

An algorithm for estimating time-varying commodity price models

Boris I. Godoy, Graham C. Goodwin, Juan C. Agüero and Alejandro J. Rojas

Abstract— Given the current financial crisis, there is renewed interest in modelling how the price of commodities change in the market. Traditionally, such models have assumed constant parameters. However, large and sudden changes in the parameters can also be anticipated due to market shocks. This paper is aimed at addressing this issue. We first describe a bias-variance trade-off in parameter estimation when sudden changes are considered. We then propose a mechanism to achieve a compromise between the observed bias and variance. A key ingredient of this mechanism is to use an estimator having a variable memory length.

I. INTRODUCTION

Many companies are interested in predicting (or, at least, understanding) how the price of commodities change in the market. For example, in mining, knowledge (or understanding) of this phenomena is important since it underpins key decisions such as opening or closing of mine sites, annual extraction targets for the metal of interest, and others. Consequently, there is substantial interest in the modelling of commodity prices.

A standard model used for the price of commodities is that of [1], in which the parameters are assumed to be fixed. Hence, standard estimation algorithms can be used to obtain the value of the parameters [2], [3], [4]. This works well for certain systems. Also, the assumption of constant parameters is the basis for many key concepts in system identification including the analysis of asymptotic properties [3, chap.9] and convergence results in adaptive control, see for example [5]. However, in practice, the commodity price market can suffer shocks which will lead to a change in the parameters in the model, and in these cases, some form of on-line estimation is desirable [5, chap.7]. If the parameter changes are slow, then there exist strategies aimed at this case, e.g. modelling the parameters as stationary stochastic process or as (slowing varying) random walks. The Kalman Filter can then be used to design the associated estimator [6]. This approach leads to a fixed trade-off between sensitivity to parameter variations, on the one hand, and sensitivity to noise, on the other.

There are many other related ideas in the literature to track time-varying parameters. For example, the use of an exponential data forgetting factor. This effectively fixes the memory length of the estimator [7, pp.151]. However, it has the disadvantage that if the parameter is not changing, it may lead to numerical problems in the covariance matrix [7,

pp.154]. Another idea is to hypothesize a hidden Markov model for jump parameters. The parameters of the Markov model can then be estimated [8, chap.2] e.g. by using the Expectation Maximization algorithm [9], [10], [11], [12]. Other approaches can be seen in [13], [14]. There also exists a substantial literature on asymptotic properties of change detection methods [15], [16].

A key point to note, in the context of the current paper, is that each of the above procedures is based on the explicit, or implicit, assumption that the parameter variations are precisely known in a probabilistic sense. However, in practice the nature of parameter changes can be highly non-stationary i.e. there may exist periods where the parameters change frequently and there may be periods in which the parameters remain essentially constant. To deal with this kind of situation one should ideally have a variable memory length in the estimator, see e.g. [17].

The current paper combines two ideas: (i) detection of step like parameter changes in commodity price models, and (ii) a mechanism to estimate parameters when these step changes occur. The present paper differs from [2] in that we now consider the detection of infrequent step like changes in the parameters. The present paper also extends our earlier work in [17] by formalizing the bias-variance trade-off argument and propose a mechanism that selects the estimator's memory length as a compromise between bias and variance.

The main contributions of the current paper are:

- (i) We use a variable estimator memory length to achieve smooth transitions between the present estimate and the following one. The algorithm sometimes chooses a shorter memory length for the estimator (to deal with rapidly changing parameters) and, at other times, a long memory length for the estimator (to give greater immunity to noise).
- (ii) This idea is combined with a hypothesis testing procedure see e.g. [16, chap.6] to estimate when a change in the parameters has occurred. In addition, via simulation studies using commodity price models described in [1], we show that using a variable memory length estimator gives better performance than is achieved with fixed memory estimators when there are step like changes in the parameters.

The remainder of the paper is organized as follows: In Section II, we present an analysis of bias-variance trade-offs using the Least Squares (LS) estimator. In Section III, we introduce a mechanism that achieves a bias-variance compromise. In Section IV, we analyze the detection in the change of parameter. In Section V, we show a simple example. In Section VI, we describe a typical commodity

B.I. Godoy (corresponding author), G.C. Goodwin, J.C. Agüero and A.J. Rojas are with the ARC Centre of Excellence for Complex Dynamic Systems and Control, The University of Newcastle, Australia. Emails: boris.godoy, juan.aguero@newcastle.edu.au; graham.goodwin, alejandro.rojas@newcastle.edu.au

price model for the spot price. Simulation results follow in Section VII and we conclude with final remarks on the current work in Section VIII.

