Testbeds and Rapid Prototyping in Wireless System Design

Markus Rupp, Christian Mehlführer, Sebastian Caban,
Robert Langwieser, Lukas W. Mayer, and Arpad L. Scholtz*

Institute of Communications and Radio-Frequency Engineering,
Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Wien, Austria
Phone: +43 1 58801 38967, Fax: +43 1 58801 38999
Email: mrupp@nt.tuwien.ac.at
Web: http://www.nt.tuwien.ac.at/rapid-prototyping

July 19, 2006

Abstract

This tutorial paper gives an overview of requirements on testbeds and rapid prototyping suitable for MIMO transmissions. Testbeds support real-time transmissions over the air and thus allow for experimenting with true physical channels, including also an analog frontend. This makes the transmission process very realistic. On the other hand, rapid prototyping allows for sketching transmitter and receiver hardware architectures of future products. Thus, rapid prototyping is very close to the design of a final product, de-risking its financial investment. Several experiment examples demonstrate how testbeds and prototypes can support system design significantly.

*This work has been funded by the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms, http://www.nt.tuwien.ac.at/cdlab.

1 Introduction

Wireless products have become very popular due to low cost devices and services. Communicating in a wireless fashion is by far not simple, in particular when many users desire to communicate at the same time. The available spectrum is rather limited, making the design of a wireless multi-access scheme very challenging. Note that in most countries the majority of the spectrum below 1 GHz is occupied by public TV broadcasting while only little is left for cellular and WLAN services. Although recently new bands at 3.5 GHz, 5.2 GHz and 11 GHz (in some countries also 1.5 GHz, 2.6 GHz and 5.8 GHz) became available, moving to higher frequencies is not a good solution for most wireless connections because more transmit power is required to compensate for the much higher channel attenuation. Due to multipath propagation, the receivers become very complex, in particular when data is to be transmitted at high rates. Because of this high complexity in signal processing, the mobile communication devices suffer of low battery life, making it a necessity to recharge the battery quite often.

On top of all these problems, newer communication standards still require increased signal processing complexity, draining the battery power even more. Several studies [1–3] show that although Moore's law [4] predicted in 1965 is still correct after 40 years, i.e. the available complexity doubles every 16–18 months, the desired complexity follows Shannon's law and is doubling even faster. While this was not so much of a problem in the past years since the available complexity was sufficient, it becomes a permanent problem nowadays, in particular when time-to-market aspects become crucial [5]. This so-called complexity gap is accompanied by another one called design gap or productivity gap. The studies in [1, 2] show that current improvements in productivity, i.e. the designer's capability to convert algorithms into silicon, is growing much slower than the available silicon complexity predicted by Moore's law.

The reason for such a lack in productivity lies in the poor tool support. After the definition of VHDL in the mid-eighties, industry took on tool support for design engineers and came up with many useful products. At these times, only very few universities were involved. Until the mid-nineties, the quality of the products improved only very little. While many tasks could be made automatically, the remaining tasks are rather difficult to solve and many research teams at universities are currently working on solutions. In particular, there are very few tools to support float-to-fixed conversion, i.e. the conversion from floating point descriptions of high level languages like Matlab or ANSI-C to a corresponding fixed-point representation. While graphical tools from Synopsys, CoWare, and Mathworks increase visibility, they support only manual partitioning in HW (Hardware) and SW (Software) modules. However, for complex algorithms, it may be much better to have an automatic tool support, taking possible hardware platforms and communication between the modules into account when deciding for partitioning. Often, specific HW platforms are already predefined and the task is to map an algorithm onto such a platform consisting of an embedded DSP with many HW-accelerators, RAMs, ROMs, and busses. Also recently, embedded cores with power awareness were introduced [6], allowing idle algorithms to be switched off or to run on lower clock rates to preserve energy. Tools for all of these important tasks are currently missing and manual solutions are time consuming, erroneous and certainly sub-optimal.

This paper is organized as follows. In Section 2 the terminology in prototyping is defined distinguishing strictly between testbeds and prototypes. While Section 3 further explains testbeds and provides many design examples utilizing a testbed, Section 4 provides the same details for prototypes. Here, in particular, the methods for rapid prototyping are focussed on. Examples on MIMO-WLAN and adaptive predistortion techniques are provided. Section 5 discusses further challenges in rapid prototyping, and Section 6 closes the paper with conclusions.

2 Prototyping

Many years ago, prototyping was used to build a first (set of) working demonstrator(s) to prove that a new theory could really be applied and to learn how cumbersome and expensive it could once become turning it into a product. Real-time experiments with working product demonstrators were essential to understand all implications of a new technology and de-risk future decisions before marketing.

Due to tight time-to-market constraints and the permanently increasing complexity, prototyping was more and more abandoned in wireless industry. Prototyping would have required a larger and larger group of engineers, and the time to build a single prototype would have taken as long as designing the entire product. However, abandoning prototyping entirely and basing a design purely on simulations is very risky as some cases have proven(see for example [7] for details).

In order to bring prototyping back into the design chain, specific tools for prototyping must be available [7, 8], allowing for a rapid prototyping design that can be implemented much faster than the entire product with a small design team but certainly allowing for real-time experiments on a hardware very similar to the final product.

Note that the terminology in this field is often used rather loosely. In [9] it has been distinguished between demonstrators, testbeds, and prototyping. We like to repeat this distinction for convenience of easier reading (see also Merriam-Webster [10] and Funk & Wagnalls Standard Dictionary of the English language [11]):

- A *Demonstrator* mainly serves as a sales vehicle and to show technology to customers. In general it will implement a new idea, concept or standard that has already been established and has been finalized to some degree. Requirements on scalability are therefore less important than its functionality and often the required design time.

- A *Testbed* on the other hand is generally used for research. It is a vehicle for further developments or for verification of algorithms or ideas under real-world or real-time conditions. This results in the requirement for scalability, modularity and extendibility.

- A *Prototype* is the initial realization of a research idea or a standard, either as a reference, a proof of concept or as a vehicle for future developments and improvements. As opposed to a "simulation" it is not an "imitative representation" of the device. Instead it has significant similarities. In industry, a migration into a product is often intended.

