigma_a^2\{1 + (1-\theta_1)^2 + (\theta_1+\theta_2)^2 + \theta_2^2 + (M-2)(1-\theta_1-\theta_2)^2\}/M^2 \qquad (13)$$

This noise addition assumes that while θ_1 and θ_2 are known exactly, the drift term is estimated from the mean second difference time error and its variance. Appendix A contains a derivation of Eqs. 12 and 13.

It is important to realize that the actual errors in real applications are calculated on the basis of the ESTIMATED model parameters, not the "true" parameters which are unknown. This usage of the estimated parameters renders the computed errors a bit on the optimistic side because Eq. 12 does not include the errors in the estimated parameters. Figure 7a shows the theoretical contributions of the conventional Box and Jenkins analysis and the imperfect knowledge of the drift rate for various theta values. Figure 7 gives graphical views of Eqs. 12 and 13.

Figure 8a shows the theoretical forecast errors as calculated using Eqs. 12 and 13. For simulation purposes, the "true" parameter values are known. Simulation techniques allow one to verify Eqs. 12 and 13 by repeated calculation of the forecasts compared to the "true" values after the fact.

The bases for the data plotted in Fig. 8a, b, and c are:

1. ARIMA (0,2,2) model
2. Linear frequency drift
3. Data length of 100 points (Initial randomization of filter required)
4. Forecast 100 lags beyond the data length using estimated parameters
5. Repeat above for independent noise samples for at least 500 individual runs.

For Fig. 8a, the errors in the forecast for the simulated data are -.22 dB worse than the theoretical value. (This number is probably within the uncertainty limits of the experiment.) The solid line is the theoretical result of Eqs. 12 and 13. The small dots near the solid line are the results of simulation.

The next step was to estimate the theta parameters using a conventional Box and Jenkins approach. Figure 8b indicates the impact on the forecasts using imperfect parameters. The error is now about 1.01 dB worse than optimum after 100 lags. Theoretical errors (i.e., solid line) are identical in Figs. 8a, 8b, and 8c, as calculated using Eqs. 12 and 13.

The third step is to simulate the results of a "real–time" estimation procedure where the parameters are "learned" during a single pass through the 100 data points. That is, the adaptive approach is used for the forecasts. As shown on Fig. 8c, the error after 100 lags is now 1.14 dB relative to optimal or .13 dB relative to the Box and Jenkins approach. It is interesting to note that, at least for the given parameters, the adaptive forecaster is close to the Box and Jenkins forecaster in performance.

With the above approach, we can now evaluate the adaptive forecasting performance for various physical models by using different "true" values for the theta parameters, drift rate, and noise level. Further, we can estimate the relative performances of "perfect" parameters, Box and Jenkins estimations, and "real–time", adaptive method. In effect, we can evaluate the costs in accuracy of using the much simpler adaptive approach.

Reference

- G.E.P. Box and G.M. Jenkins, "Time Series Analysis", Holden–Day San Francisco, 1970.

APPENDIX A

PSI–WEIGHTS FOR AN ARIMA (0, 2, 1)

Given an ARIMA(0, 2, 1) Model:

$$X_n - 2 \cdot X_{n-1} + X_{n-2} - a_n + \theta \cdot a_{n-1} = 0$$

A-1

We define the Ψ–weights by the relation:

$$X_n = \sum_{i=0}^{\infty} \Psi_i \cdot a_{n-1} \qquad \text{for } n = 0 \text{ to } \infty$$

A-2

In order to have both (A-1) and (A-2) valid for all n, we substitute (A-2) into (A-1) and group coefficients of the a_n together and require the resulting coefficient to vanish, That is, the net coefficient of a_n is just $\Psi_0 - 1$. The first few relations for the Ψ–weights are:

$$\begin{aligned} \Psi_0 &= 1 \\ \Psi_1 &= (2-\theta) \\ \Psi_2 &= (3 - 2\cdot\theta) \\ \Psi_3 &= (4 - 3\cdot\theta) \\ &\vdots \\ \Psi_m &= (m+1) - m\cdot\theta \end{aligned}$$

