The function of the former is to calculate the filter output $\hat{s}(n)$, while the function of the latter is to adjust the set of N filter coefficients $w_i(n), i = 0,1,...N-1$ (tap weights) so that the output $\hat{s}(n)$ becomes as close as possible to a desired signal $d(n)$. The filter part processes a single input sample $y(n)$ and produces a single output sample $\hat{s}(n)$ (assuming sample per sample implementation). The filter output is calculated as a linear combination of the input sequence $y(n-i), i = 0,1,...N-1$ composed of delayed samples of $y(n)$ and the filter coefficients $w(n)$, as given in the equation below:-

$$\hat{s}(n) = \sum_{i=0}^{N-1} w_i(n) y(n-i) \quad (2)$$

Expressing the set of N filter coefficients at time index n and the sequence of delayed input samples in vector notations such that $\underline{w}(n) = [w_0(n) w_1(n) ... w_{N-1}(n)]^T$ and $\underline{x}(n) = [x(n) w(n-1) ... x(n-N+1)]^T$, where $(.)^T$ is the vector transpose operator, equation 2 can be written as $\hat{s}(n) = \underline{w}(n)^T . \underline{x}(n) = \underline{x}(n)^T . \underline{w}(n)$

The transversal filter structure is a linear temporal filter that processes the temporal samples of its input signal $y(n)$ to produce the temporally and consequently spectrally modified (filtered) output $\hat{s}(n)$. In adaptive equalizer applications, regardless of the optimization method, it is usually desired to adjust the equalizer coefficients such that the filter output $\hat{s}(n)$ resembles a desired signal $d(n)$, or equivalently, the error signal $e(n)$ must be minimized [9].

2.7 Choice of step-size

Step-size parameter i.e. μ controls how far the algorithm move along the error function surface at each update step. μ certainly has to be chosen $\mu > 0$ (otherwise we would move the coefficient vector in a direction towards larger squared error). Furthermore, too large a step-size causes LMS algorithm to be unstable, i.e., the coefficients do not converge to fixed values but oscillate. Closer analysis [10] reveals, that the upper bound for μ for stable behavior of the LMS algorithm depends on the largest eigenvalue $\lambda_{\max}$ of the tap-input auto-correlation matrix R and thus on the input signal. For stable adaptation behavior the step-size has to be:

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (3)$$

2.8 Simulation parameters of algorithms

Table 2 shows the parameters used during the simulation of algorithms.

Table 2: Parameters used for algorithms simulation

Parameters	Value
Number of bits	3000
Iterations	125
Filter length	5
Step-size	0.1

2.9 LMS algorithm

LMS algorithm is an approximation of the steepest descent algorithm which uses an instantaneous estimate of the gradient vector of a cost function. The estimate of the gradient is based on sample values of the coefficient vector and an error signal. The algorithm iterates over each coefficient in the filter, moving it in the direction of the approximated gradient [11][12]. For LMS algorithm to archive a desired output, it is necessary to have a reference signal $d(n)$ representing the desired filter output. The difference between the reference signal and the actual output of the transversal filter shown in equation 4 is the error signal.

$$e(n) = d(n) - w^T(n) y(n) \quad (4)$$

where $\hat{s}(n) = w^T(n) y(n)$

The task of LMS algorithm is to find a set of filter coefficients w that minimizes the expected value of the quadratic error signal, i.e., to achieve LMS error.

$$e^2 = (d - w^T y)^2 = d^2 - 2dw^T y + w^T yy^T w \quad (5)$$

$$E[e^2] = E[d^2] - E[2dw^T y] + E[w^T yy^T w]$$
$$= E[d^2] - w^T 2E[dy] + w^T E[yy^T] w$$

The squared error e^2 is a quadratic function of the coefficient vector w, and thus has only one (global) minimum (and no other (local) minima), that theoretically could be found if correct expected values in equation 5 were known. The gradient descent approach demands that the position on the error surface according to the current coefficients should be moved into the direction of the 'steepest descent', i.e., in the direction of the negative gradient of the cost function $J = E(e^2)$ with respect to the coefficient vector.

$$-\nabla_c J = 2E[dy] - 2E[yy^T] w \quad (6)$$

where $-\nabla_c J$ is a negative gradient of cost function J

The expected values in this equation 6, the cross-correlation vector between the desired output signal and the tap-input vector i.e. $E[dy] = p$ and the auto-correlation matrix of the tap-input vector i.e. $E[yy^T] = R$, would usually be estimated using a large number of samples from d and y. In LMS algorithm, however, a very short-term estimate is used by only taking into account the current samples: $E[dy] \approx dy$, and $E[yy^T] = yy^T$, leading to an update equation for the filter coefficients

$$w^{new} = w^{old} + \mu / 2 (-\nabla_c J(w)) \quad (7)$$
$$= w^{old} + \mu y (d - y^T w)$$
$$= w^{old} + \mu ye$$

