Proceedings of the
2nd Karlsruhe Workshop
on Software Radios

Karlsruhe (Germany)
March 20/21, 2002
Preface

During the last 10 years software radio has emerged from a niche area to become a cornerstone of modern system strategies for future-proof and truly universal mobile telecommunications. Taken into account the many contributions in magazines and journals since, we may observe a certain partitioning of the community.

On one hand, visionaries and managers continue to think about business models for systems and services, about the impact of globalization and productivity of the software radio idea. Their primary concern is to understand and to predict its implications for the future of mobile radio. On the other hand, researchers and developing engineers - sometimes the same who came up with an original system idea before - pick specific problems of transceiver design. There is a tremendous effort to technically solve software radio problems, to improve existing techniques, to propose new ones.

The noteworthy character of the community was reflected in the First Karlsruhe Workshop on Software Radios in the year 2000, and people were amazed by the richness and diversity of discussions. Both national and international academia as well as industry did contribute from their particular point of view, casting a promising light onto the event.

Today, the Communications Engineering Lab of the University of Karlsruhe is glad to welcome you to the Second Karlsruhe Workshop on Software Radios. An interesting program of 26 contributions (14 from industry and 12 from academia awaits you, where presentations are centered around six main topics

- Mobile Multipath Channel
- Selected Components
- Radio Front-End & Receiver Architectures
- Baseband Signal Processing
- Protocols
- Terminal Reconfiguration & Cognitive Radio

The authors come from eight countries (Finland, Italy, The Netherlands, United Kingdom, France, Hungary, India and Germany), thus making WSR2002 a truly international workshop. Within the wide field of software radio their contributions cover an area as extensive as two days of workshop time allow.

The organization of the workshop is meant to offer plenty of time for breaks, conversations and meetings with colleagues. We wish all participants two days full of interesting lectures, fruitful discussions, and some leisure time to become familiar with the University and the nice City of Karlsruhe.

Friedrich Jondral
Arnd-Ragnar Rhiemeier
## Contents

### Session 1  -  Mobile Multipath Channel

**High Resolution Direction Finding with Influences of Radiation Coupling**  
H. Mewes (Fachhochschule Aschaffenburg, Germany) ............................................................... 1

**Channel Diversity in a Mobile Software Radio**  
H. Schnepp, J.-F. Luy, T. Müller (DaimlerChrysler Research, Ulm, Germany), P. Russer (Technical University of Munich, Germany) ............................................................... 7

### Session 2  -  Selected Components

**Clock Generator Phase Noise in RF Sampling Receivers**  
T. Müller, S. Herter, J.-F. Luy (DaimlerChrysler Research, Ulm, Germany) ................................ 13

**Decimation by Non-Integer Factor in Multistandard and Software Radio Receivers**  
D. Babic, M. Renfors (Tampere University of Technology, Finland) ........................................ 19

**High Performance Decimation Filter for Digital Down Converters in Software Radio Applications**  
L. Lo Presti, C. Spadea, Z. Tarussio (Politecnico di Torino, Italy) ............................................ 25

**Extended Algorithms for Sample Rate Conversion**  
M. Henker, G. Fettweis (Technical University of Dresden, Germany) ........................................ 33

### Session 3  -  Radio Front-End & Receiver Architectures

**Specification for Digital Channel Selection Filters in a Bluetooth Capable HiperLAN/2 Receiver**  
F. Hoeksema, R. Schiphorst, K. Slump (University of Twente, The Netherlands) ......................... 41

**Digital I/Q Imbalance Compensation in Direct Conversion Receivers**  
M. Valkama, M. Renfors (Tampere University of Technology, Finland) ..................................... 51

**A Software Defined Radio Front-End Implementation**  
S. Abendroth (University of Karlsruhe, Germany), R. Heuze (ESIEE, Paris, France), S. Weiss (University of Southampton, United Kingdom) ......................................................... 57

### Session 4  -  Baseband Signal Processing

**Baseband Aspects of a Direct Conversion Receiver Concept Utilising Five-Port Technology**  
J.-U. Jürgensen, D. Krupezevic, M. Ratni, Z. Wang (SONY, Stuttgart, Germany) ....................... 63

**Software Radio Concept of the Receiver for Digital Broadcasting Services**  
M. Sliskovic, H. Schöpp (Harman/Becker Automotive Systems, Germany) ................................. 67

**Some Theoretical Limits for a Blind Indentification of the Telecommunication Standard in Use with RBF Neural Networks**  
C. Roland, J. Palicot (IRISA, Rennes, France) ............................................................................... 73

**Comparison of Signal Processing for Spatio-Temporal Multi-User Equalization in Multi- and Single-Carrier CDMA Systems**  
L. Brühl, B. Rembold (RWTH Aachen, Germany) ....................................................................... 81
Session 5 - Baseband Signal Processing

**Software Defined Radio Technology for Multistandard Terminals**
J. Brakensiek, R. Wittmann, M. Darianian (NOKIA Research Center, Bochum, Germany) …… 87

**Programmable Hardware Accelerator for Universal Telecommunication Applications**
B. Oelkrug, D. Bücke, D. Uffmann, A. Dröge, J. Brakensiek, M. Darianian
(NOKIA Research Center, Bochum, Germany) …………………………………………………… 93

**Bluetooth Demodulation Algorithms and their Performance**
R. Schiphorst, F. Hoeksema, K. Slump (University of Twente, The Netherlands) ……… 99

**Benefits and Limits of Parameterized Channel Coding for Software Radio**
A.-R. Rhiemeier (University of Karlsruhe, Germany) …………………………………………… 107

Session 6 - Baseband Signal Processing

**Resource Management in Software Radio Architectures**
G. Rábai, S. Imre, J. Kovács, P. Kacsuk
(MTA SZTAKI, Computer and Automation Research Institute, Budapest, Hungary) ……… 113

**Reconfiguration in Software Radio Systemss**
A. Kountouris, Ch. Moy (Mitsubishi Electric, Rennes, France) ……………………………… 119

**Management and Control of Reconfiguration Procedures in Software Radio Terminals**
S. Gultchev, K. Moessner, R. Tafazolli
(Mobile VCE Team, University of Surrey, United Kingdom) ……………………………… 125

Session 7 - Protocols & Terminal Reconfiguration

**Conformance and Interoperability – Testing Strategies for Software Dominated Digital Mobile Communication Terminals**
R. Schmale (NOKIA Research Center, Bochum, Germany) ……………………………… 131

**Supporting Programmable Handoff Architectures in Wireless Access Networks through Coordination Contracts**
S. Jamadagni, P.S. Subramanian
(Sasken Communications Technologies Ltd, Bangalore, India) ……………………………… 137

**An Efficient QoS Negotiation Protocol for Software Download Channel Establishment in Software Defined Radio**
Kishore K. R. Krishna (Sasken Communications Technologies Ltd, Bangalore, India) ……… 145

Session 8 - Terminal Reconfiguration & Cognitive Radio

**Software Download Management for Cell Broadcast Channels in WCDMA**
E. Mohyeldin, J. Luo, M. Dillinger (SIEMENS AG, Munich, Germany) …………………………… 149

**Radio Ressource Management Schemes Supporting Reconfigurable Terminals**
J. Luo, M. Dillinger, E. Mohyeldin (SIEMENS AG, Munich, Germany) …………………………… 157

**A Smart Space Enabled SDR Environment for Enterprise and Home Networks**
S. Jamadagni, S. Jamindar
(Sasken Communications Technologies Ltd, Bangalore, India) ……………………………… 163
ABSTRACT

Due to the progress in the signal processing hardware, high resolution spectral estimation algorithms now become applicable in practical systems. One favor application is radio direction finding (DF) used for navigation and electronic reconnaissance. Compared with classical DF techniques the advantage of the high resolution methods is their better resolution capability in interference fields and the ability to use nearly every possible antenna array geometry. Nevertheless, the better resolution capability may decrease in practice due to model and DF system errors. In this paper the influence of radiation coupling on the well known MUSIC algorithm (MUltiple Signal Classification) [1] is described. The radiation coupling is simulated with an antenna calculation program (SuperNEC). The resolution capability is enhanced if the radiation coupling is taken into account.

1. INTRODUCTION

Since several years, software radio solutions play an important role in radio reconnaissance and radio direction finding. The received signals are converted to suitable frequency range and A/D-converted. Demodulation, analyzing signals and other tasks like detection of wideband signals in noise or DF algorithms are then applied in the "digital domain". This has many advantages. Different algorithms may be used with the same antenna and receiver hardware, adaptive algorithms may be used and new techniques, which do not have an analog counterpart, may be implemented. Much attention has been payed on high resolution spectral estimation techniques for direction finding in the last two decades. Compared with classical approaches like the periodogram [2], which is based on the Fourier transform, these methods show a better resolution capability. Depending on the signal (time or spatial) to be analyzed, high resolution methods improve the capability to distinguish between two closely spaced sources in the frequency (time signals) or wave vector/angle domain (spatial signals). In radio direction finding (DF) applications the high resolution methods may be used to resolve interference fields with more than one incoming wave, even when the directions of arrival (DOA) are not well separated. Due to improvements in the technology of A/D converters and modern signal processing hardware the algorithms now become available for practical implementation. A problem in practical applications is the mutual coupling between different antenna elements and other conducting structures in the vicinity of the DF antenna array. Antenna simulation programs may be used to calculate the antenna characteristics. The results of the calculations can be used as additional information for the DF algorithm.

In the first part of the chapter a short review of the high resolution method MUSIC will be given. A typical multichannel DF software radio system will be shown, followed by an example showing the difference between classical and high resolution DF algorithms. The influence of radiation coupling and the difference between DF results with and without consideration of coupling effects is considered. The paper ends with a short summary.

2. THE MUSIC ALGORITHM

The high resolution method MUSIC was first published by R. O. Schmidt in 1979 [1]. It may be used to estimate the following parameters of the wave field:

- number of incoming waves
- directions of arrival
- signal power and signal correlation
- polarization

We consider the narrow band case and assume that the antennas are located in the far field of the sources. Figure 1 shows an antenna array with one incoming wave from direction $\alpha$ with wavelength $\lambda$. The $M$ antennas are sampling the wave field at various locations. Although a linear array is shown, the antennas may be located arbitrarily. For $D$ incoming waves with complex signal amplitudes $s_j$ forming the vector $\mathbf{s}$ the received vector of antenna voltages $\mathbf{x}$ can then be modeled as

$$
\begin{align*}
\mathbf{x} &= \begin{bmatrix}
\mathbf{e}_1(\alpha_1) & \mathbf{e}_2(\alpha_2) & \cdots & \mathbf{e}_D(\alpha_D)
\end{bmatrix} \\
\mathbf{e}_2(\alpha_1) & \mathbf{e}_2(\alpha_2) & \cdots & \mathbf{e}_2(\alpha_D) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{e}_M(\alpha_1) & \mathbf{e}_M(\alpha_2) & \cdots & \mathbf{e}_M(\alpha_D)
\end{bmatrix}
\begin{bmatrix}
\mathbf{s}_1 \\
\mathbf{s}_2 \\
\vdots \\
\mathbf{s}_D
\end{bmatrix}
+ \\
\begin{bmatrix}
\mathbf{w}_1 \\
\mathbf{w}_2 \\
\vdots \\
\mathbf{w}_D
\end{bmatrix}
\end{align*}
$$

(1)
The M x D matrix $E$ represents the response of the antenna array to signals incident from directions $\alpha_1, \alpha_2, \ldots, \alpha_D$ and can be found either analytically, by array calibration or by calculating the antenna characteristics with an antenna analysis program. The vector $w$ describes additional atmospheric and receiver noise. Calculating the correlation matrix $R$ of the antenna voltages and considering that signals and noise are uncorrelated gives

$$R = \mathbb{E}[xx^H] = \mathbb{E}E^H + \sigma_n^2I,$$

where $\mathbb{E} \{ \ldots \}$ denotes the expected value operator and the superscript $^H$ the conjugated transposition operator. $P = \{ss^H\}$ is the expected value of the signal amplitude vector $s$, $\sigma_n^2$ the noisepower and $I$ the identity matrix. The eigenvalues of the hermitian matrix $R$ are real valued. It can be shown [3] that they are

$$\lambda_i = \begin{cases} \lambda_{is} + \sigma_n^2, i = 1, \ldots, D \\ \sigma_n^2, \ i = D + 1, \ldots, M \end{cases}$$

Hence, we have $D$ eigenvalues greater than the noisepower and $(M-D)$ eigenvalues equal to the noisepower $\sigma_n^2$. The space spanned by the corresponding eigenvectors $q_i$, $i = 1 \ldots M$, is called the observation space. It can be partitioned into two subspaces, the signal subspace, spanned by the eigenvectors corresponding to the $D$ greatest eigenvalues, and the noise subspace, spanned by the eigenvectors corresponding to the remaining eigenvalues. All eigenvectors associated with the $(M-D)$ smallest eigenvalues of $R$ satisfy the relation

$$Rq_i = \sigma_n^2 \cdot q_i, \ i = D + 1, \ldots, M$$

$$\Rightarrow (R - \sigma_n^2I) \cdot q_i = 0, i = D + 1, \ldots, M.$$  

With (3) we have

$$\mathbb{E}E^H q_i = 0, i = D + 1, \ldots, M.$$  

Since $E$ has full column rank $D$ and the matrix $P$ is diagonal with all entries being nonzero, it follows

$$e^H(\alpha_d) \cdot q_i = 0, i = D + 1, \ldots, M; \ d = 1, \ldots, D$$

Therefore the hermitian transposed column vectors $e$ of the matrix $E$, which are called steering vectors, and the eigenvectors of the noise subspace are orthogonal. This relationship is used to form the MUSIC spectrum

$$P_{\text{MUSIC}}(\alpha) = \frac{1}{\sum_{i=D+1}^M |e^H(\alpha) \cdot q_i|}$$

where the denominator is calculated by summing over all squared products (7) of the noise subspace. The angle $\alpha$ is scanned over all possible DOA. The peaks in the spectrum denote the estimated directions of arrival. For omnidirectional antennas without mutual coupling the $m$'th element of $e$ is given by

$$e_m = \exp\left[-\frac{2\pi}{\lambda}(x_m \cdot \sin \alpha + y_m \cdot \cos \alpha)\right]$$

where $x_m$ and $y_m$ are the coordinates of the antenna element $m$. In practical applications the correlation matrix $R$ is not available, so a time averaged estimate must be used. This estimate $\tilde{R}$ is formed by taking $N$ snapshots of the antenna voltage vector $x$, which yields

$$\tilde{R} = \frac{1}{N} \sum_{i=1}^N xx^H$$

The number of incident waves $D$ can be estimated by the number of eigenvalues equal to $\sigma_n^2$. 

---

**Figure 1.** Antenna array with incoming electromagnetic wave
3. PRACTICAL APPLICATION OF THE MUSIC ALGORITHM

Figure 2 shows a DF system for applying high resolution DF algorithms. The system consists of an antenna array, a M channel receiver to convert the HF signals to an intermediate frequency, a DSP (Digital Signal Processing) unit with A/D converters and signal processors and an MMI, for example a personal computer.

Figure 2: Multichannel receiver system for DF applications

The system has been successfully used in several measurement campaigns in the HF (3 – 30 MHz) and the VHF (30 – 80 MHz) band. The antennas may be located arbitrarily. An example shows the results for a bearing of HF broadcasting stations on September 8, 1994, frequency f = 15.400 MHz. The DF equipment with a circular array (diameter 20m) was located in the north of Hamburg at 9.92° East, 53.70° North. The transmitters in table 1 are reported in a transmitter handbook for the frequency f = 15.400 MHz [4]. The transmitter Moscow was within the receiver bandwidth.

Location | Long. | Lat. | Bearing
---|---|---|---
Pori, FNL | 21.87°E | 61.47°N | 34.6°
Dubai, UAE | 55.27°E | 25.23°N | 112.0°
Ascension, ASC | 14.38°W | 7.90°S | 206.7°
Wooferton, GB | 2.72°W | 52.32°N | 264.8°

On frequency f = 15.405 MHz:
Moscow, RUS | 37.63°E | 55.75°N | 71.5°

Table 1: Locations of transmitters on frequency f = 15.400 MHz

Figure 3 shows the measurement results for the high resolution method MUSIC and for classical spectrum estimation with the Fourier algorithm. It should be noted that all spectra are calculated from the same measured antenna voltages. Table 2 shows the calculated directions of arrival (DOA) of the two methods. The MUSIC algorithm shows a very good correspondence between the measured and the desired values, the Fourier method is unable to resolve the five incoming waves.

Desired value | MUSIC | Fourier
---|---|---
34.6° | 32° | 35°
71.5° | 72° | 94°
112.0° | 114° | 191°
206.7° | 205° | 332°
264.0° | 263° | 261°

Table 2: DF Results for a circular array, five transmitters on frequency f = 15.400 MHz, MUSIC and Fourier algorithm

(a) Bearing [Deg.]

(b) Bearing [Deg.]

Figure 3: Spectra for five incoming waves, circular array with 9 antennas, diameter d = 20m, f = 15.4MHz, measurement results, dashed line gives the estimated DOA, short solid line denotes true DOA

a) MUSIC algorithm, b) Fourier algorithm
More detailed results can be found in [5], [6].

4. RADIATION COUPLING BETWEEN ANTENNAS

Modern antenna calculation programs like Super-NEC from Poynting Software, South Africa, allow the calculation of radiation couplings between antenna elements and other conducting structures. Figure 4 shows a typical antenna configuration, an Adcock DF antenna consisting of eight antenna elements. Instead of using eq. (9) for calculating the response of antennas to incoming waves from different directions, the responses are calculated with SuperNEC. This can be accomplished by creating a model of the antenna and additional conducting structures in the vicinity of the antenna. The program allows plane wave excitation of the antennas with incoming waves from all possible directions. Simulating the antenna with plane wave excitation results in a received antenna voltage for each antenna element. These voltages at the antenna feeding points are being calculated for every interesting DOA. The vectors $e$ of the voltages are then used for calculating the MUSIC spectrum (eq. (8)).

Figure 4: Defining an antenna geometry in SuperNEC

To compare the influence of radiation coupling on the performance of the MUSIC DF algorithm, we made simulations using a DF Antenna from the C. Plath GmbH, Hamburg, model A6242H8 for the VHF band. Figure 5 shows the results of a simulation. The wave field is calculated with SuperNEC, thus taking the radiation coupling between the antennas into account. The steering vector $e$ is calculated with SuperNEC with the influence of radiation coupling. As in the former example three incoming waves from $-40^\circ$, $0^\circ$ and $5^\circ$ are simulated. White noise was assumed with signal to noise ratio of 50 dB. All signals have the same power. The lines marked with a plus show the true directions of arrival, the line marked with a circle denotes the maximum in the DF spectrum, i.e., the estimated direction of arrival. It can be seen that the algorithm is not able to resolve the three incoming waves. This behaviour is caused by the difference between the true antenna voltages received by the antennas, which are influenced by the mutual coupling between different antenna elements, and the ideal antenna voltages calculated with eq. (9), which are used to calculate the MUSIC spectrum.

Figure 5: Simulation results, MUSIC algorithm, 3 incoming waves, ideal (analytically) calculated steering vector, three incoming waves from $-40^\circ$, $-5^\circ$ and $0^\circ$, DF result $-10.5^\circ$

Figure 6: Simulation results, MUSIC algorithm, 3 incoming waves, steering vector calculated with SuperNEC, three incoming waves from $-40^\circ$, $0^\circ$ and $5^\circ$, DF results $-40^\circ$, $0^\circ$ and $5^\circ$

Figure 6 shows the results of another simulation. The wave field is calculated with SuperNEC, again taking the radiation coupling between the antennas into account. The steering vector $e$ is also calculated with SuperNEC with the influence of radiation coupling. As in the former example three incoming
waves from -40°, 0° and 5° are simulated. Now the algorithm is able to resolve the three incoming waves, because the radiation coupling is now taken into account in the calculation of the MUSIC spectrum.

The next example shows the possibility to include conducting structures in the vicinity of the DF antenna. A complete ship has been simulated in conjunction with the DF antenna. Figure 7 shows the simulated example.

Figure 7: Model of a ship with an Adcock DF antenna for simulation with SuperNEC

Figure 8 shows the simulated MUSIC spectrum, which has been calculated including influences of radiation couplings between the antenna elements and between antenna elements and other parts of the ship. Three incoming waves from -5°, 0° and 40° are being simulated. It can be seen that all incoming waves are being resolved. Our simulations have shown that the MUSIC algorithm is not able to resolve the waves if the mutual coupling is not taken into account.

Figure 8: Simulation results for antenna and ship, MUSIC algorithm, 3 incoming waves, steering vector calculated with SuperNEC

5. CONCLUSIONS AND FURTHER WORK

To enhance the resolution capability of high resolution DF algorithms, it is possible to take the antenna characteristics into account. This can be accomplished by using antenna calculation programs like SuperNEC. The simulation results show, that this results into a better resolution capability of the algorithm, i.e. in better DF results. In further investigations, the simulations shall be verified by measurements with a multichannel receiver system, which is actually under development.

6. REFERENCES


Acknowledgement. The project has been funded by the German ministry for education and research Bundesministerium für Bildung und Forschung (BMBf), contract number 1703200, within the program “Anwendungsorientierte Forschung und Entwicklung an Fachhochschulen“. The author would also like to thank the C. Plath GmbH, Hamburg, Germany, for supporting the project, and Mr. Michael Dermühl for doing many simulations with SuperNEC.
CHANNEL DIVERSITY IN A MOBILE SOFTWARE RADIO

H. Schnepp¹, J.-F. Luy¹, T. Müller¹, and P. Russer²

¹DaimlerChrysler AG, Research and Technology REM/CM, 89081 Ulm, Germany
²Technische Universität München, Institute for High-Frequency Engineering, 80333 München, Germany

ABSTRACT

A novel system is proposed and demonstrated which leads to an increase in signal to noise ratio in reception of broadcasting signals. This will be done by a coherent superposition of the transmitted signals from different transmitters at different frequencies with the same information content. A demonstrator for the reception of FM broadcasting is presented.

I. INTRODUCTION

Several concepts to improve FM radio receiving conditions have been presented in literature. A major cause for low receiving quality in mobile receivers is multipath fading, which yields strong attenuation of the received signal. Methods based on the constant modulus algorithm (CMA) can avoid some of the distortion introduced by multipath effects [1]. Unfortunately these algorithms yield problems in the case of fast fading conditions, since they cannot track the fast variations of the channel parameters [2]. In the case of multipath fading antenna diversity [3] allows to increase the signal quality at the expense of the use of several antennas.

Our proposed channel diversity concept makes use of the fact that the same radio information is supplied by various transmitters at different frequencies [4]. The channel diversity approach implements a coherent superposition of these multiple information to increase the reception quality. The main advantage of the method presented is its capability to compensate fast multipath fading, since it is unlikely that the multipath fading signal minima of two channels will occur at the same location [5]. This paper presents the channel diversity concept based on a direct digitizing software radio architecture.

II. CONCEPT OF CHANNEL DIVERSITY

A. The Simple Channel Model

As depicted in Fig. 1, a mobile receiver normally can choose between several broadcast transmitters supplying the same program.

Starting with the source signal, m(t), we define the transmitted signal, s_{tx,i}(t), originating from broadcast station number i as s_{tx,i}(t) = g(m(t)), where g(t) denotes the modulation function.

At the single antenna the received signal, s_{rx,i}(t), from the i-th transmitter arrives convolved by the respective impulse response, h_i(t), yielding:

\[ s_{rx,i}(t) = s_{tx,i}(t) * h_i(t) = \int_{-\infty}^{\infty} s_{tx,i}(t') h_i(t-t') dt' \] (1)

For free space propagation the transfer function \( H_i(\omega) \) depends on the distance, \( r_i \), the gain of the transmitter antenna, \( G_t \), the effective receiver antenna area, \( A_r \), and the time delay, \( \tau(r_i) = r_i/c_0 \):

\[ H_i(\omega) = \frac{G_t A_r}{4\pi r_i^2} e^{-j\omega \tau(r_i)} = a_i(r_i)e^{-j\omega \tau(r_i)} \] (2)

where \( c_0 \) is the velocity of light. In realistic scenarios the dependence on \( a_i(r_i) \) on the the location \( r_i \) is more complicated.

The impulse response, \( h_i(t) \), is linked to the transfer function, \( H_i(\omega) \), by the Fourier transform \( H_i(\omega) \leftrightarrow h_i(t) \). The impulse response of channel \( i \) therefore is given by:

\[ h_i(t) = a_i(r_i) \delta(t - \tau_i(r_i)) \leftrightarrow a_i(r_i)e^{-j\omega \tau(r_i)} = H_i(\omega) \] (3)

Solving the integral in (1) this leads to the simplified channel model:

\[ s_{rx,i,\text{noisefree}}(t) = \int_{-\infty}^{\infty} s_{tx,i}(t') a_i(r_i) \delta(t - \tau_i(r_i) - t') dt' = a_i(r_i) s_{tx,i}(t - \tau_i(r_i)) \] (4)
Regarding the situation shown in Fig. 1 we introduce noise that comes up in the LNA of the receiver due to poor signal amplitudes and derive from (4)

\[ s_{r,1}(t) = a_1 s_{r,1}(t - \tau_1) + n_1(t) \]

where \( n_1(t) \) denotes the additional noise for each channel \( i \).

B. Achievable Improvement in Signal-to-Noise Ratio in Stationary Receivers

Assuming a moving vehicle the distances \( r_i \) change with time. As these changes are slower compared with \( (2\pi \omega)^{-1} \), we assume \( \tau_i \) and \( a_i \) locally constant for the remainder of this section.

After compensation of the time shift, \( \tau_i \), as described in section IV, it is possible to calculate the weighted sum - with coefficients \( b_i \) - of all signal components from the various transmitters \( i \):

\[ s_{r,\text{comb}}(t) = b_1 s_{r,1}(t + \tau_1) + b_2 s_{r,2}(t + \tau_2) + \ldots \]

\[ \ldots + b_N s_{r,N}(t + \tau_N) = \sum_{i=1}^{N} b_i s_{r,i}(t + \tau_i) \]

(6)

It can be shown that setting the weighting factors \( b_i \) to

\[ b_i = \frac{a_i \sqrt{S_{r,i}}}{n_i^2} \]

(7)
yields the maximum possible gain in the signal-to-noise ratio (SNR) [6].

The signal-to-noise ratio achieved by the forming of the weighted sum of the components of the received signals is equal to the sum of all \( N \) signal-to-noise-power ratios involved in the summation:

\[ SNR_{\text{comb}} = \sum_{i=1}^{N} SNR_i \]

(8)

It must be noted, that setting the weighting coefficients according to (7) only leads to the maximum increase in the SNR as stated in (8), if the additional noise terms are not correlated between the channels.

Fig. 2 displays the achieved improvement in the signal-to-noise ratio according to (8) for the case of two channels. The \( x \)-axis depicts the ratio between the SNR of channel 1 and channel 2, whereas the \( y \)-axis shows the improvement of the SNR of the optimum combined signal compared to the SNR of the single signal with the higher SNR. This figure shows the advantage of the optimum combined summation over a simple system, just choosing the signal with the better SNR.

C. Additional Advantage for FM-Modulated Signals

Modulation schemes showing threshold characteristics when the noise is increasing yield a higher improvement in the signal-to-noise ratio if the coherent superposition of the signals is done before demodulation. This is valid for FM modulated systems where the FM threshold will be improved due to the increased signal-to-noise ratio now available for the demodulator. Fig. 3 shows the expected reduction of the FM threshold. This figure was obtained by a Simulation of a FM demodulator either with a certain SNR at the input or a SNR increased by 3dB which is the maximum possible increment of channel diversity applied in a two channel case. It should be noted that the superposition before demodulation presses higher constraints on the accuracy of the calculation of the time delays \( \tau_i \).

Due to the advantages of channel diversity in connection with FM modulated signals and the widespread coverage of FM broadcast we turn our focus on a demonstration application for the use with FM radio receivers.

D. Applicability of Channel Diversity

Fig. 4 displays the area between two FM transmitters where channel diversity is applicable. This picture is based on a simple channel model only implementing the free space attenuation \( A \sim 1/r^2 \).
III. SPECIAL CONSIDERATIONS FOR MOBILE RECEIVERS

A. Extended Channel Model

The derived channel model from (5) in section II-A covers attenuation, time delay and additional noise, but not yet multipath effects. However, in mobile receivers multipath effects exhibit a strong influence on receiving signal quality.

Considering multipath fading of each individual channel the multipath model derived by Reiter [5] can be applied. The amplitude $a_{\text{multipath}}$ of the resulting multipath signal can be written as

$$a_{\text{multipath}}(r) = \sum_{k=1}^{M} a_k \cdot e^{-j \frac{2\pi}{\lambda} f_c r \cos \phi_k}, \quad (9)$$

where $M$ denotes the number of incident paths, $a_k$ the amplitude, $\phi_k$ the incident angle of the $k$th path, $r$ is the position of measurement relative to the transmitter, $f_c$ is the carrier frequency of the transmitted signal and $\lambda$ finally denotes the velocity of light.

It has to be noted, that the set of superposed waves forming $a_{\text{multipath}}$ in (9) arrive at the same frequency at the antenna. So there is no possibility for the receiver to split the incoming multipath signal into its components, save by the use of some distinct equalisation algorithms like CMA. This is in contrast to the channel diversity model depicted in Fig. 1, which describes the reception at different frequencies, and the single components of the incoming signals are therefore accessible by the receiver.

The combined model for the channel diversity channel and inclusion of multipath effects leads to the following received signal

$$s_{r,i}(t) = a_{\text{multipath}}(r_i) \cdot s_{t,i}(t - \tau_i) + n_i(t) = \sum_{k=1}^{M} a_k \cdot e^{-j \frac{2\pi}{\lambda} f_c r_i \cos \phi_k} \cdot s_{t,i}(t - \tau_i) + n_i(t) \quad (10)$$

for the channel $i$, which results from a combination of equation (5) and (9). The subscript $k$, $i$ denotes the $k$th multipath wave within the $i$th channel to be received.

B. Achievable Improvement in Signal-to-Noise Ratio in Mobile Receivers

Fig. 5 displays simulation results for the amplitude of multipath disturbed signals. As a low signal amplitude at the input of the receiver leads to additional noise that is introduced by the amplifier in the frontend, the amplitude level can be seen as a relative signal-to-noise ratio, which is displayed in this figure. Two incoming channels are shown with a difference of 10 dB in the amplitude respective SNR. The solid line depicts the SNR of the signal which results from an optimum combination of the signals.

It can be easily seen, that even a weak signal may help to improve the SNR of a strong signal, when multipath conditions occur.

A mobile receiver changes the position to the transmitter with time. Therefore the distance scale of the $x$-axis can be transformed to a time scale introducing time variant amplitudes.

Fig. 6 shows the probability function of the received SNR for the channels displayed in Fig. 5, pointing out the improvement of SNR by channel diversity in multipath environments.

Especially in moving vehicles the effect of channel diversity on the reduction of multipath distortion may lead to a higher improvement of the signal quality as the gain of up to 3 dB derived in section II-B for the stationary case.
Fig. 6. Increase in receiving quality for the two channels from Fig. 5 — depicted as probability function.

IV. TRACKING

As mentioned before, tracking has to be used to achieve a good performance for channel diversity as well to follow the changes in the time delay between the channels as for the acquisition of the signal quality.

A. Delay Tracking

Two cases have to be considered: delay tracking before and after the demodulation. The constraints on the tracking system are higher in the case of tracking before demodulation as the sampling rate of the signals has to be higher because of the frequency spreading by FM modulation.

In [7] we show that the amplitude error, $\Delta A$, denoting the deviation of the actual signal amplitude from the theoretically achievable amplitude, is caused by an error in the time delay $\Delta t$:

$$\Delta A = 1 - \cos \left( \frac{\omega \Delta t}{2} \right)$$

(11)

where $f = \omega / 2\pi$ denotes the maximum signal frequency that has to be considered. Demanding a maximum amplitude error of 0.1 dB equation (11) leads to a maximum phase error of 25° referring to the actual frequency.

Two tracking concepts will be proposed. First is the tracking by calculating the correlation between the different signals to extract the time delay as the peak in the correlation result. This concept will use a lot of computational resources as the signals are examined block by block for every computed delay value.

As the sampling ratio limits the time resolution of the correlation an oversampling ratio of about 8 should be chosen to obtain a time resolution that yields amplitude errors smaller than 0.1 dB (see equation (11)).

Fig. 7 shows a plot of the calculated correlation peak with a sampling rate of 44.1 kHz in the audio frequency range and a block size of 1024. The data was collected from measurements in an urban area at a speed of about 15 m/s.

The other concept consists of a tau-dither-loop [8] depicted in Fig. 8, which calculates the correlation at two points on both sides of the correlation peak and outputs continuously the correction for the time delay via a gradient method consuming little computational effort. Using polyphase filters, the oversampling ratio can be reduced compared to above mentioned correlation tracking algorithm.

For utilizing the benefits of both of these tracking concepts these should be used in combination. Correlation tracking will be used in the initializing phase to seek the starting values because of its large seeking range, while the tau-dither-loop will be used to track a once detected maximum.

B. Quality Tracking

Due to the movement of the mobile receiver in a multipath environment, fast fading occurs, as described in section III-B. The typical distance between two deep fades is on the order of a wavelength, which is around 3 m for FM broadcasting (Fig. 5). When the receiver moves through this waveform, the spatial separation of deep fades transfer to changes in time. At typical speeds of cars of up to 50 m/s this leads to a time separation of several 10 ms, which demands the reaction time of the quality signal detector to be on the order of ms and below to achieve a good signal quality guess.
V. IMPLEMENTATION OF CHANNEL DIVERSITY

A. Channel Diversity in a Software Radio Architecture

Channel diversity can be included in a software radio architecture as depicted in Fig. 9 [9]. The whole FM band will be directly digitized after a low noise amplifier and an antialiasing filter at the antenna.

The sampling frequency will be 56 MHz, being sufficient for direct digitizing the whole FM band via subsampling, as depicted in Fig. 10. The antialiasing filter must have enough suppression of the adjacent bands to avoid distortion in the FM-band.

For the used sampling frequency of 56 MHz the requirements on the antialiasing filter are very strict. A higher sampling frequency like for example 78 MHz — which is the next possible suitable subsampling frequency — may lower the requirements on the antialiasing filter. Moreover the usable SNR for a single channel is increased because of the higher sampling clock, which on the other side leads to higher constraints on the speed of the signal processing due to the higher data rate.

The resolution of the A/D-converter we use in our software radio architecture is 12 bit. This leads to a theoretical SNR of 74 dB at the input of the decimating stages. After decimation by a factor of about 90...180 the resulting decimation gain in the SNR is between 19.5 and 22.5 dB. This yields a dynamic range of more than 93 dB for a single FM channel. However, one has to consider the harmonic distortion and the spurious frequencies, reducing the usable SNR of the overall system.

The data rate before the demodulation should be either around 600 kbps or 300 kbps to enable proper channel filtering and demodulation of the signals.

Due to the digital signal processing of data from a single analog to digital converter, the coherency between the channels is secured. To reduce the necessary computational resources the diversity combining and tracking is moved behind the decimating filters. As the mixing and filtering is done in the digital domain this will not reduce the quality from unbalances between the channels.

B. Experimental Setup

The measurement setup is built by two linked signal processing cards implemented in a PC. These cards are both linked to an external analog-to-digital converter which is driven by a very low jitter PLL clock source. The setup is built in a way that it can be used in our test car for mobile measurements.

At this stage of our work the tracking is not yet implemented in real time and the superposition is conducted after demodulation. Both of these will be extended in the future.

VI. MEASUREMENT AND RESULTS

Fig. 7 shows a plot of the correlation peak over a measurement time of about 25 s. This graph was generated by a mobile measurement at a speed of about 15 m/s, which leads to a clearly visible correlation peak that is distorted from time to time. These distortions are typical for regions in which one of the signals has a very low SNR due to fading conditions.

Fig. 11 shows a plot of the time delay estimation for the maxima in the correlation measurement (derived from Fig. 7). As can be seen, the tracking algorithm has to be capable to follow the main maximum in spite of some emerging false peaks.
TABLE I
TIME DELAY BETWEEN DIFFERENT BROADCASTING STATIONS (MEASURED STATIONARY IN ULM)

<table>
<thead>
<tr>
<th>program</th>
<th>transmitter location</th>
<th>measured time delay relative to broadcasting station in Uml</th>
<th>calculated distance relative to broadcasting station in Uml</th>
<th>real distance relative to broadcasting station in Uml</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWR1</td>
<td>Cottgen</td>
<td>0.00 ms</td>
<td>ca. 15km</td>
<td>96km</td>
</tr>
<tr>
<td>SWR1</td>
<td>Aalen</td>
<td>0.00 ms</td>
<td>ca. 15km</td>
<td>54km</td>
</tr>
<tr>
<td>SWR3</td>
<td>Cottgen</td>
<td>10.82 ms</td>
<td>ca. 330km</td>
<td>96km</td>
</tr>
<tr>
<td>SWR3</td>
<td>Aalen</td>
<td>0.001 ms</td>
<td>ca. 12km</td>
<td>54km</td>
</tr>
<tr>
<td>Radio7</td>
<td>Aalen</td>
<td>0.26 ms</td>
<td>ca. 70km</td>
<td>54km</td>
</tr>
</tbody>
</table>

Fig. 11. Extracted time delay from Fig. 7

Tab. I shows delay measurements conducted with the test system. This yields time delays — depending on the broadcasting station and the program — of several μs up to more than 10 ms. The longer delays measured are much larger than expected by the real distance between the broadcasting stations. This is due to some digital transmission lines in the distribution network of the broadcasting companies, which may buffer the signal on its way to the transmitters. Our delay tracking algorithm therefore must cope with a large variety of possible delays. The initial search range and the signal delay within the tracking algorithm has to be larger than these measured delays.