A-3

The mean square forecast errors (See Box and Jenkins) are given by:

$$E\left\{\delta_M^2\right\} = \sigma^2 \sum_{m=0}^{M-1} \left\{(1-\theta)^2 \cdot m^2 + 2\cdot(1-\theta)\cdot m + 1\right\} \qquad \text{A-4}$$

$$= M\sigma_a^2 \{1 + (1-\theta)(M-1)\,[1 + (1-\theta)(2M-1)/6]\} \qquad \text{A-5}$$

PSI–WEIGHTS FOR AN ARIMA (0, 2, 2)

Given an ARIMA (0, 2, 2):

$$X_n - 2X_{n-1} + X_{n-2} - a_n + \theta_1 a_{n-1} + \theta_2 a_{n-2} = 0$$

A-6

We define the Ψ–weights as is (A–2). The first few relations for the Ψ–weights are:

$$\begin{aligned}
\Psi_0 &= 1 \\
\Psi_1 &= 2 - \theta_1 \\
\Psi_2 &= 2(2 - \theta_1) - (1 + \theta_2) \\
\Psi_3 &= 3(2 - \theta_1) - 2(1 + \theta_2) \\
&\vdots \\
\Psi_m &= \begin{cases} m(1 - \theta_1 - \theta_2) + (1 + \theta_2) & \text{for } m > 0 \\ 1 & \text{for } m = 0 \end{cases}
\end{aligned}$$

The mean square forecast errors become:

$$E\{\delta_M^2\} = \sigma_a^2 \left\{ 1 + (M-1) \left[(1+\theta_2)^2 + M(1-\theta_1-\theta_2) \left[1 + \theta_2 + (1-\theta_1-\theta_2)(2M-1)/6 \right] \right] \right\}$$

APPENDIX B

Mean and variance of an ARIMA (0, 2, 2)

$$Z_n = a_n - \theta_1 a_{n-1} - \theta_2 a_{n-2} + Av$$

B–1

The sum of the first M values of Z_n are:

$$\sum_{n=1}^{M} Z_n = a_M + (1-\theta_1)a_{M-1} + (1-\theta_1-\theta_2)a_{M-2} + \cdots - (\theta_1+\theta_2)a_0 - \theta_2 a_{-1} + M \cdot Av$$

B–2

Since the a_n are independent random numbers with zero mean, the mean value is obtained by dividing the sum, above, by M. The variance of the estimate is obtained by taking the expectation value of the square of (B–2). That is:

$$V^2 = \sigma^2 \left\{1 + (1-\theta_1)^2 + (\theta_1+\theta_2)^2 + (M-2)(1-\theta_1-\theta_2)^2 + \theta_2^2\right\} / M^2$$

B–3

where V^2 is the variance of the estimated mean of Z_n.