While demonstrators are most important for companies showing them to their potential customers, testbeds and prototypes are dominating the development process and are thus more of a research object. Both, testbeds and prototypes, allow for real-time experiments. However, testbeds allow only for a wireless transmission in real-time. The further processing of the received signals may be performed offline. While testbeds typically include the wireless channel, prototypes more often use channel emulators in order to provide an environment that allows for repeatable experiments. If testbeds are built properly, they may also allow for replacing the offline processing by a prototype or even the final product. While the processing of the algorithms is typically performed in a high-level language (i.e. in floating-point) when building a prototype it is important to use a hardware platform close to the final product, (i.e. processing is performed in fixed-point). The main properties and differences of testbeds and prototypes are summarized in Table 1.

Testbeds	Prototypes
Fast to program due to high level language	Considerable time required to build
Data storage and offline processing possible	Methods to speed up design are required
Includes the wireless channel	May include the wireless channel
Only wireless transmissions in real-time	Functionality in real-time
DSP functions may be replaced by prototype	Allows for design space exploration
Fixed and/or floating point algorithms	True fixed-point design

Table 1: Properties of testbeds and prototypes.

3 Wireless Testbeds

The questions that can be solved with a testbed are essentially whether the channel model assumptions are viable and what impact the analog frontend may have. Although many MIMO channel models exist, most publications assume i.i.d. (independent identically distributed) flat

Rayleigh fading channels. Measurements have proven that this is by far not true and more or less complicated models have been proposed [12, 13]. Even the most complicated models still have imperfections and do not necessarily address all points of interest. Typically, models are optimized with respect to channel capacity. However, they may fit capacity and still give incorrect behavior for BER performance [14, 15].

Typically the analog frontend of a testbed is of very high quality, satisfying linearity for a very large dynamic range of the signal. AD and DA converters provide high resolution of 14–16 bits. By introducing imperfections (cutting off bits or misadjusting the oscillator for example) one can analyze how sensitive the algorithms are on the analog part of the transmission chain. Also deriving an idea of the final product costs, based on linearity requirements for example, is possible. If it turns out that the high precision analog frontend of the wireless testbed is not sufficient for the required performance, one knows for sure that the idea is not technically sound for a final product. Alternatively, if the required analog frontend is already available (reuse of a previous product, for example), it can be used in a testbed to clarify whether it is sufficient for the new algorithms.

3.1 Commercial Tools

Since companies do not use a consistent terminology and try to offer as many features as possible, it is not easy to categorize available products into testbeds and prototypes. Most can be used for both purposes, however some are suited better for one purpose than for the other. The selection here is thus somewhat subjective. Complete out-of-the box testbeds are available from Lyrtech (www.lyrtech.com) and Signalion (www.signalion.com). Components to build testbeds and prototypes are available from Sundance (www.sundance.com), Hunt-Engineering (www.hunteng.co.uk), and Pentek (www.pentek.com)[1]. Most products lack RF-frontends making it very hard to include the true physical channel. The development of RF frontends is costly and cumbersome. In particular, available frontends are missing flexibility in terms of supported carrier frequencies and bandwidths.

Since specific design tools are typically missing in commercial testbeds and prototyping equipment, EDA (Electronic Design Automation) tools used in the chip design process are also used for prototyping. For example, EDA tools are available from Synopsys and CoWare (and many smaller companies) but the licence costs are typically too high for prototyping. In particular, when heterogeneous systems including DSPs and FPGAs are required, no supporting design tools are available. More details about tools are provided further ahead in Section 5.1.

3.2 Vienna MIMO Testbed

Due to a lack of commercially available products, we decided to develop our own testbed. This testbed was designed by a team of researchers at the Institute of Communications and Radio Frequency Engineering of the Vienna University of Technology during the last three years. Throughout the development process, emphasis was placed on scalability, modularity, and extendibility, allowing for a multitude of very different experiments. The Vienna MIMO testbed [16, 17] primarily consists of (see Fig. 1):

- a *transmit PC* that automatically preprocesses and up-converts complex baseband data samples to a low IF (Intermediate Frequency) of 70 MHz.
- an *analog transmitter-frontend*, performing linear frequency conversion of up to four low IF signal to the radio frequency of 2.5 GHz, filtering, and amplification to the transmit power level of 30 dBm.
- a *channel* realized with channel emulators or a physical radio channel. Two positioning tables are used to move the antennas in order to achieve the channel realizations needed for averaging the mean performance of a specific scenario.
- an *analog receiver-frontend* that prefilters, amplifies, and down-converts the received signals of up to four antennas to a low IF of 70 MHz.

[1] Note that we only mention the most prominent ones. There is a multitude of smaller companies providing testbed and prototyping equipment.

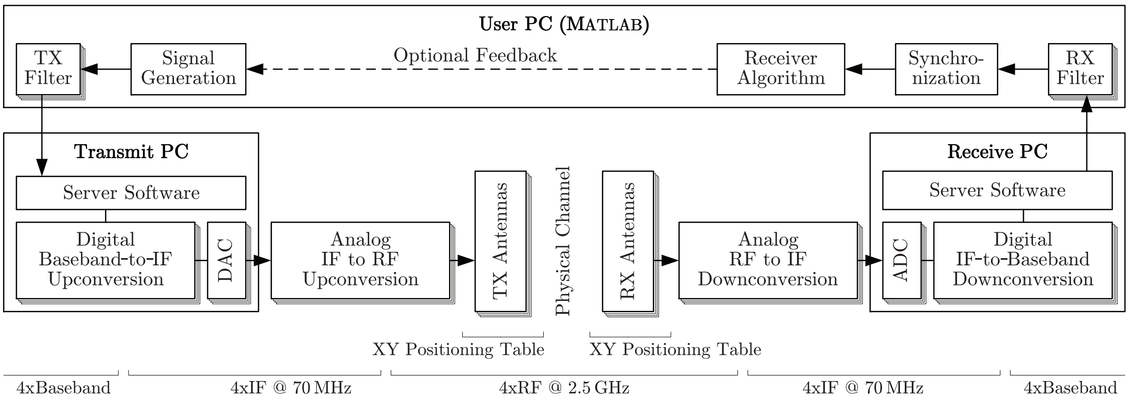

Figure 1: Block diagram of the Vienna MIMO Testbed.

- a *receive PC* that performs conversion to the digital baseband.
- *user PC*s from where algorithm designers carry out various radio transmission experiments. Using a flexible MATLAB Interface, complex baseband data samples of a 4×4 MIMO system can be easily transmitted and received.