II. ANALYSIS OF BIAS-VARIANCE TRADE-OFF

In the present section we study bias-variance trade-offs for a simple system so as to motivate the subsequent development. The system is subject to infrequent step like changes in a parameter vector θ . The LS estimator is used to illustrate the ideas.

We notice that the study of the trade-offs in variance and bias can also be related to a deficient structure of the model. Here, we focus on the bias induced by a parameter change.

For the sake of discussion, we consider θ to be a vector $\theta = [\theta_0, \dots, \theta_{m-1}]^T$ and the available data $y_k \in \mathbb{R}$ defined as a function of a measured deterministic exogenous signal x_k as follows:

$$y_k = \varphi(x_k)^T \theta + w_k, \quad (1)$$

where $\varphi(x_k)^T$ is the regressor vector and where w_k is zero mean Gaussian white noise and variance $Q = \sigma^2$. Least Squares leads to the following estimate of θ :

$$\hat{\theta}_N = \left\{ \frac{1}{N} \sum_{k=1}^N [\varphi(x_k) \varphi(x_k)^T] \right\}^{-1} \frac{1}{N} \sum_{k=1}^N \varphi(x_k) y_k, \quad (2)$$

where N is the data length. This estimate is known to be unbiased and efficient in the case when θ is constant and w_k is Gaussian white noise. Here, however, we are interested in cases where θ is time varying. One frequently used model for time variations is that of a random walk i.e.

$$\theta_k = \theta_{k-1} + v_k, \quad (3)$$

where v_k is white noise of variance S .

It is then well known (see e.g. [4, chap.7]) that the estimation problem can be solved by using a Kalman filter:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + J_k [y_k - \varphi(x_k)^T \hat{\theta}_k]. \quad (4)$$

In steady state J_k will converge to a fixed value depending on the ratio of S and Q [18, pp.703–704]. The resulting steady state estimator has a memory determined by $J_{ss} = \lim_{k \rightarrow \infty} J_k$.

Here we wish to examine a different scenario in which θ_k makes infrequent step like changes. These could be approximately modelled as in (3) with a small variance for v_k . However, this generically leads to a filter having long memory. We will argue below that a better approach is to explicitly take into account the trade-off between bias and variance in the parameter estimates. We then aim to achieve a compromise, in terms of memory length, between the two.

To develop the idea, we assume that the system is described by

$$y_k = \begin{cases} \varphi_k^T \theta_a + w_k, & \text{if } k < k_0 \\ \varphi_k^T \theta_b + w_k, & \text{if } k \geq k_0 + 1. \end{cases} \quad (5)$$

thus the parameter θ takes two values during the period $[0, N]$, namely θ_a and θ_b . Before continuing, we define the following terms

$$\begin{aligned} R &= \frac{1}{N} \sum_{k=1}^N \varphi_k \varphi_k^T, \\ R_0 &= \frac{1}{k_0} \sum_{k=1}^{k_0} \varphi_k \varphi_k^T, \quad \epsilon = \frac{1}{N} \sum_{k=1}^N \varphi_k w_k \\ R_a &= \frac{1}{N} \sum_{k=1}^{k_0} \varphi_k \varphi_k^T, \quad R_b = \frac{1}{N} \sum_{k=k_0+1}^N \varphi_k \varphi_k^T. \end{aligned} \quad (6)$$

Hence, using (2), we can write

$$\hat{\theta}_N = R^{-1} R_a \theta_a + R^{-1} R_b \theta_b + R^{-1} \epsilon. \quad (7)$$

Taking the expected value in (7) we obtain

$$\bar{\theta} = E\{\hat{\theta}_N\} = R^{-1} R_a \theta_a + R^{-1} R_b \theta_b. \quad (8)$$

If we define $\Lambda = R^{-1} R_a$, we can write (8) as

$$\bar{\theta} = \Lambda \theta_a + (I - \Lambda) \theta_b, \quad (9)$$

where Λ and I are matrices with appropriate dimensions (the latter represents the identity matrix). We then have the following result:

Lemma 1: The covariance of the estimate resulting from the LS estimation problem for the model (5) is given by $\text{cov}\{\hat{\theta}_N\} = \frac{\sigma^2 K_N}{N}$, where $K_N = R^{-1}$.

