

A study of Harmonics Estimation for Unbalanced Power Systems

Sudipta Chakraborty

Dept. of Electrical Engineering, NIT-Rourkela, India
Email: Chakraborty.sudipta02@gmail.com

Abstract— Harmonics has been present for a long time and its presence shapes the performance of a power system. Therefore, estimation of harmonics is of paramount importance while analysing a power system network. Following the inception of harmonics, various filters have been devised to achieve an optimal control strategy for harmonic alleviation. This paper introduces various algorithms to analyse harmonics in the power system. The objective is to estimate the power system voltage magnitude in the presence distortions taking into account the noise by employing different estimation approaches. We have focused our attention towards the study of Least Mean Squares (LMS) based filter, Recursive Least squares (RLS) based filter, Kalman filter (KF) and Extended Kalman (EKF) filter. For a test signal LMS, RLS, KF and EKF based algorithms have been analysed and results have been compared. The proposed estimation approaches are tested for only static signals.

Index Terms— Harmonics Estimation, LMS, RLS, Kalman Filter, Extended Kalman Filter.

1. INTRODUCTION

Harmonics became a buzzword in the early 1980s, making many people reassessed the effectiveness of their building's wiring system. Yet, many still view the concept as a relatively new occurrence. However, harmonics have been there since well before the early '80s. The associated problems existed in the electrical system way back when transistor tubes were first used in the 1930s. Aside from grounding, many consider harmonics as one of the biggest concerns for the power quality industry today. In this chapter, we'll discourse the fundamentals of harmonics and the problems it can cause in a power system. We define harmonics as voltages or currents at frequencies that are multiples of the fundamental frequency. In most systems, the fundamental frequency is 50 Hz. Therefore, harmonic order is 100 Hz, 150 Hz, and 200 Hz and so on. We usually specify these orders by their harmonic number or multiple of the fundamental

frequency. For example, a harmonic with a frequency of 150 Hz is known as the third harmonic ($50 \times 3 = 150$). In this case, for each cycle of the fundamental waveform, there are three complete cycles of the harmonic waveforms. The even multiples of the fundamental frequency are called as even-order harmonics while the odd multiples are called as the odd-order harmonics [1].

2. LMS METHOD

The least-mean-square (LMS) algorithm is a linear adaptive filtering algorithm that consists of two basic processes 1. A filtering process which involves (a) computing the output of transversal filter produced by a set of tap inputs, and (b) generating an estimation error by comparing this output to desired response. 2. An adaptive process which involves the automatic adjustment of the tap weights of the filter in accordance with the estimation error.[4] Thus, the combination of these two processes working together constitutes a feedback loop around the LMS algorithm. First we have a transversal filter around which LMS algorithm is built: this component is responsible for performing the filtering process. Second we have a mechanism for performing adaptive control process on the tap weights of the transversal filter, hence the designation "adaptive weight-control mechanism". The tap inputs $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ form the elements of M -by-1 tap input vector $u(n)$, $M-1$ is the number of delay elements; these tap inputs span multidimensional space denoted by U_n . Correspondingly, the tap weights $\hat{w}_0(n)$, $\hat{w}_1(n)$, ..., $\hat{w}_{M-1}(n)$ form the elements of M -by-1 tap weight vector $\hat{w}(n)$. During filtering process the desired response is supplied for processing, alongside the tap input vector $u(n)$. Given this input the transversal filter produces an output $\hat{d}(n/U_n)$ used an estimate of the desired response $d(n)$. We also define the estimation error as the difference between $e(n)$ as the difference between the desired response and actual filter output. The estimation error $e(n)$ and the tap-input vector are applied to the control mechanism, the feedback loop around the tap weights is thereby closed. A scalar version of

inner product of estimation error and tap input $u(n-k)$ is computed for $k = 1, 2, 3, \dots, M-2, M-1$. The result defines the correction $\delta \hat{w}(n)$ applied to weight $\hat{w}(n)$ at $n+1$ iteration. The scaling factor used here is denoted by μ . It is called the step-size parameter. The LMS algorithm uses the product $u(n-k) e^*(k)$ as an estimate of element k in the gradient vector $\nabla J(n)$ that characterizes the method of steepest descent. Accordingly the computation of each tap weight in the LMS algorithm suffers from gradient noise. The LMS algorithm involves feedback in its operation, which therefore raises the related issue of stability. In this context, a meaningful criterion is to require that $J(n) \rightarrow J(\infty)$ as $n \rightarrow \infty$ where $J(n)$ is the mean-squared error produced by the LMS algorithm at time n and its final value $J(\infty)$ is a constant. For LMS algorithm to satisfy this criterion, the step-size parameter μ has to satisfy certain conditions related to the Eigen structure of the correlation matrix of the tap inputs. To develop an estimate of the gradient vector $\nabla J(n)$, the strategy is to substitute the estimates of correlation matrix R and the cross correlation vector p

$$\nabla j(n) = -2p + 2Rw(n) \quad (1.1)$$

The simplest choice of estimators for R and p is to use instantaneous estimates that are based on sample values of the tap-input vector and desired response, as defined by, respectively

$$\hat{R}(n) = u(n)u^H(n) \quad (1.2)$$

$$\hat{p}(n) = u(n)d^*(n) \quad (1.3)$$

Correspondingly, the instantaneous estimate of the gradient vector is

$$\hat{\nabla j}(n) = -2u(n)d^*(n) + 2u(n)u^H(n) \hat{w}(n) \quad (1.4)$$

This estimate is biased because the tap weight estimator $\hat{w}(n)$ is a random vector that depends upon tap-input vector $u(n)$. Substituting the estimate for the gradient vector $\nabla j(n)$ in the steepest descent algorithm, we get the recursive relation for updating tap-weight vector

$$\hat{w}(n+1) = \hat{w}(n) + \mu u(n)[d^*(n) - u^H(n)\hat{w}(n)] \quad (1.5)$$

The result can be written in the form of three basic relations

$$\text{Filter output } y(n) = \hat{w}^H(n) u(n) \quad (1.6)$$

$$\text{Estimation error } e(n) = y(n) - d(n) \quad (1.7)$$

$$\text{Tap weight adaptation } \hat{w}(n+1) = \hat{w}(n) + \mu u(n) e^*(n) \quad (1.8)$$

