Equalisation and Frequency Offset Correction for HIPERLAN

C Tellambura
Monash University, Dept. of Digital Systems
Clayton, Victoria 3168, Australia
Email: chintha@dgs.monash.edu.au

I R Johnson, Y J Guo and S K Barton
Dept. of Electrical and Electronic Eng.
University of Bradford
West Yorkshire BD7 1DP
UK

Abstract— **To reduce the effects of inter-symbol interference resulting from the dispersive nature of the indoor radio channel most HIPERLAN receivers will incorporate an adaptive equaliser. In this paper, compuational complexity of several equaliser algorithms is estimated. The effect of frequency offset between transmitter and receiver on the performance of such an equaliser is investigated. By employing a decision feedback equaliser incorporating a second order phase locked loop, the effect of both the intersymbol interference and frequency offset can be significantly reduced. Using such a technique, the packet error ratio (PER) of a HIPERLAN radio link in a multipath channel is found by simulation.**

I. Introduction

Recently completed by the European Telecommunications Standards Institute (ETSI) [1], the HIgh PErformance Radio Local Area Network (HIPERLAN) standard intends to provide high performance, low-cost, low-power networking capability on a PCMCIA card. Its high data rate will support a variety of applications, both time-bounded and asynchronous. HIPERLAN facilitates so-called *ad-hoc* networking, where a group of users can set up a network without the requirements of a central controlling node (mesh topology) [2,3].

HIPERLAN requires low cost oscillators, with an accuracy not worse than 10 parts per million (ppm). Thus, at 5.2 GHz the output frequency of a 10 ppm oscillator will be in the range $f_c \pm 50$ kHz, where f_c is the nominal carrier frequency. Assuming the frequency offset of each oscillator is uniformly distributed over ± 50 kHz, the frequency offset between a transmitter and receiver has a triangular distribution over ± 100 kHz. The transmission rate in HIPERLAN is 23.5294 Mbit/s (bit period of 42.5 ns). At this transmission rate a worst-case frequency offset of 100 kHz corresponds to a phase rotation in the received signal of about 1.54°/bit. The modulation scheme in HIPERLAN is GMSK with pre-coding, so that data is represented by the absolute phase of constellation points and not the phase difference between successive constellation points. The frequency offset manifests itself as a phase rotation of the data. For small rotations this will have the effect of reducing the signal to noise ratio. For larger offsets the data may be taken over the decision threshold, in which case nominally correct data will be misdetected.

The high transmission rate of HIPERLAN and dispersive nature of the indoor radio channel mean that HIPERLAN equipment will usually require an equaliser to be able to successfully recover transmitted data. HIPERLAN has been designed to support limited mobility, restricted to pedestrian speeds. A maximum speed of 1.4 m/s leads to a maximum Dopper spread of ± 25 Hz. This means that in the period of the longest data packet (1ms) no path can change in length by more than $\lambda/40$, where λ is the wavelength. In other words, one assumes the radio channel is quasi-static for the duration of the longest packet. In this case it should be possible to compute the equaliser coefficients once at the start of the packet, and they will remain valid throughout the packet, i.e. it is not necessary to operate the equaliser in tracking mode. However, a frequency offset of 50 kHz produces 50 cycles of rotation in 1ms, so that some processing of the signal in the receiver must be carried out to compensate for the frequency offset.

This paper considers two solutions to the problem of frequency offset. Firstly, the equaliser coefficients can be updated on a continuous basis. This method is computationally complex and has limited performance. Secondly, joint optimisation of equaliser tap weights and demodulator phase can be achieved by incorporating a phase locked loop in the equaliser. The performance with first and second order loops is considered. Performance of this method is compared to that without any frequency offset by simulation of a HIPERLAN radio link. Reference [4] also covers these topics more extensively.

II. Continuous Update of Equaliser Coefficients

Many recursive algorithms are available for calculating the equaliser coefficients [5]. Choice of algorithm is a compromise: simple algorithms require many iterations, complex algorithms require few. The requirement of an immediate positive acknowledgement in HIPER-

The work described in this paper was supported by the UK DTI/EPSRC LINK project: PC2011 "High Throughput Radio Modem" in collaboration with Symbionics Networks Limited under EPSRC grant GR/K00318.

