A FLEXIBLE MIMO TESTBED WITH REMOTE ACCESS

Christian Mehlführer, Stefan Geirhofer, Sebastian Caban, and Markus Rupp

Institute of Communications and RF Engineering
Vienna University of Technology
Gusshausstr. 25/389, 1040 Wien, Austria
Email: {chmehl, sgeirhof, scaban, mrupp}@nt.tuwien.ac.at
Web: **http://www.nt.tuwien.ac.at/rapid_prototyping**

ABSTRACT

While wireless systems with multiple transmit and receive antennas have been thoroughly investigated over the past ten years, experimental work to corroborate theoretical results is lagging behind. A major reason for this situation is given by the fact that strong theoretical researchers often have no expertise in implementation issues. Also, building an experimental system is very time-consuming and still it is typically focussed on one transmission mode only. By the time the experimental setup is ready, hundreds of new ideas already exist that cannot be handled by the just finished setup. We thus propose a flexible MIMO testbed that allows to connect directly onto MATLAB code and thus shortens experimental times considerably. A few examples for UMTS HSDPA and for Space-Time Codes strengthen our approach.

1. INTRODUCTION

With the theoretical understanding of the nature of multiple antenna systems in a scattering environment by Foschini, Gans [1] and Telatar [2], an enormous potential for high spectral efficiency was found. However, now, more than 10 years later, still Multiple-Input, Multiple-Output (MIMO) products are missing. It is thus not simply adding a few more transmitters and receivers in order to take advantage of this potential. Many real-time experiments including physical, wireless channels are needed to understand the true advantages. Since our knowledge of wireless channels in particular for MIMO transmissions is rather poor, simulations including mathematical channel models cannot give us final answers. Simulation methods based on MATLAB and C-programming often simplify the transmission scenarios and the true problems become hidden. For example, the impact of nonlinear power amplifiers which is very typical for WCDMA as well as OFDM with their very high Crest factors is typically neglected in simulations. Also quantization effects of the AD and DA converters are not included. Simulations are typically based on floating-point (double) precision while products have to work on low cost 16 bit fixed-point processors or on dedicated HW with even lower precision.

It is thus of utmost importance to come up with real-time experiments including physical wireless channels and equipment much like the one used later in products in order to identify potential *show-stoppers* as early as possible and de-risk emerging new technologies. In this paper we present such a flexible testbed suitable for real-time experiments. In Section 2 we give a short overview of the hardware involved. Section 3 then wraps up the design methodology utilizing a MATLAB interface to the hardware that allows for quickly adapting MATLAB simulation code unto the RF front end. And finally Section 4 presents a few measurement results obtained by this testbed, concerning the UMTS HSDPA mode and Space-Time Codes for MISO transmissions.

This work has been funded by the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms.

2. MIMO PLATFORM

The main idea of our MIMO testbed (Fig. 1) is the so-called MATLAB INTERFACE, a powerful, easy to use, and still flexible interface to the MIMO testbed hardware. Using this interface, neither hardware programming skills nor a lot of time is needed to transmit and receive complex baseband data samples directly out of MATLAB via a radio frequency (RF) air interface from any PC in the local area network (LAN). This allows for a great variety of real-time experiments with minimum effort.

- Utilizing MATLAB and optional DSP+FPGA boards, the testbed user creates complex digital baseband data samples (2×14 bit) for up to four transmit antennas on his own PC. These data samples and a set of transmit options (e.g. the desired IF frequency) are then transferred to the transmit PC via the MATLAB INTERFACE. Note that the data transfer via the MATLAB INTERFACE does not require real-time ability since large amounts of transmit data can be stored on the internal harddisk of the transmit PC. The stored data samples can be transmitted in real-time at a certain instant.
- In the transmit PC, the four streams of digital baseband data samples are buffered, interpolated, digitally upconverted to the desired intermediate frequency (IF), and at last converted to the analog domain by four 14 bit 200 MHz DACs.
- IF filtering, analog upconversion to the 2.45 GHz ISM band, and power amplification are carried out in the next step.
- The channel may either be
  - a physical cannel operated by eight $\lambda/4$-monopole antennas at 2.45 GHz
  - or channel emulators (Spirent TAS4500 FLEX), allowing for repeatable experiments with a wide range of definable channel parameters.
- Low noise amplification, RF preselection, analog RF to IF downconversion, and IF filtering are carried out in the next step.