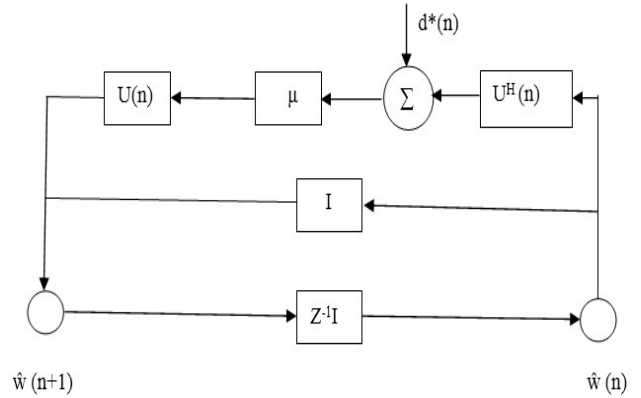


Figure 1. Signal flow graph of LMS Algorithm

The estimation vector error $e(n)$ is based on the current estimate of the tap-weight vector, $\hat{w}(n)$. The second term $\mu u(n)e^*(n)$ on the right hand-side of (1.8) represents the correction that is applied to the current estimate of the tap-weight vector, $\hat{w}(n)$. The iterative procedure is started with an initial guess $\hat{w}(0)$. The algorithm described by (1.6) to (1.8) is the complex form of adaptive least-mean-square (LMS) algorithm. At each iteration or time update, it also requires the knowledge of most recent values: $u(n)$, $d(n)$ and $\hat{w}(n)$. The LMS algorithm is the member of family of *stochastic-gradient-algorithms*. When the LMS algorithm operates on stochastic inputs, the allowed set of directions along which we “step” from one iteration cycle to the next is random and cannot therefore be thought of as being true gradient directions. The LMS algorithm requires only $2M + 1$ complex multiplications and $2M$ complex additions per iteration, where M is the number of tap weights used in the adaptive transversal filter. In other words, the computational complexity of the LMS algorithm is $O(M)$.

3. RLS METHOD

An important feature of recursive least squares (RLS) algorithm is that it utilizes the information contained in the input data, extending back to the instant of time when the algorithm is initiated.[4] The resulting rate of convergence is therefore typically an order of magnitude faster than the simple LMS algorithm. This improvement in performance is achieved at the expense of large increase in computational complexity. In the recursive implementations of the method of least squares, we start the computation with known initial conditions and use the information contained in new data samples to update the old estimates. So it is found that the length of the observable data is

variable. Accordingly, we express cost function to be minimized as $\hat{\delta}(n)$, where n is the length of the variable data. Also, we are introducing weighting factor into the definition of cost function $\hat{\delta}(n)$. We thus write

$$\mathcal{G}(n) = \sum_{i=1}^n \beta(n, i) |e(i)|^2 \quad (1.9)$$

where $e(i)$ is the difference between the desired response $d(i)$ and the output $y(i)$ produced by a transversal filter whose tap inputs (at time i) equal $u(i), u(i-1), \dots, u(i-M+1)$. That is, $e(i)$ is defined by

$$e(i) = d(i) - y(i) = d(i) - w^H(n)u(i) \quad (1.10)$$

Where $u(n)$ is the tap input vector at time n , is defined by

$$u(n) = [u(i), u(i-1), \dots, u(i-M+1)]^T \quad (1.11)$$

Where $w(n)$ is the tap weight vector at time n , is defined by

$$w(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)] \quad (1.12)$$

The tap weights of the transversal filter remain fixed during the observation interval $1 \leq I \leq n$ for which the cost function $\mathcal{G}(n)$ defined.

The weighting factor (n, i) in (1.9) has the property that $0 < \beta(n, i) \leq 1$

$$(1.13)$$

Where $i = 1, 2, \dots, n$.

A special form of weighting that is commonly used is the exponential weighting factor or forgetting factor defined by

$$\beta(n, i) = \lambda^{n-i} \quad i = 1, 2, \dots, n. \quad (1.14)$$

λ is a positive constant with value close to, but less than 1. When λ is 1, we have the ordinary method of least squares. The inverse of $1 - \lambda$ is a measure of the memory of the algorithm. The $\lambda = 1$ case, corresponds to infinite memory. Thus in the method of exponentially weighted least squares, we minimize the cost function

$$\mathcal{G}(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (1.15)$$

The optimum value of the tap weight vector, $\hat{w}(n)$ for which the cost function attains its minimum value is defined by the normal equations written in matrix form

$$\phi(n) \hat{w}(n) = z(n) \quad (1.16)$$

The M -by- M correlation matrix $\phi(n)$ is now defined by

$$\mathcal{G}(n) = \sum_{i=1}^n \lambda^{n-i} u(i) u^H(i) \quad (1.17)$$

The M -by-1 cross-correlation vector $z(n)$ between the tap inputs of the transversal filter and the desired response is defined by

$$z(n) = \sum_{i=1}^n \lambda^{n-i} u(i) d^*(i) \quad (1.18)$$

Isolating the term corresponding to $i = n$ from the rest of the summation on the right hand side of (1.17), we may write

$$\phi(n) = \lambda \left[\sum_{i=1}^{n-1} \lambda^{n-1-i} u(i) u^H(i) \right] + u(n) u^H(n) \quad (1.19)$$

The recursion for updating the value of correlation matrix of the tap inputs

$$\phi(n) = \lambda \phi(n-1) + u(n) u^H(n) \quad (1.20)$$

Where $\phi(n-1)$ the old value of correlation matrix, and the matrix product $u(n) u^H(n)$ plays the role of “correction” term in the updating operation. We may use (1.18) to derive the following recursion for updating the cross-correlation vector between the tap inputs and the desired response

$$z(n) = \lambda z(n-1) + u(n) d^*(n) \quad (1.21)$$

With the correlation matrix $\phi(n)$ assumed to be positive definite and therefore non-singular, we may apply the matrix inversion lemma to the recursive equation (1.20). We first make the following identifications

$$A = \phi(n), \quad B^{-1} = \lambda \phi(n-1), \quad C = u(n), \quad D = 1$$

Applying the matrix inversion lemma, we obtain the following recursive equation for the inverse of correlation matrix

$$\begin{aligned} \phi^{-1}(n) &= \lambda^{-1} \phi^{-1}(n-1) - \\ &\frac{\lambda^{-2} \phi^{-1}(n-1) u(n) u^H(n) \phi^{-1}(n-1)}{1 + \lambda^{-1} u^H(n) \phi^{-1}(n-1) u(n)} \end{aligned} \quad (1.22)$$