LAN means that the equaliser taps must be computed in real-time, i.e. the packet cannot be buffered whilst the coefficients are calculated. This essentially means that an iteration of the selected algorithm must be performed every 42.5 ns. In order to achieve this with current technology it was anticipated that a simple training algorithm such as Least Mean Squares (LMS) would be used. Each data packet in HIPERLAN is preceded by a pre-determined 450 bit sequence for synchronisation purposes and equaliser training.

Some analysis of computational complexity of several equalisation algorithms is provided in Fig. 1. Here, the number of computations per second needed to implement the equaliser is estimated for varying data rates and rms delay spreads. This is quite important since the equaliser performs a large fraction (e.g. 80%) of the computation needed to implement the complete demodulator.

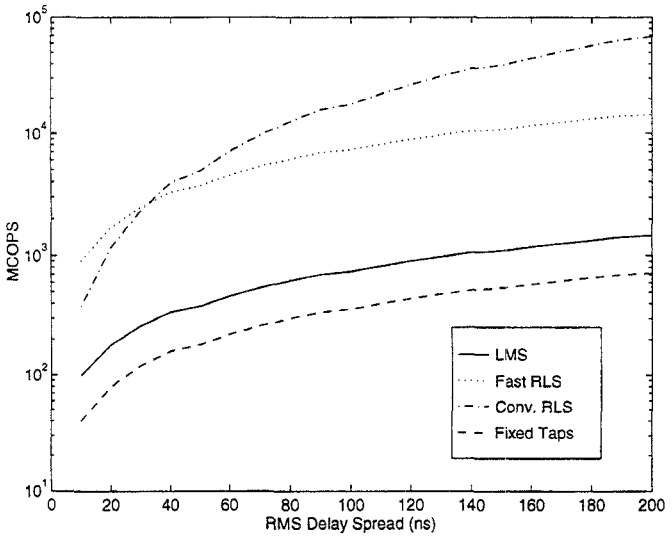

Fig. 1. Computational requirements in millions of complex operations per second (MCOPS) for LMS, RLS and fast RLS equalisers. Data rate is 20 Mbit/s.

The computational requirements depend on the equaliser length. Unfortunately, analytically determining the minimum required equaliser length is not possible in most cases. Therefore, the following simplifying assumptions are made:

- the channel impulse response length is three times the rms delay spread τ_{rms}.
- under virtually noise-free conditions, the equaliser's pulse response will approximate the inverse of the channel. Thus, the equaliser length should be several times longer than the channel impulse response length. This factor is assumed to be 3.

The computations needed for operation of an adaptive equaliser are for: updating the tap weights and computing the output. The latter needs N complex multiplications for an N-tap equaliser, regardless of the tap adjustment algorithm used. In Fig. 1 the 'Fixed Tap' curve shows the computational burden if the taps are fixed. The other three curves show the computational burden for updating and computing the output. In all cases, the

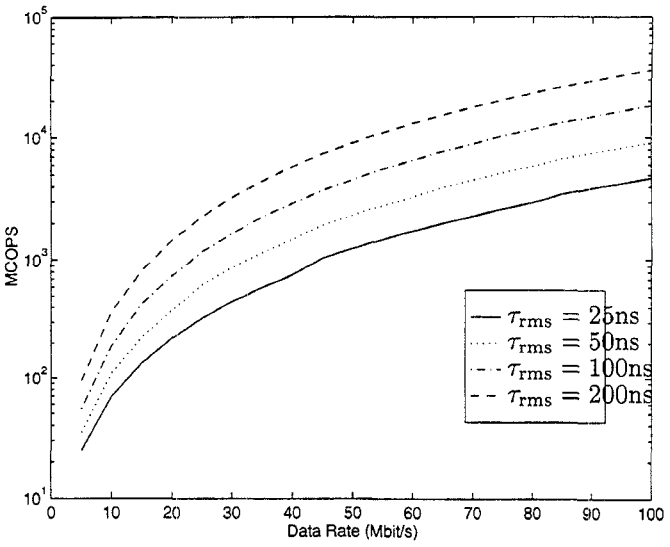

Fig. 2. Computational requirements in MCOPS for an LMS equaliser versus data rate.

computational burden increases rapidly as the rms delay spread increases. More complex algorithms such as RLS require large number of computations for tap updating. This fact can be observed in Fig. 1. For the LMS algorithm, the computational burden for varying data at a given rms delay spread is provided in Fig. 2.