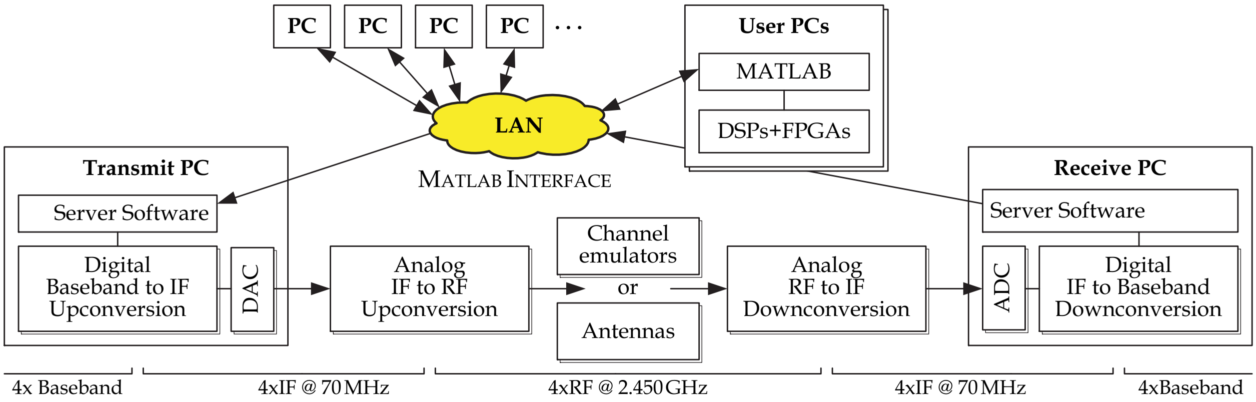

Figure 1: Measurement setup.

- After downconversion a Noise/Com UFX-EbNo noise generator can be used to impair the receive signal with additive white Gaussian noise, allowing for accurate BER measurements over a large SNR range.
- The receive PC converts the IF signals to the analog domain with a resolution of 14 bit at a sampling rate of 100 MHz. Digitally downconverted data is provided back to the user in form of complex digital baseband data samples (2×24 bit).
- These unfiltered, unsynchronized raw data samples are finally processed by the user on his own PC in MATLAB and/or optional DSP+FPGA boards.
- In order to obtain measurement results within a short period of time, cluster processing techniques were used for evaluating the performance of the different transmission systems presented in the following.

3. RAPID PROTOTYPING METHODOLOGY

The MATLAB INTERFACE, integrating the testbed into MATLAB, allows for extensive use of rapid prototyping methods. After simulating an algorithm in MATLAB, the existing code can be reused to measure the performance over physical radio channels. However, some extensions to the algorithm have to be made to adapt it to the needs of the testbed. These extensions include a pulse shaping filter at the transmitter side, and digital sampling time synchronization at the receiver side.

The efficient implementation of digital sampling time synchronization can be a real challenge and would need too much time, especially when sampling time synchronization is not the main focus of the research. Thus, we implemented a very practicable sampling time synchronization which interpolates the receive signal and correlates this interpolated signal with a training sequence. This implementation allows for adjusting the accuracy of the synchronization with the interpolation rate. Additionally, the investigation of inadequate sampling time synchronization and its consequences on the BER is possible. Detailed information about the implemented synchronization can be obtained from [3].

Once the synchronization works sufficiently well, measurements via IF and RF can be carried out and compared to reveal other effects stemming from the RF hardware (e.g. intermodulation). In addition to the testing of transmitter and receiver algorithms, the requirements for the RF hardware of the transmission systems can be determined by measuring the performance of the overall system.

After verifying the correct functionality of the algorithm by measurements with the testbed, a further step in the algorithm development process is the realization of the algorithm in a high level language like C. The flexibility of MATLAB allows to integrate this C realizations easily into the existing MATLAB code by the use of MEX-functions. The investigation of fixed-point precision issues can then be performed efficiently by comparing the double precision MATLAB measurement results with the results of the C functions.

Once the C realization of the algorithm has been successfully tested, it can be used to generate code for the above mentioned DSP boards. Running the code on a DSP allows for accurate measurements of the computational complexity of the algorithm.

4. DESIGN EXAMPLES

In this section several experiments using our MIMO testbed are outlined, and BER measurement results are presented. The results were obtained by MATLAB implementations of the algorithms and measurements via the radio frequency frontend and the channel emulators.

