Opportunities for Adaptivity in the UMTS Terminal Receiver: The Path-Search Function

Jordy Potman, Marijn Damstra, Fokke Hoeksema and Kees Slump
University of Twente, Faculty of EEMCS,
Signals and Systems Group,
P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 2773 Fax: +31 53 489 1060
E-mail: `j.potman@utwente.nl`

Abstract—**Wireless communication systems will have to become more and more flexible. In the Adaptive Wireless Networking project we try to develop such a flexible wireless communication system by developing algorithms for adaptive implementation of the digital signal processing functions in wireless communication systems. Within the limits of a wireless communication system standard there are two kinds of adaptivity: algorithm-selection adaptivity and algorithm-parameter adaptivity. In this paper we focus on algorithm-selection adaptivity in the UMTS path search function. Therefore we describe two path search algorithms: a Power Delay Profile (PDP) based algorithm and a Maximum Likelihood Estimate (MLE) based algorithm. We study and compare the performance of these algorithms under a number of multipath channel conditions by means of simulations.**

The MLE based algorithm is able to resolve a larger number of paths, especially when the paths are closely spaced, at the cost of an increased computational complexity. Therefore, using algorithm-selection adaptivity in the UMTS path search function seems to be useful. The MLE based algorithm is used when there are strong closely spaced path and the PDP based algorithm is used in all other conditions, appears to be useful. However, in order to be conclusive a few points will still have to be studied.

Keywords— **adaptive signal processing, path search, UMTS**

I. Introduction

Wireless communication systems will have to become more and more flexible for a number of reasons:
1. The number of wireless communication standards keeps increasing. Future wireless communication systems will have to support a number of these standards, for example for backwards compatibility.
2. The types of traffic that wireless communication systems have to transport cover a wide range, from relatively low-rate voice traffic to high-rate packet or multimedia data. The Quality of Service (QoS) requirements and thus the performance requirements of the wireless system that are associated with these types of traffic vary significantly. A flexible wireless communication system can thus save power by adjusting the Digital Signal Processing (DSP) it performs to the performance requirements.
3. The digital signal processing that has to be performed in wireless communication systems in order to achieve the desired performance under worst case wireless channel conditions is becoming increasingly complex. Under normal, non worst case, wireless channel conditions, simpler processing would have been sufficient and the more complex processing only leads to higher power consumption. A flexible wireless communication system can thus save power by adjusting the processing it performs to the channel conditions.

In the Adaptive Wireless Networking (AWgN) project [1] we develop such a flexible wireless communication system. The project consists of two activities. In the first activity we develop algorithms for adaptive implementation of DSP functions in wireless communication systems. This allows the wireless communication system to adjust the DSP it performs to the QoS requirements and the wireless channel conditions. Currently the focus is mainly on algorithms for adaptive implementation of the DSP functions in the Universal Mobile Telecommunication System (UMTS) terminal receiver. In the second activity the mapping of wireless communication system algorithms to a heterogeneous reconfigurable System on a Chip (SoC) architecture is studied. The reconfigurability of the SoC architecture allows the implementation of different wireless communication standards using the same hardware.

In this paper we will focus on algorithms that are suitable for an adaptive implementation of the path search function in the UMTS terminal. The paper is organized as follows: In Section II our view on adaptivity in DSP functions for wireless communication systems is given. Section III gives an overview of the

DSP functions that have to be performed in an UMTS terminal. In Section IV two algorithms for the path search function are explained in more detail and in Section V the performance of these two algorithms is compared. Finally in Section VI conclusions about the suitability of the two path search algorithms for adaptive implementation of the path search function in UMTS are given.

II. ADAPTIVITY

Wireless communication standards usually define a large number of DSP functions that have to be performed to implement that particular standard. The standards usually do not define the algorithms that have to be used to implement these functions, so the algorithms can still be chosen by the implementor of the standard. Apart from allowing products to be competitive it makes it possible to develop an implementation of the standard in which the algorithm that is used to implement a DSP function is adaptively selected based on, for example, QoS requirements or wireless channel conditions.

When a particular algorithm is selected for implementation of a DSP function, the parameters of the algorithm, if any, can still be freely selected. It is therefor also possible to adaptively select the parameter values of the algorithm based on, again, QoS requirements or wireless channel conditions.

So two kinds of adaptivity can be distinguished:

- *Algorithm-selection adaptivity*: the algorithm used to implement a DSP function is selected adaptively.
- *Algorithm-parameter adaptivity*: parameter values of an algorithm are adapted.

In this paper we focus on algorithm-selection adaptivity in the path search function of the UMTS terminal. In this paper, two algorithms to implement the path search function will be described in Section IV. But first the role of the path search function in the UMTS terminal receiver will be briefly explained in the next section.

III. UMTS TERMINAL DSP FUNCTIONS

The main processing chain of the UMTS terminal receiver roughly performs the following DSP functions [2]: receive filtering, descrambling and despreading, channel decoding and Cyclic Redundancy Check (CRC) checking, see Fig 1.

The descrambling and despreading function requires a number of supporting DSP functions: the cell search, channel estimation and path search functions. The cell search function finds the scrambling codes and timing of the cells in the neighborhood of the terminal. The channel estimation function finds the scaling and phase rotation of the received signal introduced by the wireless channel. In this paper we focus on the path search function. The path search function finds the delays and attenuations of the paths of the multipath wireless channel between base station transmitter and terminal receiver. In the next section two algorithms for the path search function will be discussed.

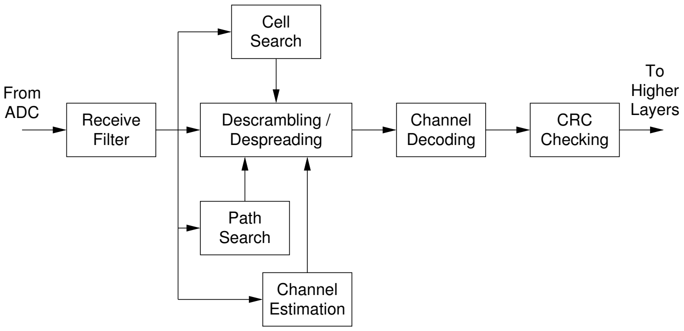

Fig. 1. UMTS terminal receiver DSP functions

IV. ALGORITHMS FOR THE UMTS PATH SEARCH FUNCTION

The path search algorithms that can be found in literature can roughly be divided in three classes: Power Delay Profile (PDP) based algorithms, Maximum Likelihood Estimate (MLE) based algorithms and subspace based algorithms [3]. In this section an algorithm from the class of PDP based algorithms and an algorithm from the class of MLE based algorithms will be described in more detail. Both algorithms make use of the fact that the transmission of the Primary Common Pilot Channel (PCPICH) in UMTS results in a transmitted signal component that only consists of the scrambling code of the base station.

A. A Power Delay Profile Based Path Search Algorithm

In [4] a PDP based path search algorithm is described. In this subsection the operation of this algorithm is summarized. Refer to Fig. 2 for an overview of the algorithm.

The multipath wireless channel will cause the arrival of multiple delayed copies of the transmitted signal at the receiver. This received signal $r(p)$, where p

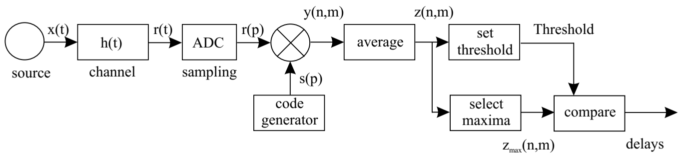

Fig. 2. Power Delay Profile based path search algorithm

is a discrete-time time index, can be expressed as

$$r(p) = \sum_{l=1}^{L} h_l(p) s(p - \frac{\tau_l}{T_c}) + w(p). \tag{1}$$

The term

$$\sum_{l=1}^{L} h_l(p) s(p - \frac{\tau_l}{T_c}) \tag{2}$$

represents the multiple copies of the scrambling code only component of the transmitted signal that arrive at the receiver. In the equations above L is the number of paths in the channel, $h_l(p)$ are the complex channel coefficients of the paths, $s(p)$ is the scrambling code, τ_l are the delays of the paths and T_c is the chip duration. The other components of the received signal, such as the user data channels, and the Additive White Gaussian Noise (AWGN) are contained in $w(p)$.

The received signal $r(p)$ is first correlated with the complex conjugate of the scrambling code $s^*(p)$

$$y(n,m) = \frac{1}{N} \sum_{p=0}^{N-1} r(nF + m + p) s^*(p), \tag{3}$$

where n is the frame index, F is the frame length, m is the correlation index, N is the correlation window size, $m \in [0, P-1]$ and P is the correlation length. The obtained correlated signal $y(n,m)$ contains a number of peaks for the different delays

$$y(n,m) = \frac{1}{N} \sum_{p=0}^{N-1} \sum_{l=1}^{L} h_l(p) \delta(m - \frac{\tau_l}{T_c}) + \tilde{w}(n,m). \tag{4}$$

Here $\delta(m - \frac{\tau_l}{T_c})$ results from the autocorrelation of the scrambling code $s(p)$ and $\tilde{w}(n,m)$ is the correlated noise.

To find a PDP $z(n,m)$ the correlated signal $y(n,m)$ is averaged with the correlated signal from the previous M frames

$$z(n,m) = \frac{1}{M} \sum_{k=0}^{M-1} |y(n-k,m)|^2 \quad m \in [0, P-1]. \tag{5}$$

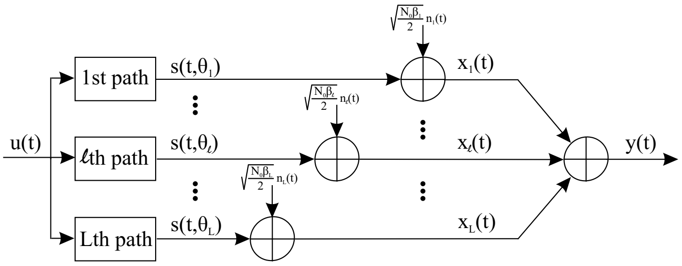

Fig. 3. Alternative view of the multipath received signal model

Local maxima in the PDP are detected by checking the height of each sample in the PDP. If it exceeds the height of both the preceding and subsequent sample a local maximum is found. A path is found when a local maximum exceeds a threshold. In [4] the threshold is set to:

$$\eta = \frac{1}{P} \sum_{m=0}^{P-1} z(n,m)(2 + \frac{4}{\sqrt{M}}). \tag{6}$$

The performance of a path searcher using a PDP based path search algorithm will be evaluated in Section V.

B. A Maximum Likelihood Estimate Based Path Search Algorithm

In [5] a MLE based path search algorithm is described. In order to be able to comprehend the operation of this algorithm as summarized in this section, it is useful to consider an alternative view of the multipath received signal model of the previous subsection.

The complex baseband signal $u(t)$ of the transmitter travels to the receiver through a channel via L paths [6], see Fig. 3. Each path contributes to the complex baseband signal $y(t)$ in the receiver. The contribution of the l^{th} path with complex gain α_l and delay τ_l is:

$$s(t, \theta_l) = \alpha_l u(t - \tau_l) \qquad \theta_l \triangleq [\alpha_l, \tau_l] \tag{7}$$

Each signal is corrupted by complex white gaussian noise $n_l(t)$, which leads to $x_l(t)$. The signal $x_l(t)$ is defined as:

$$x_l(t) = s(t, \theta_l) + \sqrt{\frac{N_0 \beta_l}{2}} n_l(t), \tag{8}$$

$$n(t) = \sum_{l=1}^{L} n_l(t) \qquad n_l(t) = \sqrt{\beta_l} n(t). \tag{9}$$

The noise $n(t)$ is divided over $x_l(t)$ by the factors β_l, with $\sum_{l=1}^{L} \beta_l = 1$. The received signal $y(t)$ is related to the signals $x_l(t)$ as follows:

$$y(t) = \sum_{l=1}^{L} x_l(t). \tag{10}$$

The complex baseband received signal $y(t)$ can therefore be described as the sum of the path signals and a complex white gaussian noise term $n(t)$:

$$y(t) = \sum_{l=1}^{L} s(t, \theta_l) + \sqrt{\frac{N_0}{2}} n(t). \tag{11}$$

Using this description a likelihood function can now be defined in terms of $y(t)$ and the path parameters $\boldsymbol{\theta} \triangleq [\theta_1, \dots, \theta_L]^T$:

$$\begin{aligned}\Lambda(\boldsymbol{\theta}; y) \;\triangleq\; & \frac{1}{N_0}\Bigg[2\int_{D_o} \Re\Big(s^*(t,\boldsymbol{\theta})y(t)\Big)dt - \\ & \int_{D_o} ||s(t,\boldsymbol{\theta})||^2 dt\Bigg]. \end{aligned} \tag{12}$$

In (12) D_o denotes the time span over which paths are searched (the correlation window), $s(t,\boldsymbol{\theta}) = \sum_{l=1}^{L} s(t, \theta_l)$. By maximizing the likelihood function the estimates of the path parameters are found:

$$\hat{\boldsymbol{\theta}}_{ML}(y) \in arg\max_{\boldsymbol{\theta}}\{\Lambda(\boldsymbol{\theta}; y)\}. \tag{13}$$

As the complexity of this calculation requires a vast amount of computational power, the likelihood function is split up into L parts (see (10)):

$$\begin{aligned}\Lambda(\theta_l; x_l) \;\triangleq\; & \frac{1}{N_0\beta_l}\Bigg[2\int_{D_o} \Re\Big(s^*(t,\theta_l)x_l(t)\Big)dt - \\ & \int_{D_o} ||s(t,\theta_l)||^2 dt\Bigg], \end{aligned} \tag{14}$$

$$\Big(\hat{\theta}_l\Big)_{ML}(x_l) \in arg\max_{\theta_l}\{\Lambda(\theta_l; x_l)\}. \tag{15}$$

The maximization (15) of the derived likelihood function (14) can be carried out by Expectation Maximization (EM). As the data $x_l(t)$ is not available, it is estimated in terms of the conditional expectation of $x_l(t)$ using $y(t)$ and a previous estimate of the channel parameters $\hat{\boldsymbol{\theta}}$. Each iteration μ of the algorithm is divided into two steps. The expectation step provides an estimate $\hat{x}_l(t)$ of $x_l(t)$ and the maximization step calculates the value of θ_l that maximizes the likelihood function.
Expectation step:

$$\hat{x}_l(t, \hat{\boldsymbol{\theta}}(\mu)) \triangleq E_{\hat{\boldsymbol{\theta}}(\mu)}\Big[x_l(t)\Big|y\Big] \qquad (l = 1, \dots, L). \tag{16}$$

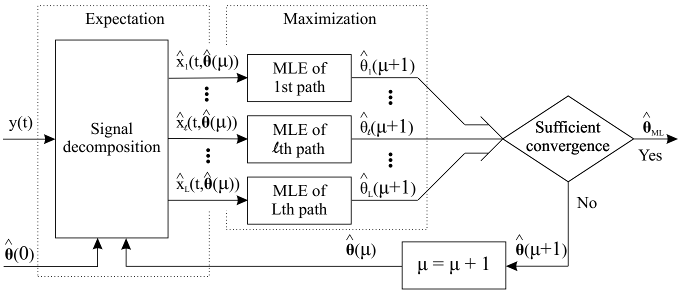

Fig. 4. MLE based path search algorithm using Expectation Maximization

Maximization step:

$$\begin{aligned}\hat{\theta}_l(\mu+1) \;=\; & arg\max_{\theta_l}\Big\{\Lambda\Big(\theta_l, \hat{x}_l\big(t, \hat{\boldsymbol{\theta}}(\mu)\big)\Big)\Big\} \\ & (l = 1, \dots, L). \end{aligned} \tag{17}$$

A general diagram of the MLE based path search algorithm using EM is shown in Fig. 4 [6]. It can be implemented by first estimating the signals $\hat{x}_l(t)$ using the estimate of $\hat{\boldsymbol{\theta}}$ from the previous iteration. This decomposition of $y(t)$ follows from (10), $\hat{\boldsymbol{\theta}}(0)$ is the initial value. The next step is to carry out L maximum likelihood estimations to find all values for $\hat{\boldsymbol{\theta}}$. If the algorithm has converged sufficiently $\hat{\boldsymbol{\theta}}_{ML}$ is assigned its value.

A more detailed description of this MLE based path search algorithm using EM can be found in [3] and the references therein. In the next section the performance of a MLE based path searcher using EM will be evaluated

V. Performance Comparison of the Path Search Algorithms

The PDP and MLE path search algorithms described in the previous section have been implemented in a UMTS physical layer simulator in order to be able to compare their performance in various multipath channel models.

Figure 5 shows the average number of path delays that the PDP and MLE path searchers estimate correctly in the Vehicular B channel model for various Signal to Noise Ratios (SNRs). See Table V for the path delays and path powers in the Vehicular B channel model. The figure shows that on average the PDP algorithm finds one path less than the MLE algorithm.

Figure 6 shows the percentage of correctly estimated paths in the Vehicular B channel model for both algorithms split out over the path delays. The figure shows that the PDP based path searcher never finds the weakest path in the channel. The PDP

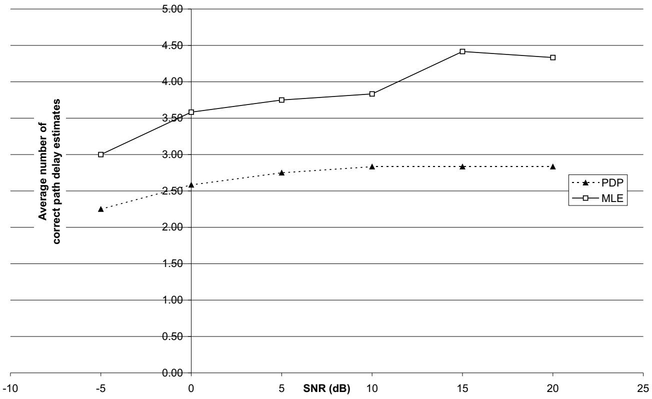

Fig. 5. Vehicular B - Average number of correctly estimated path delays (average over 12 frames; 6 paths are present)

Vehicular B	0	300	8900	12900	17100	20000	ns
	-2.5	0.0	-12.8	-10.0	-25.2	-16.0	dB

TABLE I
VEHICULAR B CHANNEL

path searcher also finds either the first or the second path in the channel, but cannot find the two paths at the same time (the sum of the detection percentages equals 100%). This explains the fact that the PDP algorithm on average finds one path less than the MLE algorithm

Figure 7 shows the average number of path delays that the PDP and MLE path searchers estimate correctly in the Office B channel model for various SNRs. See Table V for the path delays and path powers in the Office B channel model. Only the MLE algorithm benefits from the increasing Signal to Noise Ratio (SNR). On average the PDP algorithm finds about one path, whereas the MLE algorithm finds two to three paths depending on the SNR.

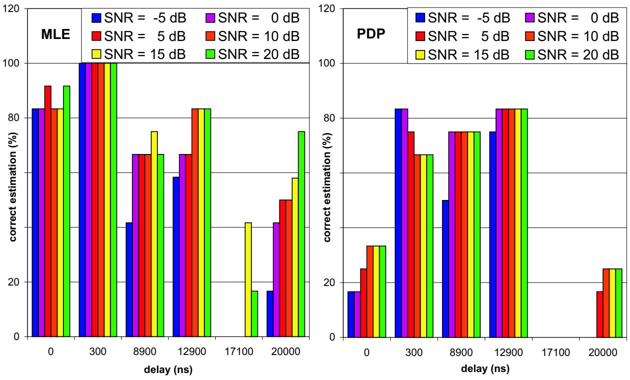

Fig. 6. Vehicular B - Percentage of correctly estimated path delays (SNR = -5 to 20 dB)

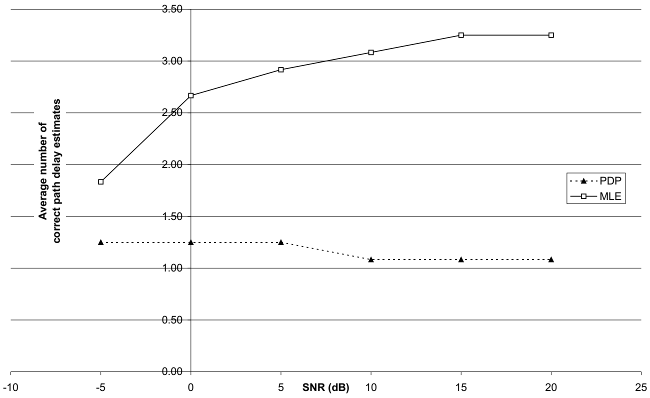

Fig. 7. Office B - Average number of correctly estimated path delays (average over 12 frames; 6 paths are present)

Office B	0	100	200	300	500	700	ns
	0.0	-3.6	-7.2	-10.8	-18.0	-25.2	dB

TABLE II
OFFICE B CHANNEL

Figure 8 shows the percentage of correctly estimated paths in the Office B channel for both algorithms split out over the path delays. It can be clearly seen that paths can only be detected when they are at least 100 ns separated in time. Both algorithms miss the paths at 100 and 300 ns. Again, the weak paths are detected better by the MLE algorithm, it does find the weak paths at 500 and 700 ns. The Office B channel clearly shows that both path search algorithms have a resolution below which they cannot separate two closely spaced paths from each other.

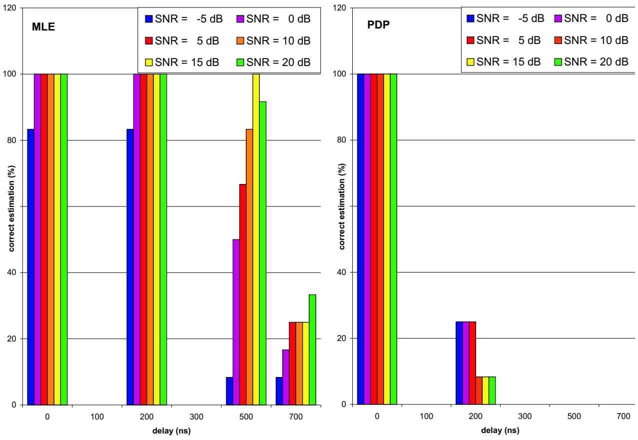

Fig. 8. Office B - Percentage of correctly estimated path delays (SNR = -5 to 20 dB)

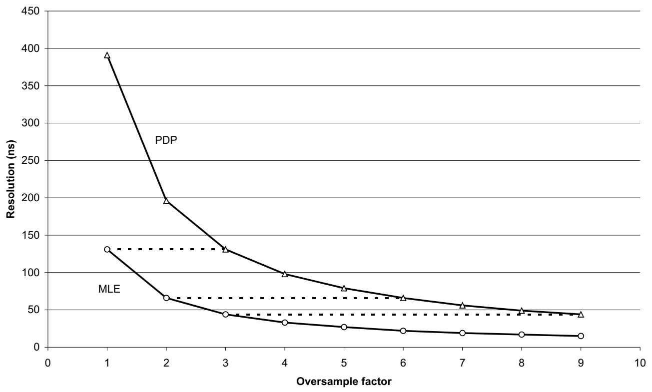

Fig. 9. Resolution as function of the oversample factor

All simulations presented thus far have been carried out with a sample rate of $1/T_c = 3.84$ MHz implying an oversampling factor of one (no oversampling). The resolution of the path search algorithms can be improved by increasing the oversample factor. Figure 9 shows the resolution of the PDP and MLE path searchers as a function of the oversample factor. It can be seen that if two paths are spaced 131 ns apart the MLE algorithm is still able to separately detect both paths with an oversample factor of one. The PDP algorithm already requires an oversample factor of three in order to be able to separate these paths.

The presented simulation results show that the MLE path search algorithm can better detect weak or closely spaced paths than the PDP based path search algorithm. Unfortunately the complexity of the MLE based path searcher is higher than the complexity of the PDP based path searcher. Figure 10 shows the ratio of the number of operations (multiply, add, etc) of the MLE and PDP algorithms, the complexity ratio, as a function of the number of paths that have to be found for various values for the maximum number of iterations μ that the MLE path searcher is allowed to perform. The figure shows that the complexity ratio increases linearly with the number of paths that have to be estimated and with the maximum number of iterations μ. Especially for larger values of μ the MLE path searcher gets excessively more complex than the PDP based path searcher.

So far all simulations of the MLE path searcher have been performed with a maximum number of iterations μ of 50. Figure 11 shows the average power estimates of the MLE for the paths in the Vehicular B channel and 15 dB SNR for various values of μ. For μ equal to ten the algorithm is not capable of finding the weakest path of the channel model at 17100 ns. For twenty iterations the power estimates differ from the power estimates for higher numbers of iterations, but all paths are detected. So the complexity of the MLE path searcher can in this case be reduced without sacrificing the performance a lot by lowering the maximum number of allowed iterations to 20.

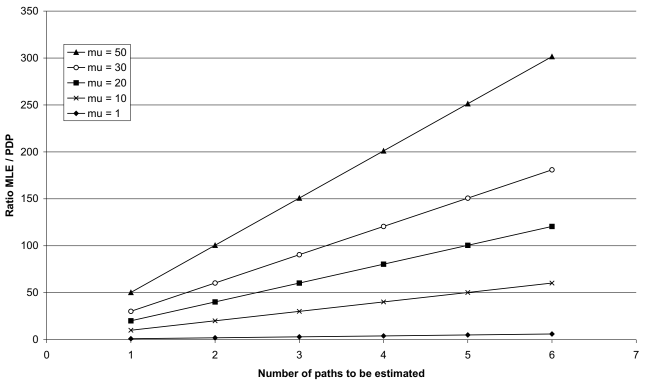

Fig. 10. Ratio of MLE/PDP complexity as function of μ ($N = 2560$, $P = 340$)

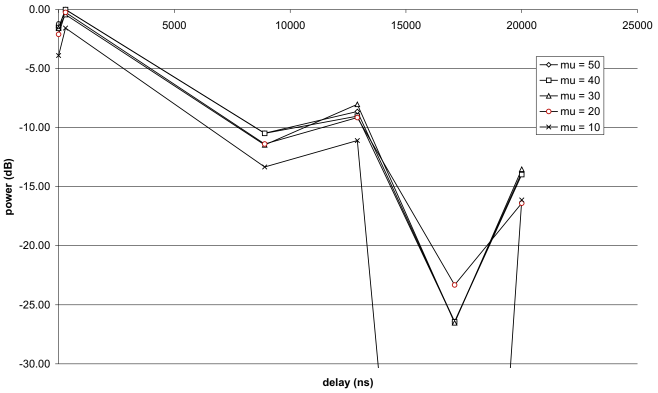

Fig. 11. Vehicular B MLE - Average power in path with correct delay estimates as function of μ (15dB SNR)

A more elaborate study of the performance and complexity of the PDP and MLE path search algorithms can be found in [3].

VI. Conclusions and Future Work

In the AWgN project we want to develop adaptive wireless communication systems. Within the limits of a wireless communication standard we distinguish two kinds of adaptivity: algorithm-selection adaptivity and algorithm-parameter adaptivity. In this paper we have studied two algorithms that can possibly be used for algorithm-selection adaptivity in the UMTS path search function: a Power Delay Profile (PDP) based algorithm and a Maximum Likelihood Estimate (MLE) based algorithm.

The MLE based path search algorithm on average detects more paths than the PDP based path search algorithm. This is due to the fact that the MLE based path searcher can detect weak paths better than the PDP based path searcher, as can be clearly seen in the Office B channel simulations of Fig. 8.

The MLE based path searcher is also better at resolving closely spaced paths. The MLE based path searcher can detect paths that are a spaced a factor three closer to each other than the PDP based path searcher for the same oversample factor as can be seen in Fig. 9.

The better performance of the MLE based path searcher comes at the cost of an increased computational complexity. The relative computational complexity of the MLE path searcher to the PDP path searcher increases linearly with the number of paths L that have to be resolved and the maximum number of iterations μ that the MLE algorithm is allowed to perform. By keeping the value of μ relatively small (say 20) the complexity of the MLE path searcher can be kept within limits.

The PDP and MLE based path search algorithms can be suitable candidates for algorithm-selection adaptivity in the UMTS path search function. From our research it can be concluded that the MLE based path search algorithm has to be used in channels with strong closely spaced paths. In channels with weak closely spaced paths the additional paths that the MLE algorithm will find will be weak and therefore will not contribute a lot to the bit error rate performance of the receiver. So in all other conditions the PDP algorithm has to be used, because the(average over 12 frames; 6 paths are present) computational complexity of the MLE algorithm is too high to justify the relatively small improvement in performance.

However in order to be conclusive a few points will have still to be studied:

- While we know what the performance of the PDP and MLE path searchers is in terms of number of resolved paths, we have not yet studied the influence of the number of resolved paths on the bit error rate performance.
- The detection of closely spaced paths by the PDP algorithm can be improved by using the Teager-Kaiser operation [7].
- There are some alternatives to the MLE algorithm, such as the SAGE algorithm [6], that have a reduced computational complexity.

In our future work we will address these issues and focus on the measurement and control functions that are required in a path searcher implementation that switches between the PDP and MLE based path search algorithms.

ACKNOWLEDGMENT

This research is supported by the Freeband Knowledge Impulse program, a joint initiative of the Dutch Ministry of Economic Affairs, knowledge institutions and industry.

REFERENCES

[1] Freeband awgn project. `http://www.freeband.nl/kennisimpuls/projecten/awgn/ENindex.html`.

[2] R. Tanner and J. Woodard, editors. *WCDMA Requirements and Practical Design*. John Wiley & Sons, 2004.

[3] M.C. Damstra. Path search algorithms for application in w-cdma systems. Master's thesis, University of Twente, 2004.

[4] K.C. Gan. Path searcher for a wcdma rake receiver. *Motorola application note*, vol. 1(no. AN2252/D):pp. 1–17, March 2002.

[5] D. Dahlhaus et al. A sequential algorithm for joint parameter estimation and multiuser detection in ds/cdma systems with multipath propagation. *Wireless personal communications: an international journal*, vol. 6(no. 1/2):pp. 161–178, January 1998.

[6] B.H. Fleury et al. Channel parameter estimation in mobile radio environments using the SAGE algorithm. *IEEE journal on selected areas in communications*, vol. 17(no. 3):pp. 434–450, March 1999.

[7] R. Hamila et al. Subchip multipath delay estimation for downlink wcdma system based on teager-kaiser operator. *IEEE communications letters*, vol. 7(no. 1):pp. 1–3, January 2003.