VII. CONCLUSIONS

We have proposed a channel diversity concept for a compensation of multipath fading in mobile radio. Based on a software radio architecture this concept may help to increase the quality of mobile reception in modern car receivers.

Simulations and first measurement results show the potential of channel diversity to increase the SNR of the received signal.

As digital signal processing parts will become cheaper and RF wiring will not, channel diversity may become an alternative or addition to antenna diversity, which may result in a reduction of the number of antennas.

ACKNOWLEDGMENTS

The authors would like to acknowledge the work of Mr. Oscar Iglesias Cid in programming and implementing the demonstrator. We also thank Dr. Konrad Böhm for his support and the useful discussions.

REFERENCES

Clock Generator Phase Noise in RF Sampling Receivers

Thomas Müller*, Stephan Herter*, Johann-Friedrich Luy*

* Daimler-Chrysler Research Ulm, REM/CM, Wilhelm-Runge-Str. 11, 89081 Ulm, Germany
e-mail: thomas.mu.mueller@DaimlerChrysler.com

Abstract – When deciding for a broadband approach of a software defined radio the influence of the sampling jitter may define the performance of the receiver. The jitter is mainly generated by the phase noise of the reference clock generator. After a definition of the jitter the calculation of the jitter out of the phase noise will be demonstrated. The sampling jitter will lead to partly correlated phase and amplitude variations. The influence on the bit error rate of BPSK based digital communication systems will be demonstrated.

I. INTRODUCTION

The trend to broadband concepts of software configurable receiver architectures is mainly driven by three reasons:
1. The complete information of a couple of channels is available in the digital domain and can be processed in parallel. Therefore the insertion of monitor- or background receivers is only a question of digital signal processing
2. The trend to higher bandwidths in new communication standards leads necessarily to a higher sampling rate of the analog to digital converter
3. Some applications need to receive a couple of standards in parallel. In an automotive environment e.g. it is necessary to receive in parallel the traffic message channel via FM, to provide the passengers with TV and the headphones are connected via Bluetooth.

The drawback of the increased sample rate are higher demands on the analog to digital converter and the clock generation circuitry. The phase noise of the clock generator and its consequences will be content of this contribution.

Although there exist similarities between the process of mixing and the process of sampling, important differences come up regarding the influence of phase noise of the clock generator. The ideal sampling process can be understood by the folding of the input spectrum with a lattice of dirac pulses with the distance of the sampling frequency (Fig. 1).

Phase noise of the sampling clock will lead to errors of the ideal sampling time

\[ t_s = n \cdot \frac{1}{f_S} \]  

where \( f_S \) is the sampling frequency and \( t_s \) is the time of sampling.

An error of the sampling point will lead in the sampled signal to a phase error and to an error of the amplitude. The amplitude error and the phase error will be discussed quantitatively in the following sections.

II. IMPACT OF THE SAMPLING JITTER

The slope of the input signal will transform the error \( \Delta \tau \) of the sampling time in an amplitude error \( \Delta y \) as shown in Fig. 2.

\[ \Delta y = \Delta \tau \frac{\partial y}{\partial t} \]  

The input signal consists in broadband sampling receivers of the desired signal and a couple of interferers. The slope of the signal can be calculated by the sum of the slopes of each partial signal, weighted with its amplitude. If one signal is dominant, it will
define the slope of the input signal. The critical case regarding the bit error rate is, if the intended signal amplitude is much below those of the interferer, so the slope of the input signal will be defined by the interferer. If the sampled signal is independent of the sampling frequency, then we have to look on the statistical properties of the input signal.

To evaluate quantitatively the bit error rate of a communication system, the statistical distributions of the errors have to be calculated. Therefore it is necessary to obtain the statistical properties of the sampling jitter and the statistical distribution of the slope of the input signal.

Fig. 2 Transformation of the sampling error to an amplitude error

As an example the variance of the slope of a band limited, gaussian noise is

$$\text{Var}\left(\frac{\partial \theta}{\partial t}\right) = P_{\text{compl}} \cdot 4\pi^2 \left( f_{\text{IF}}^2 + \frac{B^2}{12} \right)$$  \hspace{1cm} (3)

with the noise bandwidth $B$, the center frequency $f_{\text{IF}}$ and the signal power $P_{\text{compl}}$.

The probability density function (pdf) of the amplitude error $\Delta \varphi$ can be calculated by solving the integral [9]

$$p_{\Delta \varphi}(\Delta \varphi) = \int p_\varphi(x) p_{\Delta \varphi}(\Delta \varphi) \frac{1}{|x|} dx$$  \hspace{1cm} (4)

where $\Delta \varphi$ is the sampling error in seconds and $x$ is an integration variable.

With this error and with the resulting phase error the bit error probability can be calculated.

III. INFLUENCE OF THE PHASE NOISE OF A REFERENCE OSCILLATOR ON THE PHASE NOISE OF A SAMPLED SIGNAL

A phase error $\Delta \varphi_s$ of the reference signal with the frequency $f_s$ leads to an error of the sampling time of

$$\Delta t_s = \frac{1}{2\pi f_s} \Delta \varphi_s$$  \hspace{1cm} (5)

The error of the sampling time will lead to a phase error of the sampled signal.
\[ \Delta \varphi = 2 \pi f_s \Delta \varphi_s \]  \hspace{1cm} (6)

so we obtain the relation between the phase error of the clock signal and the sampled signal:

\[ \Delta \varphi = \frac{f_s}{f_r} \Delta \varphi_s \]  \hspace{1cm} (7)

The phase noise \( L(f) \) contains the signal energy of the phase variations which leads to the relationship between the phase noise of the clock signal and the phase noise of the sampled signal:

\[ L_s(f) = \left( \frac{f_s}{f_r} \right)^2 \cdot L_s(f) \]  \hspace{1cm} (8)

An increasing input frequency will lead to a stronger phase noise of the sampled signal and will reduce the dynamic of nearby a strong signal because of reciprocal mixing. The comparison of the phase noise of voltage controlled oscillators shows that their phase noise will also increase with \( f_0^2 \). In conclusion, there will be no advantage or disadvantage by using a higher sampling rate regarding the phase noise of the sampled signal.

The calculation uses a linearization of the sampled signal and of the clock signal. In the case of strong phase noise the error of linearization will lead to harmonics of the phase noise. This error will increase with the used frequencies because slope and bend of the signals are proportional to the frequency.

IV. DERIVATION OF THE SAMPLING JITTER OUT OF THE PHASE NOISE

In this section the sources for inaccuracy sampling will be discussed. Unfortunately the Allan Variance \cite{1}\cite{2}\cite{3} is not suited for this description. The reason lies in the complicated and numerically problematic transformation of the Allan Variance in phase noise or measurement units, that are important to calculate the bit error rate in communication systems. In this section a new measurement unit will be introduced whose concept is similar to that of the Allan Variance. The transformation of the phase noise in frequency domain in the new measurement unit will be described and demonstrated. Two main sampling jitter sources have to be observed: the phase noise of the reference oscillator and sampling uncertainty of the sample and hold circuit. The phase noise of the reference oscillator is usually given in the frequency domain as the power spectral density. Recently Demir et al. published some basic aspects of the phase noise in the frequency domain \cite{5}. The aperture jitter is usually defined in picoseconds as the standard deviation or the rms deviation. State of the art in monolithic A to D converters is about 0.2ps rms \cite{4}. The signal

\[ S(t) = A \cos(2 \pi f_s t + \varphi(t)) \]  \hspace{1cm} (9)

is defined in a way that the mean of the phase becomes zero:

\[ \overline{\varphi(t)} = 0 \]  \hspace{1cm} (10)

The variance of the phase variations can be described as follows:

\[ \langle \Delta \varphi^2(t) \rangle = \langle (\varphi(t) - \varphi(t + \tau))^2 \rangle \]  \hspace{1cm} (11)

By transforming we obtain

\[ \langle \Delta \varphi^2(t) \rangle = \varphi^2(t) - 2 \varphi(t) \varphi(t + \tau) + \varphi^2(t + \tau) \]
\[ = \varphi^2(t) + \varphi^2(t + \tau) - 2 \varphi(t) \varphi(t + \tau) \]  \hspace{1cm} (12)

The last term is identical to the autocorrelation function (ACF) of the phase \( \varphi(t) \). The first two terms both converge to the autocorrelation function with the offset \( 0 \). We obtain

\[ \langle \Delta \varphi^2(t) \rangle = 2 \left( ACF_{\varphi(t)}(0) - ACF_{\varphi(t)}(\tau) \right) \]  \hspace{1cm} (13)

With the Wiener-Khinchin-Theorem the fourier transformed of the ACF of \( \varphi(t) \) is equal to the power spectral density \( S_r(f) \). The power spectral density of the phase in 1/Hz is linked with the power spectral density in the baseband by the following relationship:

\[ L(f) = \frac{1}{2} S_r(f) = \frac{P_{\text{SSB}}(f)}{P_{\text{compl}}} \]  \hspace{1cm} (14)

\( P_{\text{SSB}} \) is the single side band power spectral density related to a bandwidth of 1Hz and the complete signal power \( P_{\text{compl}} \). The relationship is only valid for small angle errors \( \Delta \varphi << 1 \) rad. The calculation of variance of the phase jitter can be written in the following way:

\[ \langle \Delta \varphi^2(t) \rangle = 2 \left( ACF_{\varphi(t)}(0) - ACF_{\varphi(t)}(\tau) \right) \]
\[ = 2 \left[ FT^{-1} \left[ S_r(f) \right] \right]|_{\omega = 0} - FT^{-1} \left[ S_r(f) \right]|_{\omega = \tau} \]  \hspace{1cm} (15)

So we obtain

\[ \langle \Delta \varphi^2(t) \rangle = 2 \left[ 1 - 2 FT^{-1}[L(f)] \right]|_{\omega = \tau} \]  \hspace{1cm} (16)

In an analog to digital converter the zero crossings of the clock signal will be used as a reference to sample our signal. For the calculation of the distance of the zero crossings of the function \( S(t) \) as defined in (9)
Asin(2πf_0t + \varphi_0 + \varphi(t)) = 0  \tag{17}

the phase offset \varphi_0 can be ignored. To suppress in measurements the influence of an amplitude offset, only the distances of the rising \(\uparrow\) zero crossings are regarded.

\[ t_{\uparrow,i} = \frac{1}{f_0} - \frac{1}{2\pi f_0} \varphi(t_{\uparrow,i}); \quad i \in N \tag{18} \]

The first part of the right side of the equation describes the intended sampling period. The second part transforms the phase variations in variations of the zero crossings. A suitable definition of the variance of the sampling period

\[ \langle \Delta t_{\uparrow,i}^2(k) \rangle = \text{Var}(t_{\uparrow,i} - t_{\uparrow,i+k}) = \langle (t_{\uparrow,i} - t_{\uparrow,i+k})^2 \rangle \tag{19} \]

with \(i, k \in N^+\)

and with (18) we obtain

\[ \langle \Delta t_{\uparrow,i}^2(k) \rangle = \frac{1}{(2\pi f_0)^2} \langle (\varphi(t_{\uparrow,i}) - \varphi(t_{\uparrow,i+k}))^2 \rangle \tag{20} \]

with \(i, k \in N^+\)

For small phase errors the relationship between the variance of phase errors and the variance of sampling errors leads to:

\[ \langle \Delta t_{\uparrow,i}^2(k) \rangle = \frac{1}{(2\pi f_0)^2} \langle \Delta \varphi^2 \left( \frac{k}{f_0} \right) \rangle; \quad k \in N^+ \tag{21} \]

With formula (16) the variance of the sampling jitter can be calculated out of the power spectral density of the clock oscillator:

\[ \langle \Delta t_{\uparrow,i}^2(k) \rangle = \frac{2}{(2\pi f_0)^2} \left( 1 - 2 FT^{-1}[L(f)]_{f_0} \right); \quad k \in N^+ \tag{22} \]

The standard deviation of the aperture jitter is therefore

\[ \Delta t_{\uparrow,10} = \frac{1}{\pi f_0} \sqrt{1 - FT^{-1}[L(f)]_{f_0}}; \tag{23} \]

As an example Fig. 4 shows the jitter of an oscillator with a phase noise of about \(-90\text{dBc/100kHz}\) coupled to a reference clock with a PLL bandwidth of 10kHz. Beside the quantitative calculation of the jitter, the result shows that with increasing center frequency and identical phase noise \(L(f)\) the jitter improves with \(1/f_0\). When using the complete spectrum or the double side band (DSB) phase noise of the reference clock instead of \(L(f)\), a factor of 2 in front of the fourier transformation must be omitted because we use both sides of the spectrum which contains double the power of the signal.

A great advantage of RF sampling receivers, in comparison to receivers with an analog downconverting stage, is that the clock can be chosen fixed and the LO frequency is generated digitally and therefore nearly optimally. Fixed frequency oscillators can be produced with high Q resonators and extremely low phase noise.

V. CLOCK GENERATOR INDUCED AMPLITUDE ERRORS

As shown in the preceding sections the distribution of the amplitude error depends on the shape of the input signal with its specific pdf of the slope. The phase error is directly tied up to the phase noise of the reference clock. In this section we will first calculate the pdf of the amplitude error and depict the pdf in the complex signal space. The bit error rate of a BPSK modulated transmission will be calculated.

The amplitude error that results from a sampling error \(\Delta t\) is calculated by the multiplication of the time error with the slope in the sampling point. If sampling time and slope are uncorrelated, a pdf of the amplitude error can be calculated based on the pdf of the jitter.

\[ p_z(z) = \int p_x(x) p_y \left( \frac{z}{x} \right) \frac{1}{|x|} \, dx \tag{24} \]

where \(p_x(x)\) and \(p_y(y)\) are the pdfs of \(x\) and \(y\). For gaussian distribution of \(x\) and \(y\)

\[ p(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{25} \]
the pdf of \( z \) will be

\[
p_z(z) = \frac{1}{\pi \sigma_x \sigma_y} K_0 \left( \frac{|z|}{\sigma_x \sigma_y} \right)
\]

(26)

with the modified Bessel function of the second kind \( K_0 \).

The standard deviation of \( z \) is the product of the standard deviation of \( x \) and \( y \):

\[
\sigma_z = \sigma_x \cdot \sigma_y
\]

(27)

The resulting pdf is depicted in Fig. 5.

It can be shown that the slope of a band limited, white noise is gaussian distributed. Usually the aperture jitter of a sample and hold circuit is also gaussian distributed so that for the calculation of the pdf of the amplitude error the ordinate of Fig. 5 has to be scaled with the product of the jitter and the standard deviation of the bandlimited noise.

\[ I_2(\Delta A, \Delta \phi) = (\Delta A + A_{\text{ind}}) \cos(\phi_{\text{ind}} + \Delta \phi) \]

\[ Q_2(\Delta A, \Delta \phi) = (\Delta A + A_{\text{ind}}) \sin(\phi_{\text{ind}} + \Delta \phi) \]

(28)

The amplitude error is the product of the time error and the slope

\[ \Delta A = \frac{\partial y}{\partial t_x} \cdot \Delta t \]

(29)

the phase error has been calculated in (6)

\[ \Delta \phi = 2\pi f_s \Delta t \]

(30)

so the pdf of the phase error is the pdf of the time deviation scaled with \( 2\pi f_s \).

In Fig. 7 the pdf is shown of a sine with a frequency of 1MHz, sampled with an aperture jitter of 10ns. The walls show the numerically simulated histograms of the amplitude and phase errors.

The center of the pdf is the intended point in the complex signal space \((A_{\text{ind}}, \phi_{\text{ind}})\). When transforming the pdf according to (28) it has to be observed that some points can be reached several times when \( \Delta A > A_{\text{Soll}} \) or \( |\Delta \phi| > \pi \).

Fig. 5: Pdf of the product of two gaussian distributed values with \( \sigma=1 \)

VI. IMPACT OF THE SAMPLING INDUCED PHASE AND AMPLITUDE ERROR ON THE BIT ERROR RATE.

The sampling jitter will lead to amplitude and phase errors in the complex baseband (Fig. 6).

Fig. 6: Definition of phase and amplitude error in the complex signal space

Since both of them are caused by the shift of the sampling point, there will be a correlation between them. The phase error will lead to a rotation around the origin and the amplitude error will lead to a radial shift.
The BER can be directly evaluated in the $\Delta \phi - \Delta A$ space. For a known $A_{\text{ref}}$ and $\phi_{\text{ref}}$ the boarders of integration can be defined to calculate the probability of a right or wrong result. In Fig. 8 the integration boarders of a binary phase shift keying are shown.

VII. JITTER OPTIMIZATION

The calculated jitter can be used to optimize the phase locked loop of the reference oscillator. For this purpose the left part of the jitter curve has been weighted with $1/\sqrt{\text{measurement time}}$ and the sum of the resulting values have been used as an error function for an optimization of the PLL loop parameters. The resulting jitter and the corresponding phase noise are illustrated in Fig. 9 and Fig. 10. In both cases the PLL is locked to a noisy signal, whose jitter performance has to be improved. The results were confirmed by our realization.

VIII. CONCLUSION

The influence of the sampling clock on the sampled signal has been derived. With the aid of the demonstrated calculation scheme it is possible to evaluate the jitter out of the phase noise. When calculating the bit error rate out of the jitter it is necessary to observe the correlation between the resulting amplitude and phase error. Simulation results show the improvement of the jitter by optimizing the PLL loop parameters.
DECIMATION BY NON-INTEGER FACTOR IN MULTISTANDARD AND SOFTWARE RADIO RECEIVERS

Djordje Babic, and Markku Renfors

Institute of Communications Engineering
Tampere University of Technology, FINLAND
P.O. Box 553, FIN-33101 Tampere, {babic, mr}@cs.tut.fi

Abstract-The sampling rate conversion is a critical functionality of the software radio receiver. Because the signals of different system standards have incommensurate symbol/sampling rates and a common Analog-to Digital Converter (ADC) is to be used for all supported standards, the decimation factor may become very difficult non-integer number. The task is to find an efficient non-integer decimation structure. This paper gives overviews of two efficient fractional decimator structures based on Cascaded Integrator-Comb (CIC) filters and low order polynomial-based interpolation filters. This paper also compares the two decimator structures in terms of aliasing attenuation, and complexity.

1. INTRODUCTION

The ability to process signals corresponding to a wide range of frequency bands and channel bandwidths is a critical issue of the 3rd generation cellular multi-standard radios and impacts heavily on the design of both analogue and digital stages of the radio. Because of the need to support different wireless standards, the concept of software radio has arisen. We can define software radio as a single hardware solution adaptable to different system standards by changing software. Software radio is also known as software configurable radio, software defined radio or flexible radio. Recently, the availability of inexpensive, high performance, low power signal processing devices has made it possible to implement more and more radio functionalities within the signal processor rather than with analog components. However, the full realization of the software radio concept is not yet feasible in practice because of the limitations of today's hardware technology, especially in the analog-to-digital converter (ADC) area. A step towards its realization is a multistandard radio receiver that is designed to support several existing standards. A very primitive approach for the implementation of a multistandard receiver is based on the use of distinct receiver chains, each optimized to support one radio standard. This approach is inflexible and infeasible for the support of more than two or three standards. Therefore, it is important to develop receiver concepts which use common hardware platform for the same functionality for different standards [1]. The requirements for flexibility and adaptivity shift many of the radio functionalities from analog to digital domain. The analog components have to be designed to fulfill the constraints of all standards to be supported by the radio. The software radio concept introduces several critical radio functionalities. The practical realization of the critical functionalities is important and difficult task in receiver design. Because of the high dynamic range and wide bandwidth, ADC becomes a critical issue in the multistandard radio concept. Sampling rate reduction (decimation) from a high ADC sampling rate to a small multiple of the symbol rate is a key functionality in any digital radio receiver. The standards to be supported by a software radio platform are often based on incommensurate clock and symbol rates. This makes the sample rate adaptation and decimation the second critical functionality in multistandard radio design. Other critical functionalities are digital down-conversion and channelization, and interferer cancellation [1].

This paper discusses the problem of sampling rate adaptation and decimation in software radio receivers. The software radio concept introduces the following requirements for the decimation system [2], [3], [4]:
- decimation factor \( R \) can be a difficult fractional or even an irrational number,
- the multirate filter chain has to have good anti-aliasing and anti-imaging properties,
- the overall structure should be simple because of possible high sampling rate,
- it is desired to have a unique decimation hardware platform that is programmable and adaptable for different decimation factors and filter requirements.

In this paper two promising decimator structures are compared in terms of aliasing attenuation, complexity, and programmability.

2. SAMPLING RATE CONVERSION BY NON-INTEGER FACTOR

Sampling rate conversion is needed whenever two systems working at different sampling rates are to be interconnected [5]. A good example is a radio receiver, where typically a high sampling rate is used in the ADC due to various reasons [1] (e.g., the commonly used \( \Delta-\Sigma \) ADC principle is based on oversampling). There

This work was carried out in the project “Digital and Analog Techniques for Flexible Receivers” funded by the National Technology Agency of Finland (TEKES). It was also supported by Tampere Graduate School in Information Science and Engineering (TISE).
are several advantages in the sampling rate reduction (decimation) to the smallest possible value. The number of samples is reduced by the decimation factor, reducing the processing workload significantly. Further, a discrete-time system becomes easier to implement using the lower sampling rate in terms of filter order, finite wordlength effects, etc. Since the sampling rate conversion can be seen as a process of resampling, in spectral domain the repetitions of the input signal spectrum are expected. If the original signal is not band-limited, the different spectral replicas will overlap after downsampling. This overlapping in spectral domain, known as aliasing, changes the signal irreversibly. Similarly, after interpolation (upsampling) the spectrum repeats itself at the multiples of the sampling rate. These are called image spectra, and the phenomenon itself is called imaging [5]. The sampling rate conversion system must provide enough attenuation for aliasing and imaging in order to preserve the desired signal.

The conversion between arbitrary sampling rates includes cases where the sampling rate change (decimation) factor is integer, rational, irrational, or even a time-varying number. If the overall decimation factor is a large number, then it is more efficient to develop a multistage decimation chain. Further, the overall decimation can be divided into two main parts, decimation by an integer factor and sampling rate conversion by a non-integer factor. The decimation by integer ratio is a very well explored topic and there are many efficient solutions for this task [1].

No generally applicable solution for the problem of sampling rate conversion by a non-integer has been found yet. The type of the used filter depends on the application at hand. Different filter types are used for different input sampling frequencies, whether the non-integer sampling rate conversion block is placed at the beginning, in the middle, or at the end of the overall decimation chain. However, the placement of the non-integer block at the beginning of the overall decimation chain has several advantages; the filter requirements are relaxed due to the high ratio between input sampling rate and desired signal bandwidth.

There are several different approaches for sampling rate conversion by a non-integer factor [6], [7]. The filters intended for integer and rational sampling rate conversion factors can be considered as time varying with periodically changing coefficients. Thus, time varying filters are used to produce samples at appropriate times. There are two approaches that can be applied for these filters:

- Rational factor means storage of a certain number of filter coefficients, and the correct set is applied according to timing.
- Irrational factor means calculation of a new coefficient set as required for each new sample.

In the case of sampling rate conversion by an irrational factor, the coefficients are not updated periodically. Therefore, the required number of coefficients sets becomes infinite, or in practice with finite resolution to represent the coefficients, finite but very large. The required coefficients storage memory can become very large and the implementation of the structure impractical. In order to reduce the storage requirements, it is possible to perform the coefficient design during the filtering process based on the current relative position between input and output samples. Sometimes the complexity of coefficients calculations exceeds the complexity of filtering operations. Therefore, this method may not be feasible in real time.

Recently, polynomial-based filters have been found as an efficient solution to the problem of non-integer sampling rate conversion. The sampling rate conversion can be considered as a problem of calculating a new output sample between input samples. Therefore, it is possible to look at this problem in a sense of mathematical interpolation [8]. If the impulse response of the polynomial filter can be expressed in each sampling interval by means of the low-order polynomial, it is possible to efficiently implement this filter using the so-called Farrow structure. The main advantage of the Farrow structure is in that it has only one changeable parameter, the fractional interval \( \mu \).

3. EFFICIENT FILTER STRUCTURES

If the non-integer sampling rate conversion system is placed at the beginning of the overall decimation chain, the filter requirements are relaxed due to the high oversampling ratio. Therefore, in this case, it is possible to use simpler filter structures reducing the workload and power consumption significantly. The Cascaded Integrator-Comb (CIC) filters can be used for efficient decimation by an integer factor in the first stages as they have a simple regular structure without multipliers. The low-order polynomial-based interpolation filters are a good tool for fine tuning of the sampling rate. Therefore, one can assume that a simple and efficient system for sampling rate conversion by a non-integer may consist of a CIC filter and low order polynomial-based interpolation filter.

3.1. Cascaded Integrator-Comb (CIC) filter

CIC filters have a simple regular structure without multipliers. CIC decimation filter (see [9]) consists of \( N \) cascaded digital integrator stages operating at high input sampling rate \( F_{in} \), followed by \( N \) cascaded comb or differentiator stages operating at low sampling rate \( F_{in} / R \). Its frequency response is given by

\[
H_{CIC}(e^{j\omega}) = e^{-j\omega M/2} \left( \frac{\sin(\omega R/2)}{R\sin(\omega/2)} \right)^N, \tag{1}
\]

where \( \omega = 2\pi f/F_{in} \) is the normalized input frequency.

3.2. Linear interpolation filter

When the decimation factor is an irrational number one solution is to use polynomial-based interpolation filters. Among them, linear interpolation filter has a simple implementation structure, only one multiplication per output sample is needed. Because interpolation is
basically a reconstruction problem, polynomial-based interpolation can be analyzed using the hybrid analog/digital model shown in Fig. 1.

\[
x(n) \xrightarrow{\text{DAC}} x_d(t) \xrightarrow{h_d(t)} y_d(t) \xrightarrow{\text{Resample at } t=(n_i+\mu_t)T_{in}} y(t) \xrightarrow{\text{CIC FILTER}} x_{\text{CIC}}(m) \xrightarrow{\text{Decimation by } R_{\text{out}}} y(t)
\]

**Fig. 1.** The hybrid analog/digital model for the polynomial interpolation filter.

In this model, the interpolated output samples \( y(t) \) are obtained by sampling the reconstructed signal \( y_d(t) \) at the time instants \( t=(n_i+\mu_t)T_{in} \). Here \( n_i \) is an integer, \( \mu_t \in [0,1) \) is the adjustable fractional interval, and \( T_{in} \) is the sampling interval of the input signal \( x(n) \).

For linear interpolation, the impulse response of the reconstruction filter \( h_d(t) \) is a triangular function, and thus, its frequency response is given by

\[
H_d(f) = \left( \frac{\sin(\pi f / F_{in})}{\pi f / F_{in}} \right)^2.
\]

The digital implementation of the linear interpolation, which needs only one multiplication, can be based on the following equation:

\[
y(t) = x(n_i) + \left[ x(n_i+1) - x(n_i) \right] \mu_t.
\]

### 4. NOVEL FRACTIONAL DECIMATOR STRUCTURES

#### 4.1. Fractional decimator based on CIC filter and linear interpolation

Because we consider here a system that can be used also for irrational decimation, the overall decimation factor is determined by

\[
R = \frac{F_{in}}{F_{out}} = R_{\text{int}} + \varepsilon,
\]

where \( F_{in} = 1/T_{in} \) and \( F_{out} = 1/T_{out} \) are the input and output sampling frequencies, whereas \( R_{\text{int}} \) is the integer part and \( \varepsilon \in [0,1) \) is the decimal part of the overall decimation factor. Fig. 2 illustrates the proposed structure for the decimation filter [3]. The input signal \( x(n) \) is divided into polyphase components \( x_k(m) \) for \( k = 0, 1, \ldots, R_{\text{int}}-1 \) by using delay line and parallel CIC filters having the decimation factor of \( R_{\text{int}} \). Only few of these \( R_{\text{int}} \) parallel CIC filters are to work at the same time, hence, the number of parallel CIC filters is reduced. The sampling rate at the output of the CIC filters is \( F_{out}/R_{\text{int}} \). After this integer decimation we still have to decimate by \( 1+\varepsilon / R_{\text{int}} \) in order to have overall decimation by \( R \). This final, possibly irrational decimation is done using linear interpolation between some of the two signal pairs \( x_k(m) \) and \( x_{k+1}(m) \), where \( \oplus \) denotes a modulo \( R_{\text{int}} \) summation.

**Decimation by \( R_{\text{out}} \)**

\[
x(n) \xrightarrow{\text{Decimation by } R_{\text{out}}} x_{\text{int}}(m) \xrightarrow{\text{CIC FILTER}} x_{\text{CIC}}(m) \xrightarrow{\text{Decimation by } 1+\varepsilon / R_{\text{out}}} x_{\text{out}}(m)
\]

**Fig. 2.** Model of proposed decimation filter.

The linear interpolation block in Fig. 2 is shifted by one branch when needed, according to a certain condition (see [3] for details). Because of the modulus \( R_{\text{int}} \) summation mentioned above, the next signal pair for linear interpolation after \( x_{\text{int}}(m) \) is \( x_{\text{int}}(m) \) and \( x_{\text{int}}(m) \). The fractional interval \( \mu_t \) is re-calculated for each output sample \( y(l) \) for \( l = 0, 1, 2, \ldots \).

The time interval between samples \( x_k(m) \) and \( x_{k+1}(m) \) equals to \( T_{in} \) and, thus, the linear interpolation is done at the high input sampling frequency \( F_{in} \). This means better image attenuation. The CIC filters attenuate the disturbing channels and noise which would cause aliasing in linear interpolation. In other words, the CIC filters and linear interpolation take care of anti-aliasing and anti-imaging property, respectively.

As an example, Fig. 3 shows input and output signals as well as some of the polyphase signals for the decimation factor of \( R = 3.3 \). After four first samples, the next output sample \( y(4T_{out}) \) falls outside the interval \( x_k(m) \) and \( x_{k+1}(m) \). When this occurs, the linear interpolation is shifted by one interval (as indicated by an arrow in Fig.
2) and the interpolation is done between signals $x_i(m)$ and $x_j(m)$.

4.1.1. The frequency response of the overall system.

The overall frequency response of the decimation filter structure in Fig. 2 is a product of the frequency responses of the parallel CIC filters and linear interpolation filter. Note that the former response is a periodical whereas the latter is not. The frequency response of the parallel CIC filter stage is simply the same as the response of one CIC filter given by Eq. (1), where, however, $R$ has to be replaced by $R_{out}$. Since the linear interpolation is done at the higher input rate $F_{in}$, its frequency response is given by Eq. (2). Consequently, the overall zero-phase frequency response of the proposed decimation filter, relative to the input sampling frequency, is given by

$$H_f(\omega) = H_{CIC}(\omega)H_s\left(\frac{\omega F_{in}/2\pi}{R}ight) = \left(\sin\left(\frac{\omega R_{in}}{2}\right)\right)^N\left(\sin\left(\frac{\omega}{2}\right)\right)^2$$

(5)

where $\omega = 2\pi F_{in}/F_{out} = 2\pi f/(RF_{out})$.

4.1.2. Implementation considerations.

The actual implementation may be based on the block diagram shown in Fig. 4. In the general case, for any $\epsilon$, the number of needed comb branches is $B = N + 2$. Two of them are used for the actual interpolation and the remaining $N$ branches are used for initializing the state-variables of the branches needed later.

Fig. 4. Efficient implementation structure for $N^{th}$ order CIC filter for any non-integer decimation rate $R$. $H(z)$ denotes integrator stages of the CIC filter, $H_C(z)$ denotes comb stages of the CIC filter. The commutators COM1 and COM2 are used to select the correct input branch for the $B$ comb sections and for the linear interpolation, respectively.

The details of the control logic are given in [3]. Here we stress that the linear interpolation works at the output sampling rate. However, virtually it works as if it was at the beginning of the overall structure having high input sampling rate. The main advantage of this lies in that only one multiplication is needed per each output sample. The linear interpolation filter contains one multiplier while the CIC part is multiplierless. However, this structure is not easily programmed as the attenuation of the aliasing bands severely depends on the decimal part of the sampling factor $\epsilon$.

4.2. Fractional programmable CIC decimation filter

Even though it provides enough attenuation for aliasing components and it has low power consumption, the structure presented above is not programmable enough. By using non-integer delay in the feed-forward branch of comb stage and polynomial interpolation filter between integrator and comb stages of the CIC filter, we achieve improved attenuation for the aliasing frequency components. The main advantages of the fractional programmable CIC are high flexibility and possible programmability as the position of zeros in frequency response of the overall structure can be easily adjusted.

$$\frac{F_{in}}{x(n)} \rightarrow H_I^N(z) y(n) \rightarrow \text{PIF} \rightarrow H_C(z) \rightarrow F_{out}$$

Fig. 5. The fractional programmable CIC filter.

The fractional programmable CIC decimation filter structure [4], shown in Fig. 5, consists of $N$ integrator stages operating at input rate $F_{in}$, polynomial interpolation filter (PIF in Fig. 5) $h_D(t)$, resampler, and $N$ comb stages operating at output sampling rate $F_{out}$. The role of polynomial interpolation filter is to provide the samples to the input of the comb stage. It should be noted that the interpolation filter does not work all the time, as it prepares the samples for the output stages that work at the output sampling rate. The control logic of the interpolation filter is very similar to one presented in [3]. In this way the workload is reduced as the interpolation filter in practical realization works at the output sampling rate. In the case of non-integer decimation factor $R = F_{out}/F_{in}$, we can realize the frequency response (3) by placing a non-integer delay $D$ in the feed-forward branches of the comb stages. $D$ is determined by the desired length of the moving average (CIC) filter $K$ and overall decimation ratio $R$ as $D = \frac{R}{K}$. The moving average filter length $K$ has influence on the frequency response of the overall structure. The value of $K$ determines the positions of the zeros in the overall frequency response. The frequency response of the overall system is a product of two frequency responses of the systems in cascade, that is

$$H_m(e^{j2\pi F_{in}}) = H(e^{j2\pi F_{in}})H_a\left(\frac{\omega}{R}\right)$$

(6)

The main complexity of the structure is related to the non-integer delay filter, which contains a certain number of multipliers. As non-integer delay
approximation we use the FIR fractional delay filter designed using Lagrange interpolation method. In this method the delay $z^{-D}$ is approximated by

$$H_D(z) = z^{-D} = \frac{L}{n=0} h_D(n) z^{-n},$$

where $L$ is the filter length, and filter coefficients $h_D(n)$ have the explicit form as

$$h_D(n) = \frac{1}{L} \prod_{k=0, k\neq n}^{L} \frac{D-k}{n-k}, n=0, 1, \ldots, L.$$ (8)

The transfer function of the comb stage is expressed as the transfer function of a FIR filter as follows

$$H_C(z) = 1 - z^{-D} = 1 - h_D(0) - \sum_{n=1}^{L} h_D(n) z^{-n}.$$ (9)

It has been shown that there is a small degradation in the overall frequency response for the higher frequencies as a consequence of the selected method for the fractional delay filter realization, but this does not change overall performance of the structure. Thus, in the case when the linear interpolator filter is used, the overall frequency response becomes

$$H_N\left(e^{\frac{2\pi f}{F_{in}}}\right) = e^{-j\pi \frac{2\pi f}{F_{in}}} H_C\left(e^{\frac{2\pi f}{F_{in}}}\right) \left(\frac{\sin(\pi fK/F_{in})}{\pi fK/F_{in}}\right)^2.$$ (10)

The fractional programmable CIC filter has relatively simple structure. The polynomial filter between integrator and comb stages works at the output sampling rate as it prepares samples for the later stages that operates at the low output rate. The FIR delay filter has $L$ multipliers working also at the output sampling rate. In the case of $N^{th}$ order CIC filter and linear interpolation filter total number of multiplications per output sample is $NL+1$.

5. CASE STUDIES

In this example, the bandwidth of the input signal is $f_s=0.001 F_{in}$ and decimation factor $R=34/14$. It is required that the frequency bands that cause aliasing to the frequency band of the input signal are attenuated at least by $A_\delta=80$ dB and the passband distortion is less than $\delta_f=0.01$ (0.086 dB).

