

Baseband Processor for RF-MIMO WLAN

V. Elvira[#], J. Ibanez[#], I. Santamaria[#], M. Krstić^{*}, K. Tittelbach-Helmrich^{*}, and Z. Stamenković^{*}

[#]Communications Engineering Department, University of Cantabria
Plaza de la Ciencia s/n, 39005 Santander, Spain

^{*}Systems Department, IHP GmbH
Im Technologiepark 25, 15236 Frankfurt (Oder), Germany

Abstract — The paper describes the architecture, design, implementation, and test of a reconfigurable digital baseband processor for an RF-MIMO WLAN transceiver that performs the signal combining in the analogue domain. Description includes baseband algorithms (the main blocks being MIMO channel estimation and Tx-Rx analog beamforming), their FPGA-based implementation, a baseband PCB, and real-time tests.

I. INTRODUCTION

Current multiple-input multiple-output (MIMO) wireless systems perform the combining and processing of the complex antenna signal in the digital baseband. Since complete transmitter and receiver are required for each path, the resulting power consumption and costs of the conventional MIMO approaches [1] limit applications for ubiquitous networks. A low-power and low-cost RF-MIMO system for maximum reliability and performance (MIMAX) [2] compliant to the IEEE Standard 802.11a [3] has recently been proposed. It significantly decreases the hardware complexity by performing the adaptive weighting and combining of the antenna signals in the RF front-end [4]-[7].

The RF-MIMO analogue front-end needs new algorithms to exploit the available spatial diversity of the MIMO channel. Several challenges are addressed in the physical layer convergence protocol (PLCP). First, the impairments of the RF front-end are considered in the baseband processor. The algorithms must operate reliably and robustly with respect to the limited resolution of the RF front-end. Moreover, these algorithms must determine the optimal complex weights to be applied at each antenna (implemented by means of vector modulators). The MIMO beamforming algorithms need channel state information at both sides of the link, which is obtained by a specific training procedure. Different optimization goals can be used when determining the optimal Tx-Rx weights [5]. Because of its simplicity, the maximization of the signal-to-noise-ratio (SNR) is the criterion chosen for implementation. In the MIMAX architecture, the same beamformer is used for all subcarriers in OFDM transmissions whereas it is possible to weight each subcarrier independently in conventional MIMO schemes that operate in the baseband.

In this paper, we describe the baseband architecture and functional modules of the baseband processor capable of controlling the RF-MIMO analogue front-end.

II. BASEBAND ARCHITECTURE

The architecture of the baseband processor is shown in Figure 1. It is composed of two main parts: the baseband

processor implementing the IEEE Standard 802.11a and new MIMAX baseband modules implementing new functionalities required by the MIMAX RF front-end architecture. The new functionalities are grouped into two main modules: channel estimator and MIMAX RF weights (or beamforming) block. These MIMAX modules will be active only when a MIMAX training frame is detected by the Tx/Rx control block, which transfers the MIMAX signal field data to the MIMAX control block in order to start the procedure (i.e. the MIMAX channel estimation and beamforming).

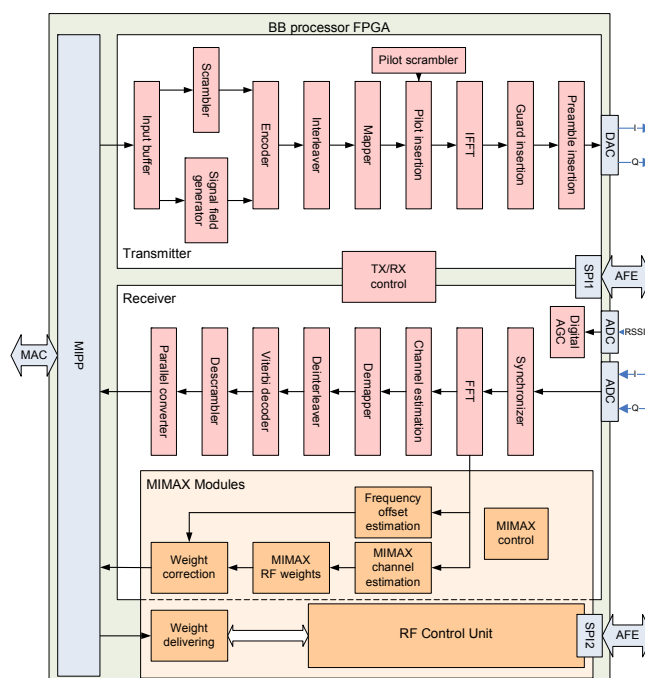


Figure 1. Architecture of the MIMAX baseband processor

More precisely, the architecture of the baseband processor integrates the following modules:

- MIMAX channel estimation: This module estimates the $n_T \times n_R$ MIMO channel based on the $n_T \times n_R$ training OFDM symbols of the received training frame. It works in the frequency domain taking the FFT signal provided by the IEEE802.11a processor as input and uses a least squares estimation method (Section III).
- MIMAX RF weights: It takes the estimated MIMO channel as input and computes the optimal Tx/Rx beamforming weights using the Max-SNR algorithm described in Section IV. It is the most important block in terms of complexity and FPGA resources.

- Frequency offset estimation: Due to the residual frequency error at the output of the conventional IEEE802.11a synchronizer, it might be necessary to include a frequency offset estimator working in parallel with the MIMAX channel estimation and RF weights modules (Section V). To estimate the frequency offset, it is necessary to transmit an additional training symbol, resulting in a training frame of $n_T n_R + 1$ training symbols.
- Weight correction: This module multiplies the weights by a unitary (e.g. rotation) matrix in order to compensate the effects of the residual frequency offset and specific Tx/Rx beamformers used during training.
- Weight delivery: It transfers the calculated optimal weights to the MAC processor (the weight updating). In addition, it allows applying (from the baseband) the predefined set of weights during training (the weight setting) and transferring (from MAC) the optimal or default weights during data transmission or reception (the weight uploading).
- MIMAX control: This module controls the signal and data flow among all MIMAX blocks. It receives from the Tx/Rx control block information included in the training frame signal field (the number of Tx/Rx antennas, the number of training symbols), as well as some activation and synchronization signals.
- RF control unit: This is a control interface between the baseband processor and analogue front-end (AFE). It is an integrated part of the baseband processor.

All the MIMAX blocks are activated only when a training frame is received. Therefore they can be powered down while either processing conventional data frames or transmitting training frames. Only the MIMAX control block, the weight delivery block, and the RF control unit remain active at any time because it must transfer and set the weights from the MAC processor to the RF control unit.

The complete baseband processor was initially designed using a Matlab model that uses floating-point operations to implement all processing stages. This floating-point model is useful to obtain an upper bound on the expected performance of the baseband processor, but cannot be used for FPGA implementation. A fixed-point Matlab model was then developed that allowed us to take design decisions with regard to the required precision (e.g., number of bits, number of iterations to be applied in the algorithms, etc.)

III. MIMAX CHANNEL ESTIMATION

The MIMAX channel estimator uses the $n_T n_R$ training OFDM symbols included in a training frame. Each training symbol is affected by a specific pair of Tx and Rx beamformers. A conventional least squares algorithm is used to estimate the $n_T n_R$ equivalent SISO channels at the 52 active subcarriers.

Some design decisions has been taken in order to simplify the implementation of the MIMAX channel estimator. First, the identity matrix has been selected for the Tx and Rx beamforming matrices used during the training stage. Second, the MIMAX training symbols will be the same as the

IEEE802.11a long training symbols composed of 52 subcarriers modulated by BPSK values.

As Figure 2 shows, the MIMAX channel estimator works in the frequency domain (i.e., after FFT) and could include an optional post filtering procedure to smooth the resulting frequency responses. From an implementation point of view, the LS estimator requires very few FPGA resources (just sign inverters and control logic), but the post filtering process could be expensive in terms of memory and MACs (while providing marginal BER improvement). For this reason, we have initially designed only the LS version of the MIMAX channel estimator block.

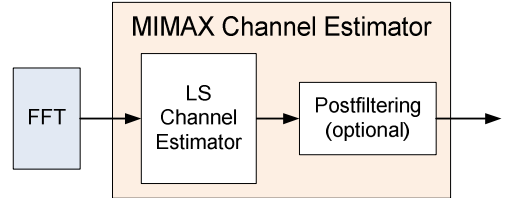


Figure 2. MIMAX channel estimation

IV. BEAMFORMING WEIGHTS CALCULATION AND DELIVERY

We have focused on the implementation of the Max-SNR beamforming algorithm. This initial algorithm has been chosen because other criteria proposed in [5] use the Max-SNR solution as a starting point. Furthermore, the choice of the Max-SNR algorithm for implementation simplifies the architecture of this block without significant deterioration of the performance of the whole system. The proposed algorithm reduces to the maximization of the energy of the equivalent SISO channel or, in other words, to the maximization of the received SNR