For convenience of computation, let

$$P(n) = \phi^{-1}(n) \quad (1.23)$$

$$k(n) = \frac{\lambda^{-1} p(n-1) u(n)}{1 + \lambda^{-1} u^H(n) p(n-1) u(n)} \quad (1.24)$$

$$p(n) = \lambda^{-1} p(n-1) - \lambda^{-1} k(n) u^H(n) p(n-1) \quad (1.25)$$

The M -by- M matrix $P(n)$ is referred to as the inversion correlation matrix. We have

$$k(n) = [\lambda^{-1} p(n-1) - \lambda^{-1} k(n) u^H(n) p(n-1)] u(n) \quad (1.26)$$

So, we get $k(n) = P(n) u(n)$

$$k(n) = \phi^{-1}(n) u(n) \quad (1.28)$$

The gain vector $k(n)$ is defined as the tap input vector $u(n)$ transformed by the inverse of the correlation matrix $\phi(n)$. To develop recursive equation for developing the least squares estimate $\hat{w}(n)$ for the tap weight vector we use equation (1.21), (1.23) and (1.24) to express the least squares

estimates $\hat{w}(n)$ for the tap weight vector at iteration at n as follows

$$\begin{aligned}\hat{w}(n) &= \phi^{-1}(n) z(n) \\ &= \lambda P(n)z(n-1) + P(n)u(n) d^*(n) \\ &= P(n) z(n)\end{aligned}\quad (1.29)$$

Substituting (1.25) for $P(n)$ in the first term only in the right-hand side of (1.29) we get

$$\begin{aligned}\hat{w}(n) &= \hat{w}(n-1) - k(n)u^H(n)\hat{w}(n-1) \\ &+ P(n)u(n)d^*(n)\end{aligned}\quad (1.30)$$

Using $P(n)u(n)$ equals the gain factor $k(n)$, we get the desired recursive equation for updating the tap weight vector

$$\begin{aligned}\hat{w}(n) &= \hat{w}(n-1) + k(n)[d^*(n) - u^H(n)\hat{w}(n-1)] \\ \hat{w}(n) &= \hat{w}(n-1) + k(n)\xi^*(n)\end{aligned}\quad (1.31)$$

Where $\xi^*(n)$ is the priori estimation error defined by

$$\begin{aligned}\xi(n) &= d(n) - u^T(n)\hat{w}^*(n-1) \\ &= d(n) - \hat{w}^H(n-1)u(n)\end{aligned}\quad (1.32)$$

$\hat{w}^H(n-1)u(n)$ Represents the estimate of the desired response $d(n)$, based on the old least squares estimate of the tap weight vector that is made at time $n-1$. The *a priori* estimation error $\xi(n)$ is different from the posteriori estimation error

$$e(n) = d(n) - \hat{w}^H(n)u(n)\quad (1.33)$$

4. KALMAN FILTER

The important feature of the Kalman filtering is the recursive processing of the noise measurement data. In power system applications, Kalman filter is used to estimate voltage and frequency variations. The Kalman filtering has also been used for dynamic estimation of voltage and current phasors [1]. This filtering technique is used to obtain the optimal estimate of the power system voltage magnitudes at different harmonic levels. The Kalman filter is an estimator which is used to

Estimate the state of a linear dynamic system influenced by Gaussian White noise, using measurement that are linear functions of the system state, but corrupted by additive Gaussian white noise. The Kalman Filter allows to estimate the state of dynamic systems with certain types of random behaviour by using these statistical information. The Kalman filter deals with the general problem of trying to estimate the state of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$X_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}\quad (1.34)$$

With a measurement $z \in R^m$ that is

$$Z_k = Hx_k + v_k\quad (1.35)$$

The random variables w_k and v_k represent the process and measurement noise and are assumed to be independent of each other. They are white noise with normal probability distributions.

$$p(w) \sim N(0, R)\quad (1.36)$$

$$p(v) \sim N(0, Q)\quad (1.37)$$

With each time step or measurement the process noise covariance R and measurement noise covariance Q matrix may change. But, here we are assuming they are constant. The matrix $n \times n$ matrix A in the difference equation and the $n \times 1$ matrix B refer to the state at the previous time step $k-1$ to the state at the current step k . Here, both A and H are assumed to be constant.

We define \hat{x}_k^- as our *a priori* state estimate at step k given knowledge of the process prior to step k , and \hat{x}_k as our *a posteriori* state estimate at step k given measurement z_k . Then, we can write *a priori* and *a posteriori* estimate errors as

$$e_k^- = x_k - \hat{x}_k^- \text{ And } e_k = x_k - \hat{x}_k$$

Then, *a priori* estimate error covariance is

$$P_k^- = E[e_k^- e_k^{-T}]\quad (1.38)$$

And the *a posteriori* estimate error covariance is

$$P_k = E[e_k e_k^T]\quad (1.39)$$

Our goal is to find an equation that computes an *a posteriori* state estimate \hat{x}_k as a linear combination of an *a priori* estimate \hat{x}_k^- and a weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k^-$.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)\quad (1.40)$$

The difference $(z_k - H\hat{x}_k^-)$ is called the measurement innovation, or the residual. The residual reflects the inconsistency between the predicted the measurement $H\hat{x}_k^-$ and the actual measurement z_k . If the residual is zero then, the two are in complete concurrence. The $n \times m$ matrix K in (2.40) is the gain or blending factor that minimizes the *a posteriori* error covariance (2.39). This minimization can be achieved by first substituting (2.40) into the above definition for e_k , substituting it into (2.39), performing the mentioned expectations, taking the derivative of the trace of the result w.r.t K , adjusting that result equal to zero, and then solving for K .

The Kalman gain calculated that minimizes P_k is given by

$$K_k = \frac{p_k^- H^T}{R + H p_k^- H^T} \quad (1.41)$$

We can observe that as the measurement error covariance approaches zero, the gain K weights the residual more heavily. Specifically

$$\lim_{R_k \rightarrow 0} K_k = H^{-1}$$

On the other hand, as the a priori estimate error covariance approaches zero, the gain K weights the residual less heavily. Specifically

$$\lim_{p_k^- \rightarrow 0} K_k = 0$$

So, we can see that as the measurement error covariance R approaches zero, the actual measurement z_k is trusted more and more, while the predicted measurement $H\hat{x}_k^-$ is trusted less and less.

