On MIMO Decoding Algorithms for UMTS

Markus Rupp[*], Suman Das[†], Maxime Guillaud[‡]
[*]Inst. of Comm. & Radio Frequency Engin., TU Wien, Vienna, Austria, mrupp@nt.tuwien.ac.at
[†]Wireless Research Laboratory, Lucent Technologies, NJ, sumand@lucent.com
[‡]Institute EURECOM, France, maxime.guillaud@eurecom.fr

Abstract

With the recent understanding of the channel capacity in wireless systems that allow much higher data rates in high scattering environments, transmission over multiple antenna systems has become an attractive option. In this paper we present candidate algorithms for decoding multiple-input-multiple-output (MIMO) systems and study their behavior and implementation for a 4-by-4 antenna system in a UMTS environment.

1 Introduction

After it was shown that the scattering wireless channel has a much larger capacity than expected, the first decoding algorithms for a MIMO system were proposed in [1]. In a recent effort (see companion paper[2]), a prototype for a UMTS system with four transmit and four receive antennas was designed, allowing to exploit four times the capacity of a standard UMTS system per user code.

The 4-by-4 MIMO system requires to measure 16 channels. For each channel, four fingers (corresponding to four significant multipath components) are selected summing up to 64 observations for four transmitted symbols. After decorrelation, the received signals can be stored in a 64×4 matrix $\mathbf{H}(k)$ for the channel information and in a 4×1 vector $\mathbf{s}(k)$ for the transmitted QPSK data elements. The observation vector is thus given by:

$$\mathbf{r}(k) = \mathbf{H}(k)\mathbf{s}(k) + \mathbf{v}(k). \tag{1}$$

2 MIMO Decoding Algorithms

The following MIMO decoding algorithms have been realized in TI-C67 DSP to decode the symbols in (1) and compared:

- ZF receiver:

$$\hat{\mathbf{s}}(k) = Q[(\mathbf{H}^*(k)\mathbf{H}(k))^{-1}\mathbf{H}^*(k)\mathbf{r}(k)].$$

- ZF-BLAST (also V-BLAST[3, 4]) receiver: successively using LS with matrix $\mathbf{H}(k)$ of decreasing dimension, based on the smallest row norm of $(\mathbf{H}^*(k)\mathbf{H}(k))^{-1}\mathbf{H}^*(k)$. Note that such an iterative method is also known in literature as down-dating (see, for example, chapter 2.7 in[5]).

- MMSE receiver:

$$\hat{\mathbf{s}}(k) = Q[(\mathbf{H}^*(k)\mathbf{H}(k) + \sigma_v^2\mathbf{I})^{-1}\mathbf{H}^*(k)\mathbf{r}(k)].$$

- MMSE-BLAST receiver: successively using MMSE with matrix $\mathbf{H}(k)$ of decreasing dimension, based on the smallest row norm of $(\mathbf{H}^*(k)\mathbf{H}(k) + \sigma_v^2\mathbf{I})^{-1}\mathbf{H}^*(k)$. Note that this condition is equivalent to the condition in the V-BLAST algorithm for $\sigma_v^2 = 0$.

- ML receiver: Full search over all possible constellations[6, 7].

- Iterative cancellation algorithm: Starting with an initial estimate $\hat{\mathbf{s}}(k, 0)$, the estimate is iteratively improved. Details follow in the next section.

3 Analysis of Iterative Cancellation Algorithm

Assume the 64×4 channel matrix $\mathbf{H}(k)$, defining the observation vector $\mathbf{r}(k)$:

$$\mathbf{r}(k) = \mathbf{H}(k)\mathbf{s}(k) + \mathbf{v}(k) \tag{2}$$

Premultiplying with $\mathbf{H}^*(k)$ allows subtracting the entries by using the following approach:

$$\hat{\mathbf{r}}(k) = \alpha\mathbf{H}^*(k)\mathbf{r}(k) - \alpha\mathbf{H}^*(k)\mathbf{H}(k)\hat{\mathbf{s}}(k) + \mathbf{D}(k)\hat{\mathbf{s}}(k) \tag{3}$$