In equation 7, 'step-size' parameter i.e. μ are introduced, which controls the distance that the algorithm move along the error surface. In LMS algorithm the update of the coefficients in equation 7, is performed at every time instant n,

$$w(n+1) = w(n) + \mu e(n) y(n) \quad (8)$$

2.9.1 Simulation results of LMS algorithm

In figure 7 LMS algorithm shows a high rate of convergence of about 50 iterations with an approximate MSE value of -23dB, and a good BER performance on tracking mode figure 8.

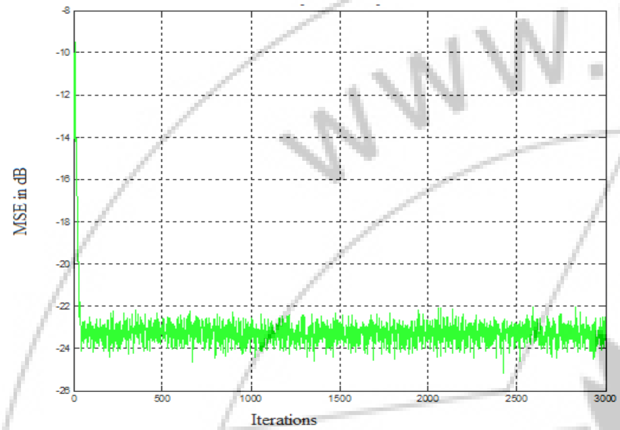

Figure 7: Convergence of LMS algorithm

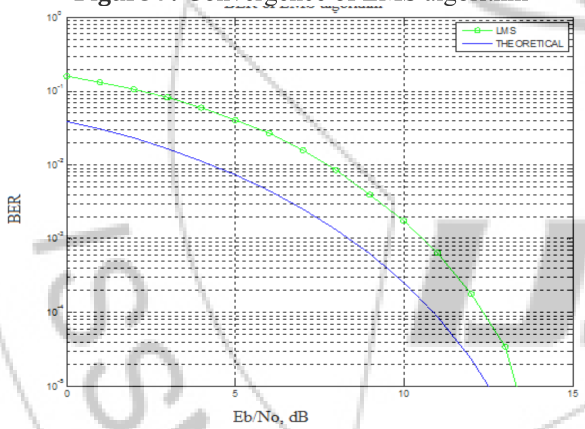

Figure 8: BER performance result of LMS algorithm

2.10 NLMS Algorithm

NLMS algorithm is a variety of LMS algorithm where the updating equation of the LMS algorithm may employ a variable convergence factor μ_n in order to improve the convergence rate. But in this case, the updating formula is expressed in equation 9 with fixed convergence factor μ so that each algorithm can be compared with the same step-size parameter.

$$w(n+1) = w(n) + 2\mu e(n) y(n) = w(n) + \Delta\hat{w}(n) \quad (9)$$

Substituting $\hat{w}(n) = w(n) + \Delta\hat{w}(n)$ and

$\Delta e^2(n) \overset{\Delta}{=} \hat{e}^2(n) - e^2(n)$ so as to achieve a good convergence rate, the motive is to make $\Delta e^2(n)$ negative and minimum by appropriately choosing μ. The objective behind this strategy is that the instantaneous squared error is a good and simple estimate of the MSE.

Also by replacing $\Delta\hat{w}(n) = 2\mu e(n) y(n)$ in equation 9, it follows that:-

$$\Delta e^2(n) = -4\mu e^2(n) y^T(n) y(n) + 4\mu^2 e^2(n) \left[y^T(n) y(n)\right]^2 \quad (10)$$

The value of μ is given in equation 10 in such a way that $\frac{\partial \Delta e^2(n)}{\partial \mu} = 0$:

$$\mu = \frac{1}{2y^T(n) y(n)} \quad (11)$$

This value of μ leads to a negative value of $\Delta e^2(n)$, and, therefore, it corresponds to a minimum point of $\Delta e^2(n)$.