On the other hand, as the a priori estimate error covariance p_k^- approaches zero the actual measurement z_k is trusted less and less, while the predicted measurement $H\hat{x}_k^-$ is trusted more and more. The Kalman filter incorporates a form of feedback control by estimating the process state at some time and then obtaining the feedback in the form of noisy measurements. The Kalman filter equations can be divided into two groups: *time update* equations and *measurement update* equations. The time update equations are accountable for extrapolating forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are accountable for incorporating a new measurement into the a priori estimate to obtain an improved *a posteriori* estimate. The time update equations can also be considered as *predictor* equations, while the measurement update equations can be considered as *corrector* equations. So, the final estimation algorithm acts as a predictor-corrector algorithm for solving various numerical problems [3].

The time update equations are

$$\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_{k-1} \quad (1.42)$$

$$p_k^- = Ap_{k-1}A^T + Q \quad (1.43)$$

The measurement update equations are

$$K_k = p_k^- H^T (Hp_k^- H^T + R)^{-1} \quad (1.44)$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (1.45)$$

$$p_k = (I - K_k H) p_k^- \quad (1.46)$$

The first step in the measurement update is to compute the Kalman gain, K_k . The next step is to measure the process to obtain z_k , and then to generate an *a posteriori* state estimate by

incorporating the measurement as in (2.45). The final step is to obtain an a posteriori error covariance estimate using (2.46). After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to predict the new a priori estimates.

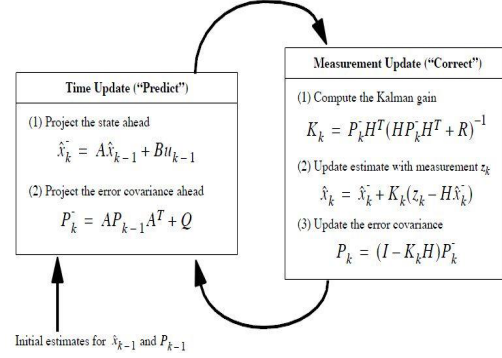


Figure 2. Kalman filter prediction Estimation Cycle

5. EXTENDED KALMAN FILTER

The Kalman Filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete time-controlled process that is governed by a *linear* stochastic difference equation. But some of the most interesting and successful applications of Kalman Filtering have been the ones when the process to be estimated and the measurement relationship to the process is non-linear. A Kalman Filter that linearizes about the current mean and covariance is known as an Extended Kalman Filter or EKF. In something similar to the Taylor Series, we can linearize the estimation around the current estimate using the partial derivatives of the process and the measurement functions to compute the estimates even in the case of non-linear relationships. This is done by modifying some of the material presented in the Kalman filtering algorithm. Let us consider that the process has a state vector $x \in R^n$ but that process is now governed by the non-linear stochastic difference equation [3].

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (1.47)$$

With a measurement $z \in R^m$ that is

$$z_k = h(x_k, v_k) \quad (1.48)$$

Where the random variables w_k and v_k again represent the process and measurement noise as in (1.36) and (1.37). In this case the non-linear function f in the difference equation (1.47) relates the state at the previous time step $k-1$ to the current time step k . It includes parameters as a driving

function u_{k-1} and the zero mean process noise w_k . The *non-linear* function h in the measurement equation (1.48) refers to the state x_k to the measurement z_k .

Actually, one does not know the individual values of the noise w_k and v_k at each time step. However, one can approximate the state and measurement vector without considering them

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (1.49)$$

$$\tilde{z}_k = f(\tilde{x}_k, 0) \quad (1.50)$$

Where \tilde{x}_k is some *a posteriori* estimate of the state (from previous time step k). One of the fundamental flaws of EKF is that the distributions (or densities in the continuous case) of the random variables remain no longer normal after undergoing respective non-linear transformations. The EKF is simply a specific state estimator that only approximates the optimality of Bayes' Rule by linearization.

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that linearize an estimate about (1.49) and (1.50)

$$x_k = \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1} \quad (1.51)$$

$$z_k = \tilde{z}_k + H(x_k - \hat{x}_k) + Vv_k \quad (1.52)$$

Where

x_k And z_k are the actual state and measurement vectors,

\tilde{x}_k And \tilde{z}_k are the approximate state and measurement vectors form

\hat{x}_k Is an *a posteriori* estimate of the state at step k

The random variables w_k and v_k represent the process and measurement noise and A is the Jacobian matrix of partial derivatives of f with respect to x , that is

$$A_{[i,j]} = \frac{\delta f_{[i]}}{\delta x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad (1.53)$$

W is the Jacobian matrix of partial derivatives of f with respect to w , that is ,

$$W_{[i,j]} = \frac{\delta f_{[i]}}{\delta w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0)$$

H is the Jacobian matrix of partial derivatives of h with respect to x

$$H_{[i,j]} = \frac{\delta h_{[i]}}{\delta x_{[j]}}(\tilde{x}_k, 0)$$

V is the Jacobian Matrix of partial derivatives of h with respect to v

$$V_{[i,j]} = \frac{\delta h_{[i]}}{\delta v_{[j]}}(\tilde{x}_k, 0)$$

It is to be noted that for simplicity in the notation we don't use the time step subscript k with the Jacobians A , W , H and V even though they are different in fact at each time step.

Now we define a new notation for the prediction error,

$$\tilde{e}_{xk} = x_k - \tilde{x}_k \quad (1.54)$$

$$\tilde{e}_{zk} = z_k - \tilde{z}_k \quad (1.55)$$

In practice, one does not have access to xk in (1.53), it is the actual state vector, i.e., the quantity one is trying to estimate. On the other hand, one does have access to zk in (1.54), it is the actual measurement that one is trying to estimate x . Using (1.53) and (1.54) we can write the governing equations for an error process as

$$\begin{aligned} \tilde{e}_{xk} &= A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \\ \tilde{e}_{zk} &= H\tilde{e}_{xk} + \eta_k \end{aligned} \quad (1.56)$$

Where ε_k and η_k represent new independent random variables with zero mean and covariance matrices WQW^T and VRV^T with Q and R as in (1.36) and (1.37) respectively.