The concept of the Vienna MIMO testbed allows the algorithm designer to focus on digital baseband data processing on his *own* PC. Next to being platform independent, also multiple users can access the testbed directly out of MATLAB from anywhere in the LAN. This not only makes it very convenient to integrate real-time air transmissions into existing MATLAB simulations but also saves a lot of costs since very expensive[2] hardware can be shared efficiently by several researchers. In the following, many examples are shown in which the testbed turned out to be useful.

3.3 Space-Time Codes

In the last years, space-time codes were thoroughly investigated and optimized [18–23]. However, hardly any results exist on how these space-time codes perform in a real system. Therefore, a MIMO testbed was used to measure the performance of an Extended Alamouti (4×1 MISO) code [17, 24]. By using code selection with just two bits feedback (i.e. selection of one out of four codes) from the receiver to the transmitter, an Extended Alamouti code transmission scheme achieves the full diversity order of four—at least in simulation. Surprisingly, the measurements with the testbed and channel emulators (which were configured to produce uncorrelated flat Rayleigh fading channels) showed an SNR loss of about 0.8 dB (see Fig. 2) when using two bits feedback. It was found that this effect is caused by slightly asynchronous transmitter outputs. A mean delay of just 20 ns (approx. 10 % of the symbol time) between the output samples of the transmitter chains caused the measured loss of 0.8 dB. For the implementation of a final product, exactly synchronous transmitter outputs are thus of utmost importance.

3.4 UMTS HSDPA Equalizers

The HSDPA (High Speed Downlink Packet Access) channels of UMTS provide high data rates to the user. These channels use a small spreading factor of 16 and are thus very sensitive to MAI (Multiple Access Interference). MAI is caused by the multipath propagation channel which destroys the orthogonality of the spreading codes and therefore degrades the performance.

The Vienna MIMO testbed, in conjunction with the channel emulators, was used to investigate the performance of different equalizer structures for HSDPA [25]. Here, the channel emulators have a major advantage over MATLAB channel models: The channel emulators allow for emulating paths with nearly arbitrary delays (0.5 ns stepsize). A MATLAB channel model would need very high interpolation factors to offer resolutions in the same dimension. Our measurement results showed

[2] The required equipment, including channel emulators and noise sources, costs approximately 500,000 €.

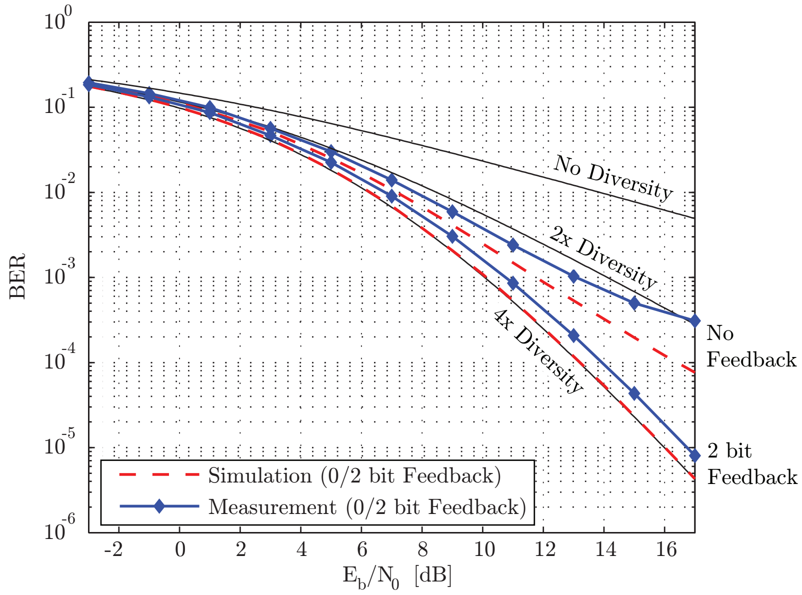

Figure 2: BER performance of the Extended Alamouti code (4.17 MSymbols/s)) [17].

that adaptive equalizer structures for HSDPA achieve the performance of the MMSE equalizer and clearly outperform the conventional RAKE receiver (Fig. 3).

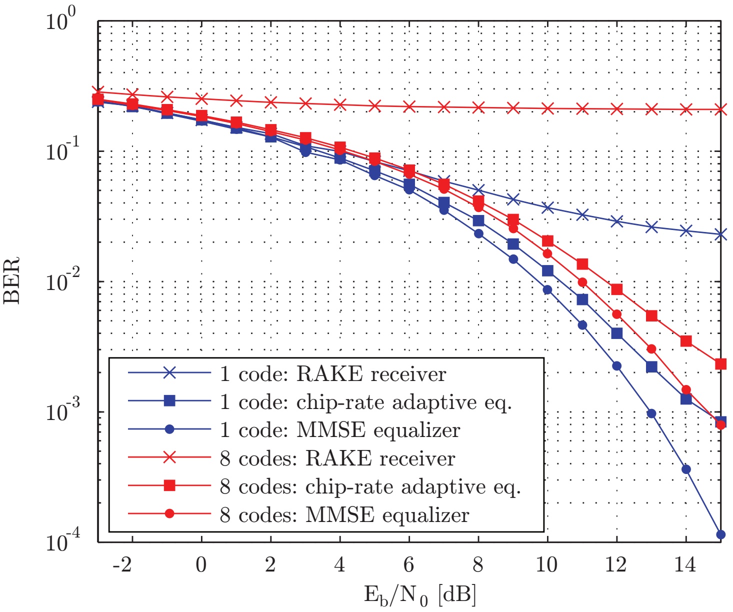

Figure 3: Comparison of different receiver structures for HSDPA [17].

3.5 MIMO Antenna Design

A general belief is that antennas for MIMO transmissions need to be at least $\lambda/4$ spaced apart, resulting in large and visible antenna constructs. A wireless testbed can also be used to determine the performance of compact and realistic MIMO antennas. One example for such an antenna design is given in the following.