$\mathbf{D}(k)$ being the diagonal elements of $\mathbf{H}^*(k)\mathbf{H}(k)$. Note that for correct values of $\hat{\mathbf{s}}(k)$ and $\alpha = 1$, $\hat{\mathbf{r}}(k) = \mathbf{D}(k)\mathbf{s}(k)$ remains. The iterative decoding algorithm then applies a nonlinear operation $Q[x]$ to map the filtered values into the nearest values in the set of transmitted symbols. The full iteration step thus reads:

$$\begin{aligned}\hat{\mathbf{s}}(k,l+1) &= Q[\alpha\mathbf{H}^*(k)\mathbf{r}(k) - \alpha\mathbf{H}^*(k)\mathbf{H}(k)\hat{\mathbf{s}}(k,l) \\ &\quad +\mathbf{D}(k)\hat{\mathbf{s}}(k,l)]. \qquad (4)\end{aligned}$$

For arguments close to the mapped values of $Q[x]$, let us assume a linear approximation, only valid in the vicinity of those values and we can re-formulate the equation by:

$$\begin{aligned}\hat{\mathbf{s}}(k,l+1) &= \alpha\mathbf{D}^{-1}(k)\mathbf{H}^*(k)\mathbf{r}(k) \qquad (5)\\ &\quad -\alpha\mathbf{D}^{-1}(k)\mathbf{H}^*(k)\mathbf{H}(k)\hat{\mathbf{s}}(k,l) + \hat{\mathbf{s}}(k,l).\end{aligned}$$

This update equation can be rewritten to the following form, displaying the transient behavior of the symbol errors:

$$\begin{aligned}\mathbf{s}(k) - \hat{\mathbf{s}}(k,l+1) &= \left[\mathbf{I} - \alpha\mathbf{D}^{-1}(k)\mathbf{H}^*(k)\mathbf{H}(k)\right] \\ &\quad \times[\mathbf{s}(k) - \hat{\mathbf{s}}(k,l)] \\ &\quad -\alpha\mathbf{D}^{-1}(k)\mathbf{H}^*(k)\mathbf{v}(k). \qquad (6)\end{aligned}$$

The stability condition can now be found immediately from this equation. The following eigenvalues all need to be upper-bounded by one:

$$\left|\mathrm{eig}\left[\mathbf{I} - \alpha\mathbf{D}^{-1}(k)\mathbf{H}^*(k)\mathbf{H}(k)\right]\right| < 1. \qquad (7)$$

By varying α the influence of the eigenvalues can be controlled as well as the noise level. For smaller values of α eigenvalues from outside of the unit circle will move towards the inside but also small eigenvalues can move from the inside toward the unit circle preventing fast convergence. Thus, a small step-size strategy requires many iterations.

A compromise can be a decaying step-size as it is common to gradient-type algorithms. In every step

$$\alpha(l) = \beta\alpha(l-1),$$

with $\beta < 1$. Figure 1 depicts the result for one value out of the time-variant channel as described in the following section when run with the iterative cancellation algorithm for various values of $\alpha(0) = \beta$.

4 Comparison

In the following simulations, two channels types have been investigated: a time-invariant channel with matrix $\mathbf{H}(k) = \mathbf{H} = \mathbf{I} + \mathbf{E}$, $\mathbf{E}$ being an all-one matrix, and a time-variant matrix $\mathbf{H}(k)$ whose entries were all