Proof: The covariance of $\hat{\theta}_N$ can be obtained as

$$\begin{aligned} \text{cov}\{\hat{\theta}_N\} &= E\{(\hat{\theta}_N - \bar{\theta})(\hat{\theta}_N - \bar{\theta})^T\} \\ &= E\{(R^{-1} \epsilon)(R^{-1} \epsilon)^T\} \\ &= \frac{R^{-1}}{N} \frac{1}{N} \sum_k \sum_j \varphi_k \varphi_j^T E\{w_k w_j^T\} R^{-1}, \end{aligned} \quad (10)$$

Finally, using ideas from [19], we can re-write (10) as

$$\text{cov}\{\hat{\theta}_N\} = \frac{\sigma^2 R^{-1}}{N} = \frac{\sigma^2 K_N}{N} \quad (11)$$

with $K_N = R^{-1}$. ■

Lemma 2: The expected value of the squared distance between the estimate $\hat{\theta}_N$ and the new value θ_b for the parameter vector is given by:

$$\begin{aligned} E\{(\hat{\theta}_N - \theta_b)^T (\hat{\theta}_N - \theta_b)\} &= \\ &= \left(1 - \frac{N_b}{N}\right)^2 \Delta^T M^T M \Delta + \frac{\sigma^2 \text{tr}\{K_N\}}{N}, \end{aligned} \quad (12)$$

where $\Delta = \theta_b - \theta_a$, $M = R^{-1} R_0$, and $N_b = N - k_0$.

Proof:

$$\begin{aligned} E\{(\hat{\theta}_N - \theta_b)^T (\hat{\theta}_N - \theta_b)\} &= E\{[(\hat{\theta}_N - \bar{\theta})^T + (\bar{\theta} - \theta_b)^T] \\ & \quad [(\hat{\theta}_N - \bar{\theta}) + (\bar{\theta} - \theta_b)]\} \\ &= E\{(\hat{\theta}_N - \bar{\theta})^T (\hat{\theta}_N - \bar{\theta})\} + (\bar{\theta} - \theta_b)^T (\bar{\theta} - \theta_b) \\ &= \frac{\sigma^2 \text{tr}\{K_N\}}{N} + (\Delta)^T \Lambda^T \Lambda (\Delta), \end{aligned} \quad (13)$$

Next we observe that the term Λ can be written as

$$\begin{aligned} \Lambda &= R^{-1}R_a = \frac{k_0}{N} \left(\frac{1}{N} \sum_{k=1}^N \varphi_k \varphi_k^T \right)^{-1} \left(\frac{1}{k_0} \sum_{k=1}^{k_0} \varphi_k \varphi_k^T \right) \\ &= \frac{k_0}{N} R^{-1}R_0 = \frac{k_0}{N} M = \left(1 - \frac{N_b}{N} \right) M, \end{aligned} \quad (14)$$

where N_b represents the number of samples that have been collected since the true parameter changed to θ_b . Substituting (14) into (13), we obtain the expression given in (12). ■

Remark 1: In the subsequent analysis we utilize the fact that

$$K_N = \left(\frac{1}{N} \sum_{k=1}^N \varphi_k \varphi_k^T \right)^{-1} \rightarrow K^*, \quad (15)$$

for N sufficiently large. $\nabla \nabla \nabla$

Lemma 3: Under the conditions given in Lemma 1 and 2 and using (15), we see that the estimator memory length, N^o , which gives the best trade-off between bias and variances is:

$$N^o(N_b) = \begin{cases} g\left(N^-, N^+, \frac{2N_b^2 \Delta^T M^T M \Delta}{2N_b \Delta^T M^T M \Delta - \sigma^2 \text{tr}\{K^*\}}\right) & \text{for } N_b > \frac{\sigma^2 \text{tr}\{K^*\}}{2\Delta^T M^T M \Delta} \\ \infty & \text{for } N_b \leq \frac{\sigma^2 \text{tr}\{K^*\}}{2\Delta^T M^T M \Delta} \end{cases} \quad (16)$$

where the function $g(x, y, z)$ denotes the nearest value of $\{x, y\}$ to z . The terms N^- and N^+ denote the immediately inferior integer value of N^* and the immediate superior integer value of N^* , respectively, where

$$N^* = \frac{2N_b^2 \Delta^T M^T M \Delta}{2N_b \Delta^T M^T M \Delta - \sigma^2 \text{tr}\{K^*\}}, \quad \text{for } N_b > \frac{\sigma^2 \text{tr}\{K^*\}}{2\Delta^T M^T M \Delta}. \quad (17)$$