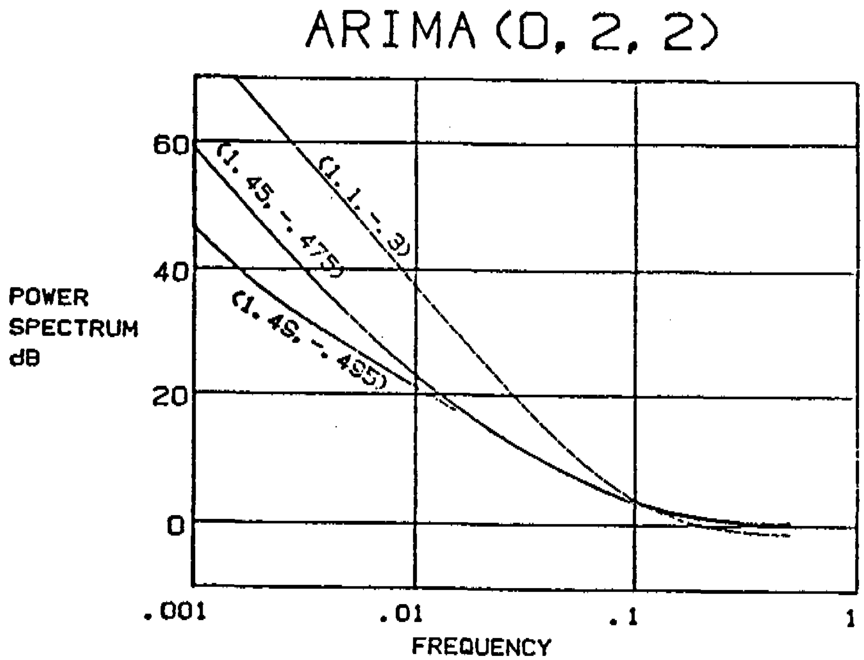
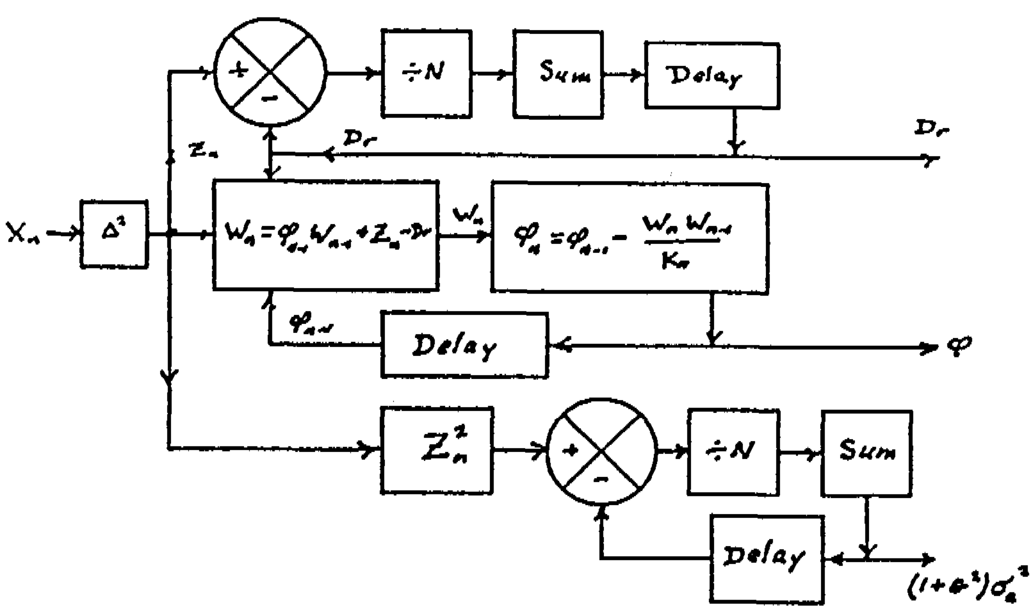


Fig. 1 Power Spectra for θ_1, θ_2.

FIG 2, ARIMA(0,2,1) Parameter Estimation



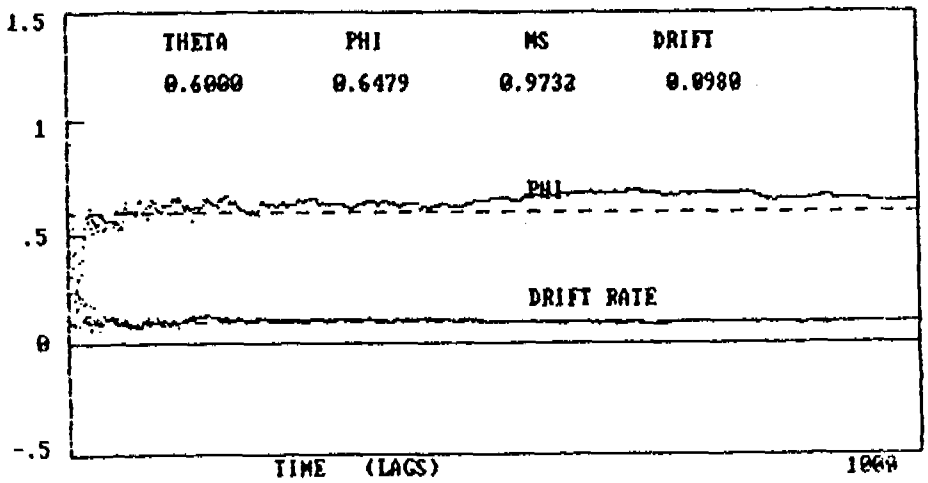


Fig. 3 Transient Response for an ARIMA(0,2,1)