The LMS algorithm for an adaptive equaliser may be summarised as follows: the current equaliser output, y_n, is given by

$$y_n = \mathbf{C}_n \mathbf{R}_n^T$$

where $\mathbf{C}_n$ is a vector representing the equaliser coefficients, $\mathbf{R}_n$ is the current input vector, T represents the transpose of the vector $\mathbf{R}_n$, and the equaliser coefficients are updated according to

$$\mathbf{C}_{n+1} = \mathbf{C}_n + \beta e_n \mathbf{R}_n$$

where e_n is the error signal and β is the step size parameter. For a linear equaliser, $\mathbf{R}_n$ contains received signal samples only, whereas for a decision feedback equaliser (DFE), $\mathbf{R}_n$ also contains past detected symbols. Clearly, for the time-variant channel caused by the frequency offset, the optimum tap vector will also be time variant. The LMS algorithm attempts to minimise the output MSE, but always lags because of noisy estimates. Thus, the output MSE increases by two quantities, J_β due to noisy estimates, and J_l due to the lag. Decreasing β reduces J_β but increases J_l. Conversely, increasing β reduces J_l but increases J_β. However, the maximum possible step size, $\beta_{\max}$, is limited by the channel eigenvalue spread and equaliser length [5,6]. As the frequency offset increases, J_l dominates the equaliser performance, even the use of $\beta_{\max}$ being insufficient, and the LMS equaliser breaks down.

The selection of the step-size parameter to allow tracking of the phase rotation places additional constraints on system design - not only does one need to select a step size parameter suitable for the multipath channels one

may encounter (as indicated by the eigenvalue spread of the channel correlation function) [5] but it must also be selected to allow tracking of any frequency offset. Selection of a step size parameter which fulfils both of these requirements may not always be possible - a large eigenvalue spread dictates the step-size parameter must be small but this will not allow suitable tracking of a large frequency offset.

III. Joint optimisation of equaliser tap weights and demodulator phase

The packet duration in HIPERLAN was selected to make the assumption of a quasi-static channel valid, even during the longest packets. The only channel variation which a receiver sees during a packet is that due to transmitter/receiver frequency offset. This can be compensated by continuously updating the equaliser coefficients or by varying the demodulaor phase. An efficient method for joint optimisation of equaliser tap weights and demodulator phase has been given by Falconer in his classic paper [7]. Figure 3 shows a linear transversal equaliser combined with a phase tracking demodulator. The output of the equaliser is rotated back by the demodulator phase $\hat{\theta}_n$, which is an estimate of the phase rotation, and a hard decision is made on this signal. The equaliser output is then subtracted from the hard decision which has been rotated by the phase estimate $\hat{\theta}_n$, the resulting signal being the error signal used in updating the equaliser coefficients.

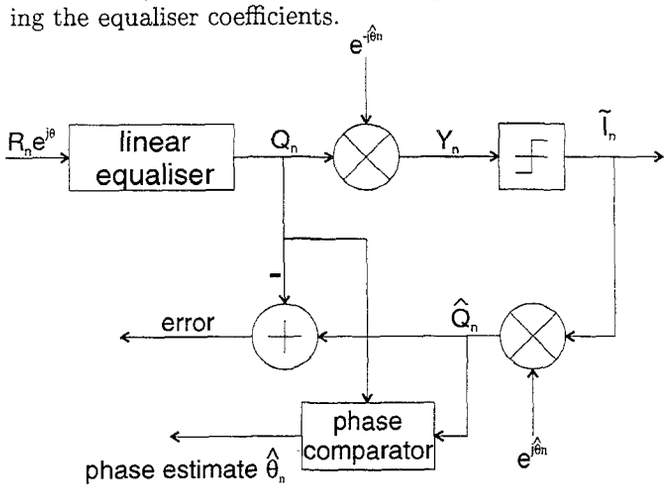

Fig. 3. Linear equaliser with phase estimation

The equations used to update the coefficients and to estimate the demodulator phase $\hat{\theta}_n$ for the LMS algorithm are the coupled stochastic difference equations [7]

$$\mathbf{C}_{n+1} = \mathbf{C}_n - \beta \mathbf{R}_n (Q_n^* - \hat{Q}_n^*) \quad (1a)$$

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \alpha \mathrm{Im}(Q_n \hat{Q}_n^*) \quad (1b)$$

where α is a phase update parameter and β is the usual step size.