It is to be noted that the equations (1.55) and (1.56) are linear, and that they closely represent the difference and measurement equations (1.34) and (1.35) from the Kalman Filter. This motivates us to use the actual measurement residual \tilde{e}_{zk} in (1.54) and a second (hypothetical) Kalman Filter to estimate the prediction error \tilde{e}_{xk} . This estimate, call it \hat{e}_k , could then be used along with (1.53) to obtain a posteriori state estimates for the original non-linear process as

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \quad (1.57)$$

The random variables of (1.55) and (1.56) have approximately the following probability distributions

$$p(\tilde{e}_{xk}) \approx N(0, E[\tilde{e}_{xk} \tilde{e}_{xk}^T])$$

$$p(\varepsilon_k) \approx N(0, W, Q_k, W^T)$$

$$p(\eta_k) \approx N(0, VR_k V^T)$$

Given these approximations and letting the predicted value of \hat{e}_k to be zero, the Kalman filter equation used to estimate \hat{e}_k is

$$\hat{e}_k = K_k \tilde{e}_{zk} \quad (1.58)$$

By substituting (1.58) back into (1.57) and making use of (1.54) we see that we do not actually need a second (hypothetical) Kalman Filter

$$\hat{x}_k = \tilde{x}_k + K_k \tilde{e}_{zk} = \tilde{x}_k + K_k (z_k - \tilde{z}_k) \quad (1.59)$$

Equation (1.59) can now be used for the measurement update in the Extended Kalman Filter,

with \tilde{x}_k and \tilde{z}_k coming from (1.49) and (1.50) and the Kalman gain K_k coming from (1.44) with the appropriate substitution for the measurement error covariance.

A. EKF TIME UPDATE EQUATION

$$\hat{x}_k^- = f(\hat{x}_{k-1}^-, u_{k-1}, 0) \quad (1.60)$$

$$P_k^- = A_{k-1} P_{k-1}^- A_{k-1}^T + W_k Q_{k-1} W_k^T \quad (1.61)$$

This basic discrete Kalman Filter the time update equations project the state and covariance estimates from the previous time step $k-1$ to the current time step k . Also, f in (1.60) comes from (1.49), A_k and W_k are the process Jacobians at the step k , and Q_k is the process noise covariance (1.36) at step k

B. EKF MEASUREMENT UPDATE EQUATION

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (1.62)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (1.63)$$

$$P_k = (I - K_k H_k) P_k^- \quad (1.64)$$

This measurement update equations correct the state and covariance estimates with the measurement z_k . Also, h in (1.63) comes from (1.50) H_k and V are the measurement Jacobians at step k , R_k is the measurement noise covariance (1.37) at step k . The basic operation of the EKF is same as that of the linear discrete Kalman Filter.

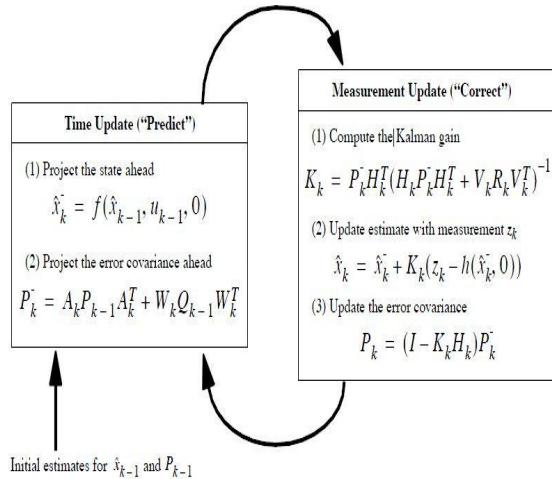


Figure 3. Operation of Extended Kalman Filter

Now, a signal containing N sinusoids has been considered as

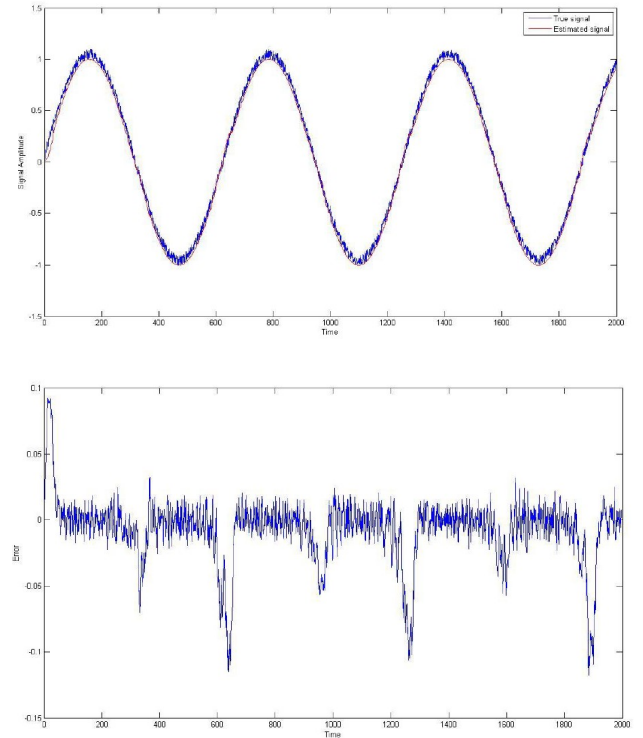
$$X = \sum_{i=1}^N A_i \sin(w_i t_k + \phi_i) + v_k$$

Where $k = 1, 2, \dots, N$

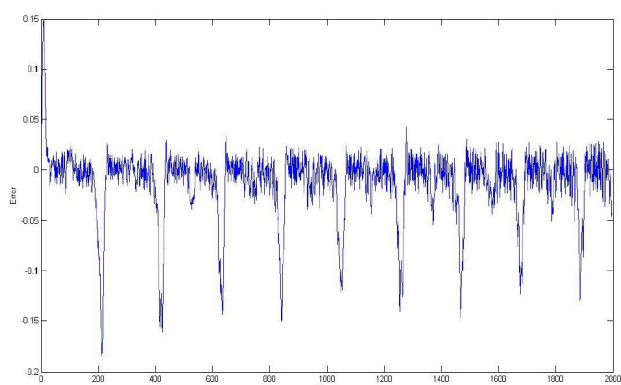
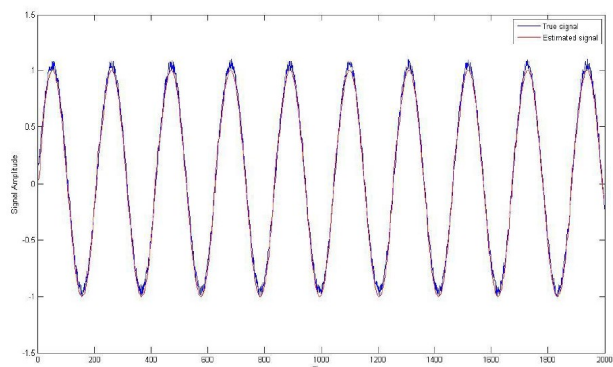
Where A_{ik} , w_i and ϕ_i is the amplitude, frequency and phase of the i^{th} sinusoid respectively. t_k is the k th sample of the sampling time and v_k is a zero mean Gaussian white noise. In this paper, we have used this signal having amplitude of 1p.u (for LMS & RLS) and 20p.u (for KF) and frequency of 50Hz. The process noises are generated using the random number generator with the help of Matlab command "randn". The amplitude estimation of the signal (where the estimated is the filtered one in case of LMS and RLS) has been carried out at different harmonic levels starting from fundamental to 5th harmonic signal. The simulation results have been shown in the subsequent pages which compare between the original signal and estimated signal. A static test signal corrupted with non-linearities and Gaussian noise has been used and the estimation of amplitude is done using Extended Kalman filtering algorithm.