A compact antenna with low constraints on its feasibility, designed for mobile communication equipment, is the so-called inverted-F antenna [26, 27]. It consists of a metal plate that is aligned in parallel to one side of a conducting box. One end of the metal plate is bent towards the box and connected to it (Fig. 4). The feed consists of a wire that extends from the box and connects to the metal plate. Matching is achieved by choosing the position of the feeding wire and the width of the metal plate. The center frequency of the antenna is determined by the length of the metal plate (approx. $^{\lambda}/_{4}$). Bandwidth can be tuned by changing the distance between the metal plate and the box.

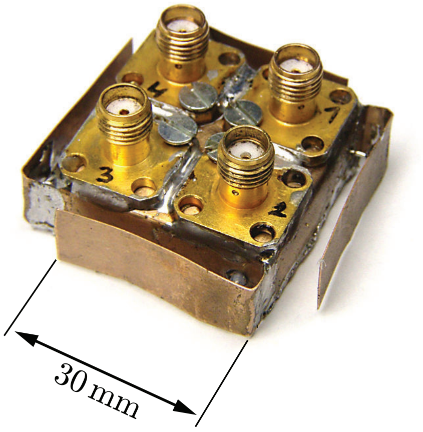

Figure 4: Photograph of the quad inverted-F antenna. The four coaxial signal connectors seen on the top serve for interfacing to the RF frontend.

Radiation of the inverted-F antenna is mainly caused by the currents in the feed wire and the part of the metal that is parallel to the feed wire. Furthermore, currents in the resonating metal plate and on the box surface cause radiation with additional arbitrary polarization states.

To evaluate the performance that can be achieved by using MIMO techniques in a cell phone, a quad inverted-F antenna was built [28]. It consists of a metal box ($28 \times 28 \times 8\,\mathrm{mm}^3$ for the carrier frequency of 2.5 GHz) which is equipped with four inverted-F antenna elements (Fig. 4). Due to the perpendicular or opposite alignment, the antennas are well decoupled from each other (S-parameter: $|S_{m,n}| \leq -15\,\mathrm{dB}$ for $m \neq n$). Furthermore, the perpendicularly aligned antenna elements emit most of the power into separate directions and employ different polarizations. All these effects enhance the MIMO performance of the quad inverted-F antenna because very different combinations of incident waves that reach the antenna in a multi path propagation scenario are presented at the four outputs. The overall size of $34 \times 34 \times 8\,\mathrm{mm}^3$ of the quad inverted-F antenna allows for integration into cell phones. Furthermore, the RF frontend of the cell phone can be placed within the metal box of the antenna. The performance of such an antenna was evaluated with MIMO HSDPA throughput experiments reported in the following section.

3.6 MIMO UMTS HSDPA Throughput

The above mentioned quad inverted-F MIMO antenna was used to directly investigate the impact of the antenna configuration on the throughput of a MIMO UMTS system [29]. The system was implemented according to the DSTTD-SGRC (Double Space-Time Transmit Diversity with Sub-Group Rate Control) proposal of Mitsubishi [30, 31]. The performance of the quad inverted-F antenna was compared to a linear antenna array consisting of four $^{\lambda}/_{4}$-monopole ground-plane antennas with an element spacing of $^{\lambda}/_{2}$.

Fig. 5 shows the results of throughput measurements plotted over E_c/I_{or} (energy of the transmitted chip-stream over total available transmitter energy). Surprisingly, the rather small quad inverted-F antenna shows approximately the same performance as the linear antenna array. The fourth antenna element of the quad inverted-F antenna does not add a significant performance gain (approx. 0.4 dB). Nevertheless, the measurement results show that small and compact user equipments capable of MIMO transmission are feasible.

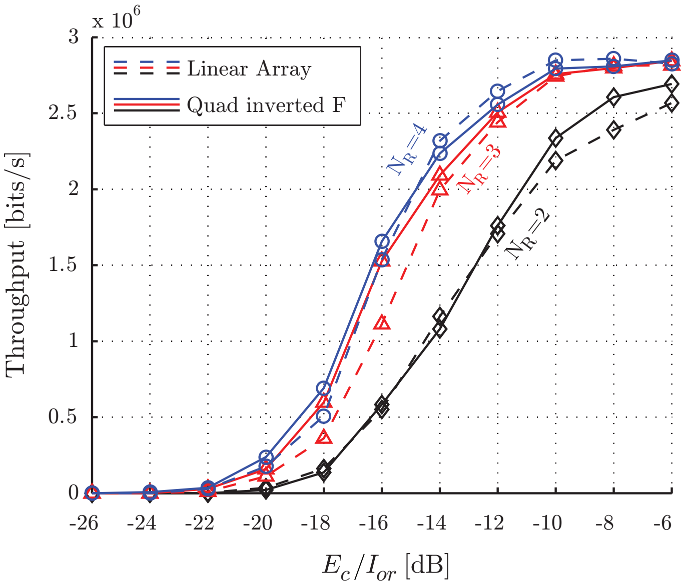

Figure 5: Throughput measurement for varying number of receive antenna elements N_R [28].

4 Rapid Prototyping

As mentioned in Section 2, conventional prototyping becomes more and more abandoned because of tight time-to-market constraints. Nevertheless, prototyping is an important means to de-risk a new technology and should be applied somehow. A possibility is to overcome the slow prototyping development by including a particular *rapid* prototyping design methodology and thus speeding up the development considerably. Since very likely a future product will not only consist of a single DSP, it is of utmost importance to support DSPs as well as FPGAs, thus offering HW/SW partitioning. Communication between connected blocks needs to be taken into account since it may be rather complicated to move functions from DSPs to FPGAs while it is usually relatively simple to stay on an FPGA or on a DSP when moving from block to block (as long as resources are available). An important feature to speed up development time is to use high level languages wherever possible. For example, many DSPs from Texas Instruments (C6x) and Analog Devices (SHARK) can be programmed fully in C including driver functions and DMA transfer. Also many tools exists that convert C into VHDL [32] or SystemC, both allowing to generate hardware. Most importantly, these tools allow to satisfy the *one-code paradigm* [7], claiming that most errors can be avoided when sticking to a single code for description in high level design and converting such code to lower design levels by automatic tools only. In this section, we begin by summarizing available commercial design tools and discussing their properties. Then we introduce the Vienna Prototyping Environment (VPE) and finally show two examples where it has been applied successfully.