A. First Order Phase Locked Loop

The system in Figure 3 when used with equations (1a) and (1b) is an equaliser incorporating a standard, first order phase locked loop (PLL) for estimating the phase of the signal. Changing the step size parameter changes the rate of convergence of the equaliser, as would be the case for an equaliser with no frequency offset. Correct operation of a first order PLL requires a non-zero output from the phase comparator, otherwise there is no signal to drive the phase accumulator. A first order PLL thus always operates with some phase lag. This phase lag can be overcome by using a second order PLL.

A.1 Analysis of 1st order Discrete PLL

Eqs. (1a) and (1b) describe a joint LMS equaliser and data-directed PLL [7]. The exact performance analysis of this structure is extremely difficult. However, insight into the operation of the PLL can be obtained by the following approximate analysis using two simplifying assumptions. Firstly, the effect of the equaliser is ignored by decoupling the two equations. Secondly, the PLL operates with a small error signal (linearised model) and $\mathrm{Im}(Q_n \hat{Q}_n^*) \approx \theta_n - \hat{\theta}_n$. Then, (1b) is a discrete first-order PLL given by

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \alpha(\theta_n - \hat{\theta}_n). \quad (2)$$

Taking z-transforms of both sides gives

$$H(z) = \frac{\hat{\theta}(z)}{\theta(z)} = \frac{\alpha}{z - 1 + \alpha}. \quad (3)$$

$H(z)$ has a pole $p = 1 - \alpha$. To ensure stability, $|p| < 1$ giving $0 < \alpha < 2$. The variance of phase error due to additive noise is quite important in practice. Loop band width, a measure of this variance, is commonly used in linear PLL theory, and is given by the area under the curve $|H(\omega)|^2$. However, for a discrete PLL, $H(e^{j\omega})$ is periodic. Thus, the loop bandwidth in this case can be defined as

$$B_L = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \, d\omega. \quad (4)$$

Evaluating this for the first order PLL (3) gives

$$B_L = \frac{\alpha}{2 - \alpha}. \quad (5)$$

If $\alpha \to 0$, then $B_L \to 0$ and $p \to 1$. The PLL response to a sinusoid contains the transient term $a_0 p^n$ $(n = 1, 2, \ldots)$, where a_0 is a constant. Thus, reduced noise bandwidth leads to larger lock-in time. The phase error sequence is defined as

$$e_n = \hat{\theta}_n - \theta_n, \quad (6)$$

and taking z-transforms of both sides yields

$$e(z) = (H(z) - 1)\theta(z). \quad (7)$$

It is expected that the phase error, e_n, vanishes as $n \to \infty$. Thus, $\lim_{n\to\infty} e_n$ is a useful performance indicator. Using the final value theorem for z-transforms gives

$$\lim_{n\to\infty} e_n = \lim_{z\to 1} (1 - z^{-1}) e(z). \quad (8)$$

Consider a constant phase sequence, i.e. $\theta_n = \theta_0$ for $n = 0, 1, \ldots$. Then $\theta(z) = \theta_0/(1 - z^{-1})$, and

$$\lim_{n\to\infty} e_n = \lim_{z\to 1}(1 - z^{-1})e(z) = 0. \quad (9)$$

Thus this PLL tracks a constant phase sequence without any error. Similarly, for a constant frequency input, $\theta_n = n\delta\omega$ for $n = 1, \ldots$, where $\delta\omega$ is the frequency offset, and evaluating (8) gives

$$\lim_{n\to\infty} e_n = -\frac{\delta\omega}{\alpha}. \quad (10)$$

Thus, this loop produces a steady-state phase error in the presence of a frequency offset. Again as $\alpha \to 0$ (so that $B_L \to 0$), the phase error increases.