These requirements are met by a fractional decimator filter (the first presented structure) having the CIC filter of order $N=3$ and linear interpolation filter. Fig. 6 presents the bands that cause aliasing to the desired band. As it can be seen, the minimum attenuation of these bands is 84.4 dB. Because the over-sampling factor is still high after decimation, the worst case passband distortion caused by the proposed filter structure is only 0.06 dB.

The same requirements are met by the fractional programmable CIC filter with $N=3$, $K=69$, thus $D=2.07658$ (Lagrange FIR fractional delay filter of length $L=6$), using the linear interpolation filter. The aliasing bands of this structure are shown in Fig. 7. In this case, the worst case aliasing attenuation is 87.2 dB, and the passband distortion is the same.

![Fig. 7. Aliasing bands of the overall structure of the fractional programmable CIC filter.](image)

**TABLE I**

<table>
<thead>
<tr>
<th>The minimum attenuation of the aliasing bands (in dB) for the fractional decimator filter.</th>
<th>CIC filter order</th>
</tr>
</thead>
<tbody>
<tr>
<td>The passband edge, normalized to the input sampling rate</td>
<td>$N=1$</td>
</tr>
<tr>
<td>$\varepsilon=0.05$</td>
<td>$f_s=0.001$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.002$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.005$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.01$</td>
</tr>
<tr>
<td>$\varepsilon=0.5$</td>
<td>$f_s=0.001$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.002$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.005$</td>
</tr>
<tr>
<td></td>
<td>$f_s=0.01$</td>
</tr>
</tbody>
</table>

5.1. Comparisons

The minimum attenuation of the aliasing bands occurs at the edge of the first aliased band, and it depends on
CIC filter order, decimal part of the decimation factor, and relative bandwidth of the desired input signal. TABLE I gives values of the minimum attenuation of the aliasing bands as function of CIC filter order, decimal part of the decimation factor, and relative bandwidth of the desired input signal for the case of the fractional decimator. These results are given for the integer part of decimation factor $R_{int}=34$. We can compare these data to the data given in TABLE II for the fractional programmable CIC filter. We can notice that the fractional programmable CIC filter attenuates more aliasing frequency components, and more important, the attenuation does not depend on decimal part of decimation factor $e$. This structure is easily programmed as the positions of the zeros in frequency response can be adjusted by changing the coefficient of the non-integer delay filter. However, the complexity of the fractional programmable CIC filter is considerably higher than complexity of the fractional decimator, in terms of multiplications per output sample.

### TABLE II

<table>
<thead>
<tr>
<th>Decimal part of the decimation factor</th>
<th>The passband edge, normalized to the input sampling rate</th>
<th>CIC filter order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_p=0.001$</td>
<td>$f_p=0.002$</td>
</tr>
<tr>
<td>$e=0.05$</td>
<td>29.1</td>
<td>58.2</td>
</tr>
<tr>
<td></td>
<td>22.8</td>
<td>45.6</td>
</tr>
<tr>
<td></td>
<td>14.2</td>
<td>28.4</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
<td>14.9</td>
</tr>
<tr>
<td>$e=0.5$</td>
<td>29.0</td>
<td>57.9</td>
</tr>
<tr>
<td></td>
<td>22.7</td>
<td>45.4</td>
</tr>
<tr>
<td></td>
<td>14.1</td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td>7.6</td>
<td>14.7</td>
</tr>
</tbody>
</table>

### 6. CONCLUSIONS

We have overviewed two structures for the sampling rate conversion by a non-integer factor. Both structures are developed as the first stage in multirate decimation chain of a multistandard radio receiver. The fractional decimator structure composed of CIC filter and linear interpolation filter has very simple structure, and thus lower power consumption. However, since the stopband attenuation depends heavily on the decimal part of the decimation factor this structure does not provide high reconfigurability, which is a very important issue of any software radio component. The fractional programmable CIC filter provides higher attenuation of the aliasing components, which is less dependent on the value of the decimal part of the decimation factor. This structure is reconfigurable but considerably more complex than the fractional decimator structure. It remains as a future topic to find the minimum required complexity of the fractional delay filter.

### References:


HIGH PERFORMANCE DECIMATION FILTER FOR DIGITAL DOWN CONVERTERS IN SOFTWARE RADIO APPLICATIONS

L. Lo Presti, C. Spadea, Z. Tarussio

Dipartimento di Elettronica - Politecnico di Torino
Corso Duca degli Abruzzi 24 - 10129 Torino - Italy
Ph.: +39-011-5644033 - Fax: +39-011-5644099
spadea(tarussio)@mail.tlc.polito.it
lopresti@polito.it

Abstract—In this paper a method to design a decimation filter for digital down converters for Software Defined Radio (SDR) applications is described. The advantage of the proposed design approach is the possibility of avoiding the final FIR stage (generally present in other schemes), and the good performances in the aliased bands. First preliminary experiments, related to the simulation of a complete transmission chain with PAM signals, have produced very promising results.

I. INTRODUCTION

This paper is devoted to the description of an efficient scheme for DDCs (Digital Down Converters) for SDR (Software Defined Radio) receivers, [1], [2]. As it is well-known, the state-of-the-art of the SDR technology is related to transceivers able to handle IF (Intermediate Frequency) digital signals, even if the ultimate goal of such a technology is the direct conversion from RF (Radio Frequency) to baseband. However by using DSP (Digital Signal Processor) or FPGA (Field Programmable Gate Array) chips at IF, as well as at baseband, the main desired characteristics (flexibility and reconfigurability) of a SDR system are generally achieved. Moreover the availability of chips implementing the direct conversion is foreseen in a very near future.

A digital IF signal is generally a wide-band signal containing several narrow-band channels.

The extraction of a single narrowband channel can be performed by using a DDC or a transmultiplexer [3], [4]. If only few narrowband channels (for example, a single one) must be extracted, a DDC scheme is preferred, [5], [6]. Notice that a DDC could be present also in the case of direct conversion if the channel to be recovered is not exactly centered at the baseband origin.

A DDC is one of the most critical components in a SDR receiver, as it has to handle high speed data, obtained by sampling the wide-band IF signal. Once the digitized wide-band signal is down-converted, it must be low-pass filtered and decimated in order to extract the baseband signal of interest. This low-pass filter, called decimation filter, performs two important tasks: it isolates the channel of interest from the adjacent channels, and acts as anti-aliasing filter before decimation.

Because of these two very strict requirements, the decimation filter specifications are very demanding: no in-band degradation and a high level of out-of-band rejection. This last point is very critical, because the aliasing depends on the filter stop-band characteristics and the alias adverse effect on useful data can be recovered or mitigated after decimation in no way. On the contrary the in-band degradation can be compensated by applying a proper equalizing filter to the decimated samples. Even if this equalizing filter has to manage low-rate decimated samples, it could represent a serious bottleneck of the DDC system, if its complexity is too high. For this reason the in-band degradation of the
decimation filter should be reduced as much as possible. In contrast with these strict requirements the decimation filter, in practical applications, can be implemented only by using low-complexity structures.

The solutions proposed in literature, [7], [8], [9], are generally based on modifications of a basic multistage structure, known as CIC (Cascaded-Integrator-Comb) [10], where the core element is a filter with a rectangular impulse response. In the frequency domain this filter exhibits a very poor out-of-band rejection and some degree of in-band distortion. To overcome the passband droop, CIC filters are usually cascaded with a second decimation stage. Programmable lowpass FIR filters are generally used for this stage.

To simplify the complexity of the second stage programmable filter, Kwentus [7] proposed a variant of the CIC filter (called sharpened CIC), able to significantly reduce the passband droop caused by CIC filtering. An interesting alternative, called ISOP, is proposed in [8]. These CIC modifications have been envisaged to mitigate the passband droop, while the capability of out-of-band rejection, which is responsible for the aliasing, is not considered.

In this paper a new scheme is proposed, able to lead to a more balanced trade-off point between complexity and performances. In fact the continuous progress in the programmable hardware platforms make feasible the addition of some degree of complexity also in the high rate section of an SDR terminal. Another application of the proposed scheme is in conjunction with a direct conversion front-end. In this case less severe constraints on the complexity of the decimation filter can be accepted, due to the fact that all the operations are performed at the baseband sampling rate.

The basic idea of the proposed scheme is to add a moderate degree of complexity to the Sinc-based filter, in order to mitigate both the in-band distortion and the alias degradation. The proposed structure can be applied to each stage of a multistage Sinc-based architecture. It consists in a modified basic stage, where the Sinc-filter is cascaded to a “pre-shaping” IIR filter, designed so that to realize the desired characteristics. With this structure, called P-CIC (where P stands for “preshaping”), it is possible to eliminate the final FIR stage.

This paper is organized as follows. Section II introduces the decimation concept. In section III the P-CIC structure is presented. Some results and comparisons are shown in section IV.

II. GENERAL SCHEME OF A DIGITAL DOWN CONVERTER

A typical scheme of a DDC is shown in Fig.1a, where the input signal is a digitized wide-band signal obtained at the output of a Analog-to-Digital Converter (ADC), and the output signal is a decimated baseband version of a narrow-band signal (the channel of interest, around the frequency $f_0$). Note that the frequency values indicated in this paper are always normalized digital frequencies, referred to the sampling frequency of the wideband ADC.

The scheme shown in Fig.1b is the digital implementation of the functional scheme of Fig.1a, where $r[n]$ is the received wideband sample sequence, $H(z)$ is the transfer function of the channelization filter, whose main task is to isolate a signal of interest $y[n]$, and $y_d[n]$ is a decimated version of $y[n]$. The decimation block produces a folding effect of the spectrum of $y[n]$ around the frequency points $f_k = k/D$, where $D$ is the decimation ratio. For this reason the task of $H(z)$ is also to act as anti-aliasing filter, and the channelization filter is also called decimation filter.

To adequately accomplish both tasks $H(z)$ must introduce no distortion in the bandwidth $B_0$ of the signal of interest, and high attenuation in the aliased bands, that is in the frequency ranges $(f_k - B_0, f_k + B_0)$. As the complexity is a very strict constraint, a typical choice is a transfer function of the type

$$H_C(z) = \frac{1}{N} \frac{1 - z^{-N}}{1 - z^{-1}}$$

where $N$ is a generic decimation ratio. This transfer function produces a Sinc-like ($\sin N x/\sin x$) frequency response with scarcely attenuated side lobes, and some degree of in-band distortion. With this filter the trade-off between complexity and performance is completely unbalanced towards a very low complexity. This transfer function is generally implemented by means of a very efficient structure, called CIC, [10].
III. Preshaping filter

In the Sinc-based architectures the alias introduced after decimation, even if around the null points of the Sinc frequency response, cannot be mitigated in any way. The idea here is to add a moderate degree of complexity to the CIC structure in order to mitigate both the in-band distortion and the alias degradation. The proposed structure can be applied to each stage of a multi-stage CIC architecture. It consists in a modified basic stage, as shown in Fig.2, where \( H_a(z) \) represents the transfer function of a generic filter able to modify the CIC stage behavior. This scheme is de facto adopted in other schemes proposed in literature, as it will be shown later. In our proposal the generic \( H_a(z) \) is realized by means of a “preshaping” filter, with transfer function \( H_p(z) \), designed so that to realize the desired characteristics. In order to reduce the implementation complexity \( H_p(z) \) is realized by means of second-order IIR filter with transfer function

\[
H_p(z) = \frac{(1-p_1)(1-p_1^*)}{(z-p_1)(z-p_1^*)} \tag{2}
\]

where the pole \( p_1 \) can be written as

\[
p_1 = m_p e^{j\phi}
\]

With this definition two degrees of freedom are available for determining the filter parameters: the phase \( \phi \) and the amplitude \( m_p \) of the complex pole \( p_1 \). The associated filtering equation \( y[n] = b_0 x[n] + a_1 y[n-1] + a_2 y[n-2] \) requires three multipliers, that is two more multiplications with respect to a situation without preshaping. In fact the multiplication by \( b_0 \) can be merged together with the CIC multiplier. The values of filter coefficients \( b_0, a_1, \) and \( a_2 \) are easily obtained from Eq.(2).

It is interesting to compare the filter performances with the ones achieved with the sharpened CIC and the ISOP structures. To this purpose it is sufficient to compare the transfer function of a first stage decimation filter composed of the two blocks shown in the figure 2. In fact all the proposed schemes can be reduced to this common structure, where \( L \) represents the number of cascaded CICs and \( H_a(z) \) is the additional block, whose task is to modify the CIC characteristics.

The example 1 described in section IV of [8] has been chosen for the comparison. In this example the filter described in [12] has been used as reference, with the specifications:
- Sampling rate: 39 Msps,
- Passband edge of the channel of interest: 90 kHz from the carrier, corresponding to $B_0 = 90/39000 = 0.0023$,
- Stopband edge of the channel of interest: 115 kHz from the carrier, corresponding to $B_1 = 115/39000 = 0.00295$,
- Final decimation ratio: 72
- First stage decimation filter: five cascaded CICs with $D = 18$.

In figure 3a curve $a$ represents the transfer function $|H_C(e^{j2\pi f})|$ of a single CIC stage with $D = 18$, curve $b$ represents the transfer function $|H_a(e^{j2\pi f})|$ of a sharpened correction, curve $c$ represents the transfer function of an ISOP correction, curve $d$ represents the transfer function $|H_p(e^{j2\pi f})|$ of a preshaping filter acting in the bandwidth $B_0 = 0.0023$. The null points of curve $a$ represent the folding points $f_k$. In figure 3b curve $a$ represents the transfer function of a two-stage CIC, that is $|H_C(e^{j2\pi f})|^2$, curve $b$ represents the transfer function $|H_C(e^{j2\pi f})||H_a(e^{j2\pi f})|$ of a sharpened correction cascaded with a CIC, curve $c$ represents the transfer function of an ISOP cascaded to a CIC, curve $d$ represents the transfer function of a preshaping filter acting in the bandwidth $B_0 = 0.0023$ cascaded to CIC. Figure 3c is the same of Figure 3b with two CIC stages. Figure 3d is the same of Figure 3b with three CIC stages.

In the figures 3b,c,d curve $a$ represents the transfer function of the cascade structure of figure 2, where $H_a(z) = H_C(z)$, that is a CIC structure with no correction. The out-of-band performances of the preshaping filter are evident.

A. Parameters of the preshaping filter

Two degrees of freedom are available for the choice of the pole location of the preshaping filter. Denoted as $2B_0$ the two-side bandwidth of the signal of interest to be down-converted, the condition

$$|H_p(z_B)| = \frac{1}{|H_C(z_B)|^L}$$  \hspace{1cm} (3)

can be imposed, where $z_B = e^{j2\pi \gamma B_0}$, and $\gamma$ is a design parameter. This condition has the goal of reducing the passband degradation in the bandwidth $(0, \gamma B_0)$. An example is shown in the curve $a$ of Figure 4, where a bandwidth $B_0 = 0.0023$ is equalized by using a P-CIC. Curves $b$, $c$, $d$ refer to CICs with $L = 1, 2, 3$.

Having two degrees of freedom and only one condition, one of the two pole parameters ($m_p$ and $\phi$) can be arbitrarily chosen, unless a further condition is introduced. For example a condition could be introduced to keep the distortion of the curve phase (unavoidable with an IIR structure) under a preassigned threshold.

Another possibility is to try a combination of $m_p$, $\phi$, and $D$ so that to make the preshaping filter coefficients in form of powers of two.

IV. FIR-free stages

One of the advantages of the P-CIC structures is the possibility of eliminating the final FIR stage, generally used in the other CIC-based structures. In fact the preshaping filter behaves also as a lowpass filter, and a sufficient out-of-band attenuation can be achieved if a multistage P-CIC structure is used.

Several experiments have been performed to validate this idea of employing a FIR-free system. The system used for the simulation runs is shown in figure 5; the scheme adopted for the decimation filter is drawn in the lowest part of the figure, inside a dashed box containing a three stage P-CIC structure. This particular structure has been chosen on the basis of simple experiments based on the comparison between an "analog" baseband signal and its Nyquist-sampled version, obtained by using several types of P-CIC configuration. The analog signal has been simulated by means of a very oversampled signal.

The three stage structure shown in the dashed box has given very good results from the point of view of both computational complexity and performances. An example is shown in figure 6,
A design parameter. This condition has the goal of reducing the passband degradation in the bandwidth (0–1 MHz). Sampling rate: 39 Msps, A. Parameters of the preshaping filter.

Denoted as two degrees of freedom are available for the condition of an ISOP correction, curve (a) in Figure 3a represents the transfer function of the cascade structure of Figure 3b with two CIC stages. The null points of curve (a) are equalized by using a P-CIC. Two degrees of freedom and only one condition is introduced. For example a correction cascade with a CIC, curve (b) in Figure 3b represents the transfer function of a sharpened CICs with $\alpha = 18$, curve (c) in Figure 3b represents the transfer function $H_2(z)$ of a three stage P-CIC structure. This particular box has given very good results from the point of view of both computational complexity and of P-CIC configuration. The analog signal has been down-sampled version, obtained by using several types of P-CIC sampled, and $\alpha = (3)$.

Figure 5: Cascaded structure

where the samples (denoted by $u[m]$) of the recovered signal $x_K[n]$ are superimposed on the continuous line representing the analog signal.

A. Data transmission example

To test the performances of the three-stage scheme of figure 5 a complete transmission chain has been simulated. Five channels are allocated in a generic frequency range $(f_1, f_2)$ inside the normalized range $(0, 0.5)$, down converted and the middle channel is extracted. Each channel contains a binary PAM modulated signal, bearing a sequence of symbols $I_n$.

To confine the spectrum of each channel into a preassigned bandwidth a raised-cosine shaping (evenly split between transmitter and receiver) has been adopted. As it is well known, this allows a zero-ISI (Intersymbol Interference) nar-
rowband data transmission. To reduce the effects of the interchannel interference (ICI), the technique described in [13] has been used to design the coefficients of FIR filters with raised-cosine characteristics. In figure 5 the two blocks $G_T(f)$ and $G_R(f)$ represent respectively these TX and RX shaping filters. The RX filter is placed after the decimation filter, in order to perform the raised-cosine shaping at the lowest rate. Notice that this final FIR filter has been designed only considering zero-ISI and low ICI requirements, as the decimation filter does not introduce a bandpass distortion to be compensated. This allows an independent design of the two filters (decimation and shaping for narrowband zero-ISI), so exploiting the results available in literature for the design of optimal zero-ISI filters for data transmission. In our case it was possible to use the method proposed in [13].

Each binary symbol is represented by using a limited number of samples per period $N_s$. Experiments with $N_s = 4, 8, 12$ have been performed.

The DAC has been modeled as an interpolator, in order to simulate the analog section between the two antennas. In this way the transmitted signal is generated with a low value of $N_s$, as it generally happens in the actual implementation, while the analog wide band signal at the receiver antenna is represented by means of an oversampled time-discrete signal.

In a data transmission, if the transmission chain introduces some distortion, an ISI effect appears, typically measured by means of the eye pattern. Another cause of impairment affecting the eye pattern is the ICI effect due to both a priori interference (at the TX end) and an imperfect narrowband isolation at the RX end. In our application ICI and aliasing are in same way combined, and must be carefully avoided.

A method of measurement of the eye opening, called peak distortion, has been used to evaluate the system performances. The peak distortion, $D_p$, is a measure of the amount of ISI at the optimum sampling instants (that is at the instants corresponding to the maximum opening of the eye pattern), where the information symbols $I_n$ are extracted. By denoting by $x_i$ the values of the signal at these instants, $D_p$ is defined as

$$D_p = \frac{\min |x_i| - m}{m} \quad (4)$$

where

$$m = \frac{1}{K} \sum_{i=1}^{K} |x_i| \quad (5)$$

and $K$ is the number of received symbols.

Several simulation runs have been performed by using the system shown in figure 5, with both P-CIC and CIC stages, and with different values of $N_s = 4, 8, 12$. The other tested parameters are:

- The pole location of the preshaping filters in the P-CIC stages, defined through the parameter $\alpha = \phi/2\pi B_0$.
- The length $N$ of the Sinc filter.
- The decimation factors.

In the table I four sets of parameters are indicated. Each one identifies a decimation filter configuration, denoted by the index $I_f$.

The curves in figure 7 represent the peak distortion as a function of the configuration index $I_f$, for three values of $N_s$, and with a single channel (no ICI is present). The curves have been obtained by interpolating the values of $D_p$ found in each experiment. The interpolation has been performed only to make the representation more intelligible.

The minimum value of peak distortion is obtained with the first configuration for $N_s = 8$ and 12, while a worse result is achieved with $N_s = 4$. Notice that the eye pattern obtained with $N_s = 4$ exhibits an apparent degradation due to the rough sampling (only four samples per period) of the waveform bearing the binary symbols. Therefore it is possible to find opposite results with another configuration, due to a
more favorable sampling of the received waveform. This effect tends to disappear as $N_s$ increases, as it can be observed in the figure.

Figure 8 shows the peak distortions as a function of $N_s$. In this figure the configuration $I_5$ is the same of $I_f = 1$, but applying only CIC stages. The configuration $I_6$ is the same of $I_f = 4$, but without preshaping.

By observing the values of $D_p$ in the figure, it can be concluded that in the cases of only CICs the distortion is always greater than that obtained with the corresponding P-CIC structures. This means that the preshaping approach is in general advantageous.

These preliminary results with a PAM signal encourage to continue with this study. Work is in progress to test the method in presence of noise and with more complex modulation schemes.

At the moment other preliminary results are available in the case of a multichannel wide-band received signal, composed of a set of narrow-band channels, where the selected channel is the one with center frequency equal to zero. First experiments only at baseband (that is, without considering the double frequency allocation due to the digital mixer) have been performed, in order to test the capability of the RX zero-ISI filter of extracting the channel of interest, and, at the same time, rejecting the adjacent channels. The curves of the peak distortion are not reported, because they are very similar to the ones obtained in the case of a single channel.

### TABLE I
PARAMETERS OF THE SIMULATION EXPERIMENTS

<table>
<thead>
<tr>
<th>Decimation filter index $I_f = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st stage</td>
</tr>
<tr>
<td>$H_p(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_1(z)$, $N1 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_1$, $3$</td>
</tr>
<tr>
<td>2nd stage</td>
</tr>
<tr>
<td>$H_{p2}(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_2(z)$, $N2 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_2$, $4$</td>
</tr>
<tr>
<td>3rd stage</td>
</tr>
<tr>
<td>$H_{p3}(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_3(z)$, $N3 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_3$, $2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decimation filter index $I_f = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st stage</td>
</tr>
<tr>
<td>$H_p(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_1(z)$, $N1 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_1$, $3$</td>
</tr>
<tr>
<td>2nd stage</td>
</tr>
<tr>
<td>$H_{p2}(z)$, $\alpha = 2$</td>
</tr>
<tr>
<td>$H_2(z)$, $N2 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_2$, $4$</td>
</tr>
<tr>
<td>3rd stage</td>
</tr>
<tr>
<td>$H_{p3}(z)$, $\alpha = 1$</td>
</tr>
<tr>
<td>$H_3(z)$, $N3 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_3$, $2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decimation filter index $I_f = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st stage</td>
</tr>
<tr>
<td>$H_p(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_1(z)$, $N1 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_1$, $3$</td>
</tr>
<tr>
<td>2nd stage</td>
</tr>
<tr>
<td>$H_{p2}(z)$, $\alpha = 1$</td>
</tr>
<tr>
<td>$H_2(z)$, $N2 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_2$, $4$</td>
</tr>
<tr>
<td>3rd stage</td>
</tr>
<tr>
<td>$H_{p3}(z)$, $\alpha = 1$</td>
</tr>
<tr>
<td>$H_3(z)$, $N3 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_3$, $2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decimation filter index $I_f = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st stage</td>
</tr>
<tr>
<td>$H_p(z)$, $\alpha = 3$</td>
</tr>
<tr>
<td>$H_1(z)$, $N1 = 0.25/B_0$</td>
</tr>
<tr>
<td>$M_1$, $3$</td>
</tr>
<tr>
<td>2nd stage</td>
</tr>
<tr>
<td>$H_{p2}(z)$, $\alpha = 1$</td>
</tr>
<tr>
<td>$H_2(z)$, $N2 = 0.5/B_0$</td>
</tr>
<tr>
<td>$M_2$, $4$</td>
</tr>
<tr>
<td>3rd stage</td>
</tr>
<tr>
<td>$H_{p3}(z)$, $\alpha = 1$</td>
</tr>
<tr>
<td>$H_3(z)$, $N3 = 0.5/B_0$</td>
</tr>
<tr>
<td>$M_3$, $2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fig. 7. Peak distortion as a function of the configuration index $I_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_p$</td>
</tr>
<tr>
<td>$N_s = 4$</td>
</tr>
<tr>
<td>$N_s = 8$</td>
</tr>
<tr>
<td>$N_s = 12$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fig. 8. Peak distortion as a function of $N_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_p$</td>
</tr>
<tr>
<td>$N_s = 4$</td>
</tr>
<tr>
<td>$N_s = 8$</td>
</tr>
<tr>
<td>$N_s = 12$</td>
</tr>
</tbody>
</table>
This is due to the design approach used for the TX and RX shaping filters, based on [13]. In fact this method is especially tailored to the design of low-ICI raised-cosine filters.

V. CONCLUSIONS

In this paper an efficient modified decimator filter for DDC applications has been shown. The first preliminary results show that the proposed “pre-shaping” filter is able to perform two fundamentals tasks: it equalizes the overall frequency response of the decimation filter inside the bandpass of the signal of interest and behaves as a low-pass filter in the aliased bands. In this way the final FIR decimation filter, generally used in other schemes proposed in the literature, can be avoided. When a final filter is necessary (as in the PAM case considered in this paper), the design of this filter can be performed ignoring the decimation filter, so focusing the design on other system requirements.

This work is part of the study of the ground terminal of the MARVEL project (IST-1999-29043).

REFERENCES


Extended Algorithms for Sample Rate Conversion

Matthias Henker and Gerhard Fettweis
Technical University of Dresden,
Mannesmann Mobilfunk Chair for Mobile Communications Systems,
D-01062 Dresden, Germany
henker@ifn.et.tu-dresden.de

Abstract
The idea of software radio (SWR) implies the capability of changing the air-interface just by down-loading the respective software. Since analog components (e.g., for pre-filtering and digitization) are difficult to parameterize these tasks have to be moved to the digital domain, or have to be done in a standard independent way. In such receivers the task of sample rate conversion (SRC) is essential and has to be performed in an adaptable manner. Polynomial filters are a very suitable choice for sample rate conversion. Since the support (length) of the filter determines the effort and costs implementing this filter, minimizing the support is an important task. It will be shown that using a more general approach to interpolation leads to filters with minimal support for a given accuracy. This results in a considerable reduction of the effort.

1 Introduction
The current mobile communications market presents a diversity of indoor and outdoor standards. This does not agree with the mobility proposals of a seamless, any-time, and anywhere communication required in modern wireless services. In order to overcome this lack of uniformity the concept of software radio has emerged. Thus, upon this revolutionary paradigm multimode wireless communications devices have to be designed, which are able to implement the different physical layer functionalities on a reconfigurable hardware platform by simply running the corresponding software.

Mobiles will always be restricted in terms of hardware resources and power consumption. That requests simple algorithms. Furthermore, a common fixed hardware should be preferred enabling a reconfiguration of the terminal by means of software only. Another point is how to cope with the diversity of the symbol rates of different standards. One solution is to clock the ADC at an integer multiple of the master clock rate of each standard. Generating all these clocks with very low jitter is the resulting new problem. A better way out is to perform the AD-conversion at a fixed rate and adapt this process to the different symbol rates by sample rate conversion [1].

Sample rate conversion can be modeled as a process of sampling, reconstructing the signal, and resampling it again. Therefore the well-known effects of imaging and aliasing occur. The received signal is assumed to be sufficiently bandlimited by means of an analog anti-aliasing filter prior the first sampling process within the ADC. However, this is not the case for the resampling process at the standard-specific sample rate. The first sampling step has caused a spectral repetition of the spectrum of the received signal. This is called imaging. Some parts of the images will become aliasing components with respect to the resampling process at the standard specific sample rate. Therefore the effect of aliasing has to be taken into account before resampling. That is the reason why a reconstruction filter is required. One way is to reconstruct the signal to obtain a copy of the original signal. This is often meant by the term interpolation. This filtering undoes the process of sampling which has caused the periodic images. In fact, exactly these images are removed by interpolation filters. Therefore, these filters are also called anti-imaging filters.

But we do not really need a completely reconstructed signal. This is because we are mostly interested in only one channel of the received signal which occupies only a part of the whole signal bandwidth. So the most important constraint on the reconstruction filter is to prevent aliasing. This results in an anti-aliasing-filter in contrast to interpolation.

If the ratio of the specific target sample rate and the fixed digitization rate can be expressed by two integer numbers \( L \) and \( M \) like

\[
\frac{f_{\text{target}}}{f_{\text{ADC}}} = \frac{L}{M}
\]

sample rate conversion can be performed by a cascade of the following operations (see figure 1)

1. up-sampling by \( L \) (imaging occurs)
2. filtering (rejection of potential aliasing components)
3. down-sampling by \( M \) (hopefully no aliasing destroys the channel of interest)

\[ s(k\cdot T_{\text{ADC}}) \xrightarrow{\text{L up}} \text{reconstruction filter} \xrightarrow{M \downarrow} \tilde{s}(n\cdot T_{\text{target}}) \]

Figure 1: Model of sample rate conversion as a cascade of an up-sampler, reconstruction filter, and a down-sampler
2 Traditional interpolation

Interpolation means the calculation of in-between values of a sampled signal \( s(kT) \). The thus reconstructed signal \( s_{rek}(t) \) can be expressed as a weighted sum of the samples \( s(kT) \) of the original continuous-time signal \( s(t) \).

\[
s_{rek}(t) = \sum_{n=-\infty}^{\infty} s(nT) h_{\text{int}}(t-nT) \tag{2}
\]

The weights of the samples \( s(nT) \) are given by the values of the interpolation function \( h_{\text{int}}(t-nT) \). To fulfill the condition of interpolation in a strict sense (also called exact interpolation), the equation \( s_{rek}(kT) = s(kT) \) must hold. This requirement is equivalent to the condition

\[
h_{\text{int}}(kT) = \begin{cases} 
1 & \text{for } k = 0 \\
0 & \text{otherwise}
\end{cases} \tag{3}
\]

which is also called the interpolation constraint. To be able to reconstruct the sampled signal at arbitrary time instances, we need to know the complete function \( h_{\text{int}}(t) \). To easily evaluate \( h_{\text{int}}(t) \) at run-time as well as easily implement it in hard- or software simple descriptions of the interpolation function are required. Typically, the interpolation function is built up from polynomial or trigonometric functions [2, 3]. In this paper only polynomial interpolation will be treated. The interpolation function is described as a sum of piecewise polynomials

\[
h_{\text{int}}(t) = \sum_{i=-N_p/2}^{N_p/2-1} p_i(t) \tag{4}
\]

with \( N_p \) - number of concatenated polynomials, \((N-1)\) - highest degree of polynomials, \( c_{ij} \) - adjustable coefficients, and

\[
p_i(t) = \begin{cases} 
\sum_{j=1}^{N} c_{ij} \cdot \left(\frac{t}{T}\right)^{j-1} & \text{for } iT \leq t < (i+1)T \\
0 & \text{else}
\end{cases} \tag{5}
\]

Ideal interpolation means that \( s_{rek}(t) \) becomes identical to the original signal \( s(t) \) before sampling. This is only possible if \( s(t) \) is band-limited with a cut-off frequency of \( \frac{1}{2T} \). In this case error free reconstruction is possible using the well known sinc-function as the interpolating function \( h_{\text{int}} \).

3 Spline-Interpolation

In contrast to the traditional interpolation (see eq. (2)) the desired signal value \( s_{rek}(t) \) is no longer a weighted sum of the samples \( s(nT) \) but can formulated as a weighted sum of general coefficients \( c(nT) \) [4].

\[
s_{rek}(t) = \sum_{n=-\infty}^{\infty} c(nT) h(t-nT) \tag{6}
\]

This form is called generalized interpolation and is carried out in two separate steps. Equation (6) is only the second step. It requires a pre-filtering step, namely the determination of the coefficients \( c(nT) \) from the samples \( s(kT) \). This is the first step. The intention is to observe the above mentioned interpolation constraint:

\[
s_{rek}(kT) = s(kT) = \sum_{n=-\infty}^{\infty} c(nT) h((k-n)T) \tag{7}
\]

That means the non-interpolating reconstruction filter \( h(t) \) becomes interpolating due to the pre-filtering step. The equation above can be written as a convolution of the pre-filter output \( c(kT) \) and the sampled version \( h_S \) of \( h \) with \( h_S(kT) = h(t)|_{t=kT} \) as follows

\[
s(kT) = (c * h_S)(kT) \tag{8}
\]

This enables us directly the determination of the pre-filter output \( c(kT) \).

\[
c(kT) = (s * h_S^{-1})(kT) \tag{9}
\]

To characterize the required pre-filter \( h_{PF}(kT) = h_S^{-1}(kT) \) the Z-transform is determined.

\[
Z \left[ h_S^{-1}(kT) \right] = Z \left[ h_S(kT) \right]^{-1} = \frac{1}{H_S(z)} = H_{PF}(z) \tag{10}
\]

with \( Z \left[ h_S(kT) \right] = H_S(z) \). The reconstruction function \( h_{\text{int}}(t) \to h(t) \) is no longer bounded to the interpolation constraint (see eq. (3)). This allows an extended choice for \( h(t) \) with possibly better performance.

The most familiar example of generalized interpolation is the spline interpolation. It is a very powerful and well-investigated interpolation algorithm based on polynomials which is used in image processing often [4, 5]. Until now it has been scarcely used in the field of sample rate conversion by rational factors.

Signals \( s_{rek}(t) \) reconstructed by means of spline interpolation are piecewise polynomials concatenated at so-called knots \( s(kT) \) with the special property of being \((N-2)\) times continuously differentiable, where \((N-1)\) is the highest degree of the polynomials. Thus, \( h(t) \) must be \((N-2)\) times continuously differentiable, too. The shortest possible reconstruction function \( h(t) \) can be obtained for \( N_p = N \). It must be noted that the support or length of \( h(t) \) determines directly the required hardware effort. Hence, the search for short efficient filters \( h(t) \) is a demanding problem. For a certain degree \((N-1)\) there exists exactly one solution, namely the B-spline of degree \((N-1)\):

\[
h(t) = \beta^{N-1}(t) \tag{11}
\]

with

\[
\beta^{N-1}(t) = (\beta^0 \ast \beta^1 \ast \ldots \ast \beta^N)(t) \tag{12}
\]

\[
\beta^0(t) = \begin{cases} 
1 & \text{for } -T/2 < t < T/2 \\
1/2 & \text{for } |t| = T/2 \\
0 & \text{otherwise}
\end{cases} \tag{13}
\]
The B-spline \( \beta^N \) is very similar to the nearest neighbor interpolation (also called zero-order hold), \( \beta^N \) is equivalent to the linear interpolation. Both B-splines fulfill the interpolation constraint. This is not true for \( N \geq 3 \) anymore. Thus, a pre-filter \( H_{PF}^N(z) = 1/H_2(z) \) of order \( N \) is required (see tab. 1).

<table>
<thead>
<tr>
<th>( N )</th>
<th>( H_{PF}^N(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>( 8/(z + 6 + z^{-1}) )</td>
</tr>
<tr>
<td>4</td>
<td>( 6/(z + 4 + z^{-1}) )</td>
</tr>
<tr>
<td>5</td>
<td>( 384/(z^2 + 76z + 230 + 76z^{-1} + z^{-2}) )</td>
</tr>
<tr>
<td>6</td>
<td>( 120/(z^2 + 26z + 66 + 26z^{-1} + z^{-2}) )</td>
</tr>
</tbody>
</table>

Table 1: Transfer function of pre-filter \( H_{PF}^N(z) = 1/H_2(z) \) of order \( N \) for B-splines of degree \( (N-1) \)

For the case of increasing the sample rate of \( s(k\cdot T) \) by an integer factor \( L \) only some samples of \( h(t) \) are required but not the whole function \( h(t) \). In the case of spline-interpolation the reconstruction function becomes

\[
\begin{align*}
h(k\cdot T/L) &= \beta^{N-1}(t)|_{t=kT/L} \\
h_2(k\cdot T) &= \beta^{N-1}(t)|_{t=kT}
\end{align*}
\]

Interpreting equation (6) as a convolution we obtain a digital system as shown in figure 2 where \( h(z) \) is the Z-transform of \( h(kT/L) \) and \( H_{PF}(z) \) the Z-transform of \( h^{-1}(kT) \) (see eq. (10)).

\[
s(k\cdot T) \xrightarrow{H_{PF}(z)} L \uparrow \xrightarrow{H(z)} s_{\text{rec}}(n\cdot T/L)
\]

Figure 2: Increasing the sample rate by an integer factor \( L \)

Calculating the frequency response of the sampled B-splines unveils a very interesting connection to the very simple CIC-filters (for details on CIC-filters see [6]). The Fourier transform of a continuous-time B-spline of degree \( (N-1) \) is

\[
\hat{\beta}^{N-1}(f) = \mathcal{F}\{\beta^{N-1}(t)\} = \mathcal{F}\{(\text{sinc}(fT))^N\}
\]

with \( \text{sinc}(x) = \sin(\pi x)/(\pi x) \). Since \( \hat{\beta}^{N-1}(f) \) is not band limited, sampling the B-spline causes aliasing. Hence, the frequency response of the sampled B-spline \( H_{\text{B-spline}}^N(k\cdot T/L) = \hat{\beta}^{N-1}(t)|_{t=kT/L} \) becomes

\[
H_{\text{B-spline}}^N(\Omega) = \frac{1}{L} \sum_{n=-\infty}^{\infty} \left(\text{sinc}(L\Omega/(2\pi) - n)\right)^N
\]

with \( \Omega = 2\pi f T/L \), and \( G_{\text{B-spline}}^N(\Omega) = \left|H_{\text{B-spline}}^N(\Omega)\right| \) being the frequency response of a simple FIR-filter (see table 2) [7]. The filter \( H_{\text{CIC}}^N(\Omega) = H_{\text{CIC}}^N(z)|_{z=e^{j\Omega}} \) is a simple running sum filter (causal notation)

\[
H_{\text{CIC}}^N(z) = \left(\frac{\sum_{n=0}^{L-1} z^{-n}}{1-z^{-1}}\right)^N = \frac{1}{L^{N-1}} \hat{H}_{\text{CIC}}^N(\Omega) \cdot H_{\text{B-spline}}^N(\Omega)
\]

mostly implemented as the well-known CIC-filters [6]. Hence, the implementation of a spline-interpolator can be realized very efficiently (see fig. 3).