4.1 SISO Example

Before utilizing the testbed for different measurement tasks, its correct operation had to be verified extensively. Specifically, we considered the W-CDMA setup as in the following HSDPA example (Section 4.2). The channel emulators in the block diagram in Fig. 1 were configured to an AWGN channel and a flat (1-tap) Rayleigh fading channel, respectively. The analytical results for both cases were taken as a reference.

In Fig. 2 the measurement results for both scenarios are depicted showing an almost perfect fit with the analytical results. The largest abberation occurs at $E_b/N_0 = 13$ dB and amounts to 0.15 dB. It should be noted that large amounts of data had to be transmitted in order to arrive at this degree of accuracy. For the flat Rayleigh fading channel in particular, we need to average over the channel's statistic arriving at over 10 GB of raw receive data. Cluster processing techniques had to be used in order to reduce the computation time associated with the measurements.

It should also be stressed that it is necessary to impose

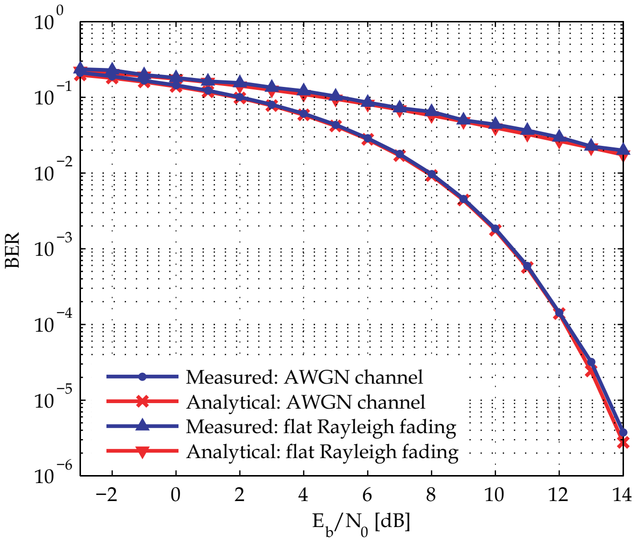

Figure 2: Measurement verification

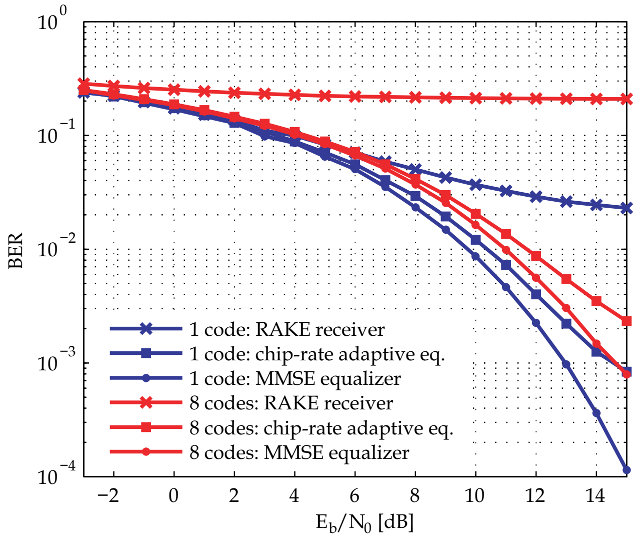

Figure 3: HSDPA example

stringent requirements on the RF-equipment and to carefully configure the transmission chain. Specifically, intermodulation has to be circumvented, and the power levels at all intermediate steps need to be carefully chosen.

4.2 HSDPA Example

The testbed has been used to measure the performance of different equalization schemes for the High Speed Downlink Packet Access (HSDPA) of UMTS. The fact that the analysis was based on *measurements* enabled us to account for many design aspects just as they would appear in a real implementation [4]. In particular, obtaining accurate synchronization and estimating the channel requires a careful setup of the measurement system and are thus challenging tasks.

In the scope of this paper it is only possible to give a brief summary of our analysis; a thorough treatment of the topic is given in [5]. Specifically, our study focused on finding appropriate schemes augmenting or replacing the conventional RAKE receiver as the standard receiver. It turned out that equalization is indeed able to ameliorate the performance significantly by reducing the interference that is generated from other user channels.

The testbed's MATLAB INTERFACE already described in Section 3 significantly accelerated the time needed for implementation and facilitated the verification of the algorithms. Additionally, for the chip-rate adaptive scheme the MEX-functionality of MATLAB had to be used to speed up the processing time (loop structures can not be omitted for this method). The C-implementation could be tightly incorporated into the measurement environment decreasing the runtime by a factor of 20-30.