Proof: Equation 17 follows by differentiating (12) with respect to N and setting the result equal to zero. Also, since the functional is quadratic in N^{-1} the critical point is the global minimum. Finally, we obtain the closest integer that minimizes the cost function. ■

Remark 2: The key conclusion from Lemma 3 is that, in order to achieve the best trade-off between bias and variance, one needs to use an estimator memory length which is a function of N_b (i.e. how many samples that have been collected since the parameter change occurred) the size of the change in parameter, Δ , and a term that depends on the noise level, $\sigma^2 \text{tr}\{K^*\}$. $\nabla \nabla \nabla$

Remark 3: We observe that Equation (17) can be rewritten as

$$N^* = \frac{N_b}{1 - \frac{\sigma^2 \text{tr}\{K^*\}}{2N_b \Delta^T M^T M \Delta}}, \quad \text{for } N_b > \frac{\sigma^2 \text{tr}\{K^*\}}{2\Delta^T M^T M \Delta}. \quad (18)$$

We see from Equation (18) that the optimal estimator memory length N^* is always greater than N_b . This means that one should always use data containing some information prior to the change. $\nabla \nabla \nabla$

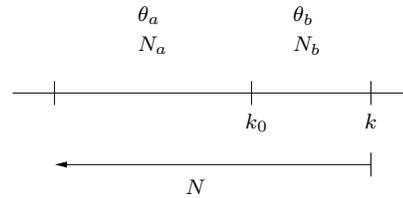


Fig. 1. Illustration of the overall time framework for Section III.

III. A MECHANISM TO ACHIEVE AN OPTIMAL BIAS-VARIANCE TRADE-OFF

In Section II we have shown that there is an optimal estimator memory length N^o that achieves the best trade-off between bias and variance when there is a step change in the parameters. We have also shown that N^o is neither N , the complete data set, nor N_b . In the present section, we suggest a mechanism to implement the choice of memory length given in (16). Notice that the main difficulty in calculating N^* (and thus N^o) is that one has to estimate (i) the time the change occurs, k_0 , and (ii) the size of change, Δ .

We assume that we have a good estimate, $\hat{\theta}_a$, for θ_a , see Figure 1, and that hypothesis tests (such as, for example, a χ^2 test) are run on a set of estimators with different memory length, refer to Figure 2.

If an hypothesis test for a particular estimator reports a parameter change, as suggested in Figure 2, we then have that the associated estimator memory length is a rough estimate for N_b . If multiple hypothesis tests report a parameter change, we choose N_b as the minimal memory length between all estimators which detect a change. Notice that once an estimate for N_b is known, then an estimate for k_0 is available since $k_0 = k - N_b$.

Next, we need to obtain a rough estimate for Δ . We can always say that equation (7) applies for any N , in particular, for N_j . Hence, from (7) we have that

$$E\{\hat{\theta}_{N_j}\} - \theta_a = (I - \Lambda)(\theta_b - \theta_a). \quad (19)$$

Therefore, from (19) we have

$$\Delta = (I - \Lambda)^{-1}[E\{\hat{\theta}_{N_j}\} - \theta_a], \quad (20)$$

and from (20) it follows that

$$\begin{aligned} \Delta &= (I - \Lambda)^{-1}[E\{\hat{\theta}_{N_j}\} - \theta_a] \\ &\approx (I - \Lambda)^{-1}[\hat{\theta}_{N_j} - \theta_a] \\ &= \left(I - \frac{N_a}{N_j} M \right)^{-1} [\hat{\theta}_{N_j} - \theta_a]. \end{aligned}$$

A crude estimate of Δ is then given by

$$\hat{\Delta} \approx \frac{N_j}{N_b} [\hat{\theta}_{N_j} - \theta_a]. \quad (21)$$

Utilizing the rough estimates of N_b and $\hat{\Delta}$ found as described above, we can then obtain N^* from (18) and thus N^o from (16). The current parameter estimate is finally chosen to be $\hat{\theta}_o$, the linear regression estimate with memory length N^o . This process is repeated iteratively, updating $k = k + 1$ and

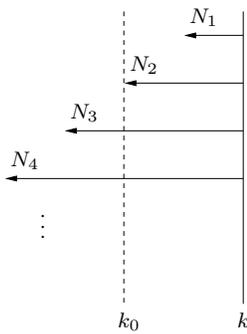


Fig. 2. Illustration of the set of available memory lengths involved in the estimate of N_b .