The combination of the B-spline with the pre-filter is called cardinal spline. Since the pre-filter is a symmetric IIR filter, the cardinal spline has infinite support (duration). Therefore there is no closed solution for the cardinal spline. The reconstruction function and the pre-filter are described separately. However, the frequency response \( H_{\text{CIC}}(\Omega) \) of the cardinal spline (overall system) can be formulated as a product of pre-filter and reconstruction function (see fig. 3)

\[
H_{\text{CIC}}^N(\Omega) = H_{\text{PF}}^N(\Omega) \cdot H_{\text{B-spline}}^N(\Omega)
\]

\[
H_{\text{CIC}}(\Omega) = H_{\text{B-spline}}^N(\Omega)
\]

\[
H_{\text{PF}}^N(\Omega) = \frac{1}{L^{N-1}} H_{\text{CIC}}^N(\Omega) \cdot G_{\text{B-spline}}^N(\Omega)
\]

with \( \Omega = 2\pi f T/L \). Figures 4 and 5 show an example of impulse and frequency response of a spline interpolation filter.

### 4 CIC-filters for interpolation and decimation

CIC-filters are very often used for interpolation and decimation, since they are quite simple [6]. They have good anti-imaging or anti-aliasing attenuation, respectively. But they lack from a severe passband droop, especially for higher orders \( N \). This is not the case for spline interpolation which provides a maximally flat passband. As it was shown in the last section the spline interpolators are closely related with the CIC-interpolators. Thus, it is worth trying to extend the CIC-interpolators by a pre-filter too. The performance will hopefully become similar to that of the spline interpolation. However, the implementation is expected to be completely different and to be simpler in many cases.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( G_{\text{B-spline}}^N(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>( (z + 6 + z^{-1})/8 )</td>
</tr>
<tr>
<td>4</td>
<td>( (z + 4 + z^{-1})/6 )</td>
</tr>
<tr>
<td>5</td>
<td>( (z^2 + 76z + 230 + 76z^{-1} + z^{-2})/384 )</td>
</tr>
<tr>
<td>6</td>
<td>( (z^2 + 26z + 66 + 26z^{-1} + z^{-2})/120 )</td>
</tr>
</tbody>
</table>

Table 2: Transfer function of the FIR-filter \( G_{\text{B-spline}}^N(z) \)
Figure 3: Spline-interpolation with digital filters

A CIC-filter for interpolation of order $N$ as well as a sampled B-spline $h_{B\text{-spline}}^N(k \cdot T/L)$ has zeros at $\Omega_0 = 2\pi n/L$ with $n \in [1, L-1]$ of order $N$. These zeros are responsible for the attenuation of the imaging components. B-splines of order $N$ have $2 \left\lfloor (N - 1)/2 \right\rfloor$ additional zeros improving the stop-band attenuation only very slightly. Therefore they should be omitted. However the thus resulting pre-filter $H_{PF}^N(z)$ depends on $L$ (see tab. 3), in contrast to the pre-filter of the B-splines (see tab. 1).

\[
\begin{array}{|c|c|}
\hline
N & H_{PF}^N(z) \\
\hline
1 & 1 \\
2 & 1 \\
3 & \left(8 \frac{L^2}{L^2 - 1}\right) / \left(z + 6 \frac{L^2 + 1}{L^2 - 1} + z^{-1}\right) \\
4 & \left(6 \frac{L^2}{L^2 - 1}\right) / \left(z + 4 \frac{L^2 + 1}{L^2 - 1} + z^{-1}\right) \\
\vdots & \vdots \\
\hline
\end{array}
\]

Table 3: Transfer function of pre-filter $H_{PF}^N(z)$ for CIC-filter of degree $N$ (In practical cases the numerator will be mostly set equal to one.)

The same principle can be used to extend CIC-decimators. In this case the pre-filter becomes a post-filter. The very good anti-imaging or anti-aliasing characteristics of CIC interpolators and decimators, respectively, are maintained. The pre- or post-filter makes the passband (theoretically) maximally flat without destroying the stopband. Since the pre- or post-filter can only be approximated a small passband ripple is superimposed. Thanks to the pre-filter the CIC-interpolator holds the interpolation constraint now (see eq. (3)).

The frequency response of the thus extended CIC-interpolator for an up-sampling factor $L$ (see fig. 6) is

\[
H_{E\text{-CIC}}^N(\Omega) = H_{PF}^N(L\Omega) \cdot H_{CIC}^N(\Omega) = H_{PF}^N(L\Omega) (\sin(L\Omega/2)^N/\sin(\Omega/2)^N)
\]

with $\Omega = 2\pi fT/L$. For the extended CIC-decimator for a down-sampling factor $M$ (see fig. 7) we get a very similar equation

\[
H_{E\text{-CIC}}^N(\Omega) = \left(\sin(M\Omega/2)^N/\sin(\Omega/2)^N\right) \cdot H_{PF}^N(M\Omega)
\]

with $\Omega = 2\pi f/T$. Impulse and frequency response of the extended CIC-filter are actually very similar to that of the spline interpolators. The influence of the dropped filter $G_{B\text{-spline}}^N(z) \rightarrow G^N(z) = 1$ is negligible.

Of course, there were several other ideas to correct the severe passband droop of CIC-filters like sharpening or using so-called ISOP-filters as pre- or post-filters, but the extension by an IIR-filter performs in a much better way [8, 9].

\[^1\] $\lfloor x \rfloor$ means the greatest integer less than or equal to $x$. 
5 CIC-filters for sample rate conversion by rational factors

Until now we have only dealt with integer factor sample rate conversion, namely interpolation and decimation. But the required sample rate conversion factor can be any rational or even irrational number. Assuming that each irrational factor can be sufficiently approximated by some rational factors, only rational factors will be treated.

To realize sample rate conversion by arbitrary rational factors, an interpolator and a decimator have to be cascaded. Using an extended CIC-interpolator of order \( N_1 \) and an extended CIC-decimator of order \( N_2 \) leads to the structure shown in figure 8. Although this solution seems simple, there is a big problem. The intermediate signal between interpolator and decimator as well as the whole integrator section is clocked \( L \)-times the input sample rate. Such implementations will overstrain any feasible hardware. But there are two facts from which it is possible to take advantage: 1) the integrator section is fed with \( L-1 \) zeros between each pair of input samples, and 2) the integrator section creates output samples which will never be used by the following differentiator section because the down-sampler drops all samples except each \( M \)th sample. The combination of the integrator section with the up-sampler and the down-sampler can be realized by a periodically time-variant system, clocked either at input or output sample rate. For a detailed discussion of the time-variant (modified) CIC-filters see [10, 14]. Meanwhile the CIC-filters have a multiplier-free structure, their time-variant implementation requires multipliers now. The required hardware effort is comparable with that of the Farrow-structure (see next section).

6 Polynomial filters for sample rate conversion by rational factors

Spline-interpolators as well as extended CIC-interpolators are special types of polynomial interpolation filters using the generalized interpolation (6). An example of polynomial interpolation filters using the traditional interpolation (2) are the Lagrange-interpolators. That means that for all these interpolators of a given order \( N \), the number of concatenated polynomials \( N_p \) and the coefficients \( c_{i,j} \) (see eq. (5)) are predefined. However, sometimes it is necessary to design a filter with certain characteristics, which are not covered by special filters. Some filter design strategies are discussed in [2] and [11]. A detailed comparison of a lot of special polynomial filters can be found in [4]. Figures 9 and 10 show an example of an optimized polynomial filter, where in contrast to the B-splines the transfer zeros in the stopband (potential aliasing components) are spread getting a wider stopband. This filter looks very similar to the so-called o-MOMS, which perform the best signal reconstruction for a given effort [15].

A cascade of up-sampler, polynomial filter, and down-sampler permits sample rate conversion by arbitrary rational factors. Still, we have the same problems as cascading CIC-interpolators and -decimators. A implementation which avoids the high intermediate sample rate and creates only the required output samples, is the so-called Farrow structure (see fig. 11) [12]. The Farrow structure consists of \( N \) FIR-filter branches and the length of each branch filter is \( N_p \). The filter coefficients \( \{c_{i,j}\} \) describe the impulse response \( h(t) = \sum_i p_i(t) \) (see eqs. (4) and (5)). More details about the Farrow structure can be found in e.g. [2, 12, 13, 14].
It is important to remark repeatedly that the length of the polynomials \( p_i(t) \) are bounded to the input sample period \( T \). That’s why, the Farrow-structure is not a very suitable choice for sample rate conversion by rational factors. This is because the Farrow-structure can only realize interpolation filters which are in fact very good anti-imaging filters, but what we really need are anti-aliasing filters. Thus, if the impulse response \( h(t) \) is built up by polynomials of length \( MT/L \) instead of \( T \), it would be possible to design anti-aliasing filters. Or equivalently, if the impulse response \( h(t) \) of the interpolation filters we dealt with in this paper, is scaled in time by the factor \( M/L \), the frequency response will be scaled by \( L/M \), thus transforming the image rejection into an aliasing rejection property. The transposed Farrow structure is exactly the looked for counterpart of the (original) Farrow structure (see fig. 12).

A very detailed derivation of this structure can be found in [14]. The transposed Farrow structure is very similar to the original one. There is only an additional integrate and dump unit required. For more details see [13, 14]. The length of the polynomials \( p_i(t) \) are now bounded to the output sample period \( MT/L \).

It is very interesting that the same filter coefficients \( c_{i,j} \) can be used for both structures, resulting in an interpolation or a decimation filter, respectively. Hence, all well investigated interpolation algorithms like spline or Lagrange interpolation can be used for decimation and sample rate conversion by rational factors, too. Of course, scaling the filter \( h(t) \) in time means that the interpolation constraint (3) is no longer hold, but it does not matter, because we do not want to reconstruct the whole signal \( s(t) \) but only prevent aliasing distortions within the channel of interest.
which is the case in image processing.

A second solution is to truncate the infinite impulse response and implement this windowed filter as a stable FIR-filter. It is quite sensible to merge this FIR-filter with a prior or following filter task (possibly matched filtering) by convolution of both impulse responses.

A third solution is to take use of the zero-pole cancellation technique. The idea is to cascade the unstable pre- or post-filter \( H_{PF}^N(z) \) and a FIR-filter having at least zeros at the unstable poles of \( H_{PF}^N(z) \). Since the frequency and phase response of the pre- or post-filter should not be deteriorated, the additional FIR-filter is a linear-phase, approximated all-pass. To show its simplicity an example for the cubic spline interpolation \((N = 4)\) is given. The transfer function of the unstable pre-filter is \( H_{PF}^N(z) = 6/(z^{-1} + 4 + z^4) \) (see tab. 1). The simplest possible transfer function of a suitable allpass approximation is

\[
H_{\text{approx. allpass}}^4(z) = \frac{1}{C_z + 2} \cdot (z^{-D} + C_z + z^{-D}) \tag{26}
\]

with \( D \in \mathbb{N}^+ \). The parameter \( C_z \) determines the placement of the transfer zeros, and \( D \) determine the number of zeros. Sensible values of \( D \) are in the range \([3 \ldots 8]\). Thus, the transfer function of the approximated pre-filter is

\[
H_{\text{approx. pre}}^4(z) = \frac{6}{C_z + 2} \cdot \frac{z^{-D} + C_z + z^{-D}}{z^{4} + 4 + z^4} \tag{27}
\]

where the scaling factor \( 6/(C_z + 2) \) can be neglected in most practical cases (see fig. 13). The coefficient \( C_z \) depends on the location of the transfer poles \( p_1 \) and \( p_2 \) and is

\[
C_z = -[p_1^D + p_2^D] \tag{28}
\]

For the cubic spline-interpolator the poles are at \( p_1 = -2 + \sqrt{3} \) and \( p_2 = -2 - \sqrt{3} \).

The drawback of this solution is that the unstable poles must be exactly cancelled by zeros. Therefore \( C_z \) must not be rounded. However, this again is equivalent to the truncation of the infinite impulse response of the pre- or post-filter. The window length is \( 2D - 1 \) samples for the example above.

### 7 Realizing the pre- and post-filter

Normally the reconstruction filter \( h(t) \) is symmetric and has finite support (duration). Therefore, the digital filter \( h_S(k \cdot T) = h(t)_{|t=kT} \) or \( h_S(k \cdot MT/L) = h(t)_{|t=kMT/L} \) respectively, is a linear phase FIR-filter. The pre- or post-filter \( h_{PF}(\cdot) \) is chosen as the inverse function of \( h_S(\cdot) \) (see eq. (10)), transforming the zeros of \( H_S(z) = \mathcal{Z}(h_S(\cdot)) \) to poles of \( H_{PF}(z) = \mathcal{Z}(h_{PF}(\cdot)) \). Thus, the pre- or post-filter is a linear-phase IIR-filter, especially an all-pole filter. Since some poles of \( H_{PF}(z) \) are outside the unit circle, the pre- or post-filter is unstable. However, there are some ways tackling this problem.

One solution is to split the pre- or post-filter in its stable and its unstable part (for details see [5]). However, this is only possible, if the input signal can be processed in blocks which is the case in image processing.

A second solution is to truncate the infinite impulse response and implement this windowed filter as a stable FIR-filter. It is quite sensible to merge this FIR-filter with a prior or following filter task (possibly matched filtering) by convolution of both impulse responses.

A third solution is to take use of the zero-pole cancellation technique. The idea is to cascade the unstable pre- or post-filter \( H_{PF}^N(z) \) and a FIR-filter having at least zeros at the unstable poles of \( H_{PF}^N(z) \). Since the frequency and phase response of the pre- or post-filter should not be deteriorated, the additional FIR-filter is a linear-phase, approximated all-pass. To show its simplicity an example for the cubic spline interpolation \((N = 4)\) is given. The transfer function of the unstable pre-filter is \( H_{PF}^N(z) = 6/(z^{-1} + 4 + z^4) \) (see tab. 1). The simplest possible transfer function of a suitable allpass approximation is

\[
H_{\text{approx. allpass}}^4(z) = \frac{1}{C_z + 2} \cdot (z^{-D} + C_z + z^{-D}) \tag{26}
\]

with \( D \in \mathbb{N}^+ \). The parameter \( C_z \) determines the placement of the transfer zeros, and \( D \) determine the number of zeros. Sensible values of \( D \) are in the range \([3 \ldots 8]\). Thus, the transfer function of the approximated pre-filter is

\[
H_{\text{approx. pre}}^4(z) = \frac{6}{C_z + 2} \cdot \frac{z^{-D} + C_z + z^{-D}}{z^{4} + 4 + z^4} \tag{27}
\]

where the scaling factor \( 6/(C_z + 2) \) can be neglected in most practical cases (see fig. 13). The coefficient \( C_z \) depends on the location of the transfer poles \( p_1 \) and \( p_2 \) and is

\[
C_z = -[p_1^D + p_2^D] \tag{28}
\]

For the cubic spline-interpolator the poles are at \( p_1 = -2 + \sqrt{3} \) and \( p_2 = -2 - \sqrt{3} \).

The drawback of this solution is that the unstable poles must be exactly cancelled by zeros. Therefore \( C_z \) must not be rounded. However, this again is equivalent to the truncation of the infinite impulse response of the pre- or post-filter. The window length is \( 2D - 1 \) samples for the example above.

### 8 Conclusions

This paper has given a short overview about traditional and generalized interpolation algorithms as well as their implementation as digital filters. It was marked that interpolation filters are anti-imaging filters undoing the process of sampling. Furthermore, it was stressed that sample rate conversion by rational factors require anti-aliasing filters more than anti-imaging filters. This is because not the whole signal \( s(t) \) needs to be reconstructed but only the channel of interest must be prevented from aliasing errors. Scaling the interpolation filters in time and frequency meets the requested requirements on the reconstruction filters.

The so far used interpolation constraint has always led to the Farrow structure. As the Farrow structure has been
turned to a transposed Farrow structure which automatically results in the required anti-aliasing properties, it is now possible to use a transposed form of spline interpolation. This automatically scales the filter properly in time and frequency, thus transforming the image rejection into an aliasing rejection property. Compared with traditional solutions this approach reduces the required effort considerably.

References


Specification for Digital Channel Selection Filters in a Bluetooth Capable HiperLAN/2 Receiver

Fokke Hoeksema, Roel Schiphorst, Kees Slump
University of Twente, Department of Electrical Engineering, Laboratory of Signals and Systems, P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 2770 Fax: +31 53 489 1060
E-mail: {f.w.hoeksema, r.schiphorst, c.h.slump}@el.utwente.nl

Abstract—In this paper we first briefly describe how the analog part of the channel selection function for a HiperLAN/2 receiver can be used as part of the channel selection function for a Bluetooth receiver. Thereafter we proceed to specify the overall requirements for the Bluetooth channel selection function. Part of these requirements are fulfilled by analog signal conditioning, part by digital processing. In order to assess the effects of analog processing early in the design process, we use a MAPLE model to assess the effect of analog filtering and sampling on a worst-case test signal. This test signal consists of the signal of the wanted channel and all possible worst-case interfering signals that, according to the standard, the system must be able to sustain. These interfering signals are not offered all at once, but one per BER-test. By interpreting results, we use the worst-case test signal to determine the cutoff frequency and order of the analog lowpass filter; the sampling frequency and resolution of the ADC and, finally, the requirements for the digital channel selection filters.

keywords: Software Defined Radio, HiperLAN/2, Bluetooth, Channel Selection requirements.

I. INTRODUCTION

In our Software Defined Radio (SDR) project we aim at combining two different types of standards, Bluetooth and HiperLAN/2 on one common hardware platform. HiperLAN/2 is a high-speed Wireless LAN (WLAN) standard (e.g. [1] and [2]), whereas Bluetooth is a low-cost and low-speed Personal Area Network (PAN) standard ([3]).

An SDR system is a flexible radio system that is re-programmable and re-configurable by software in order to cope with a multi-service1, multi-standard and multi-band environment. As can be seen in Table I the standards differ in several aspects and pose an interesting challenge for an SDR platform.

Goal of building a HiperLAN/2-Bluetooth demonstrator is to generate knowledge about designing the front end of an SDR system (from Antenna Reference Point (ARP) to including demodulation (Demodulation Reference Point (DRP), see Figure 1, taken from [4]). We are interested in the question how to use HiperLAN/2 hardware for implementation of Bluetooth functionality. HiperLAN/2 is a standard that leads to more complex implementations than Bluetooth (Table I may give an impression) - we do not expect Bluetooth capable hardware to be able to implement HiperLAN/2.

The receiver front-end we are interested in, consists of a channel-selection function and a demodulation function (see Figure 1). In this paper we present a channel selection function -partly analog, partly digital- that is suitable both for HiperLAN/2 and Bluetooth reception. However, we focus on Bluetooth reception only and specially on the channel selection system and its parameters. First the

1With a multi-service system we mean a system that is able to handle different types of data traffic: different with respect to content (email,web,video, ...), different with respect to traffic patterns and different with respect to QoS requirements.

channel selection subsystem is outlined (Section II). In order to find the requirements for the digital filters that accomplish Bluetooth channel-selection we first need

---

Table I: Bluetooth and HiperLAN/2 Parameters.

<table>
<thead>
<tr>
<th></th>
<th>Bluetooth</th>
<th>HiperLAN/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>PAN</td>
<td>WLAN</td>
</tr>
<tr>
<td>Frequency Band</td>
<td>2.4-2.4835 GHz</td>
<td>5.150-5.300 GHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.470-5.725 GHz</td>
</tr>
<tr>
<td>Access Method</td>
<td>CDMA</td>
<td>TDMA</td>
</tr>
<tr>
<td>Duplex Method</td>
<td>TDD</td>
<td>TDD</td>
</tr>
<tr>
<td>Modulation Type</td>
<td>GFSK</td>
<td>OFDM</td>
</tr>
<tr>
<td>Max. Data Rate</td>
<td>1 Mbps</td>
<td>54 Mbps</td>
</tr>
<tr>
<td>Channel Spacing</td>
<td>1 MHz</td>
<td>20 MHz</td>
</tr>
<tr>
<td>Max Power Peak</td>
<td>100 mW</td>
<td>200 mW - 1 W</td>
</tr>
</tbody>
</table>
to establish the overall requirements for the channel-selection function, so from Antenna Reference Point (ARP) to Channel Selection Reference Point (CRP), see Figure 1. These overall requirements are derived in Section III. They present the channel requirements both for the analog conditioning part of the system and for the digital channel selection filters.

Once we know the overall requirements, the issue is to determine what part of the channel selectivity is done using analog processing and what is part done using digital processing. The performance of high-speed high-resolution Analog-to-Digital Converters (ADCs) is one of the reasons to hamper the realization of the "ideal" Software Radio (in which the ADC is placed at ARP). The choice where to place the ADC has severe consequences for the power consumption of the front-end (e.g. see [5]). Early in the design process, a digital designer needs information about ADC parameters and about the signals to be processed that the analog designer cannot (yet) deliver, as the design process is ongoing. In this state of design-uncertainty, on one hand, the assumption "analog doesn't do anything", leads to a not-implementable subsystem; while the position "all channel selection needs to be done digital" may lead to too severe requirements for the digital filters (and likely too large power consumption of the front-end). So, the questions remain, where to position the ADC, what are its parameters and what are the consequences for digital filter-design?

In order to answer this question early in the design process, we developed a MAPLE [6] worksheet that enables us to assess the linear effects of analog filtering, mixing and AD Conversion on the power spectrum of the Bluetooth signal we are interested in. This system is based on the usage of a worst-case test signal, derived from the Bluetooth standard. It enables us to find parameters of the analog pre-processing and ADC.

In Section IV the worst-case test signal is defined. It needs to be interpreted in, maybe, an unexpected fashion, that will be highlighted. Our MAPLE worksheet is briefly explained here. In Section V we present results of applying our approach to the Bluetooth channel-selection function and give an example in which we find the parameters of the analog filters, the ADC parameters and the requirements for the digital filters. The paper ends with a conclusions section.

II. BLUETOOTH AND HIPERLAN/2 CAPABLE CHANNEL SELECTION

Using the HiperLAN/2 front-end for Bluetooth reception purposes, we first apply an analog bandpass filter to select the entire Bluetooth band (80 MHz wide). This signal is passed through a wide-band low-noise amplifier. Subsequently analog quadrature mixing is applied. The system is shown in Figure 2. The oscillator frequency is adjustable in steps of 10 MHz in order to follow the Bluetooth hopping pattern. In each hop-period, a so-called "chunk" of bandwidth of width $B_c = 10$ MHz is mixed toward zero. The number of chunks $N_c$ in the Bluetooth band is $N_c = 80/B_c$. The number of 1 MHz wide Bluetooth channels in a chunk is $N_b$, so $N_b * N_c = 80$.

The begin-frequency of Bluetooth channel-number $k$ is at $f_b(k)$, its centre frequency at $f_c(k)$ and end-frequency at $f_e(k)$. These frequencies (in MHz) are given by:

$$f_b(k) = 2402 + k$$
$$f_c(k) = 2402.5 + k \quad \text{for } 0 \leq k \leq 78$$
$$f_e(k) = 2403 + k$$

If the hopping-pattern of the Bluetooth receiver is at channel $k$, the channel is in chunk number $c$ ($0 \leq c \leq N_c - 1$). The relation between the channel number $k$, its position $p$ in a chunk ($0 \leq p \leq N_b - 1$) and its chunk number $c$ is given by $k = c.N_b + p$ in which:

$$p = k \mod N_b$$
$$c = \frac{k - p}{N_b}$$

All channels in chunk number $c$ are down converted with mixing frequency $f_0$ for which

$$f_0 = f_0(c) = f_b(c.N_b) = 2402 + c.N_b$$

holds. So, after quadrature mixing, we have the Bluetooth band with channel-number $k$ down-converted to a frequency band for which $p \leq f \leq p + 1$ [MHz] holds. The relevant channels for further processing are in a chunk of $0 \leq f \leq B_c = 10$ MHz. In both the in-phase and quadrature signal-path, an approximately 10 MHz wide lowpass filter is applied and an ADC with maximum sample-frequency of 65 MHz is used. In the digital domain a particular Bluetooth channel is selected and demodulation is performed. The details of the lowpass filter (exact cutoff frequency and order) and ADC (sample frequency and resolution) need to be investigated; moreover the requirements for the digital filters need to be found.

For Bluetooth signal reception, we mix at the low-side of the required 10 MHz chunk (see (3)), while for HiperLAN/2 the modulation frequency can be chosen in the centre of the required 20 MHz band. In Figure 3 the effects of central-band and low-band modulation are shown. In the picture power spectra are symbolized for, from top to bottom, the bandpass signal $x(t)$, its analytical signal $\tilde{x}(t)$, its complex envelope $\tilde{x}(t)$ and the in-phase and quadrature signals $x_r(t)$ and $x_q(t)$ (following the notation in [7]; see also [8] and [9]). It is clear that the bandwidth in the quadrature and in-phase channels...
III. Bluetooth Channel Selection Requirements

In the Bluetooth specification, [3, Chapter 4], experiments are defined that test the sensitivity and interference performance of a receiver. In such an experiment, a signal constellation, consisting of a wanted Bluetooth signal and an interfering signal, is offered to the receiver. The offered signal is demodulated and the number of erroneously received bits is counted in order to determine the BER.

In the Bluetooth standard different types of experiments are distinguished (see Table II). A first group of experiments tests for in-band interference: co-channel interference and three types of adjacent channel interference. In an experiment of this group, the interferer is one Bluetooth signal. Its power is given as a signal-to-interference ratio (\( S/I \)), (see [3, Table 4.1]). The second group tests for out-of-band blocking; there are different requirements for different frequency ranges. The interferer is a continuous wave signal of which the absolute power is given [3, Table 4.2]. An experiment in which the system achieves a BER \(< 10^{-3} \) for the specified power-level is a pass, otherwise the system fails. For a system to be conforming, it should pass all the BER-tests.

The in-band interference experiment-types differ in the distance between channel number of wanted and unwanted signal \( \Delta k \). In an experiment with channel-distance \( \Delta k \), the signal power \( P_{S \Delta k} \) is given by:

\[
P_{S \Delta k} = P_{S, \text{sens}} + 10 \text{ [dBm]}, \quad |\Delta k| = 0, 1, 2
\]

\[
P_{S \Delta k} = P_{S, \text{sens}} + 3 \text{ [dBm]}, \quad |\Delta k| \geq 3
\]

in which \( P_{S, \text{sens}} = 70 \text{ [dBm]} \)

The power of the interfering Bluetooth signal \( P_{I \Delta k} \) can be derived from the standard using the signal-to-interference ratio \( R_{S/I} \) (in [dB]). It also depends on \( \Delta k \) (see Table II):

\[
R_{S/I \Delta k} = P_{S \Delta k} - P_{I \Delta k} \quad \text{[dB]} \quad (5)
\]

Now, the issue is to derive the required interference-attenuation \( A_{\text{req} \Delta k} \) in order that the execution of the experiment leads to a pass of the BER-test. From analysis of the demodulation function, we need to find a signal-to-noise ratio \( SNR_{req} \) that is sufficient to pass the BER-test. In this paper we assume \( SNR_{req} = 21 \text{ dB} \); a refinement of this figure is given in our accompanying article, [10]. Moreover, we define a safety margin \( A_m \) that constitutes an extra suppression of the interferer of \( A_m = 3 \text{ dB} \). The attenuated interfering-power should (at maximum) equal the required noise power that follows from the required SNR by the demodulator and the safety margin:

\[
P_{S \Delta k} + R_{S/I \Delta k} A_{\text{req} \Delta k} = P_{S \Delta k} + SNR_{req} + A_m
\]

so that

\[
A_{\text{req} \Delta k} = R_{S/I \Delta k} SNR_{req} A_m \quad (7)
\]

In our case, the difference between signal-to-interference ratio and required attenuation is 24 dB (for all adjacent channels experiments, see Table II).

For a particular out-of-band blocking experiment, we assume that all interfering signal power is allocated into a 1 MHz wide band (jamming). Moreover, we assume that the wanted signal is 3 dB above sensitivity (3, Section IV.3), so at 67 dBm. We now are able to calculate signal-to-interference ratios \( R_{S/I} \) for the out-of-band experiments as well, as the absolute interfering signal power is given in the standard. As an example, consider an interfering signal within the 30 MHz-2000 MHz band. The system should pass the BER-test even if the interfering power is -10 dBm. For \( R_{S/I} \) we apply (5), we find
〈10〉 = 57 dBm. The so-found numbers are listed in the lower part of Table II. Also the attenuation the channel selection function should achieve is given there (according to (5)).

Above, we saw that for a certain type of interference, only one interfering channel per experiment is assumed. In order to be conforming, the system has to undergo all experiments and pass the BER-test for each experiment. All per-experiment requirements can be combined to one channel-selection filter that fulfills all single-experiment requirements. This filter specifies the attenuation that the functions between ARP and CRP should meet (see Figure 1). The resulting overall filter requirements (for the in-band interference experiments only) is given in Figure 4. In this figure, the centre frequency of the desired band (Δk = 0) is at 0 MHz; the centre frequency of the first adjacent channel (Δk = 1) is at 1 MHz etc. Vertically we see $A_{req}(\Delta k)$ in [dB]. Only filter requirements for adjacent channels with higher frequencies are shown. A transition band is included in the region where the Bluetooth spectrum falls off below its maximum. This transition-band specifies a don’t-care band in the filter design.

Fig. 4. Bluetooth Overall Channel Selection Requirements (based on in-band interference experiments).

**IV. A Test Signal and Model for Channel Requirements Assessment**

In the derivation of the channel selection requirements, we used the experiments that are specified in the Bluetooth standard. In an experiment (or BER-test) one interferer tests the demodulation capability of the receiver. At the end of our analysis, we defined an overall filter requirement that should enable the system to pass all tests. Repeated single-interferer experiments lead to an overall filter requirement. For analysis purposes of the front-end, we do the reverse. We conceive a test pattern that consists of all the single-experiment test signals joined together in one worst-case testsignal. We analyze the effect of this “wall of sound” on the front-end. The analysis is power-based, so a test signal is represented by its Power Spectral Density (PSD). The PSD of this test signal is implemented in a MAPLE [6] worksheet that models the signal processing operations of the propose front-end (in Figure 2). For linear filter and amplification operations the MAPLE model should mimic reality rather well, as no power in a particular frequency band of the “wall of sound” will be translated to another band. The effects on the test signal of a particular filter bandwidth and filter order can readily be interpreted.

For sampling and mixing this does not apply. In interpreting the results of the MAPLE model we have to bear in mind that not all constituting signals of the worst-case test signal are presented to the system in one ”wall-of-sound” experiment - there is no such experiment defined in the Bluetooth standard. If one, however, sees distortion in the wanted signal band, one has to keep in mind that, in that case, there is (there exists) an experiment that actually causes, for instance, the aliasing that we see.

In our current MAPLE model, no noise, no quantization effects and no non-linear effects and distortion are taken into account. The purpose of the model is in the early design phase of the project, where digital filter requirements have to be found, while the analog system is not fully specified. The ”wall of sound” test signal was created by first fitting an estimated Bluetooth power spectral density to functions that can be handled easily by MAPLE. We choose to model the PSD of a single channel using two Butterworth functions, one modelling the in-channel 1 MHz part of the spectrum, the other fitted to the tail. Subsequently we used this single channel approximation for creating the worst-case signal constellation. This constellation consists of one ”wanted” channel (at 0 dB), placed somewhere in the Bluetooth band, interfering Bluetooth channels with strength given by $R_{S/I}$ in Table II and an 100 MHz wide out-of-band signal with strength of $R_{S/I} = 40$ dB (also in Table II).

**A. Bluetooth spectrum fitting**

The normalized output spectrum of white noise, filtered by a $n_{bw}$-th order Butterworth filter with cut-off frequency $f_{bw}$ is given by

$$S_{bw}(f) = \frac{1}{1 + (f/f_{bw})^{2n_{bw}}} \quad (8)$$

The power spectrum $S_{bd}(f)$ of the complex envelope $\hat{b}(t)$ of a Bluetooth signal with frequency deviation $f_d = 0.175$ was modelled by

$$S_{bd}(f) = \frac{S_0}{\alpha + 1} \cdot (\alpha \cdot S_{bw,bw}(f) + S_{bw,bw}(f)) \quad (9)$$

in which the cutoff frequencies are given in MHz, the constant $\alpha = 1000$ and $S_0$ a normalization constant [mW/MHz], see Figure 11. The Bluetooth signal $b(t)$ itself is defined by the bandpass signal

$$b(t) = Re\{ \hat{b}(t) e^{j2\pi f_0 t} \} \quad (10)$$

in which $f_0$ is the modulation frequency.
B. Worst-case Signal Constellation

In Figure 5a the power spectral densities of a wanted channel (at 3 MHz) and its five adjacent channels \((0 \leq \Delta k \leq 3)\) are shown. It can be seen that especially the adjacent channels with \(\Delta k = 3\) introduce considerable power in the wanted channel. This in-band power is approximately 11 dB/MHz and thus comparable to the allowable level of co-channel interference in the Bluetooth standard (see also Table II).

For the "wall of sound" signal, in which the +40 dB channels shown in Figure 5 extend over the entire Bluetooth band, it was observed that the tails of these +40 dB channels add up in such a way, that the in-band power of the wanted channel was increased by approximately 3 dB. This is a consequence of the definition of the "wall of sound" signal and not a property of an interfering signal in a BER-test experiment of the Bluetooth system itself. From our "wall of sound" signal we require that the effect of additional in-band power in the wanted signal-band is negligible (no visible deviation from 0 dB). The inclusion of a transmission mask in the Bluetooth single-channel approximation solved this problem (see Figure 5b). A Butterworth filter with \(f_{bw} = 1\) MHz and \(n_{bw} = 3\) was used for this purpose. The filter changes the total power of a single channel marginally (less than 0.1%).

We have to stress the interpretation of Figure 5 (and the subsequent figures) once more. In fact we see the signal Figure 5b. A Butterworth filter with \(f_{bw} = 1\) MHz and \(n_{bw} = 3\) introduce considerable power in the wanted channel. This in-band power is approximately 11 dB/MHz and thus comparable to the allowable level of co-channel interference in the Bluetooth standard (see also Table II).

For the "wall of sound" signal, in which the +40 dB channels shown in Figure 5 extend over the entire Bluetooth band, it was observed that the tails of these +40 dB channels add up in such a way, that the in-band power of the wanted channel was increased by approximately 3 dB. This is a consequence of the definition of the "wall of sound" signal and not a property of an interfering signal in a BER-test experiment of the Bluetooth system itself. From our "wall of sound" signal we require that the effect of additional in-band power in the wanted signal-band is negligible (no visible deviation from 0 dB). The inclusion of a transmission mask in the Bluetooth single-channel approximation solved this problem (see Figure 5b). A Butterworth filter with \(f_{bw} = 1\) MHz and \(n_{bw} = 3\) was used for this purpose. The filter changes the total power of a single channel marginally (less than 0.1%).

We have to stress the interpretation of Figure 5 (and the subsequent figures) once more. In fact we see the signal constellations of 2 experiments with \(\Delta k = 3\), two experiments with \(\Delta k = 2\) and two experiments with \(\Delta k = 1\), so the test signals for a total of six experiments in one figure. We do not see the test signal of one experiment in which a wanted signal is hampered by 6 interfering channels.