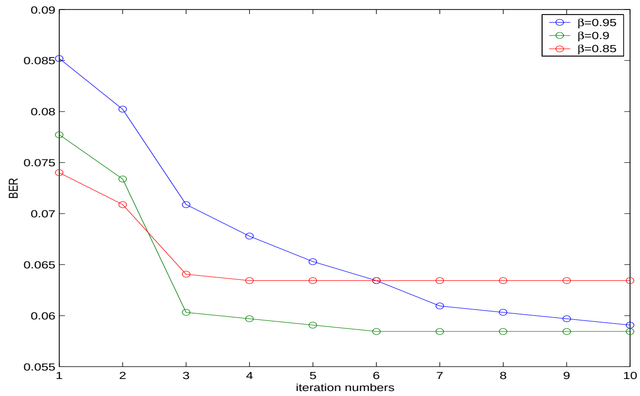

Figure 1: *Iterative cancellation algorithm over number of runs.*

independent Rayleigh fading processes with identical energy and a Doppler speed of 10kmph. Figures 2-4 depict the results on the static channel for ZF,ML and iterative cancellation receiver, respectively. The theoretical expected BER can be evaluated by classical methods and is shown for comparison. The plotted curves are shown for spreading factors 4 to 256. A very good agreement with theory has been achieved for all algorithms (only three are shown in the figures). Typically for static or almost static behavior where channel estimation can run over a longer period to improve the channel estimation quality, the differences between the best behavior (ML) and the worst (ZF) are in a range of 3dB with the BLAST-algorithms in the middle. Knowing the noise level does not show clear improvements as would be expected for MMSE schemes. The corrupted nature of the channel estimates seems to hurdle these types of algorithms. Note that the CDMA receiver front end delivers channel information on four fingers for each sub-channel. Applying low spreading factors can lead to substantial noise when the fingers do not carry information. In order to show agreement with the theoretical lines, only the first finger, matching the channel position was taken.

Figures 5 and 6 depict the results for ZF-BLAST and ML in the time-variant environment. In this simulation run, the channel was chosen to be flat but each sub-channel moves independently with a Jakes model fading. The Doppler speed was set to 10kmph, thus slow enough to track the channel movements. Assume the channel to be of the following structure:

$$H(k) = \begin{bmatrix} \alpha_{11}(k) & \alpha_{12}(k) & \alpha_{13}(k) & \alpha_{14}(k) \\ \alpha_{21}(k) & \alpha_{22}(k) & \alpha_{23}(k) & \alpha_{24}(k) \\ \alpha_{31}(k) & \alpha_{32}(k) & \alpha_{33}(k) & \alpha_{34}(k) \\ \alpha_{41}(k) & \alpha_{42}(k) & \alpha_{43}(k) & \alpha_{44}(k) \end{bmatrix} \qquad (8)$$

Each pdf is identical and of the form:

$$f_\alpha(\alpha) = \frac{\alpha}{\sigma^2} e^{-\frac{\alpha^2}{2\sigma^2}} U(\alpha), \tag{9}$$

with σ the variance of each Jakes fader and $U(\dot{)}$ the step function. Computing the BER is equivalent to solving the following integral:

$$\begin{aligned} \text{BER} &= 0.5 \int_0^\infty \text{erfc}\left(\sqrt{\frac{0.5}{\sigma_v^2 tr([H^*(k)H(k)]^{-1})}}\right) \\ &\quad \times \prod_{i=1}^{16} f_{\alpha_i}(\alpha_i) d\alpha_i. \end{aligned} \tag{10}$$

Note that the elements of $H^*(k)H(k)$ are positive real-valued on the main diagonal and complex-valued elsewhere. The integration over the non-diagonal elements is expected to add up to zero (or close to zero) while the diagonal terms have a much stronger contribution. Taking only the diagonal terms into account, the above expression can be evaluated to

$$\begin{aligned} \text{BER} &= 0.5 \int_0^\infty \text{erfc}\left(\sqrt{\frac{0.5}{\sigma_v^2 \frac{4}{\sum_{i=1}^4 \alpha_i}}}\right) \prod_{i=1}^{4} f_{\alpha_i}(\alpha_i) d\alpha_i. \\ &= 0.5 \int_0^\infty \text{erfc}\left(\sqrt{\frac{0.5 \sum_{i=1}^4 \alpha_i}{4\sigma_v^2}}\right) \prod_{i=1}^{4} f_{\alpha_i}(\alpha_i) d\alpha_i. \end{aligned} \tag{11}$$

Thus, theoretically, a four times diversity improvement is expected. Only the ML algorithm achieves this behavior, all other algorithms show a very poor performance, the ZF-BLAST algorithm being one of the better performers. Figure7 displays a comparisons of all algorithms when applying spreading factor 32.

Another channel situation was examined, in which all four fingers carry information, one chip-spaced in distance and the power profile being in the ratio [8:4:2:1]. The Doppler speed for each channel component was set to 10kmph. It can be shown that for identical power on all paths, a 16-times diversity can be expected. With this particular power profile, however, the 16-times diversity is a lower bound and the four times diversity (from the first finger only) an upper bound. The following Figure8 displays the behavior of the various algorithms using spreading factor 32.