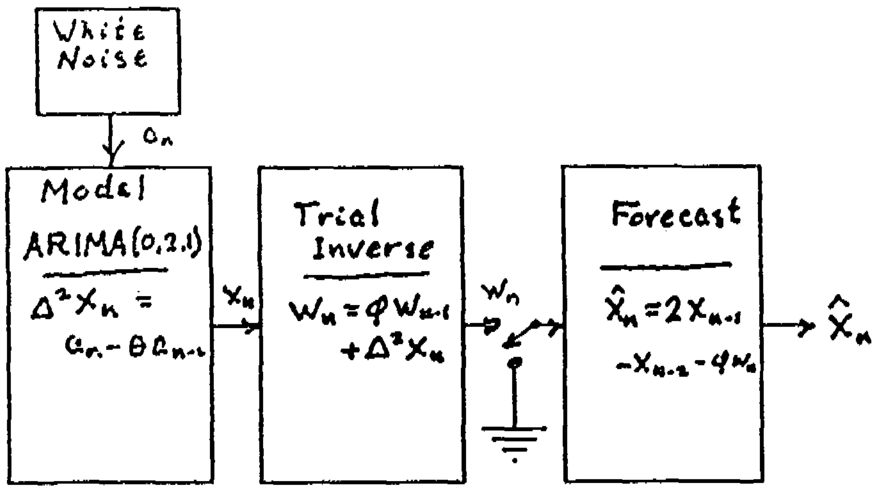


FIG. 4 ARIMA FORECAST SCHEME

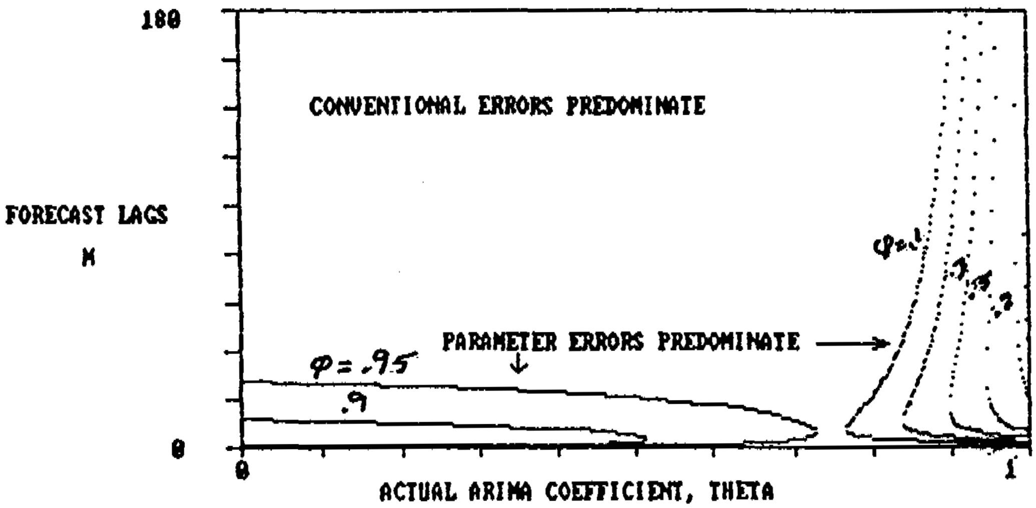


Fig. 5 Predominate Errors for ARIMA(0,2,2)