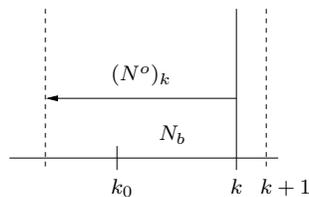


Fig. 3. Illustration of the proposed mechanism update.

$(N_b)_{k+1} = (N_b)_k + 1$ (notice that N_b is now explicitly a function of time). In pseudo-code, the algorithm can be written as follows:

- Run an hypothesis test for a set of estimators having different memory lengths.
- If an hypothesis test reports a parameter change then assign N_b to be that particular estimator memory length (if more than one hypothesis test reports a parameter change, choose the minimum between the different estimator memory lengths that report the change).
- Update $\hat{\Delta}$ by using (21) with $N_j = (N^o)_{k-1}$.
- Calculate $(N^o)_k$ by using the rough estimate for N_b and $\hat{\Delta}$.
- Select the current estimate to be $(\hat{\theta}_o)_k$, the linear regression estimate with memory length $(N^o)_k$.
- Update time $k = k + 1$, set $(N_b)_{k+1} = (N_b)_k + 1$ and recalculate the optimal memory length, see Figure 3.

Remark 4: As time evolves, and provided that no further change is detected, we see from (18) that $(N^o)_k \rightarrow N_b$.

▽▽▽

Remark 5: Notice that the above proposed mechanism is by no means unique. For example other choice could be used to determine the best bias-variance trade-off.

▽▽▽

IV. PARAMETER CHANGE DETECTION

A key aspect in the general strategy proposed in this paper is to know when a change in the parameters has occurred. There are many options, but to be definite we use the following hypothesis test:

$$\begin{aligned} H_0 : \Delta = 0, \text{ that is, the parameters are unchanged,} \\ H_1 : \Delta \neq 0, \text{ that is, the parameters have changed,} \end{aligned} \quad (22)$$

where $\Delta = (\hat{\theta}_s^b)_k - (\hat{\theta}_s^a)_k$. For a particular estimator of memory length $N_s < N$, we calculate Δ considering the

following data:

$$\text{data : } \overbrace{y_1, y_2, \dots, y_{k-N_s}}^{(\hat{\theta}_s^a)_k}, \overbrace{y_{k-N_s+1}, \dots, y_k}^{(\hat{\theta}_s^b)_k}.$$

Many tests are available in the literature, e.g. Wald test, Likelihood Ratio test and others (see e.g. [20]). However, for this specific problem where the loss function $(V_s)_k$ for each iteration k is defined as in (23), a suitable test is a $\chi^2(m)$ test [16, pp.206-207], where m is the degrees of freedom (corresponding to the dimension of the parameter vector $m = \dim(\theta)$). We test the following function:

$$(V_s)_k = \left((\hat{\theta}_s^b)_k - (\hat{\theta}_s^a)_k \right)^T P^{-1} \left((\hat{\theta}_s^b)_k - (\hat{\theta}_s^a)_k \right), \quad (23)$$

where $P = \text{cov}[(\hat{\theta}_s^b)_k]$.

A standard table can be used to design a threshold h_α to test

$$(V_s)_k \leq h_\alpha. \quad (24)$$

The hypotheses test is applied to each parallel estimator having different memory lengths. The threshold is problem specific, depending on how much sensitivity we want to have in the 'parameter change detector'. In summary, the hypothesis test can be implemented as follows

- Select from a χ^2 table the desired probability.
- Select the number of degrees of freedom, m .
- Compare the value of $(V_s)_k$ with the one obtained from a χ^2 table with α probability of false alarm, h_α .
- If $(V_s)_k > h_\alpha$ then the null hypothesis H_0 can be rejected. On the other hand, if $(V_s)_k \leq h_\alpha$ the null hypothesis H_0 is accepted.