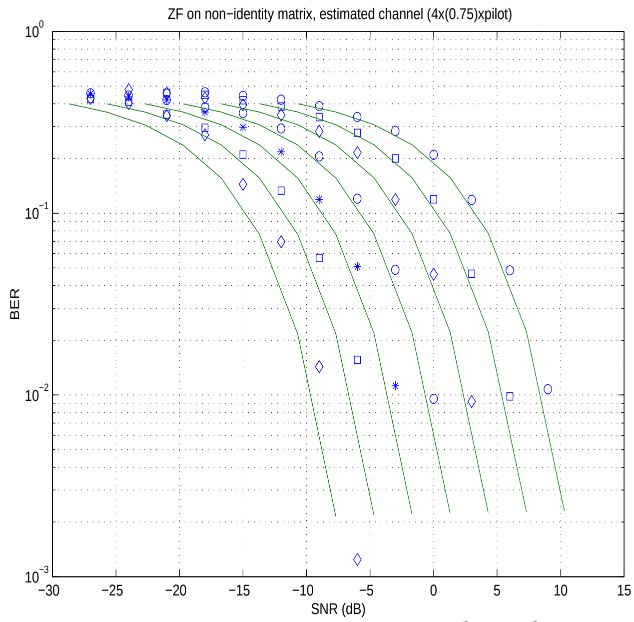

Figure 2: *ZF receiver on static channel.*

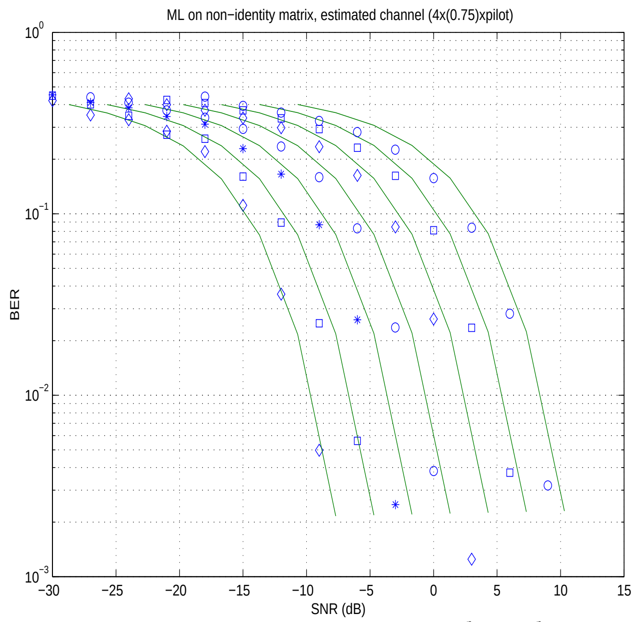

Figure 3: *ML receiver on static channel.*

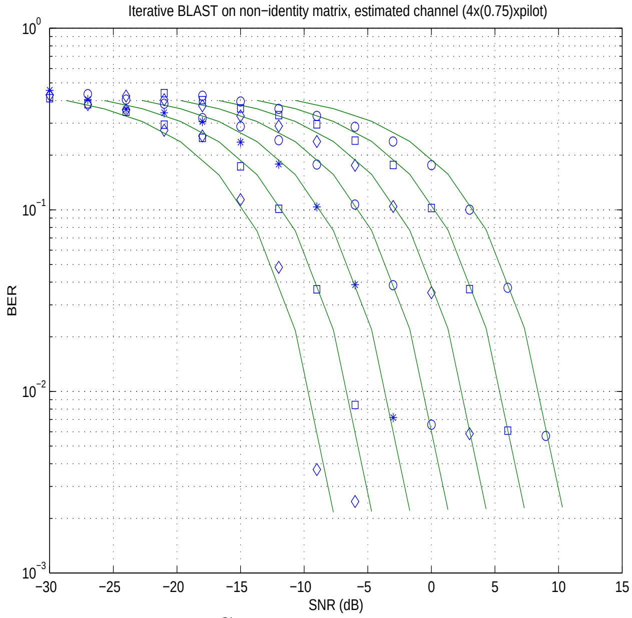

Figure 4: *Iterative Cancellation receiver on static channel.*

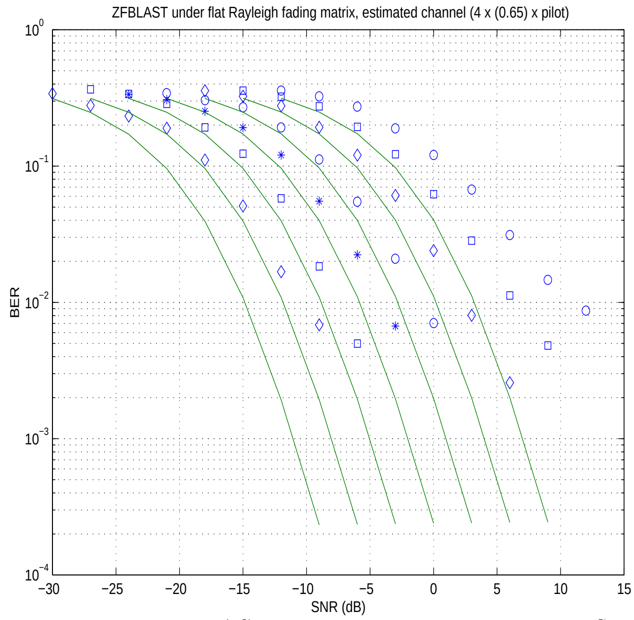