C. Bandpass Signals

In Figure 3, power spectra of a band-pass signal, the complex envelope and quadrature and in-phase signal were depicted symbolically. In our MAPLE model we compute all of these signals and can, by visual inspection, see what the consequences are of applying signal processing functions. In Figure 6 the Bluetooth band is shown (twice) as a "wall of sound" signal \(w(t)\), with PSD \(S_{bw}(f)\). In Figure 6a, we defined a worst-case test signal in which the wanted channel is identified by \(k = 20\). Its centre frequency is \(f_c(20) = 2422.5\) MHz (according to (1)). The adjacent channels behave according to \(R_{S/F}\) in Table II. In Figure 6b the wanted channel is identified by \(k = 29\) with centre frequency \(f_c(29) = 2401.5\) MHz. These two channels form the extreme bands of the third chunk (\(c = 2\)). The mixing frequency \(f_0\) is equal for both channels and is given by (3), so \(f_0(2) = 2422\) MHz.

In case of a complex or quadrature signal, we experienced difficulty in interpreting the result of a signal processing operation, using only the power spectra of the original and processed signal as a means for assessment. An example is given in Figure 7. In Figure 7a (Figure 7b) we see the spectrum \(S_{bw}(f)\) of the in-phase signal corresponding to spectrum of the bandpass signal \(S_{bw}(f)\) of Figure 6a (Figure 6b). While both the spectra in the right and left figure are symmetrical (as they should be), the wanted signal is difficult to distinguish in the right figure. This problem is caused by the fact that phase-relations between signals are lost in a power-based analysis. The question remains how to assess consequences of a signal processing operation by inspecting these pictures.

We found a workaround to this problem by inspecting the effects of a signal processing operation \(F\) in a quadrature signal path or in-phase signal path at the level of the corresponding bandpass signal. While our system performs, for example \(y(t) = F(x(t))(t)\), we inspect \(y(t)\) and compare it with \(x(t)\), see (11). So, we inspect the result of a bandpass equivalent operation \(F^{bp}\) when assessing the impact of operations on complex signals or signals in quadrature (see (11)). Examples are given in the next section. In a future paper, we plan to give details on the method and its underlying assumptions.

\[
x_c(t) \xrightarrow{F} y_c(t) \\
x(t) \xrightarrow{F^{bp}} y(t)
\]

V. The 10 MHz Chunk System Revisited

In this section we present an example of what can be done with our MAPLE model. The two worst-case signals \(w(t)\) we offer to our system (see Figure 2) are depicted in Figure 6a and Figure 6b. In Figure 8 the impact of lowpass filtering of the in-phase \(w_c(t)\) and quadrature \(w_q(t)\) components of the worst-case signal is shown. The filter is a Butterworth filter with cutoff frequency \(f_{bw} = 11\) MHz and order \(n_{bw} = 4\). It should be stressed that we do not per-se propose to use a Butterworth-type of filter in our design (as delay distortion may be a problem, see [8, p. 99]). However, the Butterworth filter does give an impression of the effects on our worst-case signal of a particular choice of filter-order and cutoff-frequency.

We see that the channel with \(k = 20\) passes more or less unaltered by the filter (Figure 8a), while the channel with \(k = 29\) experiences some skewness, as it is in the transition region of the filter (Figure 8b). Whether this effect is harmful for demodulation performance is not known to the authors; further simulation and analysis could provide an answer. In the next example, the cutoff frequency is increased to \(f_{bw} = 12.5\) MHz.

In Figure 9 we see the effect of sampling the filtered signal with a sampling frequency of \(f_{sd} = 50\) MHz. The effect of sampling is shown at the level of the processed bandpass signal. Only the first-order aliasing components were taken into account (although there is
no in-principle objection to inclusion of higher order components). Also here, the effects of sampling were computed-back from the in-phase signal to the bandpass signal (symbolized by the right downward arrow in (11)). As stated in the last paragraph of Section IV-B, we have to interpret the figure carefully. When the aliased signal overlaps with the wanted channel in the filtered signal, we must conclude that there is a BER-test experiment that causes strong interference in the band of the wanted channel. In Figure 9a we see that the wanted signal with centre frequency \( f_c(20) = 2422.5 \text{ MHz} \) can be interfered by some BER-test interferer that introduces in-band power of approximately 6 \text{ dB/MHz}. This interference is caused by a Bluetooth interferer at the border of the Bluetooth band. In Figure 9b, we see that the wanted signal with centre frequency \( f_c(29) = 2431.5 \text{ MHz} \) can be interfered by some BER-test interferer that introduces in-band power of approximately 1 \text{ dB/MHz}. The interference is caused by an out-of-band interferer. In the next example therefore, the order of the lowpass filter is increased.

In Figure 10 the effects of a 6\text{th} order Butterworth filter are shown. Indeed, there is no BER-test that will cause too much interference when selecting these parameters (although for \( k = 29 \), at \( f_c(29) = 2431.5 \), 20 \text{ dB/MHz} out-of-band interference can occur).

The pictures can also be used to make a rough guess at the required resolution of the ADC. Following the 6\text{dB/bit} rule (see for instance [11]) the resolution of the ADC can be estimated to be 40 + 20 = 60 \text{ dB}, so at least 10 bit are necessary. Precise analysis is necessary to come to a definitive conclusion with respect to the number of bits (for an approach, see [11]).

We can derive the requirements for the digital filters quite straightforwardly. As we use the test signal that is used for specification of the overall front-end as an input-signal to the system and see the effects of analog processing and ADC immediately, we see what is still lacking in selectivity. As an example, consider the vicinity of the wanted channel in Figure 10a, representing the signal that passed the ADC. It resembles the vicinity of the original bandpass signal (see Figure 6a). One may conclude that all channel-selectivity as specified in Figure 4 has to be provided by the digital processing. The analog processing achieves an ADC-passable signal, but provides no channel selectivity. On one hand, this is caused by the parameters chosen in the examples above. On the other hand, and more importantly, it is a consequence of the system design, in which a chunk of Bluetooth channels has to pass the ADC.

VI. CONCLUSIONS

In this paper we presented a method for using the HiperLAN/2 front-end for Bluetooth reception purposes. We arrive at the requirements for the overall Bluetooth channel-selection function that have to be met by cooperation of analog and digital processing.

The method, in which we use a MAPLE worksheet to assess the impact of analog processing and ADC, proved useful as our analog system was not completely specified, while digital channel filter requirements were already needed.

It was observed that the analog processing can achieve an ADC-passable signal, but that the channel-selectivity requirements have to be met by the digital processing. So, the overall requirements for the channel-selection function (Figure 4) are valid for the digital processing alone.

ACKNOWLEDGMENT

This research is carried out in cooperation with the Integrated Circuit Design Laboratory of the University of Twente.

REFERENCES

Fig. 5. **PSD of wanted channel (at 3 MHz) and nearest strong adjacent channels in the worst-case signal constellation.**

In the figures below and on the next page, on the horizontal axis the power spectral density in [dB/MHz] is given, on the vertical axis the frequency f in [MHz].

Fig. 6. **Two worst-case test signals; Wanted channels at the extreme sides of a chunk (c=2). Bandpass signals, $S_{w,w}(f)$**

Fig. 7. **Two worst-case test signals; Wanted channels at the extreme sides of a chunk (c=2). In-phase signals: $S_{w_c,w_c}(f)$.**
Fig. 8. Two times a worst-case test signal and its filtered version ($f_{bw} = 11 \, MHz$, $n_{bw} = 4$). Filter-effect shown in Bluetooth band.

Fig. 9. Two times a filtered and sampled worst-case test signal ($f_{bw} = 12.5 \, MHz$, $n_{bw} = 4$; $f_{ad} = 50 \, MHz$). Filter and aliasing effect shown in Bluetooth band.

Fig. 10. Two times a filtered and sampled worst-case test signal ($f_{bw} = 12.5 \, MHz$, $n_{bw} = 6$; $f_{ad} = 50 \, MHz$). Filter and aliasing effect shown in Bluetooth band.
Fig. 11. Bluetooth Spectrum Approximation.
DIGITAL I/Q IMBALANCE COMPENSATION IN DIRECT-CONVERSION RECEIVERS

M. Valkama and M. Renfors
Institute of Communications Engineering, Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
E-mail:{valkama,mr}@cs.tut.fi

ABSTRACT
In radio receivers, the image signal problem related to frequency translations can be solved efficiently using I/Q signal processing. In theory, no RF image reject filtering is needed which prepares the way for highly-integrated receiver implementations. With practical analog electronics, however, the unavoidable mismatches between the amplitudes and phases of the I and Q branches limit the theoretically infinite image attenuation to only 20-40 dB range. In this paper, the problem of enhancing the analog front-end image attenuation is addressed in detail for direct-conversion receivers. An effective digital compensation method based on blind signal estimation is proposed and its performance is analyzed using computer simulations.

1. INTRODUCTION
In-phase/quadrature (I/Q) signal processing is a fundamental tool in processing of bandpass signals. Theoretically, it enables one to process negative and positive frequency bands separately. Considering frequency translations and downconversion in radio transceivers, I/Q processing can thus be used to solve the inherent image signal problem without exhaustive radio frequency (RF) image reject filtering. This kind of approach results in a clearly simplified analog front-end and is utilized, e.g., in the direct-conversion and low-IF receiver architectures [1]-[5].

The infinite image attenuation is, however, realized only if the amplitudes and phases of the two signal branches (I and Q) are perfectly matched. In practical analog implementations, some imbalance will always take place. With current circuit technologies, amplitude and phase imbalances of 1-5% and 1-5°, respectively, are realistically achievable [1], [2], [5].

Considering the downconversion process, these levels of imbalance compromise the theoretically infinite image attenuation to only 20-40 dB range [5]. Therefore, digital compensation techniques enhancing this image attenuation play an important role in using simple analog front-ends in future high-performance highly-integrated wireless receivers. With this in mind, the I/Q imbalance problem in direct-conversion receivers is addressed in this paper and a novel digital compensation approach is proposed and analyzed.

The organization of the paper is as follows. In Section 2, the direct-conversion architecture is briefly reviewed and a signal model for the front-end imbalances is presented. In Section 3, I/Q imbalance compensation in direct-conversion receivers is formulated as a signal estimation task and a blind signal separation (BSS) based compensator is proposed. The compensation performance of the proposed approach is analyzed using computer simulations in Section 4 and conclusions are drawn in Section 5.

2. DIRECT-CONVERSION RECEIVER
2.1 Architecture
In the direct-conversion receiver [1], [2] (also known as the zero-IF or homodyne receiver), the received signal is I/Q downconverted (see Figure 1) to lower frequencies. More specifically, the local oscillator (LO) frequency equals the desired channel center-frequency, i.e., the desired channel signal is downconverted directly to baseband. After downconversion, the adjacent channel signals are attenuated using low-pass filtering (LPF) and the I/Q signal pair is analog-to-digital (A/D) converted for further processing.

Now, since the IF is effectively zero, the image signal is actually the desired signal itself at negative center-frequency. This is illustrated in Figure 2. As will be shown later on, the effect of imperfect self-image rejection is seen as a certain linear distortion of the original signal constellation. As a result, the image attenuation requirements are not very strict if low-level modulations are used. However, with higher-order spectrally efficient modulation methods, such as 64-QAM, the 20-40 dB image attenuation of a practical analog front-end fails to suffice. In effect, the distortion caused by the finite self-image rejection results in dramatically increased symbol error rates (SER), thus calling for some kind of compensation.
2.2 Signal Models

To reveal the I/Q imbalance effect on the desired channel signal, proper modeling of the front-end imbalances is needed [6].

Imbalanced Front-End Model: Here, for analysis purposes, we model the amplitude and phase mismatches simply as an imbalanced quadrature local oscillator signal

\[ x_{LO}(t) = \cos(2\pi f_c t) - g \sin(2\pi f_c t + \phi) \]  

where \( g \) denotes the amplitude imbalance and \( \phi \) represents the phase imbalance. Using the well-known Euler’s formulas for the cosine and sine terms, the result of (1) can also be written as

\[ x_{LO}(t) = K_1 e^{-j2\pi f_c t} + K_2 e^{j2\pi f_c t} \]  

where \( K_1 = [1 + ge^{-j\phi}] / 2 \) and \( K_2 = [1 - ge^{j\phi}] / 2 \). In other words, the pure frequency shift of perfectly matched I/Q downconversion turns into two frequency translations with different relative strengths \( K_1 \) and \( K_2 \).

Baseband Signal Model: Denoting the desired channel baseband equivalent signal by \( z(t) = z_I(t) + jz_Q(t) \), imbalanced I/Q downconversion and lowpass filtering yields

\[ z'(t) = K_1 z(t) + K_2 z'(t). \]  

This is illustrated in frequency domain in Figure 2. In terms of the I and Q branch signals \( z_I'(t) \) and \( z_Q'(t) \), the model of (3) can also be written as

\[ z_I'(t) = z_I(t), \]

\[ z_Q'(t) = -g \sin(\phi) z_I(t) + g \cos(\phi) z_Q(t). \]  

Thus, the imbalanced I/Q signal pair is a linear transformation of the original I and Q signals. An illustrative example of this effect is given in Figure 3. Obviously, the imbalances increase the sensitivity to noise and other distortions in a considerable manner. Furthermore, with a high-level constellation and/or severe imbalance levels, the I/Q imbalance itself establishes a clear SER floor. Numerical SER results will be presented in Section 4.

3. I/Q IMBALANCE COMPENSATION

The I/Q imbalance problem has received some interest in the recent literature, see, e.g., [5]-[10]. Most of the existing techniques estimate the amplitude and phase mismatches \( (g \) and \( \phi \)) based on known training data and simply invert the model of (4) as

\[ \hat{z}_I'(t) = z_I'(t), \]

\[ \hat{z}_Q'(t) = \tan(\hat{\phi}) z_I'(t) + (\hat{g} \cos(\hat{\phi}))^{-1} z_Q'(t). \]  

given, of course, that \( g \neq 0 \) and \( \phi \neq \pm \pi / 2 \). Recently in [6] and [7], alternative approaches based on blind signal estimation techniques have been proposed. The idea in these contributions is to directly estimate the baseband equivalent of the desired channel signal (not the imbalances) by properly processing the observable signals \( z_I'(t) \) and \( z_Q'(t) \).

Philosophically, this is the approach taken here as well. However, the major difference is that [6] and [7] consider receivers with a non-zero IF. On the contrary, our application here is the direct-conversion receiver for which the observations in (4) are baseband signals (not IF signals as in [6] and [7]). Thus, in this paper, the signal estimation techniques are applied directly to the imbalanced I and Q signals to cancel the mismatch effect. In other words, by directly processing the signals of (4) in a novel manner, the original I and Q signals can be recovered as will be explained in the following.

3.1 Blind Signal Separation (BSS)

Blind signal separation (BSS) [11], [12] considers recovering some interesting signals (called sources) based on observing and processing their instantaneous linear mixtures. The term blind emphasizes that the mixing system is unknown and only the statistical properties of the source signals are exploited. More specifically, the strong assumption of statistically independent source signals is commonly employed.

For notational convenience, let \( \mathbf{s}(n) \) and \( \mathbf{x}(n) \) denote vectors of the source signal and observation samples, respectively. Furthermore, let \( \mathbf{A} \) denote the (unknown)
mixing matrix. Then, the linear instantaneous mixing system is characterized as \([11], \[12]\)

\[
x(n) = As(n).
\]  

(6)

Now, the recovery of the source signals consists of another matrix multiplication

\[
y(n) = W(n)x(n) = C(n)s(n) = \hat{s}(n)
\]  

(7)

where \(C(n) = W(n)A\). Various different techniques to determine the separator \(W(n)\) are discussed in \([11]\). One interesting approach to adaptive update of the separator matrix is the so called EASI (Equivariant Adaptive Separation via Independence) algorithm \([12]\) for which the basic serial update rule is of the form

\[
W(n+1) = (I - \lambda(n)H(y(n)))W(n).
\]  

(8)

In (8), \(\lambda\) is the adaptation step-size and the adaptation function \(H(\cdot)\) is of the form

\[
H(y) = yy^T - I + f(y)y^T - yf(y)^T
\]  

(9)

where \(f(\cdot)\) is a vector-valued non-linearity which operates component-wise on its argument. Much more details can be found in the original paper \([12]\).

3.2 Imbalance Compensation

In order to utilize the previous blind signal estimation principles in the imbalance compensation task, we can write our fundamental signal model of (4) simply as

\[
x(n) = \begin{bmatrix}
z_I(n) \\
z_Q(n)
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
-g\sin(\phi) & g\cos(\phi)
\end{bmatrix} \begin{bmatrix}
\hat{z}_I(n) \\
\hat{z}_Q(n)
\end{bmatrix}
\]  

(10)

where \(z_I(n) = z_I(nT_s)\), etc. This (discrete-time) model for the imbalanced observations fits directly to the general BSS formulation with two source signals \(z_I(n)\) and \(z_Q(n)\) to be recovered based on observing two of their linear mixtures \((\hat{z}_I(n)\) and \(\hat{z}_Q(n))\). In other words, a proper BSS algorithm operating on \(\hat{z}_I(n)\) and \(\hat{z}_Q(n)\) can recover the original I and Q signals \(z_I(n)\) and \(z_Q(n)\) without any training signaling or prior knowledge of the front-end imbalances \((g\) and \(\phi))\). Thus, the model of (10) is introduced here only for analysis purposes and is not used in any way in the actual compensation processing. The baseband system model for the whole transmission chain including the compensator is presented below in Figure 4.

In general, blind separation cannot determine the order nor the scaling (power) of the separated signals \([11], [12]\). The whitening term \(yy^T - I\) in (9) results in unit variance signals and thus reduces the scale indetermination to unknown signs. In our application, with practical imbalance values and initial value \(W(0) = I\) for the separator matrix, the order and signs of the separated signals will in practice always match the true values. Therefore, no additional control considering these aspects is needed.

The fundamental assumption generally employed in blind signal separation is the statistical independence of the original source signals \([11]\). In our application, the signals to be recovered are the I and Q components of a digitally modulated signal and the validity of the independence assumption needs to be addressed with care. One example modulation type for which the resulting I and Q signals (symbol sequences) are indeed independent is QAM. More specifically, the I and Q components of \(M^2\)-QAM (\(M\) integer) signal with equiprobable symbols are statistically independent. To see this, the probability \(P_I(i,q) = 1/M^2\) for each symbol belonging to the alphabet. Furthermore, \(P_I(i) = P_Q(q) = 1/M\) and thus \(P_I(i)P_Q(q) = P_I(i)P_I(i)\) which implies independence. Thus, the proposed method is directly applicable to standard QAM signals.

On the other hand, the I and Q components of PSK modulated signals are not independent. As an example, consider 8-PSK with alphabet \(\exp(jm\pi/4), m = 0, \ldots, 7\). With equiprobable symbols, \(P_I(i,q) = 1/8\) but, e.g., \(P_I(i) = 1/8\) and \(P_Q(0) = 1/4\) and thus \(P_I(i)P_Q(0) \neq P_I(1)P_O(0)\). This implies statistical dependence. Notice, however, that with equiprobable zero mean symbols, \(E(I\cdot Q) = E(I\cdot Q) = E(I\cdot Q) = 0\) for any PSK signal and for any non-linear function \(f(\cdot)\). Thus, on average, the adaptation function \(H(\cdot)\) of (9) is zero if the matrix \(W\) is a separating matrix. This, in turn, implies that a true separating solution, if found, is a stationary point for the serial update of (8) also with PSK modulated signals, even though the I and Q signals are statistically dependent (see \([12]\) for a thorough performance analysis of the EASI algorithm).

One practical aspect related to the direct-conversion receiver is the DC-offset problem \([1], [2]\). Due to finite isolation between the mixer RF and LO ports, part of the local oscillator signal leaks into the mixer input and self-mixes itself to baseband. This self-mixing product appears directly on top of the desired signal at baseband creating low-frequency interference. Considering our imbalance compensator, since the whole formulation is for zero mean data, the DC-offset problem needs to be tackled before applying the separation algorithm. As a simple example of DC-offset compensation, block-wise estimation and subtraction of sample means of the I and Q branch signals could be used \([1], [2]\).

Another possible practical effect is related to carrier phase synchronization. In practice, the relative phases of the transmitter and receiver LOs can differ. Using \(\theta\) to denote this phase difference, the receiver imbalanced LO signal is given by
\( x_{LO}(t) = \cos(2\pi f_c t + \theta) - j g \sin(2\pi f_c t + \theta + \phi) \). (11)

Proceeding then as we did in Section 2 yields
\[ z'_i(t) = \cos(\theta)z_i(t) + \sin(\theta)z_q(t), \]
\[ z'_q(t) = -g \sin(\theta + \phi)z_i(t) + g \cos(\theta + \phi)z_q(t). \] (12)

This model, in turn, can be again formulated as a BSS signal model
\[
\mathbf{x}(n) = \begin{bmatrix} z'_i(n) \\ z'_q(n) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -g \sin(\theta + \phi) & g \cos(\theta + \phi) \end{bmatrix} \begin{bmatrix} z_i(n) \\ z_q(n) \end{bmatrix} = A \mathbf{s}(n). \] (13)

Since \( \det(A) = g \cos(\phi) \), the traditional assumptions \( g \neq 0 \) and \( \phi \neq \pm \pi/2 \) are sufficient to guarantee that also this model is invertible. Thus, a BSS algorithm can recover the original I and Q signals also in the presence of a common phase error. And even further, a separation algorithm not only unravels the imbalance effect despite the phase error but also cancels the phase error itself.

### 4. PERFORMANCE SIMULATIONS

In order to illustrate the effectiveness of the proposed compensation approach, some example simulations are carried out. A sequence of 100,000 symbols is generated with each symbol drawn randomly from a 64-QAM alphabet. To model the receiver imbalance effect, this symbol stream is then transformed according to (10) and some additive white Gaussian noise (AWGN) is added.

The ultimate performance measure of any digital communication system is the symbol error rate (SER). For reference, the SER of a traditional minimum distance symbol-by-symbol detector operating on the uncompensated signal is reported in Figure 5 as a function of signal-to-noise ratio (SNR) for different imbalance values. As is evident, I/Q imbalance if left uncompensated deteriorates the signal quality in a dramatic manner, creating a clear error probability floor.

Next, in order to illustrate the power of the BSS based compensation idea, the observable sequence (with \( g = 1.05 \) and \( \phi = 5^\circ \)) is processed using the EASI separation algorithm with update parameter \( \lambda = 5 \times 10^{-4} \). Standard cubic non-linearities \( f_1(y) = f_2(y) = y^3 \) are used and the separator matrix \( \mathbf{W} \) is initialized as \( \mathbf{W}(0) = \mathbf{I} \). With this configuration, convergence is established in approximately 5,000 iterations. The steady-state separator output sequence is then fed to the previous symbol-by-symbol detector and the resulting symbol error probability is reported in Figure 6 together with the theoretical AWGN bound. Clearly, by using the proposed signal separation based compensation scheme, the imbalance effect can be efficiently compensated and the performance approaches the theoretical AWGN bound. For further illustration, corresponding results with \( \lambda = 1 \times 10^{-4} \) are depicted in Figure 7.

One practical aspect which should be considered in this context is related to error control and channel coding. To be specific, the redundancy introduced by a channel coder to the transmitted bit stream can actually violate the assumption of statistically independent I and Q components in the corresponding symbol sequence. As an example, if a repetition code is used, not all the symbol values are equally likely, depending, of course, on the relation between the repetition rate and the size of the symbol constellation. However, practical channel codes with feasible redundancy (and combined with scrambling) will not have any notable effect on the compensation. As a practical example, we consider the rate 1/2 convolutional code utilized in the 3GPP UTRA technical specifications [13]. The generator matrix for this code is \([561, 753]\) in octal form. In the simulation, a bit sequence is encoded with this code and the resulting bit stream is mapped to a symbol stream using 64-QAM. After that, the simulation proceeds as described previously to evaluate the uncompensated and compensated symbol error rates. According to the obtained results, the redundancy and the resulting (possible) dependence between the I and Q signals have only a negligible effect on the performance.

![Figure 5: Symbol error probabilities without compensation for different imbalance levels.](image1)

![Figure 6: Symbol error probabilities with and without compensation. Adaptation step-size \( \lambda = 5 \times 10^{-4} \).](image2)
compensation method was proposed. The idea was to unravel the imbalance distortion using blind signal estimation techniques. Based on that, a novel digital receiver was addressed in detail. A signal model for the estimation techniques. As illustrated by the simulation results, though preliminary, the I/Q imbalance can be efficiently compensated using the proposed approach.

Generally, the blind signal separation methods build on the assumption of statistically independent source signals. In this paper, we showed that the two output signals of an imbalanced I/Q downconversion stage appear as linear mixtures of the corresponding ideal signals. Thus, if these ideal I and Q components are statistically independent (as is the case, e.g., for QAM signals), they can be recovered using a proper blind signal separation algorithm. This was the leading philosophy behind the proposed compensation scheme. One practical aspect related to this approach is the effect of channel distortions. In effect, a true bandpass channel cross-coupling between the channel-filtered I and Q signals. In other words, the I and Q components of the signal entering the receiver are no longer independent which complicates the separation (compensation) task. This being the case, equalization of the channel distortions and compensation of the I/Q imbalances can and should be performed jointly. As an example, instead of the instantaneous signal estimation methods, convolutive mixture separation techniques [11] could be applied. This forms an interesting topic for future work.

5. DISCUSSION AND CONCLUSIONS

The I/Q imbalance problem in direct-conversion receivers was addressed in detail. A signal model for the imbalances was given and based on that, a novel digital compensation method was proposed. The idea was to unravel the imbalance distortion using blind signal estimation techniques. As illustrated by the simulation results, though preliminary, the I/Q imbalance can be efficiently compensated using the proposed approach.

ACKNOWLEDGMENT

This work was carried out in the project “Advanced Transceiver Architectures and Implementations for Wireless Communications” funded by the Academy of Finland. The work was also supported by the Graduate School in Electronics, Telecommunications and Automation (GETA), the Finnish Cultural Foundation, and the Nokia Foundation.

REFERENCES

A SOFTWARE DEFINED RADIO FRONT-END IMPLEMENTATION

S. Abendroth\textsuperscript{1,2}, R. Heuze\textsuperscript{1,3}, S. Weiss\textsuperscript{3}

\textsuperscript{1} Dept. Electronics & Computer Science, University of Southampton, UK
\textsuperscript{2} University of Karlsruhe, Karlsruhe, Germany
\textsuperscript{3} Ecole Supérieure d’Ingénieurs en Electrotechnique et Electronique, Paris, France
\texttt{s.weiss@ecs.soton.ac.uk}

\textbf{Abstract.} We report on the implementation of a software defined radio (SDR) based on state-of-the-art digital signal processors (DSPs). While time critical operations are executed on specialised transmit and receive processors with a fixed block structure, the baseband processing is performed on highly flexible DSPs. We comment on the SDR building blocks, and the software implemented on this test-bed including blind synchronisation, equalisation, and carrier recovery for differentially encoded quadrature amplitude modulation. The functionality has been tested by audio transmission. We conclude with an analysis of capabilities and limitations of the implemented SDR structure.

\section{Introduction}

The design of highly flexible digital communication systems has become an area of considerable interest. Particularly for mobile wireless transmission via hand-held devices, size and cost competitiveness usually set limitations when trying to implement systems compatible with multiple standards that exist throughout the world. Similarly, upcoming standards will overlap with existing ones for a significant interim period, such as for example for the transition from GSM to UMTS [1, 2].

This has motivated the concept of a software defined radio (SDR), whereby the digital-to-analogue and analogue-to-digital conversion are performed as close as possible to the radio frequency. The aim of extending the digital domain is to implement modulation, demodulation, channel coding and other required processing tasks in software [3, 4]. Therefore, users, service providers, and manufacturers become more independent of the realisation of one specific data transmission standard, since by downloading appropriate software code, a different functionality can be adopted by the communications system.

In this paper we report on the implementation of a software radio test-bed with an SDR transmitter and receiver. Both functions are implemented on C6711 digital signal processors performing the baseband operations. The conversion between baseband and an intermediate frequency (IF) is achieved by separate, fast digital transmit and receive processors as suggested in [5, 6]. The transmitted and received IF signals have a resolution of 12 and 14 bits respectively, and can be sampled at a maximum of 65 MHz.

We first discuss the hardware employed in realising the SDR transmitter and receiver in Sec. 2. Software implementations, particularly for the receiver functions such as blind adaptive carrier recovery, synchronisation, and equalisation, are outlined in Sec. 3. Sec. 4 highlights the testing and running of a 96 kbit/s data link across the SDR, while in Sec. 5 conclusions are drawn.

\section{Hardware Platform}

The general setup of the implemented SDR transceiver is shown in Fig. 1. Two floating point DSPs operate as baseband units in both the transmitter and the receiver, while dedicated circuitry in form of a digital transmit processor (DTP) and a digital receive processor (DRP) are employed for the conversion and processing stages between baseband and IF. In the following, we comment first on the transmit function in Sec. 2.1 and thereafter on the receiver hardware in Sec. 2.2.

\subsection{Transmitter Circuit}

The transmitter hardware encompasses a Texas Instruments C6711 floating point DSP for baseband processing. Based on a suitable input signal, this processor provides a stream of complex valued symbols for transmission, which are then passed to a DTP, for which an Analog Devices AD9856 has been utilised. The selected DTP has an on-board digital-to-analogue converter, (DAC) to finally transform the digital data into an analogue signal at IF. As indicated in Fig. 1, the circuit is completed by an amplifier with variable gain, which can be controlled digitally.
Our selected baseband DSP is suitable for processing tasks that are highly demanding in terms of required computational complexity, and can therefore encompass tasks such as source and channel coding. The C6711 DSP used here, evaluating 900 million floating point operations per second at a clock rate of 150 MHz, resides on a low-cost DSP starter kit (DSK) board, which permits access to various ports and interfaces of the DSP. For the communication between DSP and DTP, both the DSP’s data bus and one of its two multichannel buffered serial ports (McBSP) have been utilised. Inphase and quadrature components of the complex valued symbols are presented to the DTP interleaved and in a parallel fashion over the DSP’s expansion memory interface, whereby 12 bits parallel data and a control line to enable transmission are occupied. The initialisation of the DTP and also its control during transmission are serviced via an McBSP by the DSP, over a number of control lines. Finally, the second McBSP on the baseband board is connected to a low cost onboard analogue-to-digital converter (ADC) acquiring a 12 bit single-channel audio data stream at a sampling rate of 8 kHz. In our setup, this analogue input is currently used to provide audio data for transmission as shown in Fig. 1.

The baseband data is transferred to the DTP via custom build latches, with inphase and quadrature data components interleaved. The DTP is placed on an evaluation board comprising an AD9856 quadrature digital up-converter. Within this up-converter, it is possible to download different sets of coefficients for transmit filtering and choose different oversampling ratios in the interpolation stages. At the output of the interpolation stage, a quadrature amplitude modulation is placed, where the inphase and quadrature signal components are multiplied by sine and cosine waveforms. At a maximum input bandwidth of 25 MS/sec (considering both inphase and quadrature components), the maximum sampling rate at the output of the DTP is 160 MHz, which is limited by a 65 MHz lowpass filter and the rate of the on-board digital-to-analogue converter (DAC). This DAC has a resolution of 12 bits, and is followed by a programmable gain amplifier (AD8320) for power control. This gain amplifier can be regulated through the transmit processor, which in turn is controlled via the DSP’s serial port.

As indicated in Fig. 1, the analogue IF signal could now be further modulated up to radio frequency and be transmitted via an antenna. A matching analogue circuit could receive the transmit antenna signal, and down-convert it to IF. In our test-bed, this hardware stage is currently omitted, and the IF signal from the DTP is passed straight back into a digital receive processor (DRP).

### 2.2 Receiver Circuit

On the receiver side, the analogue IF signal is fed to an AD6644ST analogue to digital converter (ADC) sampling at a maximum of 80 MS/sec with 14 bit resolution. As shown in Fig. 1, this data is passed to the DRP over a parallel port. Here, this dedicated processor is an AD6620, which has a maximum input bandwidth of 65 MS/sec for real signals or 32.5 MS/sec for complex valued data. Over two stages of cascaded-integrator-comb (CIC) filters with definable decimation ratios, finally a programmable receive filter of order 255 can be applied to the data in the down-conversion process.

From the DRP, the inphase and quadrature data is then serially transmitted to the second C6711 DSP hosting the baseband receiver functions. Although the selected DRP also allows a parallel interface, the employed DSK board permits only to write to the McBSP but not the data bus on the expansion memory interface. Through the McBSP, the DSP can also control the parameters on the DRP, including the decimation ratios as well as the selection of the CICs and the receive filter.

In the baseband receiver DSP, the oversampled data is fed into a synchronisation and equalisation stage, which will be described in Sec. 3. Thereafter, the complex data symbols are translated into a bitstream. From this bit stream, another synchronisation process is required to extract the transmitted data words. Finally as shown in Fig. 1, our implementation sends these 12 bit data words to the

![Fig. 1: Block diagram of the implemented software defined radio test-bed.](image-url)
on-board DAC via the second McBSP to output an analogue audio waveform.

3 IMPLEMENTATIONS

After setting up the software controlling the different devices as well as communication between the baseband DSPs and the transmit and receive processors, the SDR platform outlined in Sec. 2 is reasonably flexible and can be programmed in C for any functionality within the capabilities of the testbed. While in principle the bandwidth of the DTP and DRP would permit the implementation of WCDMA with a required bandwidth of 5 MHz, using the McBSP1 in the receiver DSP and the baseband processing set limitations. Hence, in the following we describe the implementation of a data link of lower bandwidth such as found in GSM using the previously defined hardware platform.

3.1 Modulation and Demodulation

In the baseband DSP of the transmitter, the 12 bit audio samples were converted into a bit stream, which in turn was subjected to a Gray encoded differential QPSK (D-QPSK) scheme. Differential encoding was selected to enable blind synchronisation and equalisation without phase ambiguity in the receiver. The D-QPSK symbols are upsampled by a factor of four, and pulse shaping with a root raised cosine filter is applied prior to transferring the data to the DTP.

The clocks in the transmitter and receiver circuits are governed by the baseband DSPs. As after demodulation a clock mismatch will lead to a carrier frequency offset, a time-varying rotation of the received D-QPSK symbols and a slight difference in the transmitted and received data rates can result. In order to mitigate these problems, the receive baseband DSP requires synchronisation and carrier offset recovery [7, 8]. This synchronisation is performed together with timing synchronisation and equalisation, which will be outlined in the next section.

3.2 Synchronisation and Equalisation

Amongst a variety of algorithms for carrier and timing offset compensation, such as phase-locked-loops and early-late-gate methods [7], adaptive filtering was chosen as a convenient way of solving the different synchronisation issues. We here apply a fractionally spaced adaptive equaliser, whereby the input is oversampled at twice the symbol rate to achieve timing greater resolution over standard symbol-spaced equalisers [9].

To recover the D-QPSK symbols, different adaptive schemes can be invoked. Standard adaptive algorithm can be employed if a training sequence is available. For our SDR testbed, a blind synchronisation and equalisation scheme was preferred, whereby the transmitted data is unknown and recovery is based on the knowledge of the transmitted symbol alphabet only. A well-behaved class of adaptive equalisers belong to the family of constant modulus algorithms (CMA) [10]. The CMA however leaves the phase of the received symbols ambiguous. Therefore, as specifically D-QPSK modulation had been selected for data encoding, a decision-directed scheme without phase-ambiguity was selected, as shown in Fig. 2.

With the oversampled received data $x[n]$, both the received data available to the equaliser at sampling period $m$ as well as the equaliser coefficients $h[m]$ can be denoted in vector notation as

$$
x_m = \begin{bmatrix} x_{0,m} \\ x_{1,m} \end{bmatrix}, \quad h_m = \begin{bmatrix} h_{0,m} \\ h_{1,m} \end{bmatrix}, \quad (1)
$$

whereby

$$
x_{i,m}^T = [x_i[m] \ x_i[m-1] \ldots \ x_i[m-L+1]] \quad (2)
$$

$$
h_{i,m}^T = [h_{i,m}[0] \ h_{i,m}[1] \ldots \ h_{i,m}[L-1]] \quad (3)
$$

with $i \in \{0,1\}$. The output $y[m]$ of the equaliser is fed into a QPSK decision device with output $g[m]$. The difference between the input and output of this slicer therefore defines the error of the equaliser,

$$
e[m] = g[m] - y[m] = g[m] - h_m^T \cdot x_m. \quad (4)
$$

In our scheme, the decision of the correct slicer output is based on the closest vicinity of a QPSK constellation point $s_{i,k} = ((-1)^j + j(-1)^k) / \sqrt{2}$, with $j = \sqrt{-1}$ and $l, k \in \{0,1\}$, to the received symbol $y[m]$,

$$
g[m] = \arg \min_{i,k} |y[m] - s_{i,k}| \quad (5)
$$

With a suitable error minimisation criterion, for example a decision-directed normalised least mean squares (NLMS) adaptive algorithm

$$
h_{m+1} = h_m + \mu \frac{x_m}{||x_m||^2} \cdot e^*[m], \quad (6)
$$

the filter coefficients in the equaliser can be automatically adjusted [9, 11].