V. A SIMPLE EXAMPLE

We investigate the application of our proposed algorithm to the system given in (1), with $\theta_a = [1 \ 0.2]^T$, $\theta_b = [1 \ 0.3]^T$, and then another change from θ_b back to θ_a , and covariance for the noise $Q = 0.1$. We change the parameter at $k_0 = 1000$, and then at $k_0 = 2000$. We choose to have two windows to detect changes, namely $N_1 = 100$ and $N_2 = 500$. We arbitrarily initialize the algorithm for $N^* = 500$. Using the χ^2 test, we detect a change at time $k_c = 1075$, and $k_c = 2100$. These changes are flagged by the shortest memory length estimator. From Figure 4 (upper plot) we observe that our algorithm achieves a trade-off between noise suppression and bias in the estimation of the parameter. We compare this estimation to the one obtained by adding one data point each time step (sequential algorithm). The estimation of Δ and N^* over time is also given in the lower plots.

VI. APPLICATION TO COMMODITY PRICE MODELLING

We next consider the continuous-time two-factor model proposed in [1]. In this model, the logarithm of the spot price is separated into two components

$$\log S_t = \xi_t + \chi_t. \quad (25)$$

The first component ξ_t (short-term component) represents the difference between the spot and the equilibrium price. The

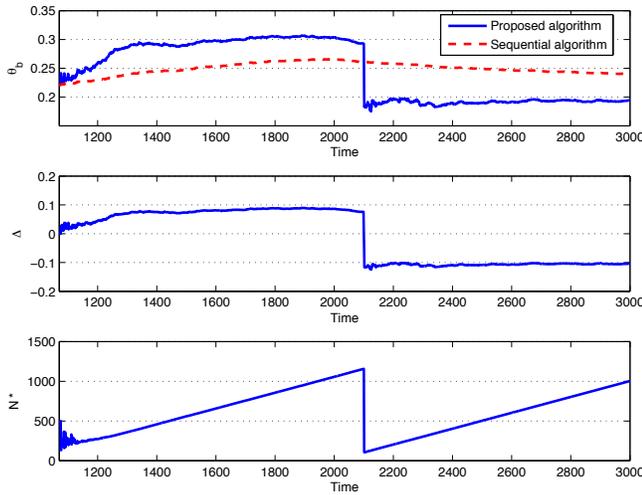


Fig. 4. Estimation of the second component of θ_b using the proposed algorithm compared to sequential algorithm (upper plot), and the estimation of Δ and N^* in the two lower plots. Change detection at $k_c = 1050$ and $k_e = 2100$.

second component χ_t (long-term component) is a geometric Brownian motion that models the behaviour of the equilibrium price. Hence, the dynamic models associated with the different components can be expressed as follows:

$$\begin{aligned} d\chi_t &= -\kappa\chi_t dt + \sigma_\chi dz_\chi, \\ d\xi_t &= \mu_\xi dt + \sigma_\xi dz_\xi, \end{aligned} \quad (26)$$

where dz_χ, dz_ξ are correlated increments of Brownian motion and $d\chi d\xi = \rho_{\chi\xi} dt$. The parameters in the model are

- (i) κ —rate of mean reversion
- (ii) μ_ξ —trend
- (iii) σ_ξ —short term volatility
- (iv) σ_χ —long term volatility

In the current paper we will work with spot prices only. Then, using spectral factorization the model can be reduced to the form [2]

$$A(q)\Delta y_k = B(q)u_k + C(q)e_k, \quad (27)$$

where e_k is white Gaussian white noise and

$$\begin{aligned} A(q) &= 1 - \alpha q^{-1}, & B(q) &= (1 - \alpha)\beta \\ C(q) &= 1 + (k_1 + k_2 - \alpha - 1)q^{-1} + (k_1 + \alpha + k_2\alpha)q^{-2}, \end{aligned}$$

with $\alpha = \exp(-\kappa h)$, $\beta = \mu_\xi h$, and h the sampling time. The values for k_1 and k_2 correspond to the steady-state Kalman Filter gains. For more details see [2].