Using this fixed convergence factor, the updating equation for LMS algorithm is then given by

$$w(n+1) = w(n) + \frac{e(n) y(n)}{y^T(n) y(n)} \quad (12)$$

Usually a fixed convergence factor μ is introduced in the updating formula in order to control the misadjustment, since all the derivations are based on instantaneous values of squared errors and not on the MSE. Also a parameter γ should be included, in order to avoid large step sizes when $y^T(n) y(n)$ becomes small. Then the coefficient updating equation of LMS algorithm becomes NLMS and is then given by

$$w(n+1) = w(n) + \frac{\mu}{\gamma + y^T(n) y(n)} e(n) y(n) \quad (13)$$

2.10.1 Simulation results of NLMS algorithm

In figure 9 NLMS algorithm shows a high rate of convergence of about 150 iterations with an approximate MSE value of 25dB, and good BER performance on tracking mode figure 10.

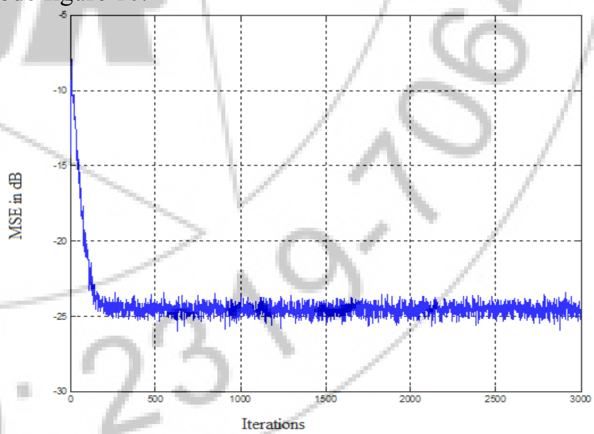

Figure 9: Convergence of NLMS algorithm

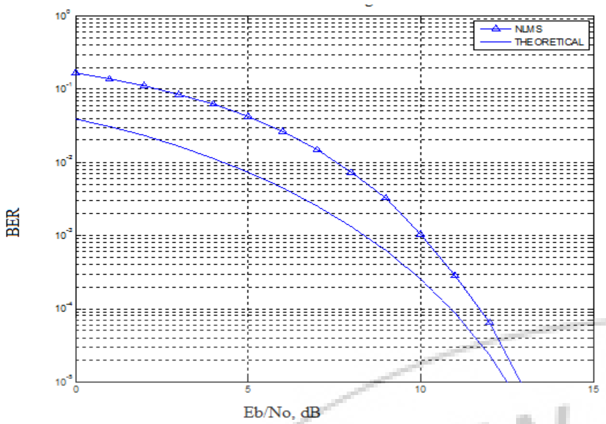

Figure 10: BER performance result of NLMS algorithm

2.11 RLS Algorithm

In contrast to LMS algorithm, the RLS algorithm uses information from all past input samples (and not only from the current tap-input samples) to estimate the (inverse of the) autocorrelation matrix of the input vector. To decrease the influence of input samples from the far past, a weighting factor for the influence of each sample is used. This weighting factor is introduced in the cost function in equation 14:

$$J(n) = \sum_{i=1}^{n} \rho^{n-i} |e[i,n]|^2 \quad (14)$$

where $e(i,n)$ is the error signal and is computed for all times $1 \leq i \leq n$ using current algorithm coefficients $w(n)$ and ρ is called the forgetting factor.

$$e(i,n) = d(i) - w^T(n)y(i) \quad (15)$$

When $\rho = 1$ the squared error for all sample times i up to current time n is considered in the cost function J equally. If $0 < \rho < 1$ the influence of past error values decays exponentially: method of exponentially weighted least squares.

Analogous to the derivation of the LMS algorithm we find the gradient of the cost function with respect to the current weights

$$\nabla_c J(n) = \sum_{i=1}^{n} \rho^{n-i} \left(-2E(d(i)y(i)) + 2E(y(i)y^T(i))w(n)\right) \quad (16)$$

We now, however, do trust in the ability to estimate the expected values $E(dy) = p$ and $E(yy^T) = R$ with sufficient accuracy using all past samples, and do not use a gradient descent method, but immediately search for the minimum of the cost function by setting its gradient to zero $\nabla_c J(n) = 0$. The resulting equation for the optimum filter coefficients at time n is given in equation 17:

$$\Phi(n) w(n) = z(n) \quad (17)$$

$$w(n) = \Phi^{-1}(n) z(n) \quad (18)$$

With $\Phi(n) = \sum_{i=1}^{n} \rho^{n-i} y(i) y^T(i)$ and $z(n) = \sum_{i=1}^{n} \rho^{n-i} d^*(i) y(i)$

Both $\Phi(n)$ and $z(n)$ can be computed recursively:

$\Phi(n) = \rho\Phi(n-1) + y(n)y^T(n)$ and $z(n) = \rho z(n-1) + y^*(n)y(n)$

To find the coefficient vector i.e. $w(n)$, we need the inverse matrix $\Phi^{-1}(n)$. Using a matrix inversion lemma, a recursive update equation for $P(n) = \Phi^{-1}(n)$ is found as:

$$P(n) = \rho^{-1}P(n-1) + \rho^{-1}k(n)y(n) \quad (19)$$

with $k(n) = \dfrac{\rho^{-1}P(n-1)y(n)}{1 + \rho^{-1}y^T P(n-1)y(n)}$

Finally, the weights update equation is

$$w(n) = w(n-1) + k(n)\left(d^*(n) - y^T(n)w(n-1)\right) \quad (20)$$

2.11.1 Simulation results of RLS algorithm

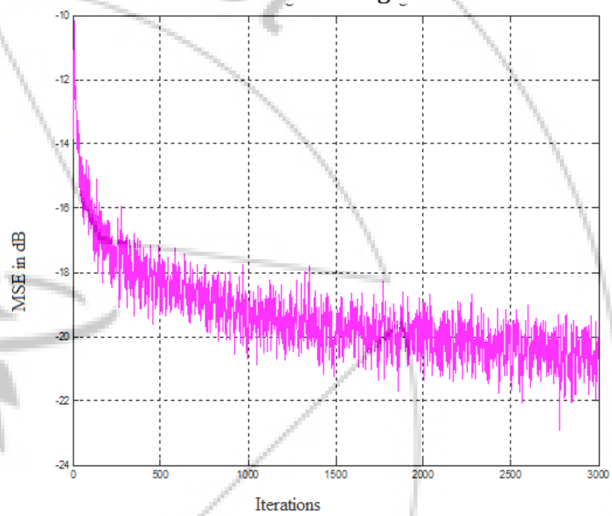

Figure 11: Convergence of RLS algorithm

In figure 11 RLS algorithm shows a low rate of convergence of about 1500 iterations with an approximate MSE value of -20dB, and poor BER performance on tracking mode figure 12.

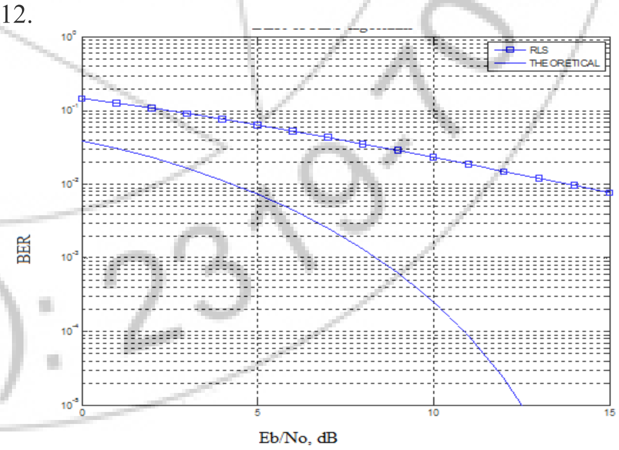

Figure 12: BER performance result of RLS algorithm

3. Convergence and BER Performance Comparison

Figure 13 compares convergence speed among all algorithms, with parameters as shown in table 2.

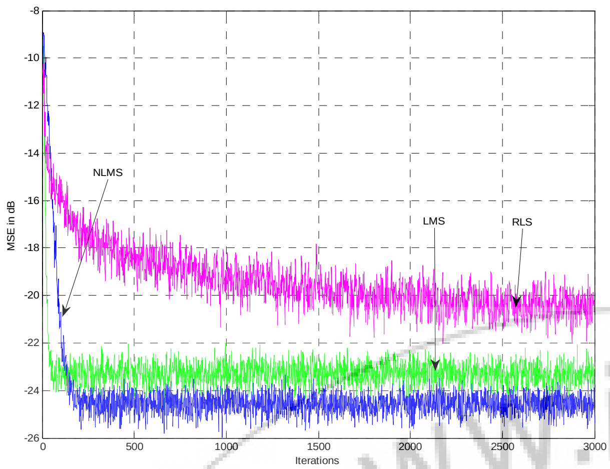

Figure 13: Convergence of LMS, NLMS and RLS algorithm

It appears that the convergence speeds of these algorithms are in the following order: LMS, NLMS, and RLS. Although the convergence rate of the LMS algorithm is slightly inferior to that of NLMS and RLS algorithms but this algorithm and RLS has much MSE value of -23dB respectively compared to NLMS of -25dB. This shows that NLMS algorithm has low MSE value of -25dB than the rest of algorithms.