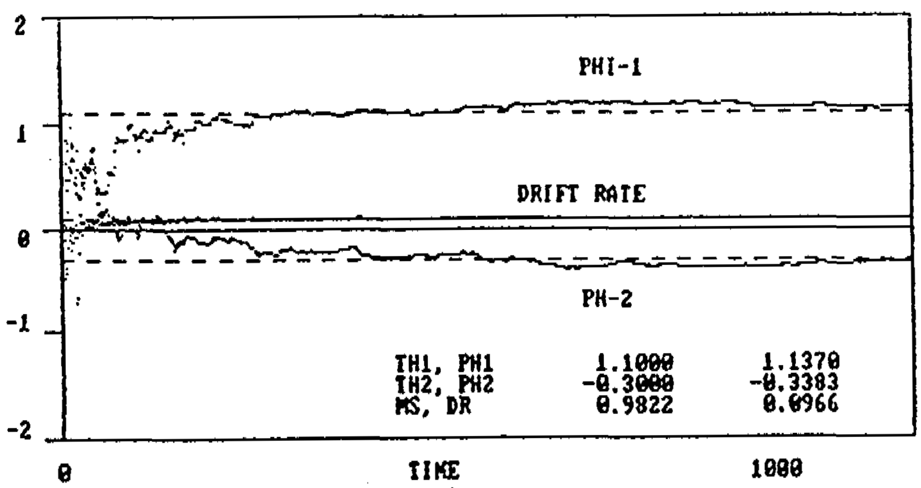
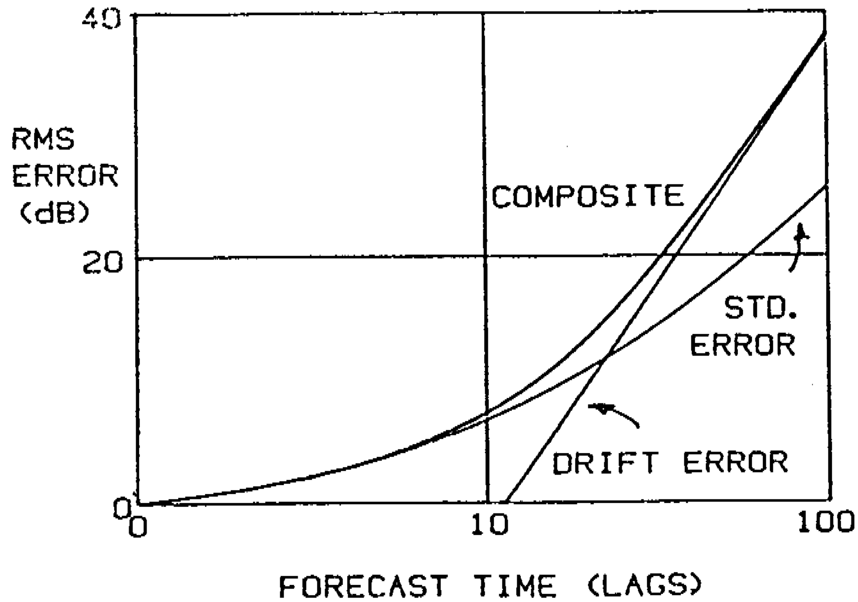


Fig. 6 Transient response for ARIMA(0,2,2)

FIG. 7 FORECAST TIME ERRORS FOR AN ARIMA (0, 2, 2)



FORECAST ERRORS FOR AN ARIMA(0,2,2)