B. Second Order PLL

Equation (1b) may be modified slightly to

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \alpha(\mathrm{Im}(Q_n\hat{Q}_n^*) + \gamma\sum_{i=0}^{n-1}\mathrm{Im}(Q_i\hat{Q}_i^*)). \quad (11)$$

The parameter γ is an integration constant. In this case the low pass filter (integrator) of Equation (1b) has been replaced by a "proportional-plus-integral" filter and the loop becomes second order. The system then forms an equaliser incorporating a second order PLL. The convergence rate for smaller β is slower so the rms phase error is higher for a given number of iterations. A high value of β corresponds to faster convergence of the equaliser but the minimum MSE achievable is relatively large.

B.1 Analysis of 2nd order Discrete PLL

Making the same simplifying assumptions as before, (11) becomes the discrete PLL

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \alpha(\theta_n - \hat{\theta}_n) + \gamma\sum_{i=0}^{n-1}(\theta_i - \hat{\theta}_i). \quad (12)$$

Taking z-transforms on both sides

$$z\hat{\theta}(z) = (1-\alpha)\hat{\theta}(z) + \alpha\theta(z) + \frac{\gamma z^{-1}(\theta(z) - \hat{\theta}(z))}{1 - z^{-1}} \quad (13)$$

leading to the transfer function

$$H(z) = \frac{\hat{\theta}(z)}{\theta(z)} = \frac{\alpha z + \gamma - \alpha}{z^2 + (\alpha - 2)z + 1 - \alpha + \gamma}. \quad (14)$$

The two poles of $H(z)$ are

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \frac{2 - \alpha \pm \sqrt{\alpha^2 - 4\gamma}}{2}. \quad (15)$$

Thus, to ensure stability of the PLL, the loop gain, α, and the integration constant, γ, should be such that

$$|2 - \alpha \pm \sqrt{\alpha^2 - 4\gamma}| < 2. \quad (16)$$

Such limits on the two parameters would not exist in a continuous-time PLL. For $H(z)$ as given in (14), and using residue techniques to evaluate (4), the loop bandwidth can be calculated [4]. Therefore, this PLL tracks a constant frequency input without a steady-state phase error.

C. Decision Feedback Equaliser Incorporating a Phase Locked Loop

Since the LTE (linear transversal equaliser) produces noise enhancement on frequency selective channels, often a DFE is used instead. In addition, for the same number of taps, a DFE is less computationally complex than a LTE when single bit decisions are fed back. It is thus useful to derive the equations corresponding to (1a) and (1b) for a DFE incorporating a PLL.

The equations used for updating the equaliser coefficients must be modified as follows. Assume the equaliser has $(N + K + 1)$ taps, $N + 1$ feedforward and K feedback. The coefficient vector is denoted $\mathbf{C}_n = [C_{-N}, C_{-N+1}, \ldots, C_0, \ldots, C_K]$. The sampled input signal in the feedforward filter and past detected symbols in the feedback filter at the nth sampling interval are $\mathbf{R}_n = [R_{n+N}e^{j\theta_n}, R_{n+N-1}e^{j\theta_n}, \ldots, R_n e^{j\theta_n}]$ and $\mathbf{D}_n = [\tilde{I}_{n-1}, \tilde{I}_{n-2}, \ldots, \tilde{I}_{n-K}]$ where θ_n is the phase rotation due to frequency offset between the transmitter and receiver at the nth symbol. It is assumed that frequency offset is sufficiently small that the same value of θ_n can be applied to all samples in the forward filter. The following relationships then hold:

$$\begin{aligned} Q_n &= [C_{-N}, C_{-N+1}, \ldots, C_0]\mathbf{R}_n^T \\ F_n &= Q_n e^{-j\hat{\theta}_n} \\ B_n &= [C_1, C_2, \ldots, C_K][\tilde{I}_{n-1}, \tilde{I}_{n-2}, \ldots, \tilde{I}_{n-K}]^T \\ Y_n &= F_n + B_n \\ \tilde{I}_n &= \mathrm{sgn}(Y_n) \end{aligned}$$

where $\mathrm{sgn}(x) = 1$ if $x > 0$ and $\mathrm{sgn}(x) = -1$ if $x < 0$. The error signal used in updating the coefficients is

$$e_n = \tilde{I}_n - Y_n.$$

For the stochastic gradient approach, the gradients of

$$|e_n|^2 = |Q_n e^{-j\hat{\theta}_n} + B_n - \tilde{I}_n|^n \quad (17)$$

with respect to $\mathbf{C}_n$ and $\hat{\theta}_n$ are used to optimise tap weights and demodulator phase. It can be shown that

$$\frac{\partial|e_n|^2}{\partial\mathbf{C}_n} = -e_n[\mathbf{R}_n^* e^{j\hat{\theta}_n} \ \mathbf{D}_n^*] \quad (18a)$$

$$\frac{\partial|e_n|^2}{\partial\hat{\theta}_n} = -2\mathrm{Im}(Y_n(\tilde{I}_n - B_n)^*) \quad (18b)$$

and the signal in (21b) can be used to drive a first-order PLL (2) or second-order PLL (13) to get the phase estimate, $\hat{\theta}_n$. The coefficients are updated as follows:

$$\mathbf{C}_{n+1} = \mathbf{C}_n + \beta e_n[\mathbf{R}_n^* e^{j\hat{\theta}_n} \ \mathbf{D}_n^*]$$

which is an extension of that for the linear equaliser case derived in [7]. Note that if the feedback tap coefficients are zero, then $B_n = 0$ and the above algorithm reverts to that for the linear equaliser case [7].

D. System Performance Using a DFE Incorporating a PLL

To assess the performance of a system incorporating a PLL, simulations of a HIPERLAN link have been carried

out. The simulated system comprises GMSK (BT=0.3) with pre-coding and (31,26) BCH code with block interleaving (each encoded block of 496 bits is interleaved using a 16×31 block interleaver). The indoor channel is modelled as an 8-tap (T-spaced) linear filter. The taps are zero mean complex Gaussian random variables with an exponentially decaying profile. The normalized delay spread $\sigma_n = \sigma/T$ is adjusted by changing the exponent. The channel is stationary over a packet of 10370 $(450 + 496 \times 20)$ bits. Average system performance is obtained over 1000 randomly generated channels. A decision-feedback equaliser (DFE) with 11 taps in its forward section and 7 taps in its feedback section is used for reducing ISI and is jointly trained with the PLL. Significantly, the DFE is trained only during the first 450 bits and the resulting tap weights are *fixed* during the remainder of the packet. However, the PLL continues to run during the entire packet.

Note that BCH(31,26) can only correct a single error in a codeword. If a double error occurs, then the decoder will introduce more errors. Thus, the average bit error ratio is not a relevant performance measure in this case. Instead, the average packet error ratio (PER) (i.e., the fraction of packets in error after decoding) is computed.

Figure 4 shows several PER curves. It is seen that at a frequency offset of 150 kHz, the 2nd order PLL suffers a negligible performance degradation relative to the zero-offset case and this holds for both $\sigma_n = 0.5$ and $\sigma_n = 1.0$. Also, the 1st order PLL is not effective at this frequency offset. Clearly, a 2nd order PLL is capable of handling any frequency offset less than 150 kHz. Note that this performance level is achieved with the equaliser taps being fixed during the data section of each packet, leading to significant savings of power.

Figure 5 shows several PER as a function of frequency offset, ΔF. The performance of a system with a 1st order PLL rapidly degrades as ΔF increases above 50 kHz, whereas for the 2nd order PLL, the performance remains roughly constant for frequency offsets up to 300 kHz.

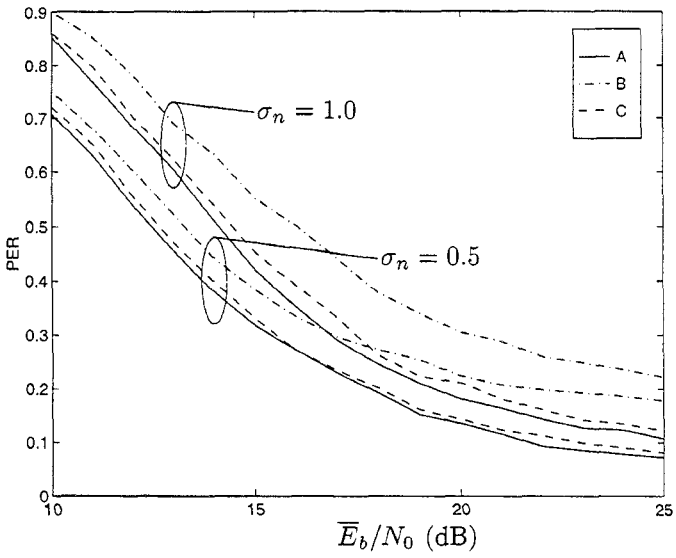

Fig. 4. Packet error rate (PER) for frequency offset correction using PLLs. Legends are A: $\Delta F = 0$; B: $\Delta F = 150$ kHz and a 1st order loop with $\alpha = 0.2$; C: $\Delta F = 150$ kHz and a 2nd order loop with $\alpha = 0.2$ and $\gamma = 0.025$. DFE(11,7) is trained using the LMS algorithm with $\beta = 0.01$.

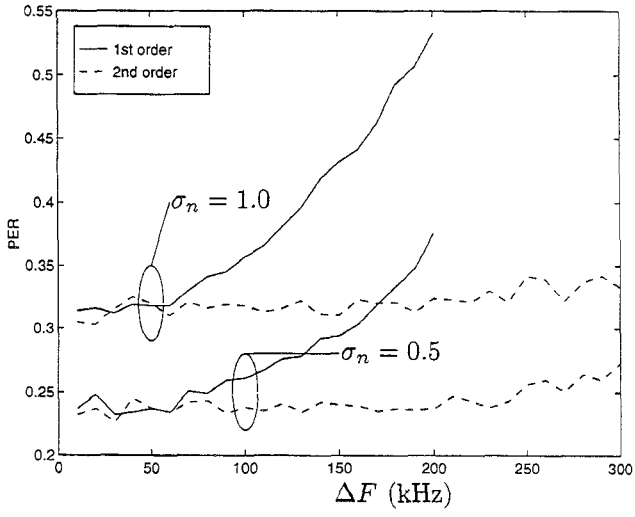

Fig. 5. Packet error rate (PER) for frequency offset correction using 1st order and 2nd order PLLs: 1st order loop with $\alpha = 0.2$; 2nd order loop with $\alpha = 0.2$ and $\gamma = 0.025$. DFE(11,7) is trained using the LMS algorithm with $\beta = 0.01$ and $\overline{E}_b/N_0 = 17$ dB.

IV. Conclusions

Equalisation techniques and transmitter/receiver frequency offset correction for HIPERLAN have been considered. Joint equalisation and phase estimation has been considered. The inherent phase lag in a first order loop limits performance. A second order loop has therefore been considered, which has significantly better performance. As the frequency offset is virtually constant over the longest packet there is no value in using higher order loops.

It is anticipated that much better performance will be achieved in HIPERLAN by using a decision feedback equaliser rather than a linear equaliser. The concept of joint equalisation/phase estimation has therefore been extended from a linear equaliser to a DFE. Simulation results of a typical HIPERLAN link using an equaliser incorporating a second order PLL show a negligible performance degradation relative to a system with zero frequency offset. The frequency offset such a system can tolerate with no decrease in performance is significantly higher than the HIPERLAN specification permits.

References

[1] ETSIprETS 300-652, "Radio Equipment and Systems (RES): High Performance Radio Local Area Network (HIPERLAN) Functional specification," April 1996.

[2] G. A. Halls, "HIPERLAN: the high performance radio local area network standard," *Electronics and Communication Engineering Journal/IEE*, pp. 289–296, Dec 1994.

[3] T. A. Wilkinson, T. Phipps and S. K. Barton, "A report on HIPERLAN Standardisation," *International Journal on Wireless Information Networks*, vol. 2, no. 2, pp. 99–119, 1995.

[4] I. R. Johnson, C. Tellambura, Y. J. Guo and S. K. Barton, "Mitigating the effect of Frequency offset in HIPERLAN," submitted to *IEE Proc. Comms.*, January 1997.

[5] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, 3rd Edition, 1996, USA.

[6] J. G. Proakis, *Digital Communications*. McGRAW-HILL, 2nd Edition, 1989, USA.

[7] D. D. Falconer, "Analysis of a Gradient Algorithm for Simultaneous Equalisation and Carrier Phase Recovery," *Bell System Technical Journal*, vol. 55, no. 4, pp. 409–428, April 1976.