4.1 Commercial Tools

Commercial tools are typically based entirely on DSPs or entirely on FPGAs. Texas Instruments C6x processors (and some others like Analog Devices' SHARK) can fully and efficiently be programmed in ANSI-C. MATLAB's Simulink supports several platforms with C6x DSPs. Thus, if it would simply be a task of mapping to a multi-DSP board, this could efficiently be achieved by such tools.

On the other hand, tools that map Simulink-code onto Xilinx or Altera FPGAs also exist, including a fixed-point library. Here the algorithm needs to be described with elements of this fixed-point library (can also be utilized in floating point). Once the algorithm runs sufficiently in simulation, it can be mapped on to the desired FPGA. However, typically such algorithms need to be of feed-forward structure. Once feedback elements are included, the simulation times become rather long due to inefficient scheduling. Also, the algorithms should be data-flow driven and not control-flow oriented. For control-flow oriented algorithms such programming is very tedious and even small algorithmic changes can require a long time.

Lyrtech supports both conversion tools for DSPs and FPGAs on their boards including combinations of DSPs and FPGAs. The communication between a DSP and an FPGA is described by a special block under Simulink. While this is not extremely efficient, it provides a means to map algorithms directly onto FPGAs and DSPs provided one has sufficient resources available.

Other companies like Sundance, Hunt Engineering, Signalion, or Pentek offer various modules with FPGAs and DSPs and driver software supporting the communication links between the various modules. However, complete prototyping tools that would allow to map from high level design to the hardware platform are typically not provided. Note that of the many providers, only Pentek and Signalion offer RF front ends (as of today), although most other companies have at least concepts for hardware they are planning to offer in the near future.

4.2 Vienna Prototyping Environment

The origin of the current Vienna Prototyping Environment (VPE) goes back to a development at Bell-labs that supported prototyping of local wireless loops [33] and early UMTS MIMO-receivers [34]. The VPE follows the one-code paradigm [7], i.e. only the high level description is refined in every design step, but no manual recoding is required. The design flow starts with COSSAP-C module descriptions, i.e. ANSI C code that is enriched with port information defining the input and output ports as well as the rates of the data streams and corresponding data types. This information can be used to convert the module to a Simulink S-function and run simulations. Once it has been decided which blocks are mapped onto DSPs and which on FPGAs, the refinement of the C-code follows different rules. For the DSP code, the variables are converted to 16 bit short types and intrinsic commands are included. With corresponding mapping algorithms the code can still run under Simulink and be tested. For FPGAs, a similar technique is used to refine the C-code so that it can be mapped automatically to an FPGA using Adelantetech's Builder-tool. Currently, many tools like this exist [32], often of academic origin and freely available. An important fact is that the algorithm in its refined form, although now suitable for a transformation to a hardware description language, can still be run under Simulink. Very significant is the possibility to co-verify the algorithms once they are converted to hardware. Many people validate their algorithms with Modelsim [35] that can be used for cosimulation with Simulink. However, Modelsim models on a fine granular level and is thus very time consuming. We developed a method that allows not only to map the high level description automatically onto DSPs and FPGAs but also to run the algorithm on the HW platform while the remaining parts are still running under Simulink. Originally developed at Rice University [36] and adopted at Bell-Labs [7], the method was made more flexible to convert it easily onto other hardware platforms and it was extended to FPGAs as well [37]. Many details of the technique can be found in [7].

4.3 WLAN Prototyping

The VPE was used to set up an 802.11n MIMO wireless LAN system utilizing the prototyping platform Smartsim by Seibersdorf Research [38, 39]. Since during the standardization of 802.11n several transmission schemes were discussed, the undefined parts of the system had to be implemented in a high level language to quickly adopt for changes after the release of the final standard.

The WLAN receiver is split into two parts. The first one consists of a receive filter, cyclic prefix removal, and a 64-point FFT. This part of the receiver was already used in the 802.11a standard and was not subject to changes to ensure downward compatibility of the MIMO extension to existing WLAN systems. Therefore, this part was implemented on a Xilinx Virtex 2 FPGA using Xilinx Core Generator which comprises highly optimized filters and FFT blocks. The functional chain, including receive filters and FFT is implemented for every receive antenna. The second part of the receiver is the channel estimation and the space-time receiver. This part of the receiver was implemented in Generic C. Using a Generic C code, an automatic code conversion tool was used to generate a Simulink model and assembler code for a TI 6416 DSP. This method allows for co-verification of the implemented Generic C receiver with a Simulink implementation. Furthermore, the method also allows to verify the code after refinement steps for improving the performance of the real-time implementation.

4.4 Adaptive Predistortion

Power amplifiers in wireless systems introduce nonlinear distortions and thus undesired spurious emissions in neighboring bands. In order to reduce such emissions, adaptive predistortion techniques are useful. The nonlinear behavior of the power amplifier is described by a parametric model. In a first step, its parameters have to be estimated. In the following, the *inverse* system is computed and the input of the power amplifier is predistorted by such an inverse system causing the output of the amplifier to be linear again. Narrow band systems (e.g. up to 1 MHz at 2 GHz) can be sufficiently described by memoryless mappings, and inverse systems can be implemented by look-up tables [40]. For systems with higher bandwidth, memory plays a crucial role. Classical approaches use nonlinear models like Hammerstein or Wiener models. However, such models typically lead to a minimization problem that is nonlinear with respect to its parameters. Thus, simple gradient type algorithms tend to run into local minima, not providing better results than memoryless approaches. A recently developed new method [41, 42] computes the predistorted signal of the amplifier directly, rather than computing the inverse explicitly. The corresponding method is an iterative method that approaches the desired quality in typically three steps. In order to check the method, a prototype based on Sundance modules was built applying the VPE. Fig. 6 depicts the results showing strong improvement and at the same time a very good agreement of simulation and real-time measurement results.

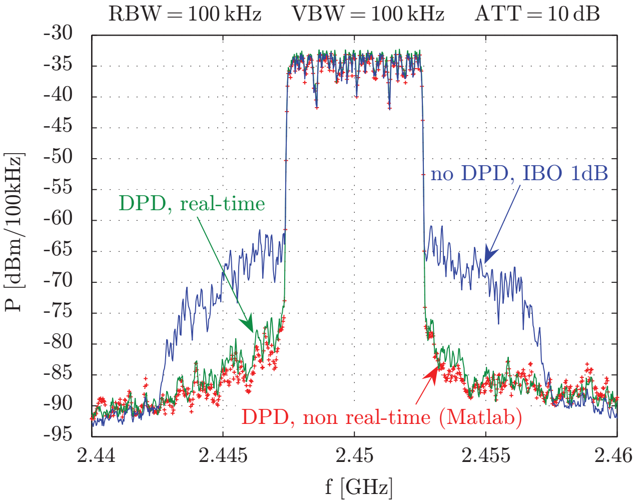

Figure 6: Improvement by predistortion of a power amplifier (DPD = Digital Predistortion, IBO = Input Back-Off).

5 Future Challenges

Modern wireless systems, with their fast exponential increase in complexity, are very challenging when it comes to converting algorithms to run on real-time hardware.

5.1 Methods

Despite many efforts, several problems remain unsolved. Including Hardware-in-the-loop (HIL) techniques for DSPs [36] and FPGAs [7, 37] certainly eases verification a lot. Some companies now offer such techniques [43] and aim in offering techniques that allow to map from Simulink to heterogeneous DSP/FPGA boards. The main problems noted in the introduction, however, float-to-fix conversion, automatic partitioning, and platform based design, remain open. Recently, some work on these topics has started [44, 45] but it will take some time until all of these aspects are included into a smooth design flow. One major problem is the conversion of block based processing, as it is given in Simulink and many other graphical tools, to HW platforms not supporting such processing with relatively large storage buffers between the blocks. While large buffers are needed for high speed simulation (we already mentioned the scheduling slow down in feedback systems in Section 4.1), small buffers are easier to implement. A solution could be utilizing variable buffer sizes when specifying the design, independent of the algorithm. The buffer size can then be set to a large value supporting fast simulation as well as to a small value supporting efficient implementation.

5.2 RF-Frontends

In modern communication systems with powerful digital signal processing, tasks like modulation and pulse shaping are more and more shifted from the analog to the digital domain. ADCs and DACs are the interfacing components of these two domains. Such converters exchange either analog baseband signals or low intermediate frequency signals with the analog frontend. The task of the RF frontend is to perform a linear frequency conversion to a carrier frequency suitable for wireless transmission. The carrier frequency conversion can be carried out in one single step (zero IF) or multiple steps (heterodyne). While highly integrated consumer products use the zero IF principle, in research often the heterodyne conversion is preferred. Due to market pressure, highly integrated consumer RF frontend chips are designed to just meet the minimum requirements of a specific final product. This makes single chip RF frontend solutions useless for wireless testbeds that should support a multitude of different present and future standards. Usually, high design flexibility and fast availability of an RF frontend can only be achieved by a heterodyne design which is inherently much more complex. Thus, the RF frontend of a wireless testbed has to be built employing the heterodyne principle from available state-of-the-art components with lower integration level (e.g. mixers, amplifiers, filters) but much better performance. The main challenge when designing a wireless testbed is therefore to build a high performance RF frontend supporting future standards with today's technology. Note that the availability of components is crucial in the design process of a testbed or prototype since these are often built years before a standard is defined or first products are launched [46].

6 Conclusions

No doubt, rapid prototyping has gained importance. In 2003 the proposal for the NoE NESAT was rejected by the European commission due to its lack of testbeds[3]. Decisions on future wireless systems require a lot of financial investment, making them crucially dependent on the utilized technology. Rapid prototyping can de-risk such decisions long before the investment is made. In this tutorial, we presented means of prototyping. In particular, we distinguished strictly between wireless testbeds and prototypes. We showed, by many examples, how testbeds and prototypes help to overcome technological hurdles without requiring a too high initial investment of time and money. Current challenges are in flexible RF frontends, as well as in the development of automatic design tools.

Acknowledgements

We like to thank many students that have taken part in the development and utilization of our Vienna MIMO Testbed and the Vienna Prototyping Environment. In particular, we like to mention Ernst Aschbacher, Werner Keim, Biljana Badic, Stefan Geirhofer, Georg Brandmayr, and Peter Brunmayr.

[3] Comment of the reviewers: "A real-time testbed is a critical tool, allowing more impact on standardization bodies, however partnership is weak in this area."

References

[1] P. Fisher and D. Cottrell, "Emerging Standards in the Electronic Design Automation (EDA) Industry," in *Electronic Systems Design Seminar*, UC Berkeley, CA, USA, Oct. 1999.

[2] R. Subramanian, "Shannon vs. Moore: the digital signal processing in the broadband age," in *IEEE Communication Workshop*, Aptos, CA, USA, May 1999.

[3] ——, "Shannon vs. Moore: Driving the Evolution of Signal Processing Platforms in Wireless Communications," in *IEEE Workshop on Signal Processing Systems SIPS02*, Oct. 2002.

[4] G. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, vol. 38, no. 8, pp. 114–117, Apr. 1965.

[5] P. Belanovic, B. Knerr, M. Holzer, G. Sauzon, and M. Rupp, "A Consistent Design Methodology for Wireless Embedded Systems," *EURASIP Journal on Applied Signal Processing*, vol. 16, pp. 2598–2612, Oct 2005.

[6] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*, 1st ed. Springer, 2002.

[7] M. Rupp, A. Burg, and E. Beck, "Rapid prototyping for wireless designs: the five-ones approach," *Signal Processing*, vol. 83, no. 7, pp. 1427–1444, July 2003.

[8] T. Kaiser, A. Wilzeck, M. Berentsen, and M. Rupp, "Prototyping for MIMO Systems - an Overview," in *Proc. of the 12th European Signal Processing Conference (EUSIPCO)*, pp. 681–688, Vienna, Austria, Sept. 2004.

[9] A. Burg and M. Rupp, *EURASIP Book on SMART Antennas, Chapter: Demonstrators and Testbeds.* EURASIP, April 2006.

[10] "Merriam-Webster OnLine Dictionary." [Online]. Available: http://www.m-w.com/

[11] *Funk & Wagnalls Standard Dictionary of the English language.* J.G. Ferguson Pub. Co, 1982.

[12] "Homepage of the I-Metra project." [Online]. Available: http://www.ist-imetra.org/

[13] W. Weichselberger, H. Özcelik, M. Herdin, and E. Bonek, "A Novel Stochastic MIMO Channel Model and Its Physical Interpretation," in *Proc. 6th International Symposium on Wireless Personal Multimedia Communications (WPMC'03)*, Yokosuka, Kanagawa, Japan, Oct. 2003.

[14] B. Badic, M. Herdin, M. Rupp, and H. Weinrichter, "Quasi Orthogonal Space-Time Block Codes on Measured MIMO Channels," in *Proc. of the SympoTIC 04*, pp. 17–20, Bratislava, Slovakia, Oct. 2004.

[15] B. Badic, H. Weinrichter, and M. Rupp, "Comparison of Non-Orthogonal Space-Time Block Codes in Correlated Channels," in *Proc. SPAWC 04*, Lisboa, Portugal, July 2004.

[16] S. Caban, C. Mehlführer, R. Langwieser, A. L. Scholtz, and M. Rupp, "Vienna MIMO Testbed," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 54868, 2006.

[17] C. Mehlführer, S. Geirhofer, S. Caban, and M. Rupp, "A Flexible MIMO Testbed with Remote Access," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO 2005)*, Antalya, Turkey, Sept. 2005.

[18] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[19] H. Jafarkhani, "A Quasi Orthogonal Space-Time Block Code," *IEEE Transactions on Communications*, vol. 49, no. 1, pp. 1–4, Jan. 2001.

[20] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[21] C. F. Mecklenbräuker and M. Rupp, "Flexible space-time block codes for trading quality of service against data rate in MIMO UMTS," *EURASIP Journal on Applied Signal Processing*, no. 5, pp. 662–675, May 2004.

[22] B. Badic, M. Rupp, and H. Weinrichter, "Adaptive Channel-Matched Extended Alamouti Space-Time Code Exploiting Partial Feedback," *ETRI*, vol. 26, no. 5, pp. 443–451, Oct. 2004.

[23] B. Badic, H. Weinrichter, and M. Rupp, "Quasi-Orthogonal Space Time Block Codes: Approaching Optimality," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO), tutorial paper*, Antalya, Turkey, Sept. 2005.

[24] S. Caban, C. Mehlführer, A. L. Scholtz, and M. Rupp, "Indoor MIMO Transmissions with Alamouti Space-Time Block Codes," in *Proc. 8th International Symposium on DSP and Communication Systems, DSPCS 2005*, Noosa Heads, Australia, Dec. 2005.

[25] S. Geirhofer, C. Mehlführer, and M. Rupp, "Design and Real-Time Measurement of HSDPA Equalizers," in *Proc. of the 6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2005)*, pp. 166–170, New York City, USA, June 2005.

[26] J. Rasinger, A. L. Scholtz, W. Pichler, and E. Bonek, "A new enhanced-bandwidth internal antenna for portable communication systems," in *Proc. of the 40th IEEE Vehicular Technology Conference 1990*, May 1990.

[27] K. Hirasawa and M. Haneishi, *Analysis, Design, and Measurement of Small and Low-Profile Antennas*, 1st ed. Artech House, Inc., 1992.

[28] C. Mehlführer, S. Caban, M. Rupp, and A. L. Scholtz, "Effect of Transmit and Receive Antenna Configuration on the Throughput of MIMO UMTS Downlink," in *Proc. 8th International Symposium on DSP and Communication Systems, DSPCS 2005*, Noosa Heads, Australia, Dec. 2005.

[29] C. Mehlführer, L. Mayer, R. Langwieser, A. L. Scholtz, and M. Rupp, "Free Space Experiments with MIMO UMTS High Speed Downlink Packet Access," in *Proc. of the 2nd IEE/EURASIP Conference on DSP enabled Radio*, Southampton, UK, Sept. 2005.

[30] 3GPP, "Technical Specification Group Radio Access Network; Multiple-Input Multiple Output in UTRA," 3GPP, Tech. Rep. 25.876 V1.7.0, Aug. 2004.

[31] C. Mehlführer, C. F. Mecklenbräuker, and M. Rupp, "Double Space-Time Transmit Diversity with Subgroup Rate Control for UMTS: Throughput Analysis," in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005*, Pacific Grove, CA, USA, Nov. 2005.

[32] E. Casseau, B. L. Gal, P. Bomel, C. Jego, S. Huet, and E. Martin, "C-based rapid prototyping for digital signal processing," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, Sept. 2005.

[33] M. Rupp, E. Beck, and R. Krishnamoorthy, "Rapid Prototyping for a High Data Rate Wireless Local Loop," in *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems and Computers, 1999*, pp. 993–997, Monterey, CA, USA, Oct. 1999.

[34] A. Adjoudani, E. Beck, A. Burg, G. M. Djuknic, T. Gvoth, D. Haessig, S. Manji, M. Milbrodt, M. Rupp, D. Samardzija, A. Siegel, T. S. II, C. Tran, S. Walker, S. A. Wilkus, and P. Wolniansky, "Prototype Experience for MIMO BLAST over Third-Generation Wireless System," *Special Issue JSAC on MIMO Systems*, vol. 21, pp. 440–451, Apr. 2003.

[35] M. Huemer, M. Lunglmayr, and M. Pfaff, "A lecture course series: from concept engineering to implementation of signal processing algorithms with FPGAs," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, Sept. 2005.

[36] B. Jones, S. Rajagopal, and J. Cavallaro, "Real-time DSP multiprocessor implementation for future wireless base-station receivers," in *TI DSPS Fest, Wireless Applications*, Aug. 2000.

[37] G. Brandmayr, G. Humer, and M. Rupp, "Automatic Co-verification of FPGA Designs in SIMULINK," in *Proc. MBD Conference 2005*, Munich, Germany, June 2005.

[38] G. Meindl-Pfeiffer, R. Kloibhofer, F. Kaltenberger, and G. Humer, "Multistandard Development platform for MIMO Software Defined Radio," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, Sept. 2005.

[39] C. Mehlführer, F. Kaltenberger, M. Rupp, and G. Humer, "A Scalable Rapid Prototyping System for Real-Time MIMO OFDM Transmissions," in *Proc. of the 2nd IEE/EURASIP Conference on DSP enabled Radio*, Southampton, UK, Sept. 2005.

[40] K. J. Muhonen and M. K. R. Krishnamoorthy, "Look-Up Table Techniques for Adaptive Digital Predistortion: A Development and Comparison," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1995–2002, Sept. 2000.

[41] E. Aschbacher, H. Arthaber, and M. Rupp, "A Fast Algorithm for Digital Pre-distortion of Nonlinear Power Amplifiers," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, Sept. 2005.

[42] M. Y. Cheong, E. Aschbacher, P. Brunmayr, M. Rupp, and T. Laakso, "Comparison and Experimental Verification of Two Low-complex Digital Predistortion Methods," in *Proc. of Asilomar Conference*, Monterey, CA, USA, Oct. 2005.

[43] M. Ahmadian, Z. Nazari, N. Nakhaee, and Z. Kostic, "Model based design and SDR," in *Proc. of the 2nd IEE/EURASIP Conference on DSP enabled Radio*, Southampton, UK, Sept. 2005.

[44] P. Belanovic and M. Rupp, "Automated Floating-point to Fixed-point Conversion with the fixify Environment," in *Proc. of RSP 05*, Montreal, Canada, June 2005.

[45] B. Knerr, M. Holzer, and M. Rupp, “Fast Rescheduling of Multi-Rate Systems for HW/SW Partitioning Algorithms,” in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005*, Monterey, CA, USA, Oct. 2005.

[46] R. Langwieser, M. Fischer, A. L. Scholtz, M. Rupp, and G. Humer, “Rapid Prototyping for RF-Transmitters and Receivers,” in *Proc. 5th IASTED International Conference on Communication Systems and Networks (CSN’06)*, Palma de Mallorca, Spain, Aug. 2006.

7 Authors

Markus Rupp was born 1963 in Völklingen, Germany. He received his Dipl.-Ing. degree in 1988 from the University of Saarbrücken, Germany, and his Dr.-Ing. degree in 1993 from the Technische Universität Darmstadt, Germany, where he worked with Eberhardt Hänsler on designing new algorithms for acoustical and electrical echo compensation. From November 1993 until July 1995, he had a postdoctoral position at the University of Santa Barbara, California, with Sanjit Mitra where he worked with Ali H. Sayed on a robustness description of adaptive filters with impacts on neural networks and active noise control. From October 1995 until August 2001, he was a member of the technical staff in the Wireless Technology Research Department, Bell-Labs, where he was working on various topics related to adaptive equalization and rapid implementation for IS-136, 802.11, and UMTS. Since Oktober 2001 he is a Full Professor of digital signal processing in mobile communications at the Vienna University of Technology and since Juli 2002 he is founder and director of the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms. He served as Associate Editor of IEEE Transactions on Signal Processing from 2002-2005, and is currently associated editor of EURASIP Journal on Applied Signal Processing, of EURASIP Journal on Embedded Systems and is an elected AdCom Member of EURASIP. He authored and coauthored more than 200 papers and patents on adaptive filtering, wireless communications, and rapid prototyping.

Christian Mehlführer was born 1979 in Vienna, Austria. In 2004 he received his Dipl.-Ing. degree in electrical engineering from the Vienna University of Technology. Besides his diploma studies he worked part time at Siemens AG where he performed integration tests of GSM carrier units. After finishing his diploma thesis on implementation and real-time testing of space-time block codes at the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, for which he received the Vodafone “Förderpreis 2006” (together with Sebastian Caban), he now is member of the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms and works towards his doctoral thesis at the same institute. His research interests include rapid prototyping, experimental investigation of MIMO systems, the upcoming UMTS MIMO HSDPA mode, the MIMO extension for WLAN (802.11n), and WiMAX (802.16).

Sebastian Caban was born 1980 in Vienna, Austria. Studying electrical engineering at the Vienna University of Technology, he received several scholarships as the best student of the college. After a year at the University of Illinois at Urbana-Champaign, USA, he is currently working as a teaching assistant on his Ph.D. thesis at the university’s Institute of Communications and Radio-Frequency Engineering. On June 23, 2006, he was awarded the “Förderpreis 2006” of the German Vodafone Stiftung für Forschung for his research which focuses on rapid prototyping in wireless communications and multiple antenna systems.

Robert Langwieser was born 1972 in Linz, Austria. He studied telecommunications at the Vienna University of Technology and earned his master degree in April 2004 with distinction. At present Mr. Langwieser is employed as a research assistant and is member of the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms. He currently works on the development of a modular radio-front-end for MIMO measurements.

Lukas W. Mayer was born 1980 in Vienna, Austria. He finished his studies at the Vienna University of Technology in 2005. In his diploma thesis he worked on the radio frequency frontend of the institutes “Vienna MIMO Testbed”. His research interests focus on rapid prototyping in wireless communications, antennas, and MIMO antenna systems. The task of his PhD thesis is to develop small antennas that will be used for a novel RFID system.

Arpad L. Scholtz was born 1947 in Kecskemet, Hungary. In 1956 he moved to Austria and received the Austrian citizenship in 1960. He studied telecommunications at the Vienna University of Technology, where he earned his Masters degree in 1972 and the Ph.D. degree in 1976, both with distinction. From 1972 to 1982, Mr. Scholtz worked as an Assistant Professor for radio-frequency engineering at the Institute of Communications and Radio-Frequency Engineering. Since 1982, he additionally has been serving at the same Institute as a Lecturer, and in 1992 he was given the title of Associate Professor. Arpad L. Scholtz is author or coauthor of more than a hundred scientific publications. He teaches radio-frequency engineering with emphasis on electronic circuit design, antennas, and point-to-point communications.

Reference:

M. Rupp, C. Mehlführer, S. Caban, R. Langwieser, L. W. Mayer, and A. L. Scholtz, "Testbeds and rapid prototyping in wireless system design,"EURASIP Newsletter, vol. 17, no. 3, pp. 32–50, Sept. 2006. [Online]. Available: http://publik.tuwien.ac.at/files/pub-et_11232.pdf

BibTeX:

```
@article{Rupp06_Testbeds,
  author =        {Markus Rupp and
                   Christian Mehlf\"uhrer and
                   Sebastian Caban and
                   Robert Langwieser and
                   Lukas Walter Mayer and
                   Arpad L. Scholtz},
  title =         {Testbeds and Rapid Prototyping in Wireless System Design},
  journal =       {EURASIP Newsletter},
  volume =        17,
  number =        3,
  year =          2006,
  month =         sep,
  pages =         {32--50},
  ISSN =          {1687-1421},
  url =           {http://publik.tuwien.ac.at/files/pub-et_11232.pdf}
}
```