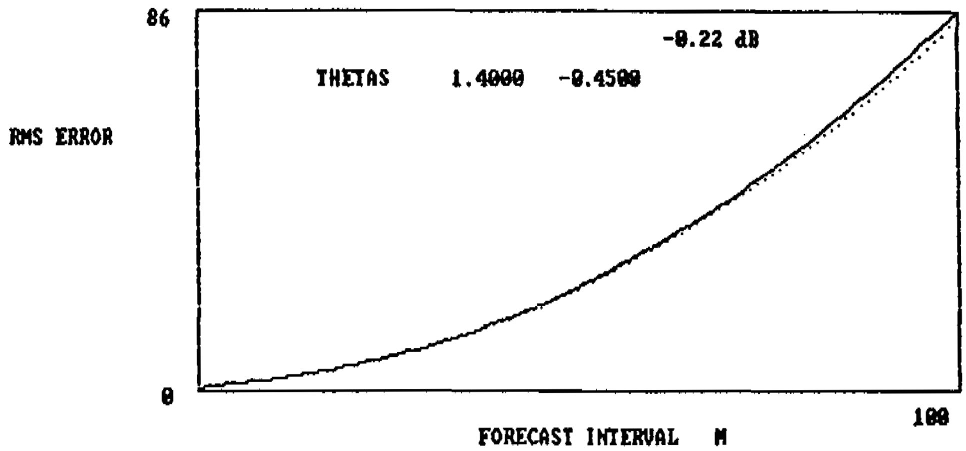


Fig. 8a Errors Using Exact Parameters

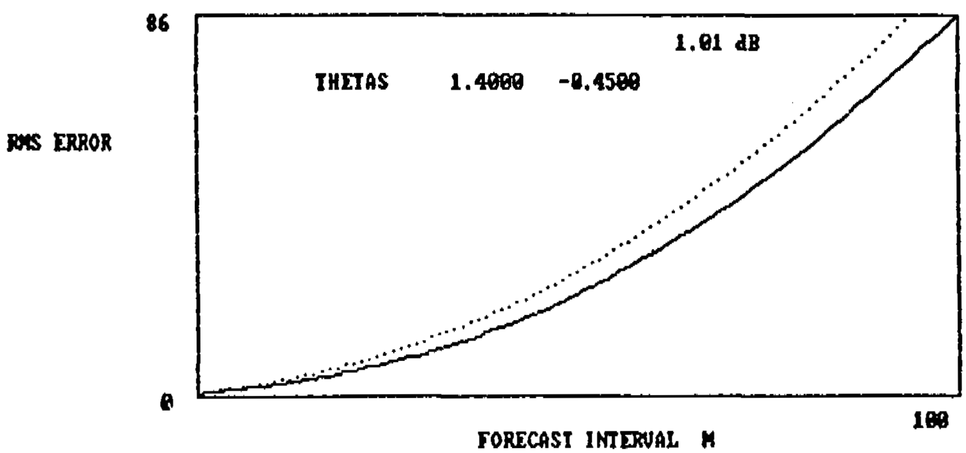


Fig. 8B Errors Using Box/Jenkins Estimation

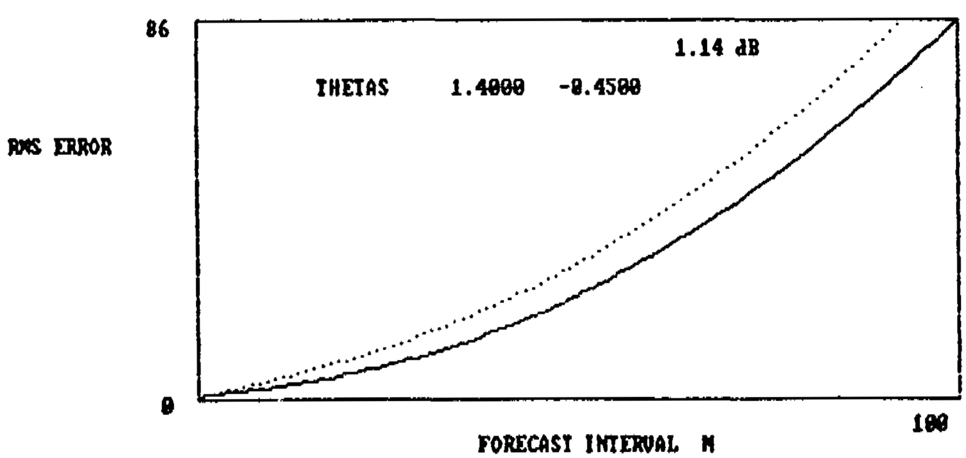


Fig 8C Errors Using Adaptive Estimation

QUESTIONS AND ANSWERS

DR. GERNOT WINKLER, USNO: Since you can do your parameter estimation on the run, you can, of course, allow a change in the characteristics of your frequency standards with time. Its an adaptive method.

DR. BARNES: That is true. It will adapt to the new value if the standard changes. I guess that I should point out on the last plot, I am assuming that I have 100 points of data and I am forecasting for the next 100 points. It is not a lot of data that I am working with. We are still seeing agreements to a fraction of a dB in this approach.