$$\arg \max_{\mathbf{w}_T, \mathbf{w}_R} = \sum_{k=1}^{N_k} \left| \mathbf{w}_R^H \mathbf{H}_k \mathbf{w}_T \right|^2, \text{ s.t. } \|\mathbf{w}_T\|^2 = \|\mathbf{w}_R\|^2 = 1,$$

where the $n_T n_R$ matrix \mathbf{H}_k is the MIMO channel response at the k -th subcarrier, and $\mathbf{w}_T, \mathbf{w}_R$ are the transmit and receive analog beamformers, respectively. These are complex vectors containing the RF weights to be applied by the AFE.

The input signals of the MIMAX RF weights block come from the channel estimator whose outputs are the 52 subcarrier samples for each one of the 16 (considering a MIMAX link with four antennas at the transmitter and receiver sides) equivalent SISO channels. Notice also that all operations are carried out with complex numbers. Specifically, the pseudocode for implementing this algorithm can be summarized in the following steps:

Step A:

- Create 52 column vectors \mathbf{x}_k (dimensions 16×1) where the i -th element of \mathbf{x}_k is the sample of the k -th subcarrier for the i -th equivalent SISO channel. Create 52 16×16 matrices $\mathbf{X}_k = \mathbf{x}_k \mathbf{x}_k^H$.
- Add the 52 matrices $\rightarrow \mathbf{Y} = \sum \mathbf{X}_k$

Step B:

- Calculate the dominant eigenvector \mathbf{z} of the matrix \mathbf{Y} using a fixed number of iterations of a power method

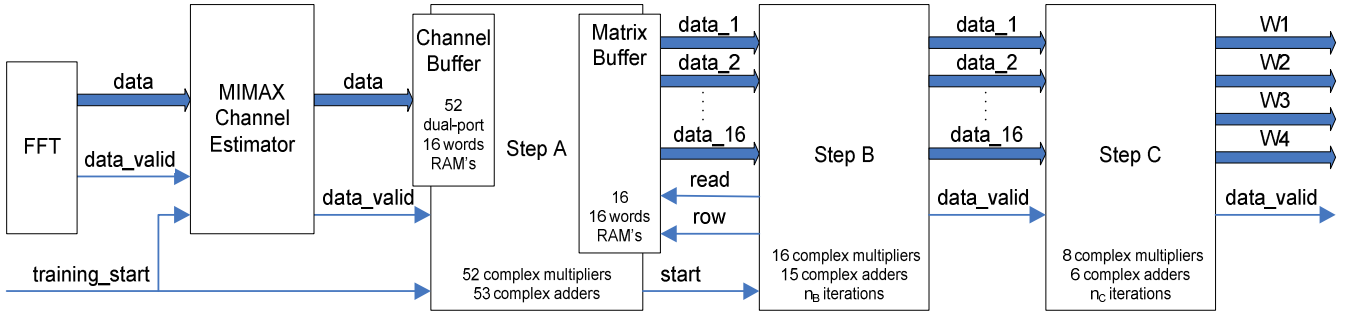


Figure 3. Max-SNR beamforming weights calculation

Step C:

- Construct \mathbf{Z} as the 4x4 matrix resized from the 16x1 vector \mathbf{z}
- The Max-SNR Rx beamformer \mathbf{w}_R is the left singular vector of \mathbf{Z} , which is obtained applying again a fixed number of iterations of a power method

A schematic diagram of the Max-SNR implementation steps is shown in Figure 3. Step A is creation of the 52 column vectors \mathbf{x}_k where the i -th element of \mathbf{x}_k is the sample of the k -th subcarrier for the i -th equivalent SISO channel. The size of \mathbf{x}_k is $n_T n_R$ (16 in this case). It also creates the 52 rank-one matrices $\mathbf{X}_k = \mathbf{x}_k \mathbf{x}_k^H$ of 16x16 dimension and adds these 52 matrices in a sum \mathbf{Y} . Step B calculates the z dominant eigenvector of the sum matrix. The common way to calculate this dominant eigenvector is to perform the singular value decomposition (SVD). However, the implementation of a complete SVD is not needed as it would use too many resources. The alternative solution is the power method which was finally implemented. This method is probably the simplest one for finding the largest eigenvector of a matrix. From the z vector of 16x1 dimension obtained by Step B, we construct the \mathbf{Z} matrix of 4x4 dimension resized by columns. Step C calculates the SVD maximum eigenvector of \mathbf{Z} in order to extract the first row of the \mathbf{U} matrix. Again, it is not necessary to perform the complete SVD. A beamforming weight coefficient can be calculated as the dominant eigenvector of the product $\mathbf{Z}\mathbf{Z}^H$ where \mathbf{Z}^H is the Hermitian of matrix \mathbf{Z} . Thus Step C can be split in two substeps: the first one is a matrix multiplication and the second is a 4x4 power method. The resultant vector of this last power method is the \mathbf{w}_R beamforming weight under the Max-SNR criterion.