The fractionally spaced equaliser described above is capable of carrier offset recovery by following a dynamic optimum solution,

$$
h_{opt,m} = c_{mv} \cdot e^{j(\omega_{cr} \cdot m + \theta_{cr})}, \quad (7)
$$

![Fractionally spaced adaptive equaliser in polyphase realisation with decision-directed updating.](image-url)
Fig. 3: Received constellation pattern (a) without and (b) with carrier offset recovery by the adaptive equaliser.

whereby $c_{inv}$ is the linear time-invariant channel inverse regularised by the channel noise, and $\Omega_{off}$ and $\theta_{off}$ are the carrier normalised angular frequency and phase offsets, respectively. Timing recovery is achieved by $c_{inv}$ modeling a fractional delay filter, whereby the equaliser benefits from an increased resolution due to fractional spacing. Therefore, as long as the initial starting value for $h_m$ is reasonably close to the true solution, and the carrier frequency offset $\Omega_{off}$ is within limits, the adaptive algorithm in (6) can converge to and track the optimum solution for the equalisation problem.

4 OPERATION AND AUDIO TRANSMISSION EXAMPLE

The operation of the SDR can be verified by accessing all memory locations and registers on the DSPs via the software tools provided for the C6711. An example for the operation of the carrier recovery and synchronisation is given in Fig. 3(a) and (b), where the received constellation over a short time interval shows the rotation of the symbols which in Fig. 3(b) has been correctly compensated to retrieve the data.

Therefore with the correct operation of the overall SDR hardware and software as outlined in Fig. 1, a 96 kbit/s transceiver for narrowband speech / audio can be performed. This data rate agrees with the 12 bit word length and 8 kHz sampling rate of the baseband unit on-board ADC/DACs. The data was modulated to and demodulated from D-QPSK symbols in both baseband DSPs and four times oversampled. Adjustments were made in the DTP and DRP such that an IF of 5 MHz was achieved. In the receive baseband DSP, all synchronisation task were performed blindly and adaptively.

5 DISCUSSION AND CONCLUSIONS

A software radio design has been implemented based on state-of-the-art DSPs. The time critical operations of the digital front-ends were delegated to specialised processors with a fixed but programmable block structure, while the baseband functionality is fully programmable on floating point DSPs. The circuit and the application to an audio signal transmission at 96 kbps has been outlined.

The limitations of the structure are currently the use of the serial port for communication between the digital receive processor and the baseband receive DSP, as well as the speed of the baseband DSPs. These limitations manifest themselves in a restriction to approximately 256 kbit/s bandwidth for transmitted D-QPSK data. While GSM or similar standards can be well implemented within such limits, for example W-CDMA has a too high bandwidth requirement for our specific SDR realisation.

The flexibility of the presented structure has to be based on variation of D-QPSK transmission, or potentially the transmit and receive filtering, or the channel coding. Some of our current work focuses on implementing a differential 16-QAM data transmission, whereby blind synchronisation schemes have to be more elaborate as in the simpler D-QPSK case. The ultimate aim is to apply adaptive QAM, whereby the modulation level is adjusted to hostility of the channel, i.e. high level QAM schemes with a high data throughput are call on whenever the channel quality permits this.

6 ACKNOWLEDGEMENT

We would like to thank Prof. L. Hanzo for encouragement and advice, and M. Schlosser and J. Stefanov for their contribution in form of an early version of the presented testbed.

7 REFERENCES


BASEBAND ASPECTS OF A DIRECT CONVERSION RECEIVER CONCEPT UTILISING FIVE-PORT TECHNOLOGY

J.-U. Jürgensen, D. Krupezevic, M. Ratni, Z. Wang

Broadband Wireless Technology Research, SONY International (Europe) GmbH
Heinrich-Hertz-Strasse 1, D-70327 Stuttgart

ABSTRACT: Five-Port technology, recently utilised for complex RF measurements, could also be utilised to build a broadband direct conversion receiver. The concept of such a direct conversion receiver based on Five-Port technology is presented and baseband implementation aspects are discussed. The functionality of this direct conversion receiver concept is verified for a multi-carrier system based on HIPERLAN/2 parameters and a single-carrier system based on UTRA-FDD parameters.

1. INTRODUCTION

Due to the increasing demand for mobile and wireless broadband services utilising various carrier frequencies and bandwidths, transmission technologies and air interface standards, future access platforms need to support multiple system functionality. One of the key issues is the integration of broadband transceivers and adaptability and flexibility in the digital processing for multiple system support. The Five-Port technology, recently utilised for complex RF measurements, can also be utilised to build a broadband direct conversion receiver (DCR) that provides the flexibility to cope with various systems in terms of carrier frequency and bandwidth [1]. Such a DCR based on Five-Port technology can operate in the range from 1GHz to 6GHz and can support up to 20MHz bandwidth. Therefore this technology could be utilised to build a receiver that supports multiple system functionality.

This paper presents the baseband aspects of a direct conversion receiver concept utilising Five-Port technology. The paper is organised as follows: Section 2 provides an overview over the Five-Port technology and the DCR concept utilising this technology. Since the Five-Port technology makes use of power detectors instead of conventional mixers, it provides power levels instead of in phase (I) and quadrature (Q) values, means digital processing is required to derive IQ values from these power levels. Section 3 presents digital processing aspect of the DCR concept, means A/D conversion of the power levels provided by the Five-Port device, the derivation of IQ values from the power levels and system specific digital processing to support multiple systems functionality. Also the way from concept to hardware implementation is described. Implementation aspects and functionality verification for a single-carrier (SC) system based on UTRA-FDD parameters [2] and a multi-carrier system (MC) system based on HIPERLAN/2 parameters [3] are described in Section 4. A Five-Port device and the rapid prototyping module utilised for the MC/SC receiver are presented. Finally, some conclusions are drawn in Section 5.

2. DIRECT CONVERSION RECEIVER CONCEPT

The proposed DCR concept provides all processing steps to facilitate direct conversion utilising the Five-Port device. The Five-Port device is a passive linear device, constituted with two input ports and three output ports. The radio frequency (RF) signal provided by the antenna and the local oscillator (LO) signal are the two input signals, three analogue power levels $P_1$, $P_2$ and $P_3$ are the output signals. The functional blocks of the Five-Port device are depicted in Figure 1. The basic functionality of the Five-Port device consists of a sum of the received RF signals with the LO signals under various phase angels. The power levels of the combined signals are measured with three power detectors and are used to calculate IQ values [4].

![Figure 1: Functional blocks of the Five-Port device.](image-url)
to be A/D converted before Five-Port related digital processing is applied. This Five-Port related digital processing computes the IQ values from these three digitised power levels.

![Figure 2: Direct conversion receiver concept utilising Five-Port technology.](image)

The system specific digital processing then applies MC or SC system specific processing to the IQ values derived from the IQ computation. This MC/SC processing consists of filtering, synchronisation, FFT and channel estimation.

### 3. FROM CONCEPT TO IMPLEMENTATION

The functionality of this DCR concept utilising Five-Port technology is demonstrated on a FPGA based hardware prototyping platform. A transmitter and receiver for the MC and SC system are implemented. The MC system utilised HIPERLAN/2 like parameters in terms of number of carriers and bandwidth. The SC system utilised UTRA-FDD like parameters in terms of chip rate and bandwidth. The transmitter is implemented on a prototyping module consisting of a FPGA and D/A converters, the receiver is implemented on a prototyping module consisting of a FPGA and A/D converters.

The schematic of the MC/SC transmitter module is depicted in Figure 3. The MC/SC transmitter block within the FPGA generates IQ values $I_{Tx}$ and $Q_{Tx}$ that are D/A converted. The outputs of the D/A converter are upconverted to the referring carrier frequency. A carrier frequency of 2.4GHz was used for the UTRA-FDD like SC system and a carrier frequency of 5.5GHz was used for the HIPERLAN/2 like MC system. The transmitter consists of chip generation and square root raised cosine filtering. The MC transmitter consists of data/pilot generation, inverse fast Fourier transformation (IFFT), filtering and cyclic prefix insertion. The modulation scheme utilised for both SC and MC system is quaternary phase shift keying (QPSK). In addition to the MC/SC transmitter, also a feed through block is needed in the FPGA to display the received and recovered IQ values $I_{Rx}$ and $Q_{Rx}$ coming from the MC/SC receiver module on an oscilloscope. The MC/SC transmitter module is connected to the MC/SC receiver module via a bus interface.

![Figure 3: Schematic of the MC/SC transmitter module.](image)

The schematic of the MC/SC receiver module including IQ computation is depicted in Figure 4.

![Figure 4: Schematic of the MC/SC-receiver module including IQ computation.](image)

The IQ computation uses the three power levels $P_1$, $P_2$ and $P_3$ provided by the Five-Port device to derive the IQ values (1,2).

\[
I = h_{i1}P_1 + h_{i2}P_2 + h_{i3}P_3 + h_{i0}
\]

\[
Q = h_{q1}P_1 + h_{q2}P_2 + h_{q3}P_3 + h_{q0}
\]
The coefficients \( h_{i,0...3} \) and \( h_{q,0...3} \) are Five-Port device specific and depend on the carrier frequency and have to be determined using calibration procedure. A non-perfect determination of \( h_{i,1...3} \) and \( h_{q,1...3} \) results in IQ imbalance and phase rotation whereas a non-perfect determination of \( h_{i,0} \) and \( h_{q,0} \) results in direct current (DC) offset. In case of non-perfect determination of these coefficients, compensation in the digital domain is required.

The SC receiver consists of square root raised cosine filtering and chip synchronisation. The MC receiver consists of filtering, synchronisation, cyclic prefix deletion, FFT, channel estimation and equalisation. The MC/SC receiver was designed to run at 80MHz for the MC system and at 30.72MHz for the SC system.

An integrated design environment was used to get from concept to hardware implementation. This environment consists of Simulink to simulate the transmitter and receiver design and verify its functionality, Xilinx System Generator to achieve a bit-true and cycle-true simulation and Xilinx Foundation ISE to map the simulated design onto hardware. The integrated design environment is depicted in Figure 5.

![Simulink diagram](image)

**Figure 5: Integrated design environment.**

All of the functional blocks of both the MC and SC transmitter and receiver have been designed utilising Xilinx System Generator blocks. Figure 6 shows an example for design of the IQ computation block.

Two’s complementary data format is used for constants, the output of all adders and multipliers. The bit-width is 12 bits, where 1 bit is utilised for the sign and 11 bits are utilised for the fractional part. The three input ports of this block are connected to the three A/D converters of the receiver module. The two output ports are connected to system specific processing blocks.

From these designs bit files are generated that are loaded into the FPGAs to demonstrate the functionality of this DCR concept based on Five-Port technology in real-time.

### 4. Implementation Aspects

A rapid prototyping board is utilised to demonstrate the functionality of this DCR concept in real-time. This board consists of two modules. One module is utilised to implement the MC/SC transmitter, the other module is utilised to implement the MC/SC receiver including IQ computation.

The transmitter module is equipped with a Xilinx XCV1000E-8FG680 FPGA [5]. The IQ values are D/A converted by Analog Devices AD9772 D/A converters [6]. The bit width of the input signal to the D/A converter is 14 bits. The output signal from the D/A converter is up-converted to a carrier-frequency of 5.5GHz for the MC system and 2.4GHz for the SC system. A bandwidth of 20MHz was transmitted for the MC system and 5MHz for the SC system.

The receiver module is equipped with a Xilinx XCV1000E-8BG560 FPGA. The power levels provided by the Five-Port device are A/D converted by Analog Devices AD9432 A/D converters. The output signal from the A/D converter is using two’s complementary format with a bit-width of 12 bits, where 1 bit is utilised for the sign and 11 bits are utilised for the fractional part.

Figure 7 shows a Five-Port device as it was used to verify functionality of this DCR concept. The inputs are RF signals from the antenna and LO signals from the local oscillator. Three analogue power levels \( P_1 \), \( P_2 \) and \( P_3 \) are the outputs.

![Five-Port device](image)

**Figure 7: Five-Port device.**
The Five-Port device is connected to the A/D converters of the MC/SC receiver module, as shown in Figure 8. Since the power detectors of the Five-Port device are working in the linear range, non-linearity compensation is not required. The structure of IQ computation that is implemented in the FPGA is independent from the supported system, means the same algorithm is used for the MC as for the SC system. Only the coefficients have to be adjusted, since they depend on the carrier frequency of the supported system. The values after IQ computation are used to derive IQ constellation points via MC/SC digital processing.

The functionality of this DCR concept utilising Five-Port technology was verified in real-time.

5. CONCLUSION

Baseband aspects of a direct conversion receiver concept utilising Five-Port technology are presented. The concept and the way from concept to hardware implementation are described and the hardware used for this concept is presented. The functionality of this concept is verified in real-time for a single-carrier system based on UTRA-FDD parameters and a multi-carrier system based on HIPERLAN/2 parameters.

Since the Five-Port device is a broadband receiver and channel selection is simply achieved by tuning the local oscillator, channel filtering and appropriate adjacent channel attenuation have to be performed in the digital domain. This topic and also the compensation of non-linearity of the power detectors need further investigation.

It is shown that the Five-Port technology can be utilised to build a receiver that supports multiple system functionality.

6. REFERENCES

[2] Universal Mobile Telecommunications System (UMTS); UE Radio transmission and reception (FDD) (3G TS 25.101 version 3.1.0 Release 1999); ETSI TS 125 101 v3.1.0 (2000-01)
[3] Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) Layer; ETSI TS 101 475 v1.1.1 (2000-04)
SOFTWARE RADIO CONCEPT OF THE RECEIVER FOR DIGITAL BROADCASTING SERVICES

M. Sliskovic, H. Schöpp
Harman/Becker Automotive Systems GmbH
Karlsbad, Germany
e-mail: SliskovicM@becker.de, SchoeppH@becker.de

ABSTRACT

As new sound and video broadcasting systems are being deployed, the need for a multistandard multiband receiver increases. The goal of this article is to extract functional blocks that can be implemented as software subroutines with variable input parameters and shared by the receivers for different broadcasting services. The advantages of the software radio approach are larger flexibility, which enables adaptation to the eventual change in the standard, the reduction of the overall system costs and the receiver physical size, and optimization of the program memory and the development time. Practical issues like computing complexity and performance are also addressed.

1. INTRODUCTION

In the last decade a number of digital sound and video broadcasting systems has been developed and standardized. They include Digital Audio Broadcasting (DAB), [1], Digital Radio Mondiale (DRM), [2], and Digital Video Broadcasting (DVB), [3]. All of these systems are based on the Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme, and for error correction a convolutional code or a concatenated convolutional and block code are being used. In the USA, the process of standardization of the In-Band On-Channel (IBOC) system is under way, and in year 2001 two privately owned satellite digital radio services were introduced – Sirius Satellite Radio and XM Satellite Radio [4]. The IBOC system is, like the above mentioned European systems, also based on the OFDM modulation scheme, and both Sirius and XM broadcasting systems use OFDM in their terrestrial repeater networks.

Due to the similarity in the physical layers of the above mentioned broadcasting services, the advantages of the software radio concept in comparison to the implementation with the dedicated chipsets are obvious. The software radio implementation is more flexible, and the overall system costs and physical size of the receiver can be reduced.

However, due to the limited computational performance of today available Digital Signal Processors (DSPs), the advantages of the parametrically defined software radio cannot be fully utilized. The signal processing algorithms have to be implemented in the most optimal way, and the optimal implementation of some algorithms depends on the system parameters. The example of such an algorithm is the Discrete Fourier Transform (DFT), which is a part of an OFDM receiver. DFT can be implemented most efficiently if the number of input signal samples is an integer power of 2 ($2^k$, $k \in \mathbb{N}$), [5]. However, out of four foreseen transmission modes in the DRM system, the number of sub-carriers (and therefore the number of input samples to the DFT algorithm) is an integer power of 2 in only one mode. For the other three transmission modes DFT has to be implemented in separate software subroutines. The paper is organized as follows. In Section 2, an overview and comparison of digital sound and video broadcasting systems will be given. Section 3 will give a short introduction to the OFDM modulation, and extract functional blocks which can be parametrically defined. The comparison and possibilities of parameterized implementation of error correcting codes will be investigated in Section 4. Section 5 will discuss the possibilities for parameterization of the frame structure, and in Section 6 conclusions are drawn.

2. DIGITAL BROADCASTING SYSTEMS OVERVIEW

The basic parameters of the digital broadcasting systems are listed in Table 1. For systems with variable bandwidth only the lower and upper bandwidth limits are given.

Most of the systems are based on the OFDM modulation scheme, which opens up a possibility to implement some receiver elements for different broadcasting systems in a single software routine with variable input parameters. A typical OFDM receiver will be presented in Section 3 and the elements that can be parametrically defined will be extracted.

Sirius and XM broadcasting systems use single-carrier modulation scheme for the satellite downlink. The modulation type (Quadrature Phase Shift Keying - QPSK) is the same both for Sirius and XM, but the bandwidth and therefore the symbol rate are different. However, the receiving filter, timing recovery unit and the equalizer can be parametrically defined.

The radio set should also be able to receive analog amplitude-modulated (AM) and frequency-modulated (FM) signals. The availability of high-speed DSP and ADC can be utilized here, too. Sampling at Intermediate Frequency (IF) level will enable to distinguish adjacent channel interference from multipath propagation, and to process the signal accordingly. The algorithms for single-carrier and analog modulation systems are not topic of this article and will not be discussed here.
Digital broadcasting system | Modulation | Error correction coding | Source coding scheme | Bandwidth
---|---|---|---|---
DAB | OFDM/π/4-DQPSK | Punctured convolutional code | MPEG audio layer 2 | 1.536 MHz
DRM | OFDM/QAM, OFDM/QPSK | Punctured convolutional code | AAC, MPEG CELP, MPEG HVXC, SBR | 4.5 – 20 kHz
DVB-T | OFDM/QAM | Reed-Solomon outer code and punctured convolutional inner code | MPEG 2 | 6 – 8 MHz
IBOC FM – hybrid mode | Combination – OFDM and analog FM | Punctured convolutional code | PAC | 400 kHz
IBOC FM – all digital mode | OFDM | | PAC | 20 – 40 kHz
IBOC AM – hybrid mode | Combination – OFDM and analog AM | Punctured convolutional code | PAC | 4.0 MHz
IBOC AM – all digital mode | OFDM | Reed-Solomon outer code and convolutional inner code | PAC | 2 × 4.2 MHz
Sirius – terrestrial network | OFDM | | PAC | 5.1 MHz
Sirius – satellite | QPSK | | PAC | 2 × 3.7 MHz
XM – terrestrial network | OFDM | Reed-Solomon outer code and convolutional inner code | PAC | 2 × 3.7 MHz
XM - satellite | QPSK | | PAC | 2 × 3.7 MHz

Table 1: System parameters of the digital broadcasting services

3. OFDM RECEIVER

OFDM is a multicarrier modulation scheme, in which sub-carriers are mutually orthogonal. In its most popular and in practice mainly used form, the sub-carriers are sine waves whose frequency is an integer multiple of the symbol rate when the guard interval length is equal to zero [6]. This form is used for all listed broadcasting services. Its advantages are insensitivity to multipath propagation (if the guard interval is long enough), and the possibility of computationally efficient receiver realization with the Fast Fourier Transform (FFT) algorithm [7], [5].

The block diagram of an OFDM receiver is illustrated in Fig. 1. The down-conversion process is not depicted, since it is the same as for any other modulation scheme.

**Frequency offset correction.** On the sampled input signal, frequency offset correction is performed. This functional block includes both carrier and sampling frequency offset correction. Carrier frequency offset correction can be performed by feeding the input signal to a Finite Impulse Response (FIR) filter with complex coefficients. The filter has to be adaptive, and the coefficient values depend only on the estimated frequency offset and on the symbol duration. These are the only two input parameters necessary to correct a carrier frequency offset, and therefore this function can be realized in a single software routine. The situation is similar with the sampling frequency offset correction. If the Farrow polynomial interpolator [8] is used, the only input parameter needed is the estimate of the sampling frequency offset. Alternatively, small sampling frequency offset correction can be compensated for by adapting coefficients of the DFT block [9].

**Frequency synchronization.** For carrier and sampling frequency offset estimation (block ‘frequency synchronization’ in Fig. 1), reference symbols are needed. To design an optimal estimation algorithm, their relationships and positions in the time - frequency lattice have to be utilized. Since the considered broadcasting systems use different reference cell positions, the complete parameterization of the considered functional block is not advisable.

Fig.1: OFDM receiver – block diagram
Symbol timing recovery. Symbol timing recovery based on the guard interval correlation provides fast and frequency insensitive timing estimation with the periodicity of one symbol. The required input parameters are guard interval length, total symbol length and signal-to-noise ratio (SNR) at the input to the estimation block.

Cyclic prefix removal. Cyclic prefix removal needs a timing information provided by the symbol timing recovery block, and the cyclic prefix length (in the number of samples).

Serial-to-parallel converter & DFT. The only input parameter necessary to describe serial-to-parallel converter (S/P in Fig. 1) and the DFT block is the sample block length, i.e. the useful symbol length (in the number of samples). S/P converter is usually realized as a buffer with variable length, and its optimal realization does not depend on its length. However, the situation with the DFT block is different. The only algorithm that supports any input block-length is the direct DFT calculation. Computational complexity of this method is proportional to $M^2$, where $M$ is the number of input samples. Fast Fourier Transform (FFT) algorithms enable reduction of the computational complexity if $M$ is a composite number. The algorithm is particularly simple and has a regular structure if $M$ is a power of an integer ($M = r^k; r, k \in \mathbb{N}$). The most popular FFT algorithm is the Cooley-Tukey FFT algorithm [5], in which $M$ is an integer power of 2 ($M = 2^k, k \in \mathbb{N}$). However, as already mentioned, for the considered broadcasting services the input block length is not always an integer power of 2. Therefore, the proposed realization for the DFT block is:

- if $M = 2^k, k \in \mathbb{N}$, use Cooley-Tukey FFT algorithm;
- otherwise, use a separate software routine computationally optimized for a given number of samples $M$.

Channel estimation and equalization. Modulation can be classified as coherent or differential. When a differential modulation is used, there is no need for a channel estimator or equalizer. The advantage is the reduced receiver complexity, and the drawback is about 3 dB noise enhancement. Differential modulation is used in DAB standard [1]. On the other hand, coherent modulation schemes (like Quadrature Amplitude Modulation - QAM, used in e.g. [2], [3]) do require channel estimation and equalization. If the cyclic prefix is long enough, all transitional effects will disappear inside the guard interval, and equalization can be realized by one complex multiplication per sub-channel [6]. Therefore, the only parameter needed to define the ‘one-tap equalizer’ block is the number of the information-bearing sub-carriers. Channel estimator needs a pilot information as a point of reference. In the considered broadcasting systems the pilot information is transmitted regularly, at predefined locations of the time-frequency lattice. High data rates and desired low bit-error rate require the use of estimators that have both low complexity and high accuracy. These two requirements are contradictory, and some compromise between them has to be made. The compromise has to be made in choosing the input parameters to the channel estimator, too. Positions of the pilot cells in the time-frequency lattice are different for different broadcasting systems, and the reduced number of parameters may lead to less accurate channel estimation.

Symbol demapping. Symbol demapper is used to demap the received frequency-domain symbol, i.e. to find the constellation point that corresponds to the recovered frequency domain sample. The constellation size and bit pattern vary inside the standard, and, of course, among the standards. This block has to be parametrically defined, and the input parameter is a particular constellation.

Parallel-to-serial converter. Similarly as for S/P converter, the only parameter needed to define a parallel-to-serial converter (P/S) is the block length, i.e. the number of sub-carriers.

Deinterleaver. Different interleaving algorithms require different deinterleaving matrices. The most adaptable deinterleaver realization is the table look-up method. However, large deinterleaving matrices occupy a lot of memory. Therefore, the most optimal solution is to make a subroutine that calculates the deinterleaving pattern only once, at the communication start-up, and stores the calculated values into a table. For algorithmically simple interleaving methods, like e.g. convolutional interleaving in DVB-T standard [3], the table look-up method may require more instruction cycles than if the algorithm were implemented in a separate software routine. In a standard DSP, the add/subtract pointer update is performed in parallel with the data fetch, therefore requiring only one instruction cycle per interleaving unit (bit or word) to perform deinterleaving.

The implementation of the channel decoder will be discussed in the next section. It is depicted in Fig. 1 because sometimes the required accuracy cannot be achieved by strict separation of functional blocks. A soft-decision decoder [10] needs an information about the reliability of the received symbol, which is provided by the demapper-P/S-deinterleaver chain and the channel estimator.

4. CHANNEL DECODER

Channel coding algorithms used by the considered broadcasting systems include convolutional codes of different constraint length, coding rate and encoding polynomials, and different types of Reed-Solomon codes. Convolutional codes are punctured by different patterns that can vary even inside a frame [1].


Therefore, the puncturing pattern has to be an input parameter to the channel decoder.

4.1. Viterbi decoder

To decode a convolutional code, Viterbi algorithm is used, Fig. 2, [10]. An important part of the Viterbi decoder is branch metric calculation unit. The output of the branch metric calculation unit is a measure of the reliability that a particular state transition really occurred at the encoder side. The branch metric depends on the encoding polynomials, number of input bits per stage, code constraint length, code rate and puncturing pattern. These parameters completely define branch metric calculation unit, and the only additional parameter needed to define the complete Viterbi decoder is the input sequence length.

A path metric is associated with each encoder state, and represents the accumulated sum of branch metrics leading from the start state to a particular encoder state. The next step in the decoding process is to select an encoder state sequence that most likely generated the encoder input, i.e. to select the path with the largest path metric. Lastly, to obtain the decoder output, the selected path is traced back to its beginning and the bit that led to a particular state transition is fed to the decoder output.

4.2. Reed-Solomon decoder

Reed-Solomon (RS) code is a block code, and coding and decoding algorithms are based on finite field computations [11]. RS code is defined by generating polynomials, but the decoder can be implemented without knowing them. The parameters required to calculate syndromes, locate error positions and to correct the detected errors are the number of errors that can be corrected and the radix of the roots of the generating polynomials. Any additional requirements, like e.g. code shortening, will represent an additional parameter.

5. FRAME PARAMETERIZATION

Functions that have to be performed at the receiver to make the service comprehensible, but cannot be classified as a part of the demodulator or channel decoder, are discussed in this section. They include frame, service and access management functions, and source decoding.

Source decoding. Source coding schemes applied in the broadcasting services are listed in Table 1. The same source coding method is used in IBOC, Sirius and XM system – Perceptual Audio Coding (PAC), and the receivers for these systems will call the same source decoding subroutine. DAB and DVB-T use the same audio coding scheme ([12], part 3), and for DRM, due to the small channel bandwidth, a set of speech and audio coding schemes is foreseen.

Frame delimiting. For frame synchronization, pilot cells should be used. Their amplitude, duration and position in the time-frequency lattice is different for each of the considered broadcasting systems, and to optimally exploit their properties and relationships, frame synchronization routines should be implemented separately.

Frame length. Frame length is different for each of the considered broadcasting systems, and in some systems it changes with the transmission mode. It can be defined by a single parameter - the number of OFDM symbols in a frame.

Transmission parameter signaling. Transmission parameter signaling (TPS) is used to inform the receiver about the transmission related signaling parameters, i.e. channel coding, modulation and spectral occupancy. The TPS information is transmitted with predefined parameters, and with very robust modulation scheme. The position of TPS sub-carriers in the time-frequency lattice, modulation scheme, error protection scheme and type of information carried by the TPS depend on the broadcasting standard. The TPS sub-carrier modulation and/or error protection scheme is, in some cases, equal to one of the possible modulation schemes for the sound/data transmission, and therefore can be parametrically defined.

Multiplex configuration. Some broadcasting systems are designed to carry a few audio channels, and even those designed for only one audio channel usually contain some digital data that has to be separated from audio/video data before feeding the demodulated data sequence to the source decoder. The information necessary to demultiplex audio and data services is called multiplex configuration information and, in some systems, it may be reconfigured on the fly. Therefore, demultiplexing at the receiver has to be defined parametrically. The parameters are start position and length (in number of bytes) of the block containing a
service chosen by the user, or time-frequency pattern of the OFDM cells containing a particular information.

**Conditional data access.** Conditional access is used to make a service incomprehensible to unauthorized users. It is supported only in some of the considered broadcasting systems. The entitlement checking function and descrambling procedure vary from one service to another, and should be implemented separately.

6. CONCLUSIONS

Most of the considered sound and video broadcasting systems are based on the OFDM modulation scheme, which opens up a possibility to implement some receiver functions in a single software routine with variable input parameters. These functions include carrier and sampling frequency offset correction, symbol timing estimation, cyclic prefix removal, serial-to-parallel and parallel-to-serial conversion, frequency domain equalization, symbol demapping, deinterleaving and channel decoding. The other necessary elements, like DFT calculation, frequency offset estimation and channel estimation, theoretically might be parameterized, but the complete parameterization would lead to the increased computational complexity (DFT) or to the reduced accuracy (estimation blocks). Channel decoder may be parametrically defined, too. The parameterized realization needs more memory and is computationally more complex than the realization of a dedicated decoding algorithm, but the overhead is not too large.

The receiver has to be able to demodulate single-carrier and analog modulation schemes, too. The functional blocks of the OFDM receiver that can be utilized for the single-carrier case include carrier and sampling frequency offset correction block, symbol demapper, deinterleaver, and channel decoder. Analog demodulators have to be implemented in separate software routines.

**REFERENCES**


[12] ISO/IEC 13818. Information technology - Generic coding of moving pictures and associated audio information - Parts 1 (Systems), 2 (Video) and 3 (Audio).
Some Estimated Limits for a Blind Identification of the telecommunication standard in use with RBF Neural Networks.

C. Roland - J. Palicot*
*IRISA
Campus Universitaire de Beaulieu
35042 Rennes–France
Jacques.palicot@irisa.fr

Abstract : In this paper we try to identify some estimated limits for blind shape recognition with RBF NN. We already defined a new receiver concept : the Self Adaptive Universal Receiver (SAUR) which uses Radial Basis Function Neural Network (RBF NN) for blind recognition of the channel Bandwidth [1], [2]. Doing that we identified perfectly the standard in use during the transmission. At that period we made pragmatic choices concerning some parameters, like the error function of the neuron as well as its good detection threshold. In this paper we confirm “a posteriori” with more solid theoretical results the validity of these choices. In addition, these results permit to give some limits in terms of Good Detection Rate versus the frequency offset (section 3.2), versus the Equivalent Noise Degradation (END) (section 3.3), versus the bandwidth compression or dilatation (section 3.3), versus the frequency offset (section 3.2), versus the frequency offset (section 3.2).

1 INTRODUCTION

Wireless communication has regained considerable interest over the last few years and nowadays is one of the fastest growing sectors of the telecommunication industry. The step forward not only expands the market for wireless communications but also creates opportunities for newer products.

The number of services on heterogeneous wireless networks (GSM, IS95, PDC, DECT, PHS and future 3G standards like the UMTS proposal in Europe) is dramatically increasing. In European countries there is an explosion in the number of consumers for mobile communications. Moreover RLAN and Hiperlan also contribute to an increase in the number of wireless services. In the broadcasting area, DAB and DVB-T are additional services. Among all the key-words behind these services the “mobility” key word is one of the most important [4]. The main drawback of this explosion of services is that consumers need more and more terminals.

Furthermore, the down-compatibility with pre-existing 2G and 2.5G networks should be insured. In addition, the interconnection (“convergence” concept, may be the second key word) among all the networks increases the complexity of the user’s environment. Consequently there is a current and growing interest in universal terminals (multi-services, multi-networks) [3]. The technical approach for these universal terminals consists in developing re-configurable terminals [5]. It seems clear that universal terminals will be the solution in a near future, and also that these terminals will include more and more “intelligence” to be user independent and network independent. In this context, we proposed in [1] a new vision for an universal terminal. It will be a self-adaptive terminal, in the sense that it will recognize in a “blind” manner the transmission standard in use and consequently it will re-configure all its architecture with the adequate software. This reconfigurable receiver will become very common in a near future, but in addition this reconfiguration will be performed in a blind manner, which is the new idea. This proposal is one possible solution for the physical layer evolution towards “cognitive” radio. As said by J. Mitola in [6], “this type of learning technique makes the Software Radio trainable in a broad sense instead of just re-configurable”.

This paper comprises two main parts. In the first one we recall briefly our proposal : the Self-Adaptive Universal Receiver (SAUR). This part is a functional description (section 2), based on a new front-end architecture that includes two functional phases, the first one being called the “Wide Band Analysis” (WBA). In the second part, we will discuss the blind recognition itself. In this second part we will present some theoretical results obtained with the RBF Neural Network : performances versus the frequency offset (section 3.2), versus the bandwidth compression or dilatation (section 3.3), versus the Equivalent Noise Degradation (END) (section 3.4). Then we will conclude this paper in verifying that these theoretical results are in accordance with those we obtain pragmatically and presented in [2].

1 GSM : Global System for Mobile Communications
2 IS95 : Interim Standard 95 (CDMA)
3 PDC : Personal Digital Cellular
4 DECT : Digital Enhanced Cordless Telecommunications
5 PHS : Personal Handy phone System
6 UMTS : Universal Mobile Telecommunication system
7 LAN : Local Area Network
8 Hiperlan : High Performance Radio Local Area Network
9 DAB : Digital Audio Broadcasting
10 DVB-T : Digital Video Broadcasting-Terrestrial
2 Our proposal for a Self-Adaptive Receiver architecture

2.1 Generalities and architecture

The solution proposed here in has two functional phases. We have called the first one “Wide Band Analysis” because it consists in the study of the complete multistandard received signal. It is represented by equation (1). This signal is the result of the summation of many standards, each standard being itself the summation of several modulated carriers (channels). The analysis of this signal should give the important information that is useful for the receiver. It could, for example, be the modulated carrier existence, the type and the position (in the frequency domain) of the standard(s) in use in the analyzed band. Concerning the WBA phase, the generic Software Radio Architecture (SWR) is well adapted. In fact, for the demodulation phase, the ADC requirements are the same as the conventional SWR problem and could not be fulfilled currently. That was the reason for our two-path architecture proposal, which is described below.

Two years ago we proposed [7] an architecture that was well-adapted for this new front-end receiver. That new receiver architecture was called "two-path architecture". It is a mix of ideal SWR and SDR techniques. In any case, the WBA phase is performed through the ideal SWR architecture path. But for the demodulation phase this behavior is possible only in some rare cases as IS95 or DECT, but in general, this is not possible and we have to reduce the signal band at the ADC input. This could be done by one of the numerous methods of Software Defined Radio. In this paper we will not describe these techniques, since, they are, for the best known described in [8], [9] and in [10].

Several variants of the “two-path architecture”, depending mainly of the number of ADC used, are possible. We present below this architecture with 2 ADCs (see figure 2).

Concerning the WBA phase, the generic Software Radio Architecture (SWR) is well adapted. In fact, under the assumption that we have sufficient time to perform powerful signal processing techniques, it is totally clear that with existing ADC, even with a low number of bits, we will reach our objective.

Unfortunately, it is not the same “story” for the demodulation phase. As is well-known, this architecture could not be implemented with today’s technology. In fact, for the demodulation phase, the ADC requirements are the same as the conventional SWR problem and could not be fulfilled currently. That was the reason for our two-path architecture proposal, which is described below.

Two years ago we proposed [7] an architecture that was well-adapted for this new front-end receiver. That new receiver architecture was called "two-path architecture". It is a mix of ideal SWR and SDR techniques. In any case, the WBA phase is performed through the ideal SWR architecture path. But for the demodulation phase this behavior is possible only in some rare cases as IS95 or DECT, but in general, this is not possible and we have to reduce the signal band at the ADC input. This could be done by one of the numerous methods of Software Defined Radio. In this paper we will not describe these techniques, since, they are, for the best known described in [8], [9] and in [10].

Several variants of the “two-path architecture”, depending mainly of the number of ADC used, are possible. We present below this architecture with 2 ADCs (see figure 2).

Concerning the WBA phase, the generic Software Radio Architecture (SWR) is well adapted. In fact, under the assumption that we have sufficient time to perform powerful signal processing techniques, it is totally clear that with existing ADC, even with a low number of bits, we will reach our objective.

Unfortunately, it is not the same “story” for the demodulation phase. As is well-known, this architecture could not be implemented with today’s technology. In fact, for the demodulation phase, the ADC requirements are the same as the conventional SWR problem and could not be fulfilled currently. That was the reason for our two-path architecture proposal, which is described below.

Two years ago we proposed [7] an architecture that was well-adapted for this new front-end receiver. That new receiver architecture was called "two-path architecture". It is a mix of ideal SWR and SDR techniques. In any case, the WBA phase is performed through the ideal SWR architecture path. But for the demodulation phase this behavior is possible only in some rare cases as IS95 or DECT, but in general, this is not possible and we have to reduce the signal band at the ADC input. This could be done by one of the numerous methods of Software Defined Radio. In this paper we will not describe these techniques, since, they are, for the best known described in [8], [9] and in [10].

Several variants of the “two-path architecture”, depending mainly of the number of ADC used, are possible. We present below this architecture with 2 ADCs (see figure 2).