6. SIMULATION RESULTS

A. LMS RESULTS

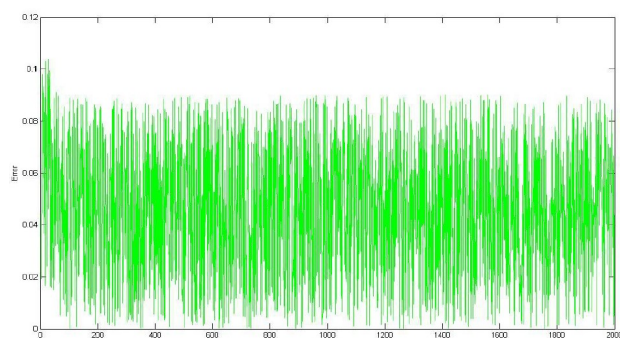
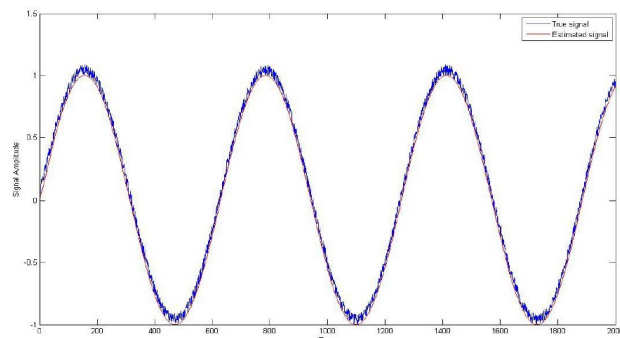


Fundamental Amplitude Estimation Using LMS

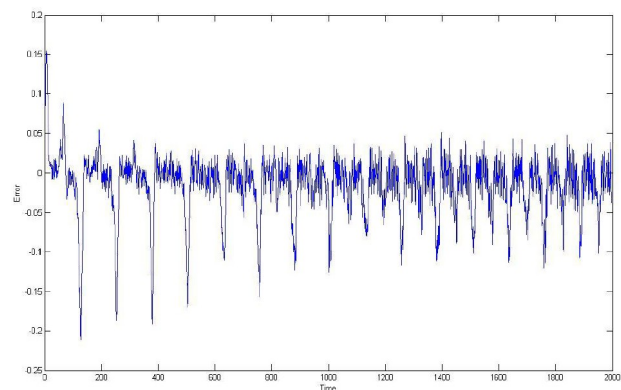
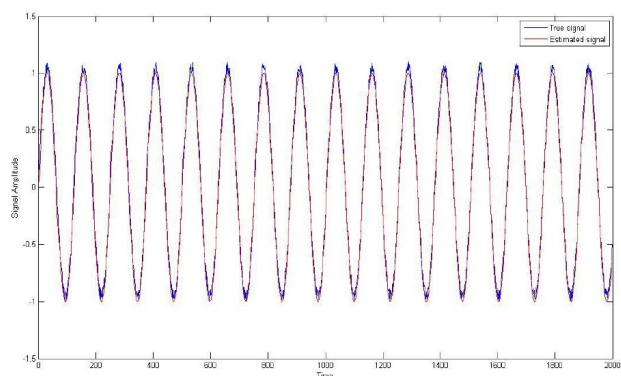


3RD Harmonic Amplitude Estimation Using LMS

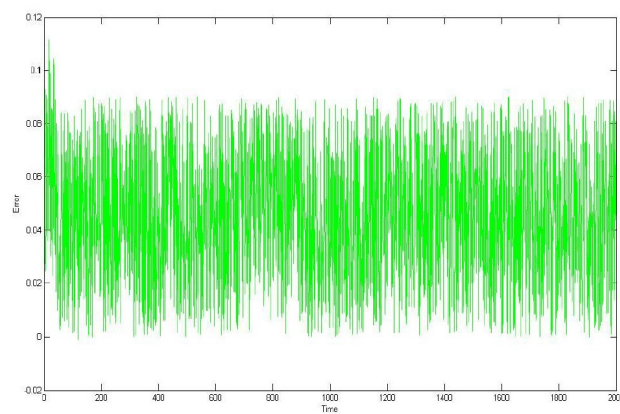
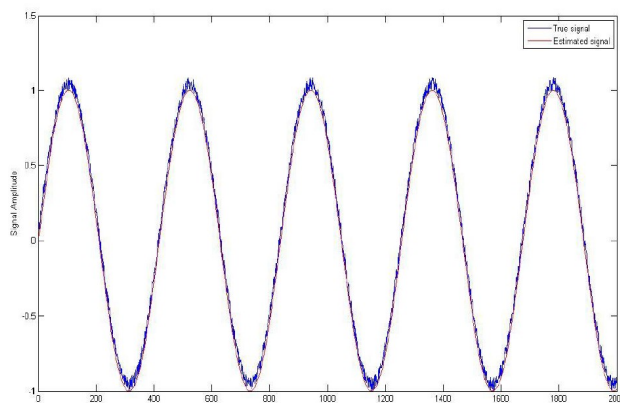
B. RLS SIMULATION RESULTS



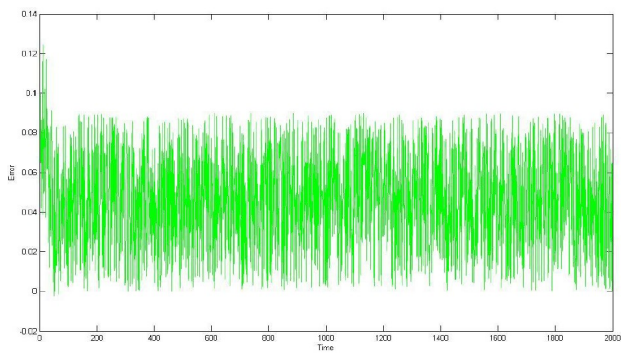
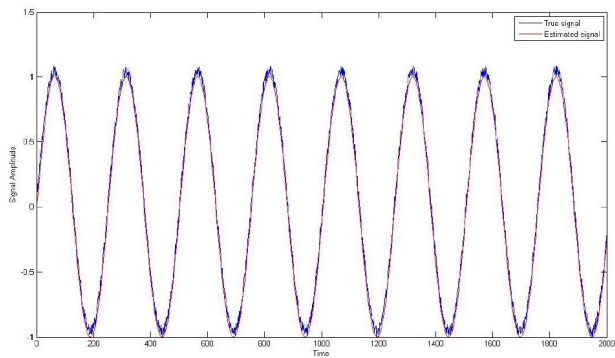
Fundamental Amplitude Estimation Using RLS



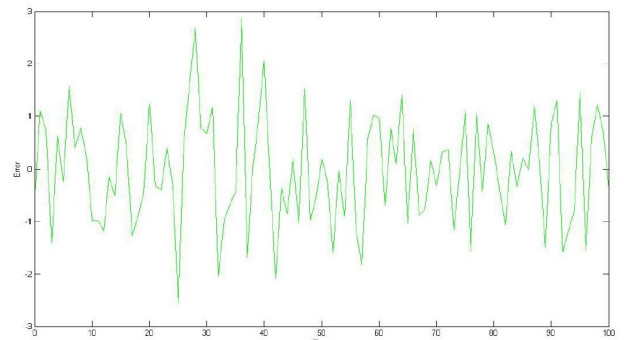
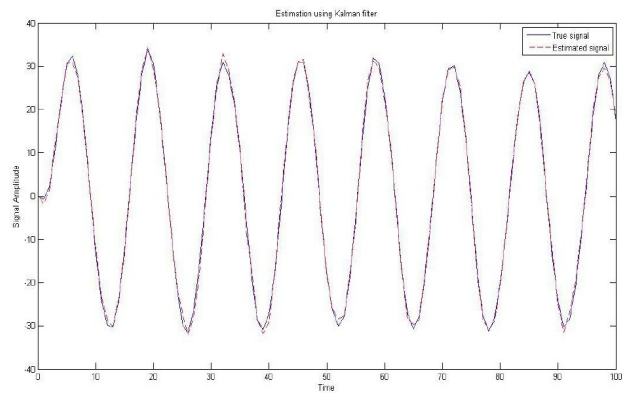
5TH Harmonic Amplitude Estimation Using LMS



3RD Harmonic Amplitude Estimation Using RLS

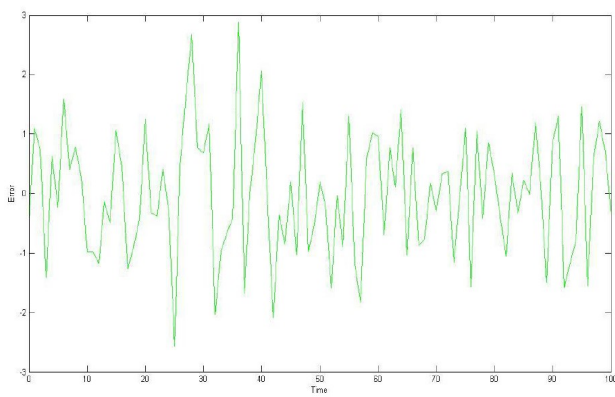
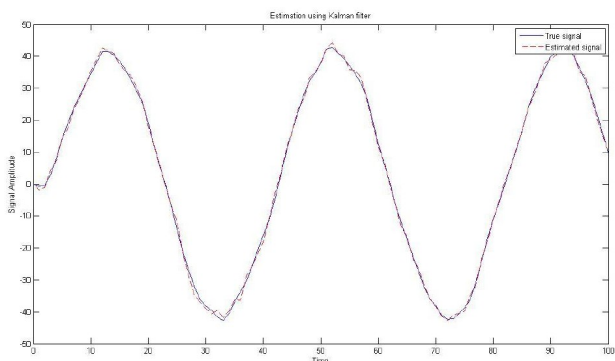


5TH Harmonic Amplitude Estimation Using RLS

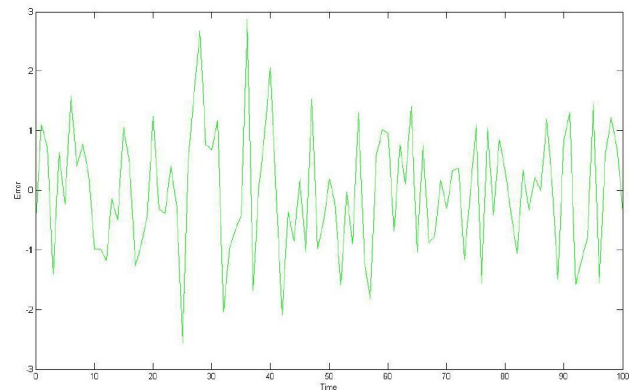
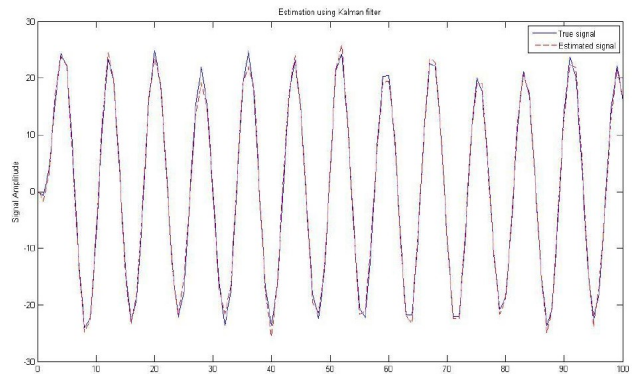


3RD Harmonic Amplitude Estimation Using Kalman Filter

C. KALMAN FILTER SIMULATION RESULTS

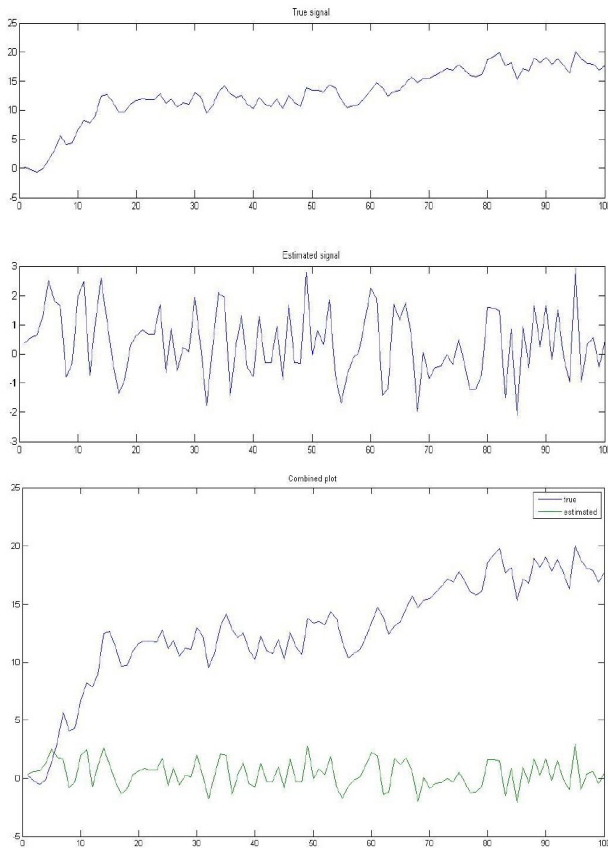


Fundamental Amplitude Estimation Using Kalman Filter

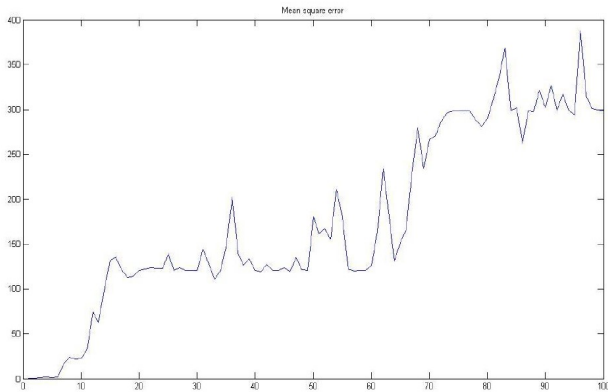


5TH Harmonic Amplitude Estimation Using Kalman Filter

D. EXTENDED KALMAN FILTER SIMULATION RESULTS



Extended Kalman Filter Output (amplitude vs. Cycles of Prediction and estimation)



Mean Square Error in EKF

7. CONCLUSION

Harmonic distortion is one of the different aspects that affect the power system efficiency. Since the harmonic content of the power circuit depends upon the load, the presence of a non-linear load and electronic converters in the system are the main cause of harmonics. Harmonics can be broadly divided into two categories: characteristic and non-characteristic. The production of non-characteristic harmonics in the circuit should be avoided as far as

the technical aspects are concerned. Characteristic harmonics are the integral multiples of the fundamental frequency and their amplitude is directly proportional to the fundamental frequency and inversely proportional to the order of the harmonic. Since it is essential to filter out those harmonics, we require an estimator to estimate the parameters of the harmonics. There are various methods for the estimation of the parameters. We have discussed about the Least Mean Squares (LMS), Recursive Least Squares (RLS), Kalman Filter(KF) and Extended Kalman Filter (EKF) algorithms in this paper. The LMS is the most commonly used algorithm used for estimation. It is a gradient descent algorithm which adjusts the adaptive filter taps changing them by a quantity proportional to the instantaneous estimate of the gradient of the error surface. The RLS algorithm performs an exact minimization of the sum of the squares of the desired signal estimation errors at each instant. The Kalman Filter is basically a recursive estimator and its algorithm is also based on the least square error. Since all the algorithms produce a noisy estimate of the filter taps, we need a low pass filter which would then process this noisy signal. The filter bandwidth of this filter should be so chosen that it compromises between eliminating the noise from the noisy estimate and preserving the original signal. This feature is only provided by the KF. The RLS algorithm is not capable of doing this since its filter bandwidth is fixed. The LMS algorithm has this feature but its quantitative values are not adequate. But one limitation of KF is that it cannot be used for non-linear systems. To work with non-linear systems we proposed the Extended Kalman Filter (EKF) .In this algorithm we need to compute the matrix of partial derivatives (Jacobians) in order to linearize the non-linear system about the current mean and co-variance. We found out that although this filter is able to estimate the parameters for non-linear systems significantly, it suffers from strategic disadvantage as the computation of Jacobians is very difficult.

ACKNOWLEDGMENT

I am very thankful to Prof Lakshi Prosad Roy for his help and Contribution. I am also thankful to Electrical Engineering Dept. to provide me Experimental Set up. At last I want to thank my parents who always support me.

REFERENCES

1. Dynamic state estimation of power system harmonics using kalman filter methodology, Husam M.

- Beides(Student member) and G.T Heydt(Senior Member), School of Electrical Engineering, Purdue University, West Lafayette, IN 47907, October 1991
2. Comparison of Estimation Techniques using Kalman Filter and Grid-based Filter for Linear and Non-linear System, Subrata Bhowmik (NIT Rourkela) and Chandrani Roy(MCKV Institute of engineering), *Proceeding of the International on Computing: Theory and Applications(ICCTA'07)*.
3. An introduction to Kalman filter, Greg Welch and Gary Bishop TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, Updated: Monday, July 24, 2006
4. Haykin Simon, Adaptive Filter Theory, Prentice Hall, 1996 (3RD Edition)
5. Michaels Ken, Bell South Corp., Fundamental of harmonics, Jun1,1999, part 1-3
6. <http://www.pge.com/includes/docs/pdfs/mybusiness/customerservice/energystatus/powerquality/harmonics.pdf>
7. Padiyar K.R, HVDC Power Transmission Systems, Technology and System Interactions, New. Age International (P) Ltd. 1990
8. Harmonic estimation in a power system using a novel hybrid Least Square- Adaline algorithm. M. Joorabian, S. S. Montazavi, A. A. Khayyami, Electrical Engineering Department, Shahid Chamran University, Ahwaz, 61355, Iran, 30TH May 2006
9. A.A. Girgis, W. B. Chang and E. B. Makram, A digital recursive measurement scheme for online tracking of harmonics, *IEEE Trans. Power Delivery* 6(July(3)) (1991), pp. 153-1160.
10. Y.N. Chang, Y.C. Hsieh and C. S. Moo, Truncation effects of FFT on estimation of dynamic harmonics on power system, *Power system technology*, 2000, *Proceedings, Power Conference 2000, International conference on*, vol.3, 4-7 December(2000), pp. 1155-1160.