The first task of the weight delivery block consists of transferring the calculated optimal weights to the MAC processor after a training frame has been received. This is so-called weight updating and it is a straightforward procedure (Figure 4). The beamforming weights are provided directly by the MIMAX RF weights block (or by the weight correction block if finally needed).

The next task is to transfer the optimal or default weights from MAC to radio-frequency control unit (RFCU) during the transmission or reception of data frames. This procedure, called weight uploading, has been easily implemented by allowing a direct connection between the MAC processor and the RFCU as shown in Figure 5. Finally, the last task is to

apply the predefined set of weights during transmission or reception of a training frame: this procedure is denoted as weight setting.

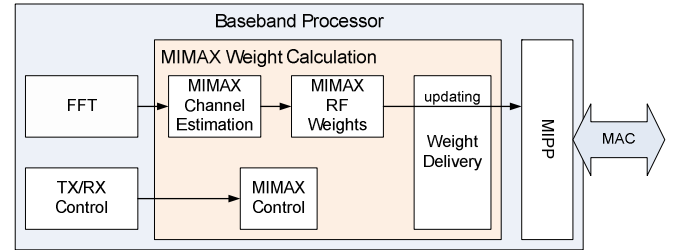


Figure 4. Illustration of the weight updating

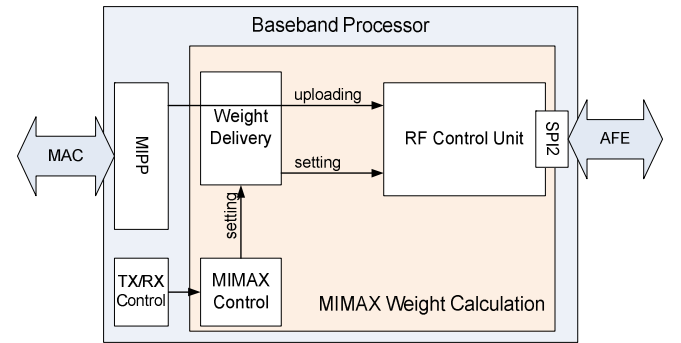


Figure 5. Illustration of the weight delivery

V. FREQUENCY OFFSET ESTIMATION

Any residual frequency offset that occurs after the synchronizer stage of the conventional IEEE802.11a receiver distorts the weight calculations during training. Therefore, it could be necessary to estimate and compensate that residual frequency offset by transmitting two training symbols using the same pair of Tx and Rx beamformers.

Under the assumption that the residual frequency offset is lower than the subcarrier spacing, the maximum likelihood frequency offset estimator is given by

$$\hat{\Delta f}_{ML} = \frac{1}{2\pi\Delta t} \angle \left(\sum_{k=1}^{N_c} s_1[k] s_2^*[k] \right),$$

where N_c is the number of active subcarriers; s_1 and s_2 are the OFDM training symbols used for frequency estimation; and Δt means the time between symbols s_1 and s_2 .

VI. FPGA IMPLEMENTATION AND TEST

Xilinx FPGAs have been selected for implementation of the reconfigurable baseband processor. In this section, the FPGA implementation process of the main MIMAX baseband modules (channel estimation, frequency offset estimation and beamforming weights calculation) is briefly described.

For the design and implementation of these blocks we have used the Xilinx System Generator tool. This tool is a plug-in to the Matlab's Simulink that enables designers to develop high-performance DSP systems to be implemented in FPGA technology. It can automatically translate designs into FPGA implementations that are faithful, synthesizable and efficient.

The chosen FPGA is a Virtex5 LX330 which has 34560 slices. Regarding the RF weights calculation block, some decisions have been taken to reach a good compromise between FPGA utilization and system performance: We used 5 iterations for each power method and 8 bits interfaces between the blocks shown in Figure 3. The conventional IEEE802.11a baseband processor occupies around 45 %, whereas the new MIMAX baseband modules occupy 33 % of the available slices. The operating clock frequency of the processor is 80 MHz.

The baseband modules are integrated in a dedicated baseband board featuring communication with the MAC processor and the analogue front-end. The baseband board incorporates, except a Virtex5 LX330 FPGA, all required interfaces, digital-to-analogue and analogue-to-digital converters for baseband signals, program flash, power and clock circuitries, and connectors. The photograph of the produced baseband board is shown in Figure 6.



Figure 6. MIMAX baseband printed-circuit-board

The baseband board was used for the real-time tests of the MIMAX baseband processor in several setups. First, we have verified the correct reading, changing, and re-reading of a few configuration parameters. Then, using the USB terminal program a few beacon, data and training frames were transmitted and the generated I/Q signals at the DAC were analyzed to verify a correct transmission. Afterwards, some data frames were generated in Matlab and downloaded to the vector signal generator. The signals generated with the E4438C RF generator were used as I/Q inputs of the MIMAX

baseband board and the correctness of the data was verified by the USB terminal program.

The most important test aimed at checking the correct real-time behavior of the developed MIMAX modules. For this test we generated training frames for a 4x4 MIMO system where each of the 16 training symbols was affected by a different SISO channel. These training frames were generated in Matlab and distorted by known MIMO channels. The training sequence was transmitted with the vector signal generator and the optimal weights calculated by the processor were provided to the USB terminal program. The beamforming weights obtained in simulation and those provided by the baseband board are compared in Figure 7. This test was repeated for different channel conditions: in all examples, a very good agreement between the weights obtained in simulation and those provided by the baseband board was observed.

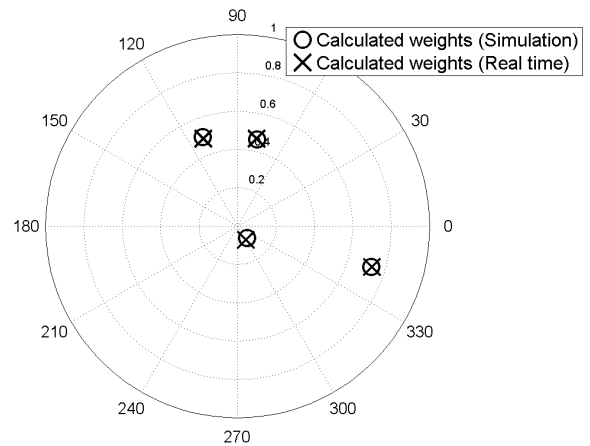


Figure 7. Comparison of RF weights calculated in simulation and in real time

VII. CONCLUSION

In this paper, we have described a new baseband processor designed and implemented according to the requirements of the RF-MIMO analogue front-end that exploits the available spatial diversity of the IEEE802.11a communication scheme.

REFERENCES

- [1] H. Boelskei, D. Gesbert, C. B. Papadias, and A.-J. van der Veen, *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, Cambridge University Press, Cambridge 2006.
- [2] Z. Stamenkovic, K. Tittelbach-Helmrich, M. Krstic, J. Perez, J. Via, and J. Ibanez, "Architecture of an Analog Combining MIMO System Compliant to IEEE802.11a," *Proc. ICT-MobileSummit 2009*, Santander (Spain) 2009, (pp. 1-8)
- [3] IEEE-SA Standards Board, "High-speed Physical Layer in the 5 GHz Band," *IEEE Standard 802.11a-1999(R2003)*, Supplement to IEEE Standard 802.11-1999.
- [4] F. Ellinger and W. Baechtold, "Adaptive Antenna Receiver Module for WLAN at C-Band with Low Power Consumption," *IEEE Microwave and Wireless Components Letters*, vol. 12, pp. 348-350, 2002.
- [5] J. Via, I. Santamaria, V. Elvira and R. Eickhoff, "A General Criterion for Analog Tx-Rx Beamforming under OFDM Transmissions," *IEEE Trans. on Signal Processing*, vol. 58, pp. 2155-2167, 2010.
- [6] R. Eickhoff, R. Kraemer, I. Santamaria, and L. Gonzalez, "Developing Energy-Efficient MIMO Radios," *IEEE Vehicular Technology Magazine*, vol. 4, pp. 34-41, 2009.
- [7] A. Jahani, F. Tzeng, and P. Heydari, "Code Modulated Path Sharing Multi-Antenna Receivers: Theory and Design," *IEEE Trans. Wireless Communications*, vol. 8, pp. 2193-2201, 2009.