When implementing an equalizer, it is indispensable to strike a balance between performance and complexity, as to allow for a real-time implementation. In our analysis, we considered MMSE equalization (as to get an impression on what can be achieved) and compared these results to adaptive techniques with little complexity.

The tradeoff between performance and complexity is illustrated in Fig. 3. We demonstrate that equalization indeed improves the BER (and thus the system performance) significantly by more than one decade in BER. Additionally, the chip-rate adaptive scheme [6] shows a performance similar to the MMSE equalizer by using only a fraction of the computational load. Adaptive equalization thus represents a very promising tradeoff between performance and complexity.

4.3 MISO Space-Time Codes

The first measurements utilizing more than one antenna on each side of the transmission system are measurements of MISO Space-Time block codes, namely the Alamouti Code (AC) [7] for a 2×1 system and the Extended Alamouti Code (EAC) [8] for a 4×1 system. Such transmit diversity schemes are very promising for the downlink channel of mobile communication systems because the costly multiple RF paths are located within the base stations.

In [8] it is shown that a feedback of channel state information (CSI) from the receiver to the transmitter can greatly increase the performance of the EAC. When using only two bit feedback (FB) per transmit block a diversity order of four can be reached in flat fading Rayleigh channels. The simulated BER curves shown in Fig. 4 were generated using a block fading channel model. For the measurements the channel emulators were used and the Doppler frequency was set to 5 Hz. This Doppler frequency in combination with short block lengths (approximately 0.4 ms at a symbol rate of 4.17 MSymbols/s) ensures a nearly constant channel during the transmission of one data block. Additionally, to obtain many different channel realizations a 10 ms gap between the transmission blocks is introduced, greatly reducing the number of data blocks to be transmitted. In contrast to the HSDPA example the Space-Time coded transmit signals employ RRC filtered (rolloff= 0.22), unspread Gray coded QPSK data symbols.

The measurement result for the EAC using the low-complexity Maximum Likelihood (ML) receiver of [9] is illustrated in Fig. 4. The three black solid lines denote perfect $1\times$, $2\times$, and $4\times$ diversity and are drawn for comparison reasons. Furthermore, the BER measurement result for the 2×1 system using the Alamouti code is depicted. The resulting BER curve perfectly matches the 2×1 diversity curve. A comparison between the AC and the EAC without feedback information reveals that the increase in diversity is only mar-

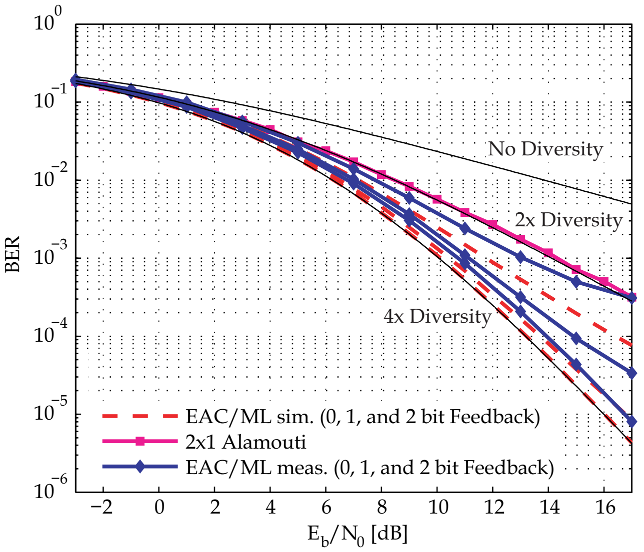

Figure 4: Comparison AC/EAC, measured and simulated (QAM, 4.17 MSymbols/s)

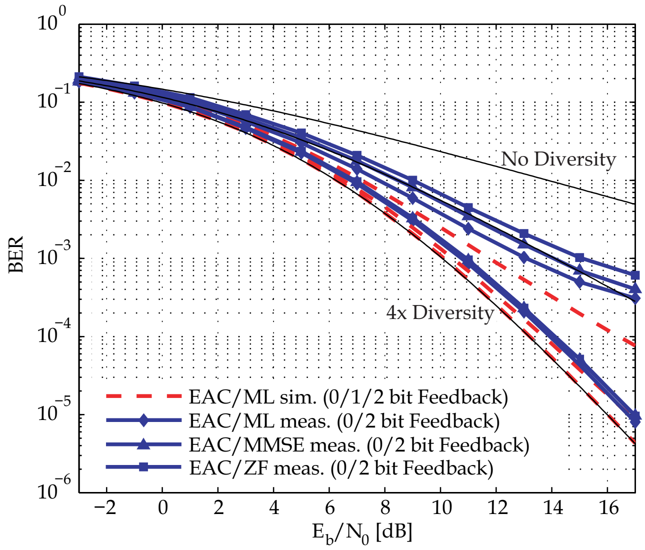

Figure 5: Different receivers for the EAC (QAM, 4.17 MSymbols/s)

ginal when considering that twice the number of TX paths have to be used. However, when using feedback information the diversity increases and reaches almost four. When comparing the simulated (1 bit FB) and measured (2 bit FB) EAC it can be concluded that one bit of feedback information has to be spent for the implementation. The reason is that in our implementation the transmitter board does not have exactly synchronous outputs making exact synchronization at the receiver impossible. The mean standard deviation of the delay between the different transmitter outputs is in the order of 8-10% of the the symbol duration. By excluding blocks with large standard deviations of the delay, the performance of the measurement could be increased.

Fig. 5 shows a comparison between the Zero Forcing (ZF), Minimum Mean Square Error (MMSE), and the Maximum Likelihood (ML) receiver. These receivers are also described in detail in [9]. The results show that the ML receiver achieves a performance up to 2 dB better than the ZF receiver when using no feedback. But as explained above, using the EAC without feedback does not make sense since we can reach almost the same performance with a 2×1 system utilizing the Alamouti code. The results also show that the three receivers perform almost identically when using a 2 bit feedback. Thus, the ZF receiver strikes the best tradeoff between performance and complexity in this case.

5. CONCLUSIONS

In this paper we proposed a very flexible testbed for wireless transmissions that allows to connect to existing MATLAB code and includes physical wireless channels. Much emphasis is put on the fact that the design of the testbed was not based on a single experiment but rather a large range of possible transmission scenarios utilizing different modulation formats and bandwidths. In the presented form, the MIMO testbed allows to quickly test new ideas, but it does not yet include all aspects of final products. Fixed-point implementations are currently only supported by MATLAB fixed-point libraries.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. A. L. Scholtz, R. Langwieser, W. Keim, and L. Mayer for supporting the measurements and insightful comments.

REFERENCES

[1] G. J. Foschini and M. J. Gans, “On limits of wireless communication in a fading environment when using multiple antennas,” *Wireless Personal Communications*, vol. 6, no. 3, pp. 311-335, 1998.

[2] I. E. Telatar, “Capacity of multi-antenna Gaussian channels,” *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585-595, 1999, Technical Memorandum, Bell Laboratories, Lucent Technologies, Oct. 1998.

[3] E. Aschbacher, S. Caban, C. Mehlführer, G. Maier, and M. Rupp, “Design of a Flexible and Scalable 4×4 MIMO Testbed,” *Proc. 11th IEEE Digital Signal Processing Workshop*, pp. 178-181, Aug. 2004.

[4] M. Harteneck, M. Boloorian, S. Georgoulis, and R. Tanner, “Practical Aspects of an HSDPA 14 Mbps Terminal,” *Proc. 38th Asilomar Conf. on Signals, Systems and Computers*, Nov. 2004.

[5] S. Geirhofer, C. Mehlführer, and M. Rupp, “Design and Real-Time Measurement of HSDPA Equalizers,” *Proc. SPAWC*, June 2005.

[6] P. Schniter and A. R. Margetts, “Adaptive Chip-Rate Equalization of Downlink Multirate Wideband CDMA,” *Proc. 36th Asilomar Conf. on Signals, Systems and Computers*, pp. 1228-1232, Nov. 2002.

[7] S.M. Alamouti, “A Simple Diversity Technique for Wireless Communications,” *IEEE J. Sel. Ar. Comm.*, vol. 16, no. 8, pp. 1451-1458, Oct. 1998.

[8] B. Badic, M. Rupp, and H. Weinrichter, “Quasi Orthogonal Space Time Block Codes for Data Transmission over four and eight transmit antennas with very low feedback rate,” *SCC 04*, Erlangen, Germany, pp. 157-164, Jan. 2004.

[9] C. F. Mecklenbräuker and M. Rupp, “Generalized Alamouti Codes for Trading Quality of Service against Data Rate in MIMO UMTS,” *JASP, Special Issue on MIMO Communications and Signal Processing*, vol. 5, pp. 662-675, 2004.