Concerning the WBA phase, the generic Software Radio Architecture (SWR) is well adapted. In fact, under the assumption that we have sufficient time to perform powerful signal processing techniques, it is totally clear that with existing ADC, even with a low number of bits, we will reach our objective.

Unfortunately, it is not the same “story” for the demodulation phase. As is well-known, this architecture could not be implemented with today’s technology. In fact, for the demodulation phase, the ADC requirements are the same as the conventional SWR problem and could not be fulfilled currently. That was the reason for our two-path architecture proposal, which is described below.

Two years ago we proposed [7] an architecture that was well-adapted for this new front-end receiver. That new receiver architecture was called "two-path architecture". It is a mix of ideal SWR and SDR techniques. In any case, the WBA phase is performed through the ideal SWR architecture path. But for the demodulation phase this behavior is possible only in some rare cases as IS95 or DECT, but in general, this is not possible and we have to reduce the signal band at the ADC input. This could be done by one of the numerous methods of Software Defined Radio. In this paper we will not describe these techniques, since, they are, for the best known described in [8], [9] and in [10].

Several variants of the “two-path architecture”, depending mainly of the number of ADC used, are possible. We present below this architecture with 2 ADCs (see figure 2).
No known Spectral Analysis techniques could be performed in these conditions, because the resolution ratio is too high. Consequently, we proposed an iterative adaptation of the Wide Band to be analyzed. Figure 4 presents this process. At each iteration, we research energy in the band with a conventional periodogram, then we filter and decimate the samples around this detected peak of energy. This process is not detailed here, we only recall briefly that, during the last iteration of the process, for recognizing the $BW_c$ we proposed to use Radial Basis Function Neural Networks. In fact, the channel bandwidth shape (pattern) depends on several parameters, like the modulation, the channel coding and the channel filtering.

This specificity will be used in our method for the recognition of the parameter. In fact we will not only measure the $BW_c$ value, which could easily be done with a threshold but we will also use all the information given by the shape of the spectrum in the bandwidth. In our approach to recognizing a $BW_c$, it is exactly like a “pattern recognition”. It is as though we were doing a recognition of several parameters in the same operation. In figure 5 our recognition method is summarized. The received signal is shared in such a way to be compared with reference signals in the NN, then the output of the neuron gives directly which standard is recognized.
2.2.2 Blind recognition

The method we proposed for identifying the $BW_c$ parameter is fully summarized in the figure 5. As it could be easily seen, the RBF NN compares a received signal (a real spectrum given by equation 5) with a reference signal (a reference spectrum given by equation 11). The neuron which is activated gives directly the standard.

Obviously, the overall process is a little more complicated, it has been fully described in [3]. To well understand the rest of the paper we will recall briefly the two mains functions of this process. It comprises firstly a pre-treatment of the received signal before the error function computation. This last one being the second function we will present.

### 2.2.2.1 The Power Spectrum Density (PSD) of the received signal

If $x(t)$ is the received signal comprises $S$ standards with $P_S$ modulated carriers for each standard then we can write $x(t)$ as presented in equation (1). We voluntarily do not take into account the channel coding aspect in this formula, under the assumption that the codes have only a slight influence on the bandwidth shape.

$$x(t) = \sum_{i=1}^{S} \left( \text{fem}_i(t) \cdot m_{p,i}(c(t)) \right) \exp(2\pi j f_p t)$$  \hspace{1cm} (1)

In this equation $m_{p,i}(t)$ is the modulation of carrier $p$, fem$_i$(t) its shape filter and c(t) its mapping. For the general case, with a monochannel modulation type, the signal at the ADC input could be written as :

$$x(t) = \sum_{i=1}^{S} \sum_{k=1}^{N_i} h_p(t) \left( \text{fem}_i(t) \cdot m_{p,i}(c(t)) \right) \exp(2\pi j f_p t) + b(t)$$  \hspace{1cm} (2)

With $h_p(t)$ the channel impulse response at frequency $f_p$, and $b(t)$ the noise at the ADC input.

Then sampling $x(t)$ at sample frequency $f_s$, we obtain :

$$x(kT) = \sum_{i=1}^{S} \sum_{k=1}^{N_i} h_p(kT) \left( \text{fem}_i(kT) \cdot m_{p,i}(c(kT)) \right) \exp(2\pi j f_p f_s) + b(kT)$$  \hspace{1cm} (3)

After quantization:

$$x(kT) = \sum_{i=1}^{S} \sum_{k=1}^{N_i} b(kT) \left( \text{fem}_i(kT) \cdot m_{p,i}(c(kT)) \right) \exp(2\pi j f_p f_s) + b(kT)$$  \hspace{1cm} (4)

In this equation the total noise $b_T$ is the summation of the input noise with the quantization noise. The Power Spectral Density of the equation (4) becomes :

$$\gamma(k) = \frac{\sum_{i=1}^{S} \sum_{k=1}^{N_i} \text{fem}_i(kT) \cdot m_{p,i}(c(kT)) |M_{p,i}(\frac{k}{L}f_s)|^2}{\sum_{i=1}^{S} \sum_{k=1}^{N_i} B_{j,i}(k) \exp(2\pi j f_p f_s)}$$  \hspace{1cm} (5)

### 2.2.2.2 The pre-treatment function

Figure 6 presents the pretreatment which could itself be shared in three different main functions (sharing the analyzed bandwidth in cuts function, suppression of some cuts, normalization of these cuts).

Figure 6 : Pretreatment functions

The sharing function

Because each reference signal is obtained with a PSD of different length, it is necessary to cut the analyzed PSD signal in many cuts of the same length in order to compare these cuts with the reference signal. The number of cuts $N_i$ for one reference is given by the ratio $\frac{M}{L_i}$, where $M$ is equal to $L_F f_s / 2$ (half the length of the PSD) and $L_i$ the number of points of the $i^{th}$ PSD of the reference $C_{j,i}$.

Suppression of cuts

We compute the minimum, maximum and mean powers of all the $N_i$ cuts. For each cut, if the ratio between the mean power and the maximum power is less than a predefined threshold, then this cut is not used. This is the case when the following equation is verified :

$$10 \log \left( \frac{1}{N_i} \sum_{k=1}^{N_i} \gamma(k) \right) < 5 \text{ dB}$$  \hspace{1cm} (6)

this operation is twofold. Firstly, it decreases the number of comparisons and then decreases notably the computation burden; secondly, it also decreases the number of "false detections". In fact after AGC (see following function), the noise could, in certain cases, be recognized as true signals.

The normalization function

We have to perform a normalization function (like an AGC) in order to have the same mean power in the analyzed signal and in the reference signal. Then it is possible to perform a normalized error function computation. Then the modified spectrum becomes:
\[ y_{\text{mod}} = y_i \left( \frac{1}{L} \sum_{l=1}^{L} C_l \right) \left( \frac{1}{L} \sum_{l=1}^{L} y_i \right) \]  

(7)

### 2.2.2.3 The error function of the \( i \)th neuron

We have compared three different error functions. First, two conventional Mean Square Error functions (11) on the differences between the two PSD on both on the linear (equation (8)) and log scales (equation (9)).

In these equations \( C_l \) are the points of the \( i \)th reference signal (equation (11)) and \( y_i \) the points of the PSD of the received signal (equation (5)) modified by the pre-treatment function.

\[ \text{MSE lin} = \frac{1}{L} \sum_{i=1}^{L} (y_i - C_l)^2 \]  

(8)

\[ \text{MSE log} = \frac{1}{L} \sum_{i=1}^{L} \left( \log y_i - \log C_l \right)^2 \]  

(9)

The third error function is given by equation (10). It corresponds to a combination of the two previous functions. It is with this last one that we obtain the best results, particularly on GSM signals.

\[ \text{MSE Comb} = \frac{1}{L} \sum_{i=1}^{L} \left[ (y_i - C_l)^2 + \left( \log y_i - \log C_l \right)^2 \right] \]  

(10)

This error function has been designed in order to benefit the noise performance of the MSE lin and the rupture detection of the MSE log.

**The reference signals:**

The reference spectrum signals are given by the following equation:

\[ \tilde{y}_{r}(k) = C_s(k) = \left| \text{Fem}_{s}(\frac{k}{f_c}) \right|^2 \]  

(11)

This corresponds to the product of the modulation PSD by the transmitter filter modulus. The modulation PSD is given by:

\[ y_{\text{mod}}(\frac{k}{f_c}) = \sum_{n=1}^{N} M_s, pu(\frac{k}{f_c}) \]  

(12)

It can be seen that equation (11) is for one standard \((S=1)\) and for one carrier \((P_c=1)\), as well as for a perfect channel equivalent to equation (5). This is exactly the equivalence we would like to recognize. In Figure 5, 2 examples of reference signal are presented (GSM for the GMSK modulations and LMDS for the QAM Nyquist filtering modulations).

### 3 Estimated limits

The rest of this paper is devoted to the “theoretical” analysis of this method. For that purpose the method we choose is presented in figure 7. It consists in setting the input signal by the reference signal itself with or without added perturbation.

**Figure 7 : The method to obtain the limits**

In order to avoid too many results and to avoid confusion with a great number of results, we voluntarily limit this study to the recognition of GSM signals, therefore to the GSM neuron.

#### 3.1 Error matrix

In this section, in order to have upper limits for the thresholds, we have computed the error obtained between one reference signal for the neuron and the other reference signals as stimuli of this neuron. We obtain, therefore, an error confusion matrix.

A column of the table (2) gives, for the combined error function, the errors of one neuron excited by all the others signals. In the same way, a line gives the errors of different neurons excited by the same signal.

The spectrum being classified in the table by ascending order of the \( BWc \), both in line and column, the smallest error values are located around the diagonal.

These values give us a good indication of the threshold level so as to obtain the desired discrimination. In fact, this smallest value corresponds to the upper limit of the threshold of the considered neuron. For the GSM neuron this value is 0.021.

Through out this theoretical limits study, we will analyze the evolution of the threshold limits (upper and lower) value for the GSM neuron. The evolution shape is sketched in figure 8. That means that to increase the good detection rate the threshold should increase from zero to the optimal value \( T_{opt} \). In fact we would recognize GSM signals which are disturbed. On the

<table>
<thead>
<tr>
<th>Combined error</th>
<th>Reference spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT2</td>
<td>GSM</td>
</tr>
<tr>
<td>CT2</td>
<td>0</td>
</tr>
<tr>
<td>GSM</td>
<td>0.013826</td>
</tr>
<tr>
<td>PHS</td>
<td>0.084595</td>
</tr>
<tr>
<td>DECT</td>
<td>0.109655</td>
</tr>
<tr>
<td>IS95</td>
<td>0.117735</td>
</tr>
<tr>
<td>DAB</td>
<td>0.117735</td>
</tr>
<tr>
<td>UMTS</td>
<td>0.117735</td>
</tr>
<tr>
<td>DVB</td>
<td>0.117735</td>
</tr>
<tr>
<td>LMDS</td>
<td>0.117735</td>
</tr>
<tr>
<td>RLAN</td>
<td>0.117735</td>
</tr>
</tbody>
</table>

Table 2 : Error confusion matrix with “combined error” function
other hand this same threshold should decrease from the upper limit (0.021 for GSM example) to the optimal value \( T_{\text{opt}} \) in order to decrease the false detection rate. It is clear that this optimal threshold will be a compromise between these two constraints. The aim is to maximize the distance \( \Delta \) between the Good Detection Rate (GDR) and the False Detection Rate (FDR).

\[ \Delta \text{Err}_{\text{sync}} = f(\gamma(l) - C(l + \epsilon)) \]  

This error depends of course on the shape of the reference and also on the content of the co-channel band. This theoretical de-synchronization is obtained by an offset of the reference signal with itself. The offset is equal to zero until the de-synchronization is complete.

\[ \text{Error} = \text{Combined} \]

\[ \text{Upper limit} \]

\[ \text{Normalized error} \]

\[ \text{Combined} \]

\[ \text{MSE} \]

\[ \text{Upper limit} \]

The error due to a frequency de-synchronization corresponds to an offset of \( \epsilon \) points of the PSD. If \( L_c \) is the number of reference signal points, then the total error for this problem is given by:

<table>
<thead>
<tr>
<th>Figure 8</th>
<th>Optimal threshold</th>
</tr>
</thead>
</table>

| Figure 9 | Synchronisation procedure |

| Figure 10 | Theoretical error for the 3 functions of the GSM neuron versus the desynchronisation value of the reference GSM signal as the stimulus |

| Figure 11 | Theoretical error for the MSE and Combined error functions of the GSM neuron with the upper limit normalized to 1 |

Figure 11 presents the same type of result with a view of the area of interest. In this figure, the upper limit is normalized to one for both error functions. We confirm with this figure that the combined error function is the best for GSM signals. We can conclude than with a 5% of desynchronization the additive error is very low: only 0.004 as it could be seen in figure 10. For this 5% desynchronization value the neuronal computation complexity is multiplied by 20. That means we need 20 neurons in parallel to deal with this desynchronization. This is the value we have chosen.
At that level we can conclude that the optimal threshold is given by the following equation:

\[0.004 < T_{opt} < 0.021\]

### 3.3 Versus the bandwidth compression or dilatation

We are now interested in the discrimination power of the neuron. In others terms, which minimum bandwidth (with the same shape) value could be discriminate versus the neuron threshold. The stimulus of the neuron is its reference multiplied by a factor between 0.5 and 2. In figure 12 it can be seen that if we would not have false detection for stimuli which have a bandwidth comprised between 125 and 360 kHz, the reference being 200 kHz, the upper limit of the threshold should be 0.015. Fortunately, the closest \(BW_c\) is for the CT2 standard with 100 kHz.

Now we can conclude that the optimal threshold is given by the following equation:

\[0.004 < T_{opt} < 0.015\]

### 3.4 Versus the Equivalent Noise Degradation

**Definition:** the Equivalent Noise Degradation (END) is an additive noise which corresponds to the total impairment on the received signal.

This impairment could comprise Interference Inter Symbol, noise, desynchronization... END is added directly on the PSD reference signal in the frequency domain. In figure 13 if we would recognize a GSM signal which is disturbed by an END of -4.78 dB, the lower threshold limit should increase to 0.011.

At that level we conclude that:

\[0.011 < T_{opt} < 0.015\]

### 3.5 Verification with “live results”

The conclusion of the previous sub-section said that the threshold should be comprise between 0.011 and 0.015. In this section we confirm with live results this value. In figure 14 we plot the GDR versus the neuron threshold for thousands of digitized received signals in our lab. It can be seen that the optimal threshold which maximizes the distance between GDR and FDR is equal to 0.0125. It is fully in accordance with the conclusion of the previous sub-section.

**Figure 12:** theoretical combined error of the GSM neuron versus the compression value of the reference signals as the stimulus.

**Figure 13:** Combined error versus END

**Figure 14:** “Live” result : Determination of the threshold for the GSM neuron

**Figure 15:** Verification between “theoretical” and “live” results
Figure 15 presents a combination of figure 13 and 14. It could easily be seen that the optimal threshold value (0.0125) gives directly the END which is accepted by the GSM neuron. END value is equal to -4.56 dB.

4 Conclusion

This new idea to perform self-adaptive terminal, thanks to blind recognition of the standard in use has been firstly proposed in [1]. At that period we made pragmatic choices concerning the neuron’s parameter. These choices were done thanks to extensive simulation results obtained with “live” signals digitized in our lab. In this paper we confirm, with more “theoretical” results, the most important choice concerning the error function. In addition, we improve the mean of choosing the neuron threshold in increasing its lower limit and in decreasing its upper limit. We can conclude that the proposed method is very efficient and robust. It is a first vision of the future fully adaptive intelligent terminals.

References:


Comparison of Signal Processing for Spatio-Temporal Multi-User Equalization in Multi- and Single-Carrier CDMA Systems

L. Brühl, B. Rembold
Institute of High Frequency Technology
RWTH Aachen, Germany
bruehl@ihf.rwth-aachen.de

Abstract

In this paper, signal processing for single- and multi carrier systems is investigated. The transmission chains from modulation to coherent symbol decision of both transmission schemes are presented and compared, with a focus on linear equalization in the frequency domain. While differences in pulse shaping, guard period length and channel estimation are present, a unitary and efficient equalization structure can be applied. An extension of the comparison to space and code division multiple access still shows only marginal differences in transmission spectra of MC-SD/CDMA and SC-SD/CDMA. A joint space-time detection with MMSE optimization criterion is proposed and implementation aspects are discussed.

1 Introduction

For broadband wireless communications in time-varying multi-path environments, channel equalization is one of the most challenging tasks. In order to perform equalization with feasible computational complexity, OFDM (orthogonal frequency division multiplex) was developed and is finally applied in most of the latest high data rate communications systems, like digital video and audio broadcasting and the wireless LAN standards Hiperlan2 and IEEE 802.11. As a competitor to the multi-carrier transmission with OFDM, single-carrier transmission with frequency domain equalization can be shown to yield a comparable spectral efficiency[2] with nearly the same computational effort. While there are differences in modulation, pulse shaping, signal dynamic and channel estimation, the equalization process is very similar in both approaches.

In this paper frequency domain equalization in single- and multi-carrier systems is compared with a focus on signal processing. For an ideal point-to-point connection, the systems only differ in the position of the inverse Fourier transformation, which is applied in the transmitter for OFDM and in the receiver, after equalization, for single-carrier transmission. In a practical application, however, there are more issues which have to be considered. Therefore in section 2 a basic single user OFDM system is introduced and realization aspects concerning spectral forming, guard period, channel estimation and equalization are discussed. In section 3 the OFDM system is then compared to an appropriate single-carrier system. Both systems will be extended to multi-user operation with code division and space division multiple access (SD/CDMA) in section 4. Finally a unitary equalization structure is shown for multi- and single-carrier SD/CDMA.

The presented work is intended to evaluate and compare the signal processing of multi- and single-carrier transmission for future communications systems. Moreover the revealed similarities in equalization may also be interesting for a multi standard chip design, providing equalization hardware for single-carrier mobile communication standards like GSM and UTRA-TDD and the multi-carrier WLAN and broadcasting standards.

2 Basic OFDM System Setup

In this section a basic OFDM system is classified. First transmitter and OFDM symbol structure are defined. Then a receiver with linear MMSE equalization and pilot based channel estimation will be introduced.

![Figure 1. Basic OFDM transmitter](https://example.com/figure1.png)

The block diagram of the transmitter is depicted in figure 1. After coding, interleaving and modulation $N_u$ complex symbols are blocked to form one OFDM symbol. In the middle of the block $N_z$ zeros are inserted and a $N_{FFT}$-point IFFT is performed, where $N_{FFT} = N_u + N_z$. Without this kind of oversampling an ideal interpolation filter would be required to remove the aliasing components after the DA conversion. By the
insertion of additional subcarriers with zero amplitude the aliasing bands are dispersed and give room for feasible transition bands. After the IFFT, the block is cyclically extended to the front and the back. Most of the extension before the block is required for the guard period. A cyclic guard period is required to prevent inter carrier interference (ICI) and inter symbol interference (ISI) in frequency selective multi path environments. To avoid ISI, the guard period length \( T_G \) must be larger than the maximum excess delay \( \tau_{\text{max}} \) of the channel. The subcarriers are orthogonal to each other only when they extend over the whole FFT duration \( T_{\text{FFT}} \) of the receiver. With multipaths of different delays this is only possible when the guard period is extended cyclically.

To reduce the out-of-band spectrum a window function is multiplied with the OFDM symbol. Windowing avoids sharp phase transitions between consecutive OFDM symbols. In fact, the ideal comb spectra of the IFFT is convoluted with the Fourier transform of the window function. Without windowing this would be a sinc function, which decreases rather slowly. To sustain orthogonality between subcarriers, the window must not alter the values within the guard period and the IFFT duration. Therefore some additional extension at the beginning and the end of the OFDM symbol is required for the slopes of the window. For a raised cosine window with rolloff factor \( \beta \) the slopes extend over \( \beta T_{\text{OS}} \) where \( T_{\text{OS}} \) is the resulting length of the OFDM symbol. The slopes of consecutive symbols could overlap because this part of the signal is not used at the receiver. The setup of the resulting OFDM symbol is shown in figure 2.

\[
T_{\text{OS}} = \frac{T_{\text{FFT}} + T_G}{1-\beta}
\]

**Figure 2. Setup of the OFDM symbol**

In a time discrete system symbol duration \( T_{\text{OS}} \), guard length \( T_G \) and window extension \( \beta T_{\text{OS}} \) must be integer multiples of the DA converters sampling period \( T_S \). By dividing \( T_{\text{OS}}, T_G \) and \( \beta T_{\text{OS}} \) by \( T_S \) we get the number of samples for the OFDM symbol \( N_{\text{OS}} \), the guard period \( N_G \) and the window extension \( N_{\text{WE}} = \beta N_{\text{OS}} \) respectively.

The appropriate OFDM receiver is shown in figure 3. We first assume the receiver uses the same sampling period \( T_S \) than the transmitter. After timing and frequency synchronization, which is not further considered, the cyclic extension is removed. The remaining samples are transformed back to the frequency domain with a \( N_{\text{FFT}} \)-point FFT. In the frequency domain the \( N_u \) non-modulated subcarriers are discarded, the \( N_u \) channel coefficients \( c(n) \), with \( n = 1 \ldots N_u \), are estimated and used in the channel equalization. After the equalization a coherent soft bit detection is possible followed by deinterleaving and decoding.

The main advantage of OFDM is the low computational complexity of the equalizer, which consists of the \( N_{\text{FFT}} \)-point FFT, the estimation and calculation of \( N_u \) equalizer coefficients \( \tilde{w}(n) \), with \( n = 1 \ldots N_u \) and \( N_u \) complex multiplications of the FFT outputs with the coefficients. For the calculation of the equalizer coefficients different optimization criteria could be applied. The most common linear approach for OFDM is zero forcing. However, in the following the more general minimum mean square error (MMSE) will be used, because it shows a better performance in single carrier transmission and multiple access. Under the assumption of white, gaussian noise the MMSE solution simplifies to

\[
w(n) = \tilde{w}(n) = \frac{h^*(n)}{h^*(n)h(n) + \sigma^2}
\]

where \( .^* \) denotes complex conjugate, \( n = 1 \ldots N_u \) is the subcarrier index and \( \sigma^2 \) is the noise variance at the receiver. In the basic OFDM setup, \( \sigma^2 \) can be set to zero, which is identical with the zero forcing solution. This simple calculation holds only for the single user OFDM system. For more complex systems with multiple access, the MMSE solution includes matrix inversions (section 4), which can easily grow critical in concern of computational complexity.

A precondition for the MMSE solution is the estimation of the channel coefficients. Channel estimation in OFDM is usually done with pilot symbols, which are distributed over the frequency bands and adjacent OFDM symbols. The number of estimates \( N_{\text{CE}} \) required within the bandwidth depends on the maximum excess delay. The repetition period \( T_{\text{CE}} \) of the estimates is determined by the maximum doppler frequency \( B_{d,\text{max}} \). By choosing

\[
N_{\text{CE}} > 2 \frac{N_u}{N_{\text{FFT}}} \frac{\tau_{\text{max}}}{T_S}
\]

and

\[
T_{\text{CE}} < \frac{1}{2B_{d,\text{max}}}
\]

the Nyquist criteria is met and all intermediate channel coefficients could be interpolated without loss of accuracy. We could define \( r_{\text{pilot}} \) as the number of symbol es-
3 Comparison with Single-Carrier Transmission

After the OFDM configuration was classified, a comparable single-carrier (SC) system will now be defined. Transmitter and receiver for OFDM and SC are compared and similarities and differences will be identified.

First we take a look at the SC transmitter (figure 4). The main difference to the OFDM transmitter is, of course, the missing IFFT, but also a totally different principle of pulse shaping. Because of the continuity of the individual carriers, in OFDM only the borders of the OFDM symbol have to be shaped by the window function. In SC transmission the data symbols vary randomly, so each symbol has to be shaped with an interpolating shaping filter. Usually this is done by oversampling and an interpolating convolution with the impulse response of a raised cosine filter (here with \( L_{\text{rcos}} \) taps and roll off factor \( \alpha \)).

![Figure 4. Single-carrier transmitter](http://example.com)

The difference in shaping also affects the choice of the guard period: the overlapping window slopes of the OFDM symbol are not needed in SC transmission, but the guard period has to be slightly longer, because the channel length, seen by the receiver, is increased by the length of the shaping filter \( L_{\text{rcos}} T_2 \). In order to use frequency domain equalization (FDE) at the receiver, the guard period of the SC burst has also to be cyclically extended. Note, that in SC transmission, in contrast to OFDM, the cyclic system model could also be forced by appending the received data burst with zeros.

To get a comparable configuration to the OFDM setup we now assume a SC burst structure with \( N_{\text{FFT}}/2 \) data symbols, a cyclically extended guard period of length \( (T_0 + \beta T_{\text{OS}})/2 \) and a pulse shaping oversampling factor of two. When comparing this configuration with an OFDM configuration with two fold oversampling \( N_u = N_{\text{FFT}}/2 \) we get a very similar burst structure with the same data throughput and nearly the same bandwidth requirements.

For this setup, we now take a look at the SC receiver with FDE (figure 5). Again we assume the same sample rate as in the transmitter, which means twice the symbol rate. In distinction to the OFDM receiver, the equalizer output has to be transformed back to the time domain before symbol decision. Because of the oversampling, fractionally spaced equalization is enabled. So the output of the IFFT has to be sub-sampled with the symbol rate (here: half the sample rate). The subsampling could also be applied in the frequency domain by accumulating the aliasing frequencies [1]. The IFFT length could then be halved.

![Figure 5. Single-carrier receiver](http://example.com)

The channel estimation usually differs from the OFDM case. Here we assume that the channel was estimated in a training burst prior to the considered data burst. A very efficient approach with cyclic midamble base codes is specified in the UTRA-TDD mode and presented in [3]. Normally the impulse response of the channel is calculated and has to be transformed to the frequency domain for FDE.

Even on the same physical channel, the channel coefficients will differ between SC and OFDM because of the pulse shaping filter included in SC transmission. Once the channel coefficients are known in the frequency domain the same MMSE optimization criterium as in the OFDM receiver could be used to equalize the channel. Even when, in contrast to OFDM, the channel coefficients for all frequency points are estimated at once, it would be sufficient to calculate \( N_{\text{CE}} \) equally spaced equalization coefficients and than interpolate all the intermediate coefficients. As quoted in section 2, huge savings in computational complexity might be achieved by doing this.

Most of the signal processing, including FFT, the calculation of the equalizer coefficients and the weighting with the coefficients, are exactly the same as for the OFDM configuration. In a hardware realization, however, different requirements on word widths have to be taken into account. In the OFDM system, the signal dynamic after equalization is quite low, because all carriers have the same known amplitude. In the SC case the dynamic...
in the frequency domain is quite undefined and could be significantly higher.

Before the configurations are extended to multiple access, a more efficient receiver configuration without oversampling is proposed. For transmission, oversampling is required to create a properly defined spectrum with limited bandwidth. At the receiver, however, it’s quite ineffective to throw away half the output values. For an oversampling factor of two, in OFDM only the lower half of the carriers are modulated. At the receiver the other $N_u$ values in the middle of the FFT output block are discarded. It’s a mathematical equality between this and sub-sampling the input values by a factor of two and making a $N_u/2$-point FFT. In the SC case the 3dB bandwidth of the shaping filter is equal to half the symbol rate. Therefore sub-sampling of the input signal down to symbol rate will introduce severe aliasing, but in frequency selective multi-path environments there is no degradation expected between fractionally spaced and symbol rate equalization. Thus the sampling rate and the FFT length of the receiver can be halved without degradation for both, the OFDM and the SC configuration. In the following, this more efficient approach will be used.

To summarize the comparison: the main differences between the signal processing of the proposed OFDM and SC configurations are the position of the IFFT, pulse shaping, guard period and channel estimation. In the receiver most of the signal processing, including FFT, calculation of the equalizer coefficients and the weighting with the coefficients, are exactly the same. In both configurations oversampling has to be applied in the transmitter and can be omitted in the receiver. In the following section the comparison will be extended to systems with multiple access.

4 Extension to Multiple Access

For both, OFDM and SC transmission, several multiple access schemes (MA) are known. In this paper a combined space and code division multiple access (SD/CDMA) will be applied. For SC a SC-SD/CDMA approach with short spreading codes, as described in [5] will be used. The OFDM counterpart is multi-carrier CDMA (MC-CDMA) [4] with multiple receive antennas (MC-SD/CDMA). It will be shown that most of the signal processing for the equalization of both transmission schemes is exactly the same, as in the single user case. First the transmitters for MC and SC are classified and their transmit signals are compared. Then a unitary receiver structure is proposed, which is able to equalize both signals.

In MC-CDMA a block of $N_u/Q$ data symbols is repeated $Q$ times to form an OFDM symbol with $N_u$ used subcarriers. Each sub block is then multiplied with it’s associated chip of a user specific spreading code of length $Q$ (figure 6). Afterwards the whole block is transformed into the time domain, cyclically extended and weighted with a window, just as described in section 2.

Figure 6. Transmission spectrum in MC-CDMA

There are different views on spreading in SC transmission. One of them is to insert $Q - 1$ zeros between adjacent data symbols and convolute the output with the spreading code. In the frequency domain, the zero stuffing means a $Q$ fold repetition of the data symbols spectrum and the convolution is equal to a multiplication with the spectrum of the spreading code. As can be seen in figure 7 the process is quite similar to that in MC-CDMA. The main difference is the shape of the spectrum of the spreading code, which in MC-CDMA is constant over the sub blocks whereas continuous in SC transmission. Like in the MC-CDMA transmitter we define $N_u/Q$ data symbols and spreading codes of length $Q$ to get the same burst structure as introduced in section 3.

Figure 7. Transmission spectrum in SC-CDMA

Further on a system setup of $K$ transmitters using orthogonal spreading codes and a base station with $M$ parallel receive antennas is regarded, where $K < QM$. With multiple receive antennas diversity gain and multi-user separation can be achieved. In a frequency selective channel the orthogonality of the spreading codes is lost. For small spreading factors the cross correlation between the codes will be unacceptable, so a joint detection has to be applied.

Assuming perfect synchronization, for each antenna $m = 1 \ldots M$ the guard period is discarded and a Fourier transformation is performed. In the frequency domain the following system model will hold for both, MC-
SD/CDMA and SC-SD/CDMA:

\[ x_n = H_n d_n + n_n \quad \text{with} \quad n = 1 \ldots \frac{N_u}{Q}, \quad (5) \]

where \( d \) is a \( K \)-length symbol vector, \( x \) is a vector of length \( QM \) containing received data values, \( H \) is a \([QM \times K]\) channel matrix, including channels and spreading codes and \( n \) represents the noise spectrum. For OFDM \( d \) contains a data symbol of each of the \( K \) transmitters and \( n \) is the data symbol index. For SC \( d \) contains one frequency slot of the Fourier transform of each transmitters data block and \( n \) means the frequency index.

For uncorrelated data symbols and white gaussian noise with variance \( \sigma^2 \) the according MMSE solution for \( d_n = W_n x_n \) is

\[ W_n = (H_n^H H_n + \sigma^2 I)^{-1} H_n^H \quad (6) \]

As we see, the equalizer coefficients \( W_n \) are now matrices of the size \( K \times QM \) and \([K \times K]\) matrix inversions are required for their calculation. For the implementation of the FDE there are two different possibilities to structure the the input data and calculate \( x_n \) and \( H_n \). One implementation requires \( M N_u \)-point FFTs and the other \( QM \) \( N_u/Q \)-point FFTs. The latter features a lower computational complexity for \( Q > 1 \) because of the \( N \log(N) \) behavior of the FFT. But in practise a hardware implementation with variable spreading codes must be designed for the worst case \( (Q = 1) \) anyway. Moreover it will always be less effective to implement FFTs with variable lengths \( (N_u/Q) \) than \( \text{fix} \)ed lengths \( (N_u) \). Therefore the \( \text{fix} \)rst approach is preferable and will be shown for the input vector \( x_n \). The channel matrices \( H_n \) can be composed in the same way out of the frequency domain channel coefficients.

One \( N_u \)-point FFT per receive antenna is calculated, resulting in \( M \) vectors \( r^{m} \). For each antenna vectors \( x^{m}_n \) are then composed by

\[ x^{m}_n = [r^{m}(n + 0 \cdot \frac{N_u}{Q}) \ldots r^{m}(n + (Q - 1) \cdot \frac{N_u}{Q})] \quad (7) \]

Now \( m \) vectors \( x^{m}_n \) of length \( Q \) are stacked to form

\[ x_n = \left[ x^{1}_n \ldots x^{m}_n \ldots x^{M}_n \right]^T \quad (8) \]

With this structure, the products \( W_n x_n \) cause an individual weighting and superposition of each of the \( Q \) repeated spectra of the transmitted symbols (fignures 6 and 7). The different weighting of the spectra is included in the channel estimation. Therefore the equalizer is identical for SC and MC transmission. The multiple receive antennas provide additional copies of the spectra, which are affected by other, more or less uncorrelated channel coefficients. Therefore the equalizer features \( QM \) degrees of freedom to separate the \( K < QM \) transmitted signals.

In the single-carrier case, \( d_n \) are the MMSE estimates for the transmitted data blocks Fourier transforms and have to be transformed back with \( K N_u/Q \)-point IFFTs. Reversed to the oversampled receiver in section 3 the spectral superposition of the \( Q \) spectra could also be made by sub-sampling the time domain signal by \( Q \). Thus instead of the variable length \( N_u/Q \)-point IFFTs could be replaced by \text{fix}x ed length \( N_u/Q \)-point IFFTs.

5 Conclusions

In this paper, the signal processing for single- and multi carrier systems was compared, with a focus on frequency domain equalization. The main differences in pulse shaping, guard period length and channel estimation were shown for a single user case. Under the only assumption of equal numbers of transmitted symbols and sufficiently dimensioned cyclically extended guard periods, exactly the same frequency domain MMSE equalization could be applied for both systems. The setup was then extended to a multi-user system with space and code division multiple access. The spreading process was depicted for MC-CDMA and SC-CDMA, with a minor difference in the weighting of the transmitted spectra. With this in mind, it is obvious that also for SD/CDMA a unitary frequency domain equalization is feasible. The MMSE solution could be implemented with \text{fix}xed FFT sizes independent of the spreading factor.

References

SOFTWARE DEFINED RADIO TECHNOLOGY FOR MULTISTANDARD TERMINALS

Jörg Brakensiek, Reimund Wittmann, Mohsen Darianian
Nokia Research Center, Bochum, Germany
jorg.brakensiek@nokia.com

Abstract – This paper will present the target and vision of the German government (BMBF) funded project Mobile, re-configurable Multistandard Terminal for UMTS and Wireless LAN (MoReTeX) as part of the joint BMBF project Software Defined Radio Based Architecture Studies for Re-configurable Mobile Communication Systems (RMS).

Starting from the vision of a heterogeneous network architecture, possible scenarios will be investigated from the terminal user perspective. The project approach towards a multi-standard terminal for UMTS and Wireless LAN is based on software defined radio (SDR). A first system architecture will be presented. Key issues are both the re-configurable digital baseband and the re-configurable analog front-end.

Keywords – Software Defined Radio, multi-standard, re-configurable, terminal, UMTS, WLAN, heterogeneous network

1. INTRODUCTION

The next generation mobile communication systems will lead to an integration and intercommunication of existing networks and access technologies. These technologies will be integrated on a common wireless network platform forming a heterogeneous network. This heterogeneous network will allow the separation of transfer requests over a wireless channel from the used transmission access technology and its transmission protocol.

1.1 Heterogeneous Network

During the last years the mobile community has introduced a wide range of communication systems and standards. They can be organized in a layered structure, shown in Figure 1.

These layers provide a hierarchical view on the achievable network structure.

- Distribution layer, based on broadcast networks (e.g. DVB-T), provides a global coverage, with full mobility and large coverage, supporting only downlink communication.
- Cellular layer, based on 2G, 2.5G and 3G systems (i.e. GSM, GPRS, EDGE, UMTS), provides full coverage, with full mobility and global roaming, supporting individual communication links.
- Hot spot layer, based on wireless local area networks (e.g. IEEE 802.11x, HIPERLAN), provides local coverage and restricted mobility, supporting individual high data rate links.
- Personal network layer, based on pico cell systems (e.g. Bluetooth), provides short-range coverage with restricted mobility, supporting point-to-point communication links.
- Fixed (wired) layer, based on fixed access (e.g. DVB-C, ISDN, xDSL and POTS), provides point access, supporting high data rate links.

In a heterogeneous network scenario, systems are connected by implementing handover procedures, which can either be horizontal, i.e. between systems belonging to the same layer (intrastandard), or vertical, i.e. between systems belonging to different layers (interstandard). The different handover possibilities are demonstrated in Figure 1 with horizontal and vertical arrows.

1.2 Application Scenario

A heterogeneous network architecture requires systems, which are supporting more than one access technology, and which are very interesting, especially from the terminal point of view. Inside a heterogeneous network, any user, application or service will be able to choose the network resource based on parameter, which are important for the current transfer request:

- available and supported access technologies
- guaranteed bandwidth or data rate
- supported mobility
- access time
- quality of service (QoS)
- transfer costs (€/Bit)
- needed power consumption (pJ/Bit)

Terminals may use the data transfer possibilities in a very flexible manner. The data transfer itself is handled transparently; i.e. the transfer is done using one or more different access technologies, invisible for the user.
Using IP as the common transfer protocol on a higher protocol layer would be the basis for an All-IP network. A heterogeneous network scenario is depicted in Figure 2, showing a mobile terminal (e.g. Laptop or PDA), linked to different networks, e.g. providing the following services:

- store data on file server via Bluetooth
- transfer videos to the TV via Wireless LAN
- send a note to a friend driving in a car via UMTS
- download IP data via DVB-T

**Figure 2: Heterogeneous Network Application Scenario**

Those kinds of new network architectures require mobile terminals, which are supporting different transmission standards.

2. DEFINITION OF A RE-CONFIGURABLE SYSTEM

Re-configuration is seen as the key technology allowing the implementation of heterogeneous network terminals. Therefore a re-configurable system will be defined in more detail subsequently.

2.1 Definitions of multi-X

Re-configuration in this case means the support of different system properties, which are often described as multi-X, where X defines a special system characteristic, providing multiple views of the functionality. Some of those multi-x definitions are given in the following:

- **Multi-band**
  Multi-band systems are supporting more than one dedicated frequency band, which is used by one wireless standard (e.g. GSM 900, GSM 1800). Multi-band is only important for the RF-Front-End, as the digital baseband should be independent from the transmission frequency.

- **Multi-standard**
  Multi-standard systems are supporting more than one air interface standard. They can be inside one standard (e.g. UMTS/TDD, UMTS/FDD), inside a network layer (e.g. GSM, UMTS) or across different network layers (e.g. UMTS, WLAN).

- **Multi-function**
  Multi-function is a system property, which is related to higher layers (application layer), providing different services (e.g. telephony, data services and video streaming).

- **Multi-carrier**
  Multi-carrier systems are supporting more than one independent transmission and receiving channel at the same time. (In this context multi-carrier is not related to the modulation scheme, e.g. OFDM.)

- **Multi-mode**
  Multi-mode is a combination of a dedicated multi-standard and a multi-band system: Multi-mode = Multi-standard + Multi-band. (Other definitions of multi-mode are based on different modulation schemes.)

Based on these definitions, the projected terminal will be a multi-standard one, as this is the most interesting scenario from the terminal perspective. Multi-carrier is more related to basestations, doing parallel processing of independent channels in adjacent frequency bands. Multi-band functionality is somehow included in this approach, as UMTS is operating at 2 GHz and Wireless LAN at 5 GHz.

2.2 Levels of re-configuration

Re-configuration can be done on different timelines, i.e. the re-configuration strategy is heavily dependent on the number of expected re-configurations during the product lifetime and the time period between two consecutive re-configurations.

The following levels of re-configuration can be defined:

- **Commissioning**
  Configuration is done only at the time of product shipping, when the customer has asked for a dedicated mode (i.e. standard and frequency band). Therefore this would be more a configuration than a true re-configuration.

- **Re-configuration with downtime**
  Re-configuration is done only a few times during the product lifetime, e.g. change of the network infrastructure. Re-configuration will take some time, where the system is switched off. This may include the exchange of some hardware components (e.g. antenna).

- **Re-configuration on a per call basis**
  Re-configuration is done dynamically on a per call based decision. That means that any downtime is not acceptable, i.e. only parts of the system (e.g. RF-Front-End, digital baseband part) can be rebooted, whereas other has to operate normally (e.g. GUI and running application).

- **Re-configuration per timeslot**
  Re-configuration is done highly dynamically, with a very fine time granularity, i.e. re-configuration can be done often during a single call or data connection. This type of re-configuration does not look reasonable, otherwise the framing structure of independent systems would have to be synchronized.

Based on these definitions, the multi-standard terminal will do a re-configuration on a per call basis. Re-
configuration with downtime will restrict the usage and will be comparable with two or more exchangeable modules (e.g. PCMCIA cards), not allowing independent transmission, optimized for the current requirements. For basestations re-configuration with downtime or even commissioning is suitable, but a multi-standard UMTS/WLAN basestation will mostly implement both standards in parallel, otherwise no system handover procedures can be realized.

2.3 Re-configuration strategy

Having the target to develop a multi-standard terminal, which can be re-configured for UMTS and Wireless LAN on a per call basis, there are three different possible re-configuration strategies, which will be examined in the following.

2.3.1 One bit re-configuration

The simplest re-configuration would be to switch between standards on a very high level. This would require two independent baseband architectures implemented side by side. At some place the incoming data stream is multiplexed to the dedicated baseband processing chains. Then the processed data is given to the data sink for further data processing. The amount of needed information for re-configuration is at minimum one single bit, defining the multiplexer behavior. This approach has the one main advantage, which makes it feasible to think about. The system design time, i.e. the time-to-market, can be very short from the ASIC development point of view, if both systems are already developed, verified and implemented inside the company or can be bought from a 3rd party IP vendor.

Though this approach may be good enough for some dedicated basestation applications, it is not applicable for terminal applications as power consumption and area efficiency are the dominating factors in ASIC design for high volume, mobile devices.

2.3.2 Software Defined Radio

The most flexible configurable solution is Software Defined Radio (SDR), where the same dedicated device can be re-programmed to support various standards. Software defined radio (or just software radio) is the emerging technology to build flexible radio systems, introducing re-programmable and re-configurable hardware.

It represents an ideal that may never be fully implemented but that nevertheless simplifies and illuminates tradeoffs in radio architecture [2]. There are two trends in software defined radio system development

- Moving the border between the digital and the analog domain to higher frequency, using wideband, high-speed A/D converters.
- Replacing dedicated hardware (i.e. ASIC) by digital signal processors (DSP) for baseband processing.

The ideal software radio receiver architecture is shown in Figure 3. Only the bandpass filter (BPF) and the low noise amplifier (LNA) are analog components. A/D conversion is done directly on the RF signal.

![Figure 3: Ideal Software Radio receiver [3]](image)

The main target of software defined radio architectures is to be a general-purpose communication platform for existing wireless access technologies and future extensions or enhancements. Threfor the re-configurable/re-programmable hardware has to provide enough processing power and communication bandwidth. New features and access technologies should be downloadable, e.g. via the air interface or serial I/O. Hardware resource managers have to check if a new baseband processing can be realized on the given hardware resources.

The main advantage of this approach is that the system designer can change parts of the system algorithms, without having to do a re-design of the entire ASIC. This will lead to shorter development times, if especially the system is not fixed or standards are changing. But the available system resources are limiting the possible algorithmic changes. The main disadvantage is that a general-purpose architecture has to take into account future changes and enhancements of standards and systems, which results in a huge overhead of available resources and performance. In addition, high-bandwidth communication systems such as Wireless LAN IEEE 802.11a with 54 MBit/s would require a massively parallel implementation of microprocessor and DSP resources in order to fulfill the performance requirements. They will contribute mainly to the power consumption and area overhead of the system.

![Figure 4: Re-configuration Strategies](image)

(a) One bit re-configuration, (b) Software Defined Radio, (c) Re-configuration by parameterization
2.3.3 Re-configuration by parameterization

The terminal approach, which is targeted in the RMS project, lies in-between the previous two concepts. Similarities and differences between the standards are identified and parameterized. The goal is to develop dedicated hardware for the common baseband structures and to accommodate the differences by reconfiguration based on parameters.

Therefore this approach can be seen as a Software Defined Radio approach restricted towards fixed supported functionality (i.e. multi-standards/multi-band) for a number of given access technologies (i.e. standards). This approach will be named as a Software Re-configurable Radio (SRR).

It is not intended for this approach to be open for each new feature, standard and complex system enhancement. Re-configuration of the implemented system will be possible later on, but these changes will be restricted to parameterization of basic algorithms and operations, which will not be changed themselves. Also the execution order and the data flow can be re-configured. The idea is to provide as much re-configuration capabilities as needed, but as few as possible.

The system architecture, which is underlying this multi-standard/multi-band radio, has to be scalable and modular in order to allow future extension of the system towards new standards, standard enhancements and additionally allocated frequency bands. Therefore SRR is the way towards SDR, which can be implemented efficiently and power optimized in the near future.

3. RE-CONFIGURABLE HARDWARE ARCHITECTURE

The proposed approach of a software re-configurable radio will require new hardware architectures, supporting both a re-configurable front-end and digital baseband.

On the one hand the developed system architecture should be suitable for the intended multi-standard system, i.e. it should support UMTS and WLAN applications. On the other hand the architecture should lead to a new future proven architecture, which could be easily extended and enhanced towards other standards or functions.

System analog part

RF

BB

A/D

D/A

Crkt.

Crkt.

Figure 5: Re-configurable architecture

Therefore the system architecture has to be flexible, scalable and modular enough to serve the different system requirements concerning computational power, system constraints, functionality and efficiency.

The very high level architecture for the multi-standard terminal is shown in Figure 5. The front-end and digital baseband architectures are described in more detail in the following sections.

3.1 Front End Architecture

As mentioned above, software defined radio approaches are targeting to move the border between analog and digital domain towards higher frequencies. The digitizing of the analog band is done at an intermediate frequency (IF) with a single A/D converter. This will lead to the superheterodyne receiver with IF sampling as shown in Figure 6.

![Figure 6: Superheterodyne Receiver Architecture](image)

After the RF-stage with bandpass filtering and low noise amplification (LNA), a mixer and local oscillator (LO) are used to down convert the desired signal band to the IF frequency. [5]

One major problem in the superheterodyne receiver is the high performance requirements of the RF bandpass filter to provide sufficient image band suppression. This will lead to increasing IF-frequencies or a two IF receiver at the cost of added receiver complexity. Normally IF filters cannot be integrated, resulting in off-chip components, increasing power consumption, receiver sizes and cost.[4]

Only the further integration of components will allow reaching the sophisticated power consumption requirements and cost constraints for a high volume, mobile terminal.

Therefore the direct conversion receiver architecture as shown in Figure 7 seems to be the most suitable architecture and will be investigated for the multi-standard terminal.

![Figure 7: Direct conversion receiver architecture](image)

This architecture provides the lowest number of external components (high integration) and the A/D converter can operate at much lower frequencies (power consumption).
The direct conversion receiver will introduce several noise sources that either do not exist in superheterodyne receivers or can be neglected there:

- I/Q imbalance
- DC-offset
- Second order harmonics
- 1/f noise

They have to be corrected in the digital baseband and the front-end. Non-ideal signal behavior, which is mostly based on technology constraints and imperfections, e.g. process variations, will be adjusted in the front-end directly using signal path optimization.

To match the two standard requirements in terms of frequencies and bandwidth, the front-end components have to be re-configured. Feedback from the digital baseband will provide adjustment possibilities of front-end functions in order to optimize the signal performance, e.g. DC-offset decrease. Therefore we will introduce signal path optimization into the front-end architecture.

The signal path optimization will have three different possibilities to improve the signal quality:

1. Calibration of the FE component in the front-end itself (self calibration)
2. Calibration of the FE component from the digital baseband (calibration with digital feedback)
3. Correction of the signal in the digital baseband based on measured imperfections

### 3.2 Digital Baseband Architecture

The digital baseband architecture supporting a software re-configurable terminal for UMTS and Wireless LAN has to deal with the following challenges:

- Different modulation schemes and underlying basic algorithms: OFDM with FFT for Wireless LAN and W-CDMA with RAKE for UMTS
- Different data rates, up to 54 Mbit/s for Wireless LAN and 2 Mbit/s for UMTS
- Optimized power consumption

Re-configuration of the digital baseband in this case means using the same hardware platform for both standards, with as little parallel hardware as possible. Different re-configurable/re-programmable technologies are available supporting re-configuration:

- Digital signal processors (DSP) provide a basic instruction set with special instructions for dedicated applications. The instruction flow is stored in the memory and can easily be changed. Even though the achievable clock frequency is very high, the overall speed is not sufficient to handle these complex high-speed applications
- Field programmable gate arrays (FPGA) are now also available as embedded devices. They consist of a field of basic logic cells with programmable interconnection. FPGAs provide a lot of flexibility but the achievable speed is restricted and power consumption is high.
- Application specific integrated circuits (ASIC) are optimized for speed and power consumption, but provide less or even no re-configuration. ASICs can be made re-configurable to some extend, introducing basic programmability. This will not change the underlying algorithms itself, but can do some parameterization.

Currently there is heavy research on re-configurable architectures and start-up companies are providing first results. The main direction is implementing parallel processing elements with flexible interconnections, targeted for general-purpose applications in wireless communication systems. The main disadvantage of these approaches is that the general-purpose approach introduces heavy overhead and first test samples are showing high power consumption (several Watt), which is not acceptable for mobile terminals.

The proposed approach, which will be used in the RMS project, is based on arithmetic units that are tailored to meet the demands of the basic signal processing algorithms. The idea is to provide as much reconfiguration capabilities as needed but as few as possible. These arithmetic units, called hardware accelerators, are supporting a microprocessor or a DSP doing independent block based complex signal processing.

These hardware accelerators are tailored towards similar algorithms, which are based on the same fundamental calculations, forming a class of algorithms. This approach is shown in Figure 9. These accelerators allow a scalable and modular realization of the re-configurable digital baseband. Basic re-configuration possibilities of these accelerators allow an optimal configuration for dedicated algorithms. Details of this architecture are described in [6].

System partitioning between DSP/µP, hardware accelerators and dedicated user logic (UL) will be a
major topic. New approaches have to be investigated, which will support the system designer to decide whether to implement a function in software (i.e. on a microprocessor or DSP), in re-configurable hardware or fixed ASIC. The flexible architectures require a flexible interconnection of system components, to serve different standards and modes. Therefore the on-chip interconnection and communication between different system components will be the critical factor in this software re-configurable architecture.

4. DEMONSTRATOR

The developed re-configurable architecture will be demonstrated, showing the re-configuration of main digital baseband parts. The demonstrator will concentrate on the receive path of the terminal, as this part fulfills the more complex signal processing tasks compared to the transmitter. The demonstrator functional block diagram is shown in Figure 10.

![Figure 10: Digital Baseband Demonstrator](image)

The UMTS rake receiver part and the Wireless LAN FFT part will be implemented including synchronisation and channel estimation parts in a re-configurable manner most probably on a prototype environment based on FPGAs, microprocessor and a DSP. Channel multiplexing, interleaving and convolutional encoding will be done in a software simulation environment (e.g. CoCentric System Studio).

The additional front-end demonstrator will consist of real silicon components for the multi-standard receiver chain, proofing the re-configurable properties of key components and the signal path optimization. It is planned to have prototypes in Infineons latest 0.13 μm RF-CMOS technology.

5. SUMMARY

As communication networks will merge together in the future, building heterogeneous networks, new system concepts have to be developed, supporting multi-standard functionality inside terminals. Target of the Nokia Research Center work in the BMBF funded project RMS is the development of a re-configurable multi-standard terminal for UMTS and Wireless LAN. Inside this paper the vision and the approach of this project work including first results have been presented. The front-end will be based on a direct conversion receiver, whereas the digital baseband architecture will implement a re-configurable hardware accelerator concept. Final project result will be demonstrated.

6. ACKNOWLEDGEMENTS

The RMS project is part of the BMBF funded Mobile on Chip program. Other project partners are Alcatel and Lucent, concentrating on basestation issues and Infineon providing the latest silicon technologies. The project has been started last year and will be finished in summer 2004.

The research work of the Nokia Research Center is done in cooperation with the universities of Dortmund (Prof. Götze), Karlsruhe (Prof. Jondral) and Berlin (Prof. Böck).

7. REFERENCES

Abstract — Algorithms in telecommunication applications today require more and more calculations to be executed, maximizing the data transmission efficiency by using complex source and channel coding techniques. In most cases the computational power needed exceeds the performance of general purpose as well as digital signal processors (DSPs) by magnitudes. Application specific integrated circuits (ASICs) can cope with these requirements but are a costly and man-power consuming alternative when designed for only one purpose. Programmability after production is restricted heavily, which does not meet the demands of faster product-to-market time when standards are not fully fixed yet. In this paper we will describe a universal hardware accelerator for telecommunication applications to be used in a System-on-Chip (SoC) environment. The architecture exploits parallelism by using multiple processing elements (PEs) and efficiently computes the large variety of algorithms, which can be mapped to a field of those PEs.

Keywords — DSP, ASIC, hardware accelerator, SoC, processing element, algorithm classes, reconfigurable hardware

I. INTRODUCTION

The concept to accelerate complex algorithms can be derived by looking at a relatively simple example as shown in figure 1, which represents the signal flow graph of an 8-point Fast-Fourier-Transformation (FFT) as it is found in other variants as a component in OFDM transmission systems. On the left side eight time domain data samples enter the signal flow graph and after a number of computations, the eight transformed frequency domain data samples are available on the right side. It is obvious that this signal flow graph consists of many atomic but common operations, which are circled in this graph and are the so-called “butterfly” operations combining two complex input operands and producing two complex results. In the case of the 8-point FFT, 12 butterfly computations are encountered, which are grouped in three stages (see figure 2). In each stage of the FFT four butterfly computations could be done in parallel and therefore it is possible to have four parallel hardware butterfly execution units to achieve a speedup of 4 compared to a serialized computation of this algorithm as it would usually have to be done on a single-issue general purpose processor or DSP - of course only under the circumstance that the processor has such a complex execution unit, which they usually do not and have to split up this butterfly in even simpler instructions. If these execution units are also interconnected, a general structure as in figure 3 is the outcome, not showing the possible fully-meshed interconnection scheme for reasons of viewability. For many algorithms this field of processing elements (PEs) can be relatively large because of the algorithmical complexity - here only an 8-point-FFT was used as an example - with a large number of PEs and therefore not always practical to be implemented completely in hardware, also regarding the complex interconnect-structure between the PEs, which could only be solved by a sophisticated and
high-bandwidth bus structure. Many different approaches have been and are still investigated in university and commercial research projects and have led to different accelerator architectures, optimized for respectively one specific purpose. As an example, the RAW project ([1],[2]) is shortly described, which originates from the Massachusetts Institute of Technology (MIT) in Boston. The basic idea is to design an array of quadratic tiles, which allows direct placing in the layout phase. The cost for testing is greatly reduced, as only one tile has to be examined. Figure 4 shows the architecture of one tile element. The main components of each tile are a tile processor, a static switch processor and a dynamic router. The tile processor uses a 32-bit slightly modified MIPS instruction set, while the static switch processor also incorporates a MIPS-like instruction set reduced to move, branch and jump instructions and has a route component to transfer data to other switch processors. The dynamic router itself runs without user intervention.

Each tile element also includes 32 kByte SRAM data memory and 32 kByte SRAM instruction memory dedicated to the tile processor, whereas the instruction memory is uncached while the data memory can be cached or uncached. Additional memory of 64 kByte is used by the switch processor as instruction memory.

The RAW chip containing 16 tiles arranged in a 4x4 matrix was implemented using IBM’s SA-27E process, which is a .12 micron, 6 layer, copper interconnect process to be demonstrated in a handheld device as well as a RAW supercomputer board containing 16 RAW chips on one board.

This RAW approach shows a great overhead for routing of data signals in order to achieve a possible fully-meshed interconnections scheme, but is a mighty architecture for many different tasks to be executed because the inherent PEs are very complex and closer to generic purpose processors.

Our approach focuses on Multi-Input-Multi-Output (MIMO) processing to also cope with functions inside the PE that produce multiple results. The problem of interconnectivity between such a large number of PEs is minimized in partly sequentializing the operations, which would actually be performed in the virtual large field. A major advantage compared to having a large processing field is the fact that the resource efficiency of course is higher, because using a only-real PE-field approach demands to design for the worst-case of a large field with all PEs working in parallel and therefore the problem of keeping all the PEs busy with computations when solving smaller problems arises.

In each timestep, in our terms called a “sweep”, the complete virtual field would perform one computation and new temporary results are created inside the field. This sweep is operated by using a subset of real PEs and stepping through the virtual field while storing the temporary results in memory. Figure 3 shows how two PEs, circled here, out of the large virtual field are grouped and computed by real PEs in one activation out of many of the complete sweep. The accelerator is based on an approach of a virtual field of PEs, allowing all subsets of a full-meshed connection scheme. The size of the virtual field is dependent on the amount of memory allocated to the accelerator purpose. The operations are done by a subset of only a few parallel operating but higher-complexity PEs, which are specifically designed for a distinguished class of algorithms and in contrast to
the RAW architecture specialized for only one class of algorithms in telecommunication applications. Different classes of algorithms, whereas one specific algorithm can also belong to more than one class, have been identified and can be efficiently processed with one optimized PE for each class. In the following some examples of these overlapping classes are given:

- Linear transformations, matrix multiplication (FFT, DCT, FIR)
- Matrix based digital signal processing (least square adaptive filtering, subspace extraction)
- Digital filtering (FIR, IIR, Up/Down-Sampling)
- Galois field arithmetic (e.g. Reed-Solomon-Codec)

### II. Architecture

The real PEs subsequentially compute the complete virtual field whereas the intermediate results are stored in memory and the connection scheme is accomplished by memory addressing to feed the PEs with data. Figure 5 shows the proposed hardware architecture and how it is embedded in a microprocessor and peripherals environment. It can be accessed via the standardized PVCI interface and therefore allows to be connected to any bus system using a Peripheral Virtual Component Interface (PVCI, see [3]) compliant bus wrapper in a SoC system, here shown connected to a microprocessor via the processor bus.

Performance comparisons to leading edge DSPs have proven that the algorithms using specialized PEs are computed much more cycle effective and the scalable architecture allows even greater speedups compared to DSPs with only few execution units. Further explanations will be done on our proposed architecture, which in this configuration contains two PEs, PE1 and PE2. These cells perform the computations on the data and have an additional configuration input, e.g. to allow switching to different operation modes.

All data-values needed for computation are stored in a block called Output-Buffer-Interface (OBI). The OBI also contains memory for storing intermediate values, that are created during run-time. All data-values resulting from the PEs are stored in the OBI. On the other hand, all input data values for the PEs come from the OBI. The Output-Buffer-Interface is described in more detail in section II A. The addresses for the OBI memory access actions, which provide the data from its buffers to the PE inputs are stored in the NetRAM, which operates as an address generator connected to the OBI. The NetRAM contains a memory module, where all addresses needed for the computation of the given algorithm are stored. The NetRAM structure is described in more detail in section II B.

OBI and NetRAM each contain memory, but no control logic. Therefore, another block is necessary, which controls the actions of NetRAM and OBI. This is handled by the ControlFSM. It acts as a finite-state-machine (FSM). This block is described in more detail in section II C.

#### A. Output Buffer Interface (OBI)

The Output Buffer Interface (OBI) inherits the necessary memory to store the temporary results inside the virtual field. In a single sweep, which is defined by a complete computation cycle of the virtual field and many processing steps of the subset of real PEs, the PEs are fed from the OBI out of the active-memory banks and at the same time the PE results are stored in the shadow-memory banks. After every complete sweep the functionality of both memory banks are swapped.

Additionally to storing the intermediate values in the virtual field, also two memories are dedicated as input value and output value memories, called InRAM respectively OutRAM. A special software driver, also implementable in hardware of course, takes the input stream of data to be computed, sorts it into the InRAM and also reads the results out of the OutRAM to stream them back to the microprocessor.

#### B. NetRAM

The addressing of the OBI-inherent RAM blocks to feed the PEs as well as the configuration of the PEs in one specific processing step is done by the NetRAM, which holds a long configuration word. The NetRAM is a memory block storing the read addresses for the Output Buffer Interface, as well as the configuration vectors for the PEs. Figure 6 shows how the NetRAM is organized. Each column in the NetRAM contains the address information for one single activation of the OBI and the two PEs. The read address for the NetRAM is provided by the ControlFSM (see section II C).
C. Control Finite State Machine

The ControlFSM operates as a finite state machine controlling the memory actions of NetRAM and OBI. It is activated externally, by the microprocessor for example, and also signals the finalization of a complete “Megasweep”, containing several programmed sweeps, via an interrupt signal. The block itself contains a memory table storing the behaviour of the FSM, which has an entry for each sweep to be performed containing information about the NetRAM start address and the number of NetRAM accesses to be performed (equals to the number of activations of both PEs).

The NetRAM is addressed by the ControlFSM and its programmability allows multiple sweeps of the same NetRAM address ranges or the sequential execution of different virtual field connection schemes. Both together, NetRAM and ControlFSM, then manage an equivalent to multiple different subprogram executions and also loops as they can be found in standard microprocessors.

The ControlFSM also provides the write addresses for the result data coming from the PEs to the OBI, which are written in a linear fashion.

III. Design Flow

Two different designs flows have to be differentiated:
- hardware design flow of new PEs for different algorithm classes
- software design flow of NetRAM and ControlFSM configuration files

A. PE design flow

The PEs are usually first designed in a C-level simulation environment of the accelerator platform, which is available for Synopsys CoCentric System Studio. Here different PE internal architectures can be implemented fast and tested for algorithmical functionality, first in floating-point implementation, then considering the effects of fixed wordlengths.

The next step is the implementation in a hardware description language and co-simulations with the System Studio model will follow. Here also a simulation environment of the complete accelerator is available and the configuration can be done with the necessary microcode files, which are created in the parallel software design flow.

B. Accelerator microcode design flow

The microcode for the accelerator, containing the configuration information for NetRAM and ControlFSM can be extracted using two different attempts: a bottom-up approach, which allows to design the virtual field architecture and interconnection-scheme and automatically produces the necessary microcode files for configuration of the accelerator. A screenshot of the bottom-up design environment can be seen in figure 7. On the other hand a software macro approach based on Mathworks Matlab has been developed to map various algorithms like FFT, IFFT, DCT, FIR, matrix-matrix and matrix-vector-multiplications.

For this macro-based approach first the properties of the underlying hardware accelerator architecture must be defined, which is shown in figure 8.

These properties contain the buffer size inside the OBI, which limits the number of temporary results.
inside the field, which can be stored by the OBI and therefore of course also limits the size of the virtual field. The second parameter is the bit-width of each operand applied to each of the three inputs of every PE.

Figure 9 shows the graphical user interface of the FFT mapping tool, which was chosen after the properties were already set. First the FFT length has to be specified (in this case a 128-point-FFT is chosen). On the right side the user can specify, which configuration files are created, NetRam for the NetRAM contents, FSMBehav for the configuration of the ControlFSM and ARMProgram, which outputs the configuration file for the software driver. When the button “Create” is pressed, the mapping procedure is started, the configurations are created, the chosen output is shown as text output in the window and by pressing “Save” they can all be saved and used for setting up the hardware.

IV. PROCESSING ELEMENT LIBRARY

At synthesis time the type of PE to be used can be chosen out of a library of different PEs specialized for a certain application including e.g.

- PE for matrix and vector algorithms, which very efficiently computes vector-matrix and matrix-matrix multiplications, butterfly and FIR-stage operations
- CORDIC PE for efficient trigonometric calculations and vector operations
- PE for efficient Galois-field operations used for example in the Reed-Solomon-Codec

V. DEMONSTRATOR PLATFORM

The described accelerator architecture was demonstrated on the ARM Integrator/AP AHB ASIC Development Platform, which is designed for hardware and software development of devices and systems based on ARM cores and the AMBA bus specification. This platform supports up to 5 ARM/FPGA modules and in our configuration one core module and four logic modules are fitted onto the development mainboard.

The mainboard consists of the connectors to fix the core and logic modules, Boot ROM, 512KB SSRAM, 32MB of Flash memory, a PCI host-bridge controller and a PCI/PCI bridge, keyboard and PS/2 connectors, PCI v2.1 slots and the possibility to expand the memory with SDRAM modules - all integrated on an ATX form factor mainboard.

The core module includes an ARM920T microprocessor core, a FPGA containing the AHB system bus bridge, a SDRAM controller, reset and interrupt controller, status and configuration RAM, up to 256MB SDRAM and 256KB local SSRAM as well as clock generators and a Multi-ICE debug connector.

The four logic modules we have mounted on the mainboard include a FPGA with large gatecount resources, 1MB SRAM, clock generators, general-purpose LEDs, switches, push button, connectors for Multi-ICE and logic analyzer and a full connection to the developer mainboards AMBA system and control busses. In the figure 10 the complete demonstrator platform can be seen. It consists out of one core module with the ARM920T integrated on it, four logic modules, which each can hold one hardware accelerator for the ARM processor and the mainboard onto which these modules are mounted.

The PCI slots contain three PCI peripheral cards: an Ethernet network, graphics and a sound controller. The ARM is running the linux operating system and device drivers allow the user to access the accelerator by copying files to a special device, which is the data to be computed, and reading the results from the device. Each logic module containing one accelerator can be separately addressed in a global address space and the driver supports to handle all four accelerators as separate devices.
VI. Conclusion

The proposed accelerator for telecommunication applications is parameterizable by choosing the PEs for one specific algorithm class and the amount of memory dedicated to the virtual fields temporary results at synthesis time. It is programmable at runtime concerning the virtual field connections via the NetRAM and execution control via ControlFSM configuration.

This very flexible approach and after-production reconfigurability allows the usage as an accelerator block in larger hybrid SoCs, to be a driver in software defined multistandard radios [4].

Due to limiting the number of parallel operating PEs and using intelligently addressed memory, algorithms, which can be mapped to large virtual fields of PEs can be executed very efficiently minimizing the overhead for inter-PE value routing, contributing to power and die area efficient computations.

References

Bluetooth demodulation algorithms and their performance

Roel Schiphorst, Fokke Hoeksema and Kees Slump

University of Twente, Department of Electrical Engineering,
Laboratory of Signals and Systems,
P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 2770 Fax: +31 53 489 1060
r.schiphorst@el.utwente.nl

Abstract—In our Software Defined Radio SDR project we aim at combining two different types of standards, Bluetooth and HiperLAN/2 on one common hardware platform. Goal of our project is to generate knowledge about designing the front end of an SDR system (from the antenna signal to the channel bit stream) where especially an approach from both analog and digital perspective is essential.

Bluetooth uses Gaussian Frequency Shift Keying (GFSK) as modulation technique. As QAM and GFSK signal can be demodulated by using the same type of algorithms, the outer receiver of HiperLAN/2 can be used as a Bluetooth demodulator.

So, whereas most commercial Bluetooth chips are low cost and inflexible, in our project flexibility and re-use of hardware is important. It is for that reason that a part of the channel selection and demodulation will be done in the digital domain.

The choice of the demodulation algorithm determines the channel selection requirements (better demodulation algorithms require less SNR). This work analyzes the performance of several Bluetooth (GFSK) demodulation and decision algorithms using single bit decision. A simulation model was built to measure the performance of these algorithms. In our simulation model, low-IF or zero-IF Bluetooth signals are sampled with 80 MHz to get a realistic setting. The synchronization word in the Bluetooth packet is used for bit synchronization.

To obtain a BER (Bit Error Rate) of 0.1%, specified by the Bluetooth standard, the best combination of demodulation and decision algorithm requires an SNR of about 15 dB.

Ongoing research focusses on several subjects: multibit decision, advanced demodulation algorithms (such as (adaptive) decision feedback equalizers) and bit-synchronization algorithms.

keywords: software-defined radio (SDR), Bluetooth, GFSK, demodulator.

I. INTRODUCTION

In our Software Defined Radio (SDR) project [1] we are focussing on the the front end of an SDR system (from antenna signal to the channel bit stream). Furthermore, our SDR design should be feasible within a few years, so power consumption is an important issue.

The vehicle of our project is a notebook to which we add the SDR functionality. This has three advantages. First, we can use the processing capabilities of the general purpose processor for digital signal processing. Second, in comparison to SDR for mobile phones, our demonstrator can consume much more power (in the order of 1 W). Third, a notebook is very suited for demonstration purposes. In order to generate knowledge about SDR systems we decided to implement (for our demonstrator) two standards on one common platform: HiperLAN/2 and Bluetooth.

Whereas Bluetooth is a low-cost, low data-rate standard ('1-dollar chip'), HiperLAN/2 is a high-speed wireless LAN standard (up to 54 Mbit/s). Consequently, as the application of the two standards is completely different, the technical parameters are also different. Bluetooth [2] uses the Gaussian Frequency Shift Keying (GFSK) modulation technique in the 2.4 GHz band and HiperLAN/2 [3] on the other hand uses Orthogonal Frequency Division Multiplexing (OFDM) in the 5 GHz band. In this modulation technique, basically \( n \) baseband Quadrature Amplitude Modulation (QAM) symbols are translated to time signals by performing an inverse FFT. The \( n \) complex modulated carriers created by the inverse FFT are up converted and transmitted. At the receiver the inverse process takes place. So the receiver can roughly be divided into two parts, an inner receiver and an outer receiver. The inner receiver performs the FFT and the outer receiver is a QAM demodulator.

Two functions in the digital part of the SDR front end can be distinguished [4]: channel selection and demodulation. As HiperLAN/2 is a high-speed wireless standard, it is expected that this standard will require more processing power than Bluetooth. So, whereas most commercial Bluetooth chips are low-cost and inflexible, in our project flexibility and re-use of hardware is important. Therefore...
we aim to use the (digital) HiperLAN/2 hardware for (a part of the) channel selection and demodulation of Bluetooth signals.

This paper will discuss several demodulation algorithms for Bluetooth GFSK signals. In order to evaluate the performance of the algorithms, a Bluetooth simulation model has been built. In this model, Bluetooth packets are generated and transmitted and demodulated by different demodulation algorithms. First this paper will discuss the Bluetooth GFSK modulation technique. The demodulator can be split up into two parts, the demodulator and the decision function. The demodulator converts the incoming GFSK signal into a Non-Return-to-Zero (NRZ) signal. As we want to use the HiperLAN/2 hardware (with a fast AD converter) for our Bluetooth receiver, we can use for the decision function more samples per symbol.

The Bluetooth standard requires a maximum Bit Error Rate (BER) of $10^{-3}$. So the performance of the different demodulation and decision algorithms is evaluated by looking at the required Signal to Noise Ratio (SNR) for a BER of $10^{-3}$. In literature [5] we found an SNR value of 21 dB for a BER of $10^{-3}$. By using more advanced algorithms, described in this paper, this value can be lowered significantly. The performance of these algorithms is discussed in the results section. Finally some conclusions will be drawn.

II. BLUETOOTH GFSK MODULATION

In normal continuous phase Frequency Shift Keying (FSK) a ‘0’ is represented by an harmonic signal with frequency $f_0$ and a ‘1’ by frequency $f_1$, both per interval of T s. Continuous FSK uses an Voltage-Controlled Oscillator (VCO) that is driven by the bit signal. In this implementation no phase shifts occur between bit transitions, which explains the name continuous phase FSK. However due to the binary nature of the input signal, fast frequency transitions occur and therefore results in a large bandwidth [6]. It is for that reason that GFSK uses a Gaussian pre-modulation filter.

Fig. 1 shows a GFSK modulator. First the bits are converted to signal elements. A ‘0’ is being represented by a signal with value -1 and a ‘1’ by a signal with value 1, each with a duration of T seconds. The filter output is then connected to an Voltage Controlled Oscillator (VCO) that translates the amplitude of the filtered bits into an frequency shift. In Fig. 2, the effect of the Gaussian filter is shown. The Gaussian filter reduces the bandwidth of the input signal of the VCO. This reduces also the bandwidth of the output signal and therefore GFSK is more spectrum efficient compared to normal Frequency Shift Keying (FSK) at the cost of an increased BER [7], although the noise is also reduced by the smaller band.

For FSK signals with a modulation index, $h = 0.3$ in an Additive White Gaussian Noise (AWGN) channel, the required SNR for a BER of 0.1% is about 12.5 dB [8]. The Gaussian pre-modulation filter, however, removes higher frequencies of the modulating signal (as can be seen in Fig. 2). This reduces the bandwidth of the VCO output signal but also reduces the bit energy which has a negative effect on the BER. In our literature search for GFSK demodulation we did not find any analytical relation between the BER and SNR reported. However most designers assume 21 dB [5].

In Bluetooth systems, the modulation index $h$ may vary between 0.28 and 0.35 [2]. The modulation index $h$ is defined as:

$$ h = \frac{2f_d}{R} = 2f_dT $$

where $f_d$ is the frequency deviation, $R$ the bitrate and $T$ the symbol time [9]. The frequency deviation ($f_d$) is the maximum frequency shift with respect to the carrier frequency, if a ‘0’ or ‘1’ is being transmitted.

For Bluetooth signals $f_d$ may vary between 0.140 and 0.175 MHz (according to Eq. (1)). Fig. 3(a) shows the power spectrum of a Bluetooth signal at 2 MHz with an $f_d = 0.175$ and Fig. 3(b) shows the power spectrum for $f_d = 0.140$. As expected, the power spectrum of Fig. 3(a) is a little wider and more flat than the one of Fig. 3(b). Visual inspection of both filters shows that the signal strength has dropped approximately 40 dB at the border of the channel (nominal channel width = 1 MHz [2]). Due to the relative small modulation index of Bluetooth GFSK, the signal energy is concentrated in a small band.

Fig. 4(a) shows the power spectrum of two neighