A new approach to parameter estimation of time–varying systems, applied to fading digital mobile radio channels

Mikael Sternad, Anders Ahlén and Lars Lindbom
Automatic Control and Systems Analysis Group
Department of Technology, P. O. Box 534
S–751 21 Uppsala, Sweden

Abstract

Parameters of time–varying dynamic systems may sometimes be modelled as deterministic functions of time. This is shown to be feasible, and to improve the estimation accuracy, in the estimation of fading digital mobile radio channels. Such systems may be described as finite impulse responses, with coefficients varying, on a short time–scale, as *sinusoids*. We propose to estimate the phase, amplitude and frequency of these sinusoids, in each data burst in a TDMA system.

A prediction error identification algorithm has been developed, assuming each data burst to begin with a known training sequence. The parameter estimation is separated from the equalizer design and the equalization phase. The parameter estimation was tested on simulated data, generated by a two ray channel model, with 10 dB SNR. The proposed algorithm provided very accurate channel models, also in time intervals with severe fading. In contrast, the alternative approach of using recursive identification with a sliding data window was found not to behave well.

1 INTRODUCTION

Parameter estimation of time–invariant dynamic systems from noisy data is a relatively mature research area. Widely used algorithms, based on well–developed theoretical foundations, exist. See, for example, [8] or [13]. However, for systems having time–varying parameters, estimation algorithms are less satisfactory and their properties are far from completely understood. The conventional approach is to track time–variations of the system, using recursive identification with a receding data window [9].

Shortening of the data window improves the tracking capability but results in increasingly noisy parameter estimates. A tradeoff has to be made between the bias and the variance of the estimate. When the required tracking accuracy is high, and the measurements are noisy, there may exist no satisfactory compromise.

One way of avoiding this dilemma is to consider the physical cause of the parameter variations, and the typical way in which the parameters change, if such information is available. In particular, if the parameter vector $\theta(t)$, for a given data batch, can be described by a *deterministic function of time*, $\theta(t) = h(\theta_d, t)$, off–line identification of the parameter vector of this function, θ_d, may be considered. Compared to the tracking of $\theta(t)$ itself, by recursive estimation, this promises to improve the estimation accuracy, if the data batch is longer than the sliding window used by a tracking algorithm. Furthermore, the mean trajectories of estimates obtained with a sliding window follow the system parameter variations with a time lag. The proposed approach eliminates this bias.

While the ideas described above have general application, they have resulted from our work on digital mobile radio channels. This report describes some preliminary results of the work, and illustrates the parameter estimation approach.

When digital data are sent over a 900 MHz mobile radio channel, the signal travels along the line of sight between transmitter and receiver. Due to reflections from buildings and other obstacles, many other signal paths also contribute to the received signal. This *multipath propagation* has the advantage of making signal reception possible when the line of sight path is obscured. Time delays between different paths do, however, create problems; they may correspond to one or several symbol lengths, depending on the transmission rate.

This problem, known as *intersymbol interference*, causes decoding problems at the receiver. The most straightforward way to avoid it would be to decrease the symbol rate, so that path delay differences become small, relative to the duration of one digital symbol. For obvious reasons, this is unattractive from an economical point of view. A superior alternative is the use of *equalizers* at the receivers. An equalizer estimates the channel input, i.e. the transmitted digital data sequence, from the noisy received signal. See Figure 1.

In some situations, *linear equalizers* [3], [6] are appropriate. Significantly lower error rates can, however, often be achieved by the use of nonlinear algorithms, such as *decision feedback equalizers* (DFE's), see [2], [10] and [14]. *Viterbi equalizers* [4] are another alternative. The optimization of an equalizer requires reasonably accurate models of the transmission channel and the noise properties. Since channels are often time–varying, equalizers are frequently implemented as adaptive filters. See, for example, [5]. A good survey was given by Qureshi [12]. The channel modelling is quite often implicit, i.e. the parameters of the equalizer are estimated directly. It should be emphasized that the channel estimation problem is far from trivial. The input is unknown most of the time, yet the amplitude *and phase* properties of the channel must be estimated from noisy output measurements only.

Figure 1. Simplified description of the signals in the baseband of a digital communication system. The transmitted data $u(t)$ propagate through a dispersive channel, which is linear, but often time–varying. The received signal $y(t)$ is corrupted by noise $v(t)$. An equalizer reconstructs the data sequence.

With a mobile transmitter or receiver, the signal power through different propagation paths changes continuously. These variations have a strong *sinusoid* component, with frequency close to the Doppler frequency for the moving vehicle. See, for example, Figure 1–17 in [7]. The existence of such a regular, quasi–deterministic variation forms the basis of our channel parameter estimation approach.

2 MODELLING ASSUMPTIONS

With the modulation schemes used in digital mobile radio communication, signals and channel parameters in the baseband are represented as complex numbers. See e.g. [11]. In this report, the discussion of the method will be based on real–valued signals. Generalization to the complex case is rather straightforward.

In discrete time and in the baseband, the channel is described by a finite impulse response filter, with time–varying coefficients. These time–variations are well described by the Rayleigh fading model. For details of this model, see e.g. [7].

Let v be the velocity of the vehicle, c the speed of light and f_c the carrier frequency. The Doppler frequency for motion parallel to the signal path is

$$f_d = \frac{v f_c}{c} \tag{2.1}$$

In a Rayleigh fading environment, many different secondary signal paths contribute to each of the channel impulse response coefficients. The time variation of these coefficients may then be described as stochastic processes with significant energy around the frequency f_d, but insignificant energy above f_d. On short time–scales, it is thus reasonable to model them as sinusoids. Our basic modelling assumption is as follows:

> *In time intervals up to half a Doppler period, or $\tau \leq 1/2f_d$, the time variation of each channel impulse response coefficient is assumed to be well described by a sinusoid, with frequency $\leq f_d$.*

The received signal is corrupted by noise. The noise is coloured, in general mainly because of the effect of the receiver filter. Since this filter is known, its discrete time inverse may be applied to the sampled data. We assume that $y(t)$ has been prefiltered in this way. The noise $v(t)$ can then be assumed to be white, and the effect of the receiver filter on the channel impulse response is eliminated. [1]

Thus, let the channel, with a discrete time dispersion of m symbols, be modelled as

$$y(t) = b_o(t)u(t) + b_1(t)u(t-1) + \ldots + b_m(t)u(t-m) + v(t) \tag{2.2}$$

where the noise $v(t)$ is zero mean and white, with $\rho \triangleq Ev(t)^2/Eu(t)^2$, and the time–varying parameter vector

$$\theta(t) = (b_o(t) \ldots b_m(t))^T \tag{2.3}$$

is to be estimated. The coefficients vary approximately as sinusoids

$$\begin{aligned} b_o(t) &= K_o \sin(\omega_o t + \varphi_o) + e_o(t) \\ &\vdots \\ b_m(t) &= K_m \sin(\omega_m t + \varphi_m) + e_m(t) \end{aligned} \tag{2.4}$$

where $|e_i(t)|$ are assumed small, compared to $|K_i|$. The model is thus specified by the time–invariant parameter vector [2]

$$\theta_d = (K_o \ldots K_m \;\; \omega_o \ldots \omega_m \;\; \varphi_o \ldots \varphi_m) \tag{2.5}$$

[1] Incidently, it is optimal to include such noise–whitening prefiltering in Viterbi equalizers and also, as has been shown by Sternad and Ahlén [14], in realizable decision feedback equalizers.

[2] We have allowed the approximating frequencies $\{\omega_i\}$ in (2.4) to differ. Since all these frequencies are expected to be close to $2\pi f_d$, and thus close to each other, a model with only one common frequency might be adequate. The number of parameters is then reduced from $3(m+1)$ to $2(m+1)+1$.

With an estimate of θ_d, (2.4) provides an estimate of $\theta(t)$, for *any* t.

The signal $\{y(t)\}$ is assumed to be received in blocks of N data, corresponding to a time interval $\tau = N/f_s$, where f_s is the symbol rate. The input $\{u(t)\}$ is assumed known for the first $L \ll N$ samples of each data block (the *training sequence*), and unknown thereafter. We make the following assumptions.

Assumption 1. The burst length N corresponds to at most half a period of the maximal Doppler frequency (2.1). Then, sinus modelling of (2.3) is reasonable, according to our basic assumption. This implies

$$\frac{N}{f_s} \leq \frac{c}{2 f_c v_{\max}} \tag{2.6}$$

where f_s is the symbol rate, f_c is the carrier frequency and $v_{\max}$ is the maximal vehicle velocity.

Assumption 2. The length of the training sequence, L, is not smaller than the number of parameters in θ_d. Thus,

$$L \geq 3(m+1) \tag{2.7}$$

After a complete data block has been received, the proposed algorithm runs through three phases.

- In Section 3.1, an algorithm used on the training sequence is presented. It is denoted *Phase I*, and has the purpose to provide reasonable initial values for the second phase. Assumption 2 prevents the optimization problem from being under–determined.

- In *Phase II*, described in Section 3.2, the estimate of θ_d is improved using the rest of the data block, where $u(t)$ is unknown. Instead of the true $u(t)$, an estimate, obtained from a simple linear equalizer, is utilized in the algorithm. Although the estimation of θ_d is an off–line problem, the estimators in Phases I and II have been implemented as recursive algorithms, which traverse the data one or several times.

- In *Phase III*, which is not discussed in this report, a final pass through the data block is made by an equalizer, to estimate $u(t)$. The equalizer is a time–varying filter, which is optimized based on estimates of ρ and of (2.2), (2.4). The equalization could be performed by a DFE, a Viterbi algorithm, or any other kind of equalizer. (The effect of the receiver filter must be taken into account in this phase.)

The performance of the algorithms for Phases I and II is illustrated by a simulated example in Section 4.

3 THE ALGORITHM

3.1 Phase I: utilizing the training sequence

It is relatively straightforward to develop a recursive prediction error method for the model structure (2.2)–(2.5), with known input $u(t)$, and assuming $e_i(t) = 0$. Since the noise $v(t)$ is assumed to be white, the output predictor is

$$\hat{y}(t) = K_o \sin(\omega_o t + \varphi_o) u(t) + \ldots + K_m \sin(\omega_m t + \varphi_m) u(t-m) \tag{3.1}$$

The prediction error is denoted

$$\varepsilon(t) = y(t) - \hat{y}(t) \tag{3.2}$$

Assume that the parameter vector θ_d is held fixed. The negative gradient of $\varepsilon(t, \theta_d)$, with respect to θ_d, is then

$$\Psi(t) = -\left(\frac{\partial \varepsilon(t)}{\partial K_o} \cdots \frac{\partial \varepsilon(t)}{\partial K_m} \; \frac{\partial \varepsilon(t)}{\partial \omega_o} \cdots \frac{\partial \varepsilon(t)}{\partial \omega_m} \; \frac{\partial \varepsilon(t)}{\partial \varphi_o} \cdots \frac{\partial \varepsilon(t)}{\partial \varphi_m} \right)^T \tag{3.3}$$

where

$$\begin{aligned}
-\frac{\partial \varepsilon(t)}{\partial K_k} &= \sin(\omega_k t + \varphi_k) u(t-k) \quad , \quad k = 0, \ldots, m \\
-\frac{\partial \varepsilon(t)}{\partial \omega_k} &= K_k t \cos(\omega_k t + \varphi_k) u(t-k) \quad , \quad k = 0, \ldots, m \\
-\frac{\partial \varepsilon(t)}{\partial \varphi_k} &= K_k \cos(\omega_k t + \varphi_k) u(t-k) \quad , \quad k = 0, \ldots, m
\end{aligned}$$

Based on these relations, a recursive prediction error method for minimizing the criterion $J = E\varepsilon(t)^2$ can be obtained immediately, as explained, for example, in [9]. We have implemented the algorithm as a *repeated* recursive identification algorithm, because this improves the accuracy, in particular for short data series. (For a discussion of the repeated use of recursive identification algorithms to solve off–line problems, see [15].) The algorithm is listed on the next page.

.

Algorithm for Phase I

Below, t is the time index, pointing on the data, while i is a running index. Start with $t = m+1$, $i = 1$, with $\theta_d(1)$, $P(1)$ and $\lambda(1)$ given

1. Read $u(t)$ and $y(t)$.

2.
$$\begin{aligned} s_k &= \sin(\hat{\omega}_k^{(i-1)} t + \hat{\varphi}_k^{(i-1)}) u(t-k) \quad , \quad k = 0, \ldots, m \\ c_k &= \hat{K}_k^{(i-1)} \cos(\hat{\omega}_k^{(i-1)} t + \hat{\varphi}_k^{(i-1)}) u(t-k) \quad , \quad k = 0, \ldots, m \end{aligned} \tag{3.4}$$

3.
$$\varepsilon(i) = y(t) - \sum_{k=0}^{m} \hat{K}_k^{(i-1)} s_k \tag{3.5}$$

4.
$$\Psi(i) = (s_o \ldots s_m \;\; c_o t \ldots c_m t \;\; c_o \ldots c_m)^T$$

5. Update the forgetting factor $\lambda(i)$. (See (3.8) below.)

6.
$$P(i) = \frac{1}{\lambda(i)} \left(P(i-1) - \frac{P(i-1)\Psi(i)\Psi(i)^T P(i-1)}{\lambda(i) + \Psi(i)^T P(i-1)\Psi(i)} \right) \tag{3.6}$$

7.
$$\begin{aligned} \hat{\theta}_d(i) &\triangleq (\hat{K}_o^{(i)} \ldots \hat{K}_m^{(i)} \;\; \hat{\omega}_o^{(i)} \ldots \hat{\omega}_m^{(i)} \;\; \hat{\varphi}_o^{(i)} \ldots \hat{\varphi}_m^{(i)}) \\ &= \hat{\theta}_d(i-1) + P(i)\Psi(i)\varepsilon(i) \end{aligned} \tag{3.7}$$

8. .
If $t < L$, then $t = t+1$, $i = i+1$, go to step 1.
If $t = L$, either repeat: set $t = m+1$, $i = i+1$, and then go to step 1, or stop, if number of repeats or the accuracy is adequate.

Remarks.

Since the training sequence is assumed to be short, a couple of passes through the data are necessary to eliminate the effect of the initial value $P(1)$, so that the same accuracy as with an off–line algorithm is obtained. Only a few passes are necessary, in general. The variance of $\varepsilon(i)$, measured in the last pass, can serve as an estimate of the noise variance, to be utilized in Phase II.

Note that the model structure is nonlinear in the parameters. The elements of θ_d enter nonlinearly in $\Psi(t)$ and $\hat{y}(t)$.

The time–varying forgetting factor is initialized at 0.95 and approaches a final value $\lambda_\infty < 1$ exponentially.

$$\lambda(i) = 0.95\lambda(i-1) + 0.05\lambda_\infty \;\; ; \;\; \lambda(1) = 0.95 \tag{3.8}$$

This forgetting factor is *not* used because we wish to track time–variations. It is just a well–known trick to speed up convergence. (See, for example, [9], chapter 5.)

We suggest $\hat{\omega}_k$ and $\hat{K}_k$ to be initialized as the values obtained from the previous data bursts, since it is reasonable to assume the Doppler frequencies and the amplitudes not to change much between two consecutive bursts. The phases φ_k must, however, be considered to be completely unknown. [3]

The P–matrix, of dimension $\dim \theta_d | \dim \theta_d$, is initialized as a diagonal matrix. The diagonal elements are chosen proportional to rough guesses of the squared expected error of the initial estimate $\theta_d(1)$. This has worked well in the simulation experiments. [4] In an implementation in a signal processor, the P–matrix update (3.6) should be performed in factorized form, to obtain good numerical conditioning.

Note, that magnitude, frequency and phase decribing each b_i are coupled to each other, but orthogonal to the parameters describing other b–coefficients, if $u(t)$ is white. This should be of value in the design of fast algorithms, such as ladder or lattice algorithms.

After a few passes through the training sequence, the algorithm will provide a parameter vector θ_d, which approximates $\{b_k(t)\}$ reasonably well, for $t \in [1, L]$. The parameter estimates $\hat{\theta}_d$ will, however, in general be far from the true values, when (2.4) holds exactly. The model approximates the true sinus curves progressively worse for higher t. This is the case in particular in experiments with a high noise level. This is not surprising. The parameters cannot be accurately estimated from noisy measurements of a short data sequence, which constitutes only a very small fraction of one sinusoid period.

However, the main purpose of Phase I is merely to provide a good initial value for Phase II. The model of $\{b_k(t)\}$ should be reasonably accurate in the time interval $t \in [L, 2L]$, immediately after the training sequence. If this is not the case, the estimates of $u(t)$, utilized instead of the true $u(t)$ in Phase II, will frequently be erroneous in the beginning, with the possible risk that this algorithm diverges.

[3]During "start up", i.e. the first burst, it may be advantageous to use a somewhat longer training sequence. As a general recommendation, increase of the training sequence lenght is beneficial from an accuracy point of view.

[4]The initial diagonal values of the P–matrix can be seen as step length parameters; they determine the length of the steps taken in different directions in the parameter space, during the first few updates.

3.2 Phase II: estimation with an unknown input

The algorithm in Phase II uses the remaining $N - L$ data, which constitute the major part of the data block. It is identical to the scheme described in Section 3.1, except for the following modifications.

1. Instead of the unknown transmitted data sequence $\{u(t)\}$, estimates $\{\tilde{u}(t)\}$, obtained from a linear recursive equalizer, are used. The linear equalizer is a smoother, followed by a nonlinear memoryless decision module. It is re–calculated, based on the latest estimate of $\theta(t)$, as t increases. [5] For an example, see Section 4.

2. As initialization, $t = L + 1$ and $i = 1$, while $P(1)$ and $\hat{\theta}_d(1)$ receive the values obtained at the end of Phase I. In general, it is sufficient with one pass through the $N - L$ data in Phase II. The forgetting factor $\lambda(i)$ should be close to unity. We aim at maximal accuracy during this phase, so no information should be discarded.

3. To decrease the effect of occasional single incorrect estimates $\{\tilde{u}(t)\}$, the parameter updating law (3.7) should be robustified. We suggest the use of

$$\hat{\theta}_d(i) = \hat{\theta}_d(i-1) + P(i)\Psi(i)\bar{\varepsilon}(i)$$

   where

$$\bar{\varepsilon}(i) = \begin{cases} +\alpha & \text{if } \varepsilon(i) > \alpha \\ \varepsilon(i) & \text{if } -\alpha \leq \varepsilon(i) \leq \alpha \\ -\alpha & \text{if } \varepsilon(t) < -\alpha \end{cases} \tag{3.9}$$

   The parameter α may be set in the range $0.3\sigma_y - 0.7\sigma_y$, where σ_y is the standard deviation of $y(t)$ in the data batch. The length of updating steps caused by large (possibly erroneous) prediction errors is thus limited. (This corresponds to the use of a criterion $J(\varepsilon(t))$ which depends linearly, as opposed to quadratic, on $\varepsilon(t)$, above $|\varepsilon(t)| > \alpha$. See e.g. [9], chapter 5.)

4. The adaptation may be discontinued during *flat fading*, i.e. when all $|\hat{b}_k(t)|$ are small, relative to the estimated standard deviation of the noise. Under such periods, there is maximal risk of frequent erroneous estimates $\{\tilde{u}(t)\}$. Note that this does *not* imply that the resulting model (2.4) will be inaccurate during flat fading phases. On the contrary, with our method, it is possible to optimize the estimate θ_d only during the periods of the data burst when the local signal to noise ratio is high. The resulting sinusoid models will, however, be valid over the *whole* burst of length N. This is why the strategy can provide channel estimates which are highly accurate during phases of flat fading.

[5] The computations involved in optimizing linear recursive equalizers have been discussed by Fitch and Kurz [3] and by Ahlén and Sternad [1].

4 A TWO RAY MODELLED TRANSMISSION CHANNEL

Consider the general channel description adopted in Section 2. In this section, the channel will be characterized by the following properties. The carrier frequency is 900 MHz. The symbol time and the time dispersion is 40 μs. Thus, the discrete time channel has two coefficients. A time division multiple access (TDMA) system is used for scheduling of different messages. A message is received in bursts containing 170 data. The training sequence comprises the first 20 data of each burst. The signal to noise ratio varies from 10 dB to 20 dB. The transmitted sequence is binary and antipodal. Thus, $u(t) \in [-1, 1]$, with zero mean and unit variance. The measurement noise $v(t)$ is assumed white, zero mean and with variance ρ. As a nominal (highest) Doppler frequency for the main ray, $f_d = 83$ Hz is assumed, corresponding to a vehicle driving at 100 km/h.

We thus have the following specifications: $m = 1$, $N = 170$, $L = 20$, $f_s = 25$ kHz, $Ev(t)^2 = \rho$ and $Eu(t)^2 = 1$. The condition (2.7) is satisfied. Although the condition (2.6) is almost, but not quite, satisfied, we will assume the data to be generated by a system in which the sinusoid description (2.4) holds *exactly.*

Thus, the measured data are assumed to be generated by

$$y(t) = b_o(t)u(t) + b_1(t)u(t-1) + v(t) \tag{4.1}$$

where the coefficient variations are described exactly by

$$\begin{aligned} b_o(t) &= K_o \sin(\omega_o t + \varphi_o) \\ b_1(t) &= K_1 \sin(\omega_1 t + \varphi_1) \end{aligned} \tag{4.2}$$

Thus,

$$\theta_d = (K_o, K_1, \omega_o, \omega_1, \varphi_o, \varphi_1) \tag{4.3}$$

The SNR is $E(b_0^2 + b_1^2)/\rho$. Straightforward use of the algorithm in Section 3 gives

$$\varepsilon(t) = y(t) - \hat{K}_o \sin(\hat{\omega}_o t + \hat{\varphi}_o)u(t) - \hat{K}_1 \sin(\hat{\omega}_1 t + \hat{\varphi}_1)u(t-1) \tag{4.4}$$

in Phase I, while in Phase II

$$\tilde{u}(t) \triangleq \text{sign}(\hat{u}(t)) \tag{4.5}$$

is substituted for $u(t)$.

The estimate $\hat{u}(t)$, utilized in Phase II, is obtained from a linear smoother. This smoother can be optimized by solving a spectral factorization and a linear polynomial equation. See [1] or [14]. In this particular case, a one step smoothing estimate $\hat{u}(t|t+1)$ turns out to be sufficiently accurate. For this simple channel model, we can express its coefficients as explicit functions of $\hat{b}_o$, $\hat{b}_1$ and $\hat{\rho}$.

The smoothing input estimator so obtained is given by

$$\hat{u}(t|t+1) = -\frac{\hat{b}_o\hat{b}_1}{r}\hat{u}(t-1|t) + \frac{\hat{b}_1}{r}\left(1 - \frac{\hat{b}_o^2}{r}\right)y(t+1) + \frac{\hat{b}_o}{r}y(t) \tag{4.6}$$

where

$$r \triangleq \frac{g}{2} + \sqrt{\frac{g^2}{4} - \hat{b}_o\hat{b}_1} \qquad g \triangleq \hat{b}_o^2 + \hat{b}_1^2 + \hat{\rho}$$

Instead of a linear equalizer, the Viterbi algorithm could just as well have been used. However, it is *not* advisable to use a decision feedback equalizer in Phase II. Its tendency to generate occasional long bursts of errors may throw the adaptation completely off course. (This suspicion has been verified in simulated examples.) The ability to avoid adaptation on DFE–decisioned data is a main advantage of the approach.

The Phase II–algorithm is described schematically in Figure 2. The following example illustrates the properties of the algorithm.

Figure 2. A schematic view of the steps involved in Phase II, where θ_d is updated recursively by means of a nonlinear prediction error algorithm. An LFE estimate $\tilde{u}$ provides inputs to the predictor. A new prediction error is found, and the algorithm takes a new step towards the minimum of the criterion. The forward and backward shift operators are denoted q and q^{-1}, respectively.

Example 1. Let the true parameter vector in (4.3) be

$$\theta_d = (\ 1.00 \ \ 0.50 \ \ 0.0176 \ \ 0.0176 \ \ 1.00 \ \ 2.00\ )^T$$

which corresponds to a Doppler frequency of 70 Hz and a second ray damped 6 dB. We simulate (4.1) with a signal to noise ratio of 10 dB. In Figures 3a and 3b, we see the received signal $y(t)$ and the transmitted data sequence $u(t)$, during one burst.

Figure 3a. The received signal.

Figure 3b. The transmitted data.

In Phase I, five iterations are used. (Thus, the time index t goes from 2 to 20, while the running index i goes from 1 to 95.) We choose the forgetting factor $\lambda(i)$ as in (3.8), with $\lambda_\infty = 0.98$. Initial values of θ_d and P are set to

$$\theta_d(1) = (\ 0.5 \ \ 0.2 \ \ 0.01 \ \ 0.01 \ \ 0, 0\ )^T$$

$$P(1) = \text{diag}(\ 10 \ \ 10 \ \ 0.1 \ \ 0.1 \ \ 100 \ \ 100\ )$$

Use of the Phase I provides the following estimate

$$\hat{\theta}_{dI} = (\ 0.93 \ \ 0.72 \ \ 0.0058 \ \ -0.0018 \ \ 0.99 \ \ 2.30\ )^T$$

In Figures 4a and 4b, we see how the channel coefficients in (4.2) are approximated during the first 40 data. Note that the approximation is reasonable, not only during the training sequence, but also during the next 20 data points.

Figure 4a The Phase I–estimate of $b_o(t)$ (dotted), and the true value (solid). Training sequence $t \in [1, 20]$.

Figure 4b The Phase I–estimate of $b_1(t)$ (dotted), and the true value (solid).

In Phase II, we apply the modified Phase I algorithm, described in Section 3.2. The robustifying parameter is $\alpha = 0.3$. After the 150 data run, an improved $\hat{\theta}_d$–vector

$$\hat{\theta}_{dII} \quad = \quad (\ 0.95 \ \ 0.53 \ \ 0.0181 \ \ 0.0206 \ \ 0.87 \ \ 1.80\ )^T$$

is obtained. Figure 5 shows the channel coefficients and their approximation after Phase I and Phase II. The incorrect decisions generated by the LFE are shown in Figure 6. Only a few errors were obtained, and they did not affect the adaptation significantly. The risk of error increases during fading of the main ray b_o. This is evident in this example, where $b_o(t)$ is small around $t = 120$.

Figure 5. The true impulse response coefficients as functions of time (solid lines), their approximation after Phase I (dotted line) and the final approximation after Phase II (dashed line).

Figure 6. The input estimation errors $u(t) - \tilde{u}(t)$ in Phase II, caused by incorrect decisions by the linear equalizer (4.5), (4.6).

Example 2. Using the same received and transmitted data as in Example 1, we apply the recursive least–squares (RLS) algorithm. It tracks changes of the impulse response coefficients using a forgetting factor. This corresponds to the use of an exponentially weigthed sliding window. The parameter vector to be estimated will then be

$$\theta_{RLS} = (\ b_o \ \ b_1\ )^T$$

In this special case, we choose the forgetting factor $\lambda = 0.90$, which seems to be the least unsatisfactory compromise between fast tracking and noise sensitivity. Initial values of the θ_{RLS}–vector and P–matrix are chosen as

$$\hat{\theta}_{RLS}(1) = (\ 0 \ \ 0\ )^T$$

$$P(1) = \text{diag}(\ 100 \ \ 100\ )$$

In the first simulation, we use all the 170 transmitted data, *as if they were known*, to see how well changes of the coefficients could be tracked, in an ideal case. Figure 7 shows the result.

Figure 7. True parameters (solid lines) and the RLS tracking estimates (dotted), with $\lambda = 0.90$. The input $\{u(t)\}$ is assumed known for the whole data sequence. Compare the estimates from our algorithm in Example 1 (dashed lines).

Next, a more realistic case, where only the first 20 transmitted data are known, is investigated. In the remaining data series, the LFE (4.5), (4.6), updated in each sample based on the estimated channel parameters, provides data estimates. Figure 8 shows the true parameters and their estimates. The incorrect decisions generated by the LFE are shown in Figure 9. Because of the bad quality of the channel estimate in the fading phase around $t = 120$, the equalizer is incorrectly tuned and generates frequent errors. These destroy the parameter tracking, which causes still more errors, until the end of the data batch.

.

Figure 8. The true channel coefficients as functions of time (solid lines), and the RLS–estimates (dotted). Dashed lines are the estimates obtained by our new algorithm.

Figure 9. The input estimation errors $u(t) - \tilde{u}(t)$, caused by incorrect decisions by the linear equalizer (4.5), (4.6), when RLS–tracking is utilized.

5 CONCLUSIONS

We have considered the problem of estimating fading radio channels, with properties which are encountered in digital mobile radio communication in the 900 MHz band.

This has turned out to be a rather demanding system identification problem. The transmitted data sequence, i.e. the channel input, is known only for a small fraction of each data block of a TDMA system. The channel impulse response is time–varying. It has sometimes maximum phase and sometimes minimum phase. Frequent intervals of fading can be expected. The received signal power is then low, compared to the noise power. To reduce the error rate in equalizers, accurate channel models are required *in particular* around fading intervals. Unfortunately, the received signal contains little information about the channel properties in precisely these intervals.

We have succeeded in designing a channel estimation algorithm which promises to overcome these difficulties. It is based on three design principles, listed below. For systems with properties as those discussed in Section 4, we believe that it would be very hard to construct an equalizer which works acceptably, without utilizing at least the first and the third of these principles.

1. The channel estimation (Phase I and II) is *decoupled* from the final equalization (Phase III). We do not utilize decision–directed adaptation of equalizer parameters. Instead, the channel estimate is optimized using the whole data burst. A (preliminary) equalizer is used to estimate unknown transmitted data. The adaptation is sensitive to long bursts of errors, but insensitive to single errors. Thus, we utilize a linear equalizer, which has a higher error rate than a DFE, but does not generate long error bursts. (In the final Phase III, a DFE might be used.)

2. The effect of single decision errors (and of outliers due to noise) is reduced by robustifying the criterion, yielding (3.9). The parameter α is used to tailor the sensitivity.

3. Instead of tracking the channel parameters, their time variations are modelled by *deterministic functions*. In a fading mobile radio enviroment, sinusoids are the obvious choice. Data from time intervals with acceptable local signal to noise ratio are used to tune the sinusoid model parameters. The optimized model will then also describe the channel properties in intervals with low signal power.

The present report is an outline of the main ideas involved in our approach. Much work remains to be done. In particular, the following questions need to be adressed.

- The error in the sinusoid modelling assumption (2.4) needs to be investigated, for different fading enviroments and data burst lenghts N/f_s.

- The scheme needs to be generalized to, and tested for, the more realistic complex channel case, obtained e.g. in differential QAM. This seems to be straightforward.

- It should also be straightforward to include estimation of the properties of coloured noise. The noise might be coloured for other reasons than known effects of the receiver filter, for example co–channel interference. The model (2.2) can be generalized to

$$y(t) = B(q^{-1})u(t) + \frac{M(q^{-1})}{N(q^{-1})}v(t)$$

  which includes a (low order) ARMA model of the noise, describing other effects than the receiver filter.

- The worst possible situation confronting our algorithm (as well as any other scheme) would be severe flat fading during the training sequence. With a bad model at the beginning of Phase II, the algorithm may then diverge. Methods for avoiding this need to be studied. An increase of the lenght of the training sequence would reduce this problem significantly.

- A theoretical analysis of the algorithm might provide valuable design guidlines. A challenging complication is that the elements $-\partial\varepsilon(t)/\partial\omega_i$ of $\Psi(t)$ in (3.3) are asymptotically unbounded, when $t \to \infty$. Thus, the very powerful ODE method [9] cannot be applied directly.

- Different alternative equalizers need to be compared, to obtain the best design for Phase III.

Finally, we believe that the concept of modelling time–varying systems by estimating the parameters of deterministic functions, which describe the time–variations, can be applied fruitfully on numerous problems.

References

[1] Ahlén A and M Sternad (1989)
Optimal deconvolution based on polynomial methods. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol ASSP–37, pp 217–226.

[2] Belfiore C A and J H Park (1979)
Decision feedback equalization. *Proc. IEEE*, vol 67, pp 1143–1156.

[3] Fitch S M and L Kurz (1975)
Recursive equalization in data transmission – A design procedure and performance evaluation. *IEEE Transactions on Communication* vol COM-23, pp 546–550.

[4] Forney G D (1972)
Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory*, vol IT-18, pp 363–378.

[5] Haykin S (1986)
Adaptive Filter Theory, Prentice Hall, Englewood Cliffs, NJ.

[6] Lawrence R E and H Kaufmann (1971)
The Kalman filter for the equalization of a digital communication channel. *IEEE Transactions on Communication*, vol COM-19, pp 1137–1141.

[7] Lee W (1982)
Mobile Communication Engineering, McGraw–Hill, New York.

[8] Ljung L (1987)
System Identification. Theory for the User. Prentice–Hall, Englewood Cliffs, NJ.

[9] Ljung L and T Sderstrm (1983)
Theory and Practice of Recursive Identification. MIT Press, Cambridge, MA.

[10] Monsen P (1971)
Feedback equalization for fading dispersive channels. *IEEE Transactions on Information Theory*, vol IT–17, pp 56–64.

[11] Proakis J G (1984)
Digital Communications. McGraw–Hill, New York.

[12] Qureshi S U H (1985)
Adaptive equalization. *Proc IEEE*, vol 73, pp 1349–1387.

[13] Sderstrm T and P Stoica (1990)
System Identification. Prentice Hall International, Hemel Hempstead.

[14] Sternad M and A Ahlén (1990)
The structure and design of realizable decision feedback equalizers for IIR channels with coloured noise. *IEEE Transactions on Information Theory*, vol IT-36, no 4.

[15] Solbrand G, A Ahlén and L Ljung (1985)
Recursive methods for off–line identification. *International Journal of Control*, vol 41, pp 177–191.