Figure 6: *ZF-BLAST receiver on time-variant flat channel.*

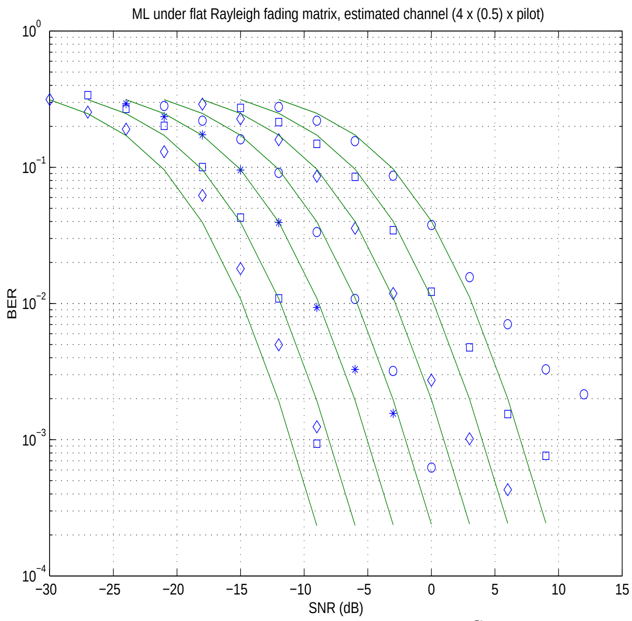

Figure 5: *ML receiver on time-variant flat channel.*

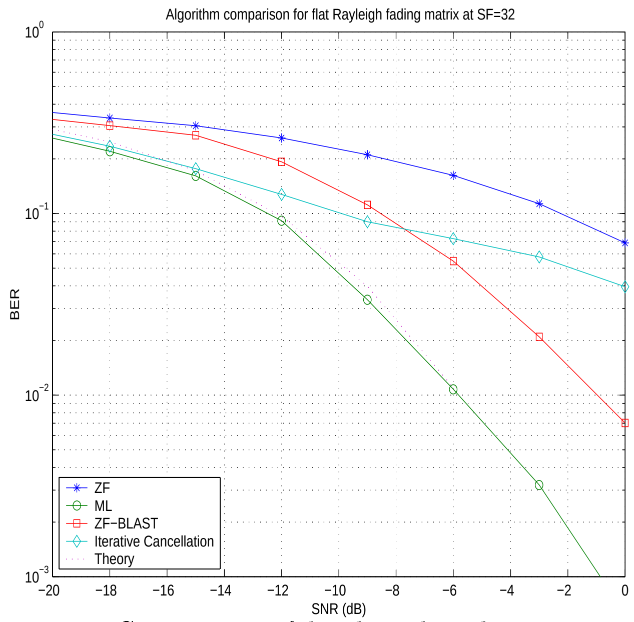

Figure 7: *Comparisons of decoding algorithms on time-variant flat channel with spreading factor 32.*