Remark 6: We note that the model in (27) is not precisely of the form proposed in (1) because the latter considers past values of y_k . However, the extension of the analysis proposed in Section II for the model used in (1) can be done straightforward to a more general case. $\nabla\nabla\nabla$

The identification of the parameters in the model for commodity prices can be carried out using the three-step approach proposed in [2]. However, since we are not including future contract prices in the analysis in the current work,

Parameters	Value
κ	13.8087
μ_ξ	-1.25%
h	1/48
σ_ξ	14.5%
σ_χ	28.6%
$\rho_{\chi\xi}$	0.3
σ_v	0.5

TABLE I

VALUE FOR THE PARAMETERS IN THE SCHWARTZ-SMITH MODEL USED TO GENERATE THE OIL SIMULATED DATA. THIS CORRESPONDS ROUGHLY TO THE OIL DATA IN [1].

the model estimate is based on the first step described in [2], which uses information about the spot price only. This first step is briefly describe as follows:

The model using the spot price only is converted into an input-output representation to find initial values for κ and μ_ξ . These initial conditions are found using the Prediction Error Method (PEM) applied to the ARMAX model given in (27).

A detailed explanation of all this step can be found in [2].

VII. APPLICATION OF THE SUGGESTED ALGORITHM TO SIMULATED DATA

Here, we apply the idea developed in the previous sections. We use a variable length estimator and simulated commodity price data. To illustrate the ideas, we begin by choosing estimator memory lengths of 150, 250, 350 samples (weeks) for the parameter change detection stage. This selection is, of course, problem specific. In particular, we note that the minimum data necessary to obtain reasonably parameter estimates is given, in this case, by 50 samples.

In the present example, the results obtained using an estimator with memory length $(N^o)_k$ (as explained in Section III) is compared against the estimate obtained by adding new available data, but without discarding any of the older data (which we call 'sequential algorithm').

We use the model in (27) with the nominal parameter values given in Table I. We simulate the system for approximately 28 years (1500 weeks). We use the values of μ_ξ given in Table I for the first 20 years (1000 weeks). We then change this parameter to the new value $\mu_\xi = 2.50\%$. The true value of μ_ξ is shown in Figure 5 by the red-dotted line. Note that data is collected for a further 10 years (500 weeks) after the parameter change occurs. Also shown in Figure 5 is the estimate provided by the sequential algorithm. Note that 2 years (i.e. 100 weeks) after the change, the sequential estimator is still essentially giving the old (and now incorrect) value of μ_ξ . Moreover, after a further 10 years (i.e. 500 weeks) the estimator has only changed by about half of the true parameter step change.

This suggests that the sequential algorithm would give incorrect (and misleading) information if used, say, for asset valuation.

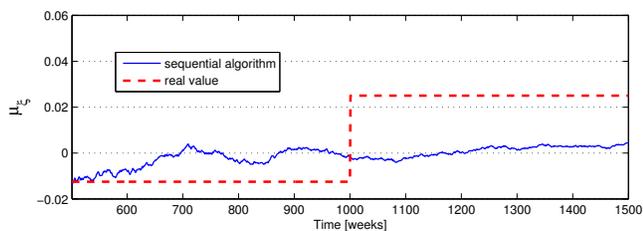


Fig. 5. Change in the parameter μ_ξ .

Figure 6 (upper plot) shows the results obtained using the estimator with memory length $(N^o)_k$. It can be seen from Figure 6 (upper plot) that this estimator responds quicker to the change and also has good noise immunity. In Figure 6 (upper plot), we notice that there is a delay when responding to the change. This delay is due to the change detector algorithm, which takes about 200 [weeks] to respond. Notice that in Figure 6 (lower plot), we observe that the estimation of Δ is zero most of the time. This is because the change has not been detected yet. Comparing the results in Figure 6 (upper plot) with those in Figure 5, we see the merits of using an estimator with memory length $(N^o)_k$.