In figure 14 comparisons of the BER performance among the algorithms are presented, it is shown that with an E_b/N_o 13dB, NLMS algorithm achieve a BER of 10^{-5}, LMS achieve a BER of 5×10^{-5} and RLS achieve a BER of 10^{-2} .

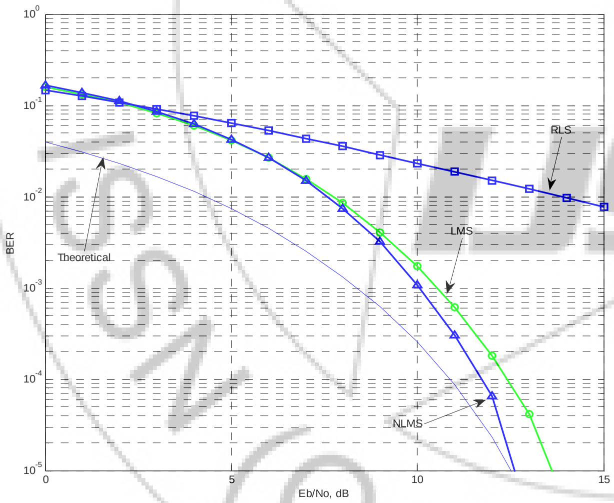

Figure 14: BER performance result of LMS, NLMS and RLS algorithm

4. Conclusion

This paper presents the data communication over the broadband power-lines networks. With the advent of new technologies in BPLC system, and in general, in the field of communications, adaptive equalizers have really proved to be very powerful in combating ISI effect. Effect of ISI from the BPL channel can be decreased (mitigated) using a NLMS algorithm rather than remaining algorithms that have been discussed on this paper. The simulations show that the effect of ISI causes an increase of BER. By applying adaptive equalizer, BER can be decreased. In Figure 13 and figure 14, shows that NLMS has better BER performance and nearly much faster rate of convergence compared to the remaining algorithms. For future work, performance of Normalized Least Mean Squares filter (NLMS) be analyzed so as to improve its BER performance and its convergence rate.

References

[1] Hrasnica, H., Haidine, A., Lehnert, R. (2004), *Broadband Powerline Communications Networks-Network Design*, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

[2] Amirshahi, P. and Kavehrad, M. (2007), "System Design Considerations for High data Rate Communications Over Multi-wire Overhead Power-lines", The Pennsylvania State University, Department of Electrical Engineering, Center for Information & Communications Technology Research (CICTR) University Park, PA 16802.

[3] Anatory, J. and Theethayi N. (2010), *Broadband Power line Communication System: Theory and Applications*, WIT Press, Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK.

[4] Malowanchuk, B. (2004), "Broadband Over Power-line (BPL) interference: Fact or Fiction?", *Canada's Amateur Radio Magazine*, Vol. , No. , pp. 39-44.

[5] Paulo S.R. D. (2008), *Adaptive Filtering Algorithms and Practical Implementation*, (Third Edition), Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA.

[6] Anatory, J., Theethayi, N., Thottappillil, R., and Mvungi, N. H. (2009), "A Broadband Power-Line Communication System Design Scheme for Typical Tanzania Low-Voltage Network", *IEEE Transaction on Power Delivery*, Vol. 24, No. 3, pp. 1218-1224.

[7] N. Mtonyole, J. Anatory and A. Mvuma, "Effects of Multipath on MC-CDMA for Broadband Power-line Communications" Journal of Informatics and Virtual Education, ISSN 1821-7087 ,Vol. 2, No. 01, November 2012 , pp 15-23.

[8] Frias-Velazquez, A. and Romero-Troncoso, R. J. (2005), "Algorithm for Convergence Criteria Simulation on LMS Adaptive Filters", *Telecommunications And Radio Engineering C/C Of Elektrosviaz' And Radiotekhnika*, Vol. 64, No. 7/12, pp. 537-542.

[9] Garas, J. (2000-2002), *The Adaptive Signal Processing Toolbox: For use with Matlab*, ASPT User Manual Version 2.1, DSP ALGORITHMS.

[10] Haykin, S. (1996), *Adaptive Filter Theory*, (Third Edition), Prentice-Hall, Inc., Upper Saddle River, NJ.

[11] Hormis, R., Berenguer, I. and Wang, X. (2006), "A Simple Baseband Transmission Scheme for Power Line Channels", *IEEE Journal on Selected Areas in Communications,* Vol. 24, No. 7, pp. 1351-1363.

[12] Huaiyu, D. and Vincent, H. P. (2003), "Advanced Signal Processing for Power line Communications", invited paper, *IEEE Communications Magazine*, Vol. 41, No. 5, pp. 100-107.