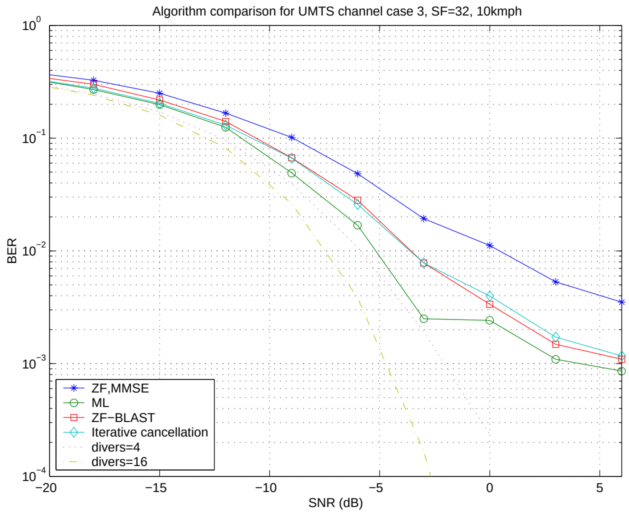

Figure 8: *Comparisons of decoding algorithms on time-variant dispersive channel with spreading factor 32.*

5 Implementation

Each algorithm is implemented and optimized to run on a TI-C67 floating point DSP. Details of the complete system are presented in the companion paper[2]. The channel estimation is performed on a set of four pilot symbols, i.e., 266μs. All algorithms have the nice property that some part is dependent on the channel information while the remaining part is not. In other words, every time the new channel information comes in, the first part needs to be updated. However, the decoding part can be applied as often as the channel information remains constant. For a spreading factor of four, for example, the decoding part can be applied 256 times before the channel is updated again. This reduces the complexity of the algorithms substantially. Furthermore, if it is known that the channel is static (or quasi-static), the channel updates can run on a much slower scale, leaving basically the decoding complexity. In the following table1, the supported spreading factors for the algorithms running in real-time (assuming only the decoding part) are listed. The (*) at ML means that the performance for spreading factor 64 could only be achieved by applying fix-point C-types (short) and operations. Fortunately, ML does not require a matrix inverse and is therefore very robust in fix-point.

algorithm	spreading factors
ZF,MMSE	32,64,128,256
ML	64*,128,256
ZF-,MMSE-BLAST	64,128,256
Iterative Cancellation	64,128,256

Table 1: Supported spreading factors in real time on 167MHz C67

6 Conclusion

Various decoding strategies for MIMO decoding in a wideband CDMA (UMTS) scenario were presented. While on static channels all algorithms show roughly the same behavior, on time-varying channels, the algorithms perform distinctly different. ML is always the best performer and does not suffer of too high complexity. Due to its numerical robustness, it can be implemented in short types with fix-point operations and thus saving complexity.

Acknowledgement

The authors would like to thank Veronique Batter for her devotion in writing DSP code and Dragan Samardzija for his many helpful comments.

References

[1] G.J.Foschini, M.J.Gans, "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas", Wireless Personal Communications, No. 6, 1998, pp. 315-335.

[2] M.Guillaud, S.Das, A.Burg, M.Rupp, E.Beck, "Rapid Prototyping Design of a 4×4 BLAST-over-UMTS Systems, submitted to Asilomar Conference 2001.

[3] P.W.Wolniansky, G.J.Foschini, G.D.Golden, R.A. Valenzuela, "V-BLAST: An architecture for achieving very high data rates over rich-scattering wireless channels," in Proc. ISSSE-98, Pisa, Italy.

[4] G.D.Golden, G.J.Foschini, R.A.Valenzuela, P.W. Wolniansky, "Detection Algorithm and Initial Laboratory Results Using V-BLAST Space-Time Communication Architecture", Electronics Letters, Vol. 35, No. 1, pp. 11-14, Jan. 1999.

[5] T.Kailath, A.H.Sayed, B.Hassibi, Linear Estimation, Prentice Hall, 1999.

[6] R.Van Nee, A.van Zelst, G.Awater, "Maximum Likelihood decoding in a space division multiplexing system," VTC Japan, May 15-18, 2000.

[7] G.Awater, A.van Zelst and R.van Nee, "Reduced Complexity Space Division Multiplexing Receivers," VTC Japan, May 15-18, 2000.