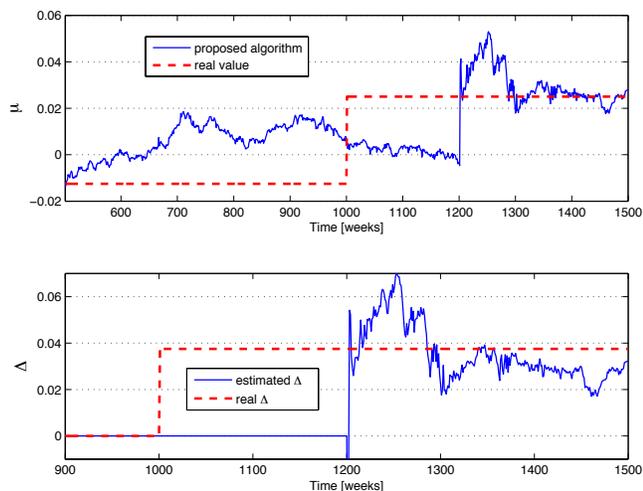


Fig. 6. Estimation of μ_ξ (upper plot) and Δ (lower plot) using the algorithm proposed in Section III.

VIII. CONCLUSION

In this paper we have studied the issue of bias-variance trade-offs associated with parameter estimation when parameters undergo infrequent step changes. We have shown that the bias-variance trade-off can be dealt with by using parallel estimators having different memory length.

This study has been motivated by commodity price models and the reasonable assumption that they undergo parameter changes in response to market shocks. Within this context, we have shown the impact of such changes on parameter estimation for the Schwartz-Smith commodity price model.

We have observed that, whilst sequential estimation schemes will eventually respond to parameter changes, the

response time can be slow. This means that predictions (or valuations) based on the current parameter estimates may be in error. As an alternative, we have proposed an algorithm having a variable memory length. We have shown that this algorithm is capable of forming a better trade-off between parameter change detection and immunity to noise. Predictions (or valuations) provided by this kind of scheme are believed to be more reliable than those obtained by standard sequential parameter estimation schemes.

Our next step in this research will be to apply these ideas to multiple time series, to account for possible coupling between the value of different commodities (such as for example copper and gold).

REFERENCES

- [1] E. Schwartz and J.E. Smith. Short-term variations and long-term dynamics in commodity prices. *Manag. Science*, 46(7):893–911, July 2000.
- [2] T. Perez, G.C. Goodwin, and B.I. Godoy. Parameter Estimation of Structural Commodity price models. In *15th IFAC Symposium on System Identification SYSID*, Saint-Malo, France, 2009.
- [3] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.
- [4] G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, 1977.
- [5] G.C. Goodwin and K.S. Sin. *Adaptive Filtering Prediction and Control*. Prentice-Hall, 1984.
- [6] D.Q. Mayne. Optimal Non-Stationary Estimation of the Parameters of a Linear System with Gaussian Inputs. *Journal of Electronics and Control*, 14:107–112, 1963.
- [7] P.E. Wellstead and M.B. Zarrop. *Self-Tuning Systems: Control and Signal Processing*. Wiley&Sons, 1991.
- [8] R.J. Elliott, L. Aggoun, and J.B. Moore. *Hidden Markov Models: Estimation and Control*. Springer, 2008.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Statistical Society, Series B*, 39(1):1–38, 1977.
- [10] J.D. Hamilton. Analysis of time series subject to changes in regime. *Journal of Econometrics*, 45(1–2):39–70, July–August 1990.
- [11] J.D. Hamilton. *Time Series Analysis*. Princeton Univ. Press, 1994.
- [12] C.-J. Kim and C.R. Nelson. *State-Space Models with Regime Switching*. The MIT Press, 1999.
- [13] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Trans. on Pattern Analysis and Mach. Intell.*, 27(12):1–15, 2005.
- [14] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: a tutorial. *European Journal of Control*, 13(2–3):242–260, 2007.
- [15] M. Basseville and A. Benveniste. *Detection of Abrupt Changes in Signals and Dynamic Systems*. Springer-Verlag, 1986.
- [16] Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. Wiley& Sons, 2000.
- [17] A.J. Rojas, G.C. Goodwin, and C.P. Renton. Short or Long Memory Estimators? In *15th IFAC Symposium on System Identification SYSID*, Saint-Malo, France, 2009.
- [18] G.C. Goodwin, S.F. Graebe, and M.E. Salgado. *Control System Design*. Prentice Hall, 2001.
- [19] X. Bombois, B.D.O. Anderson, and G. Scroletti. Open loop vs Closed-loop identification of Box-Jenkins systems in a least costly identification context. In *European Control Conference*, Kos, Greece, 2007.
- [20] Larry Wasserman. *All of Statistics. A concise course in Statistical Inference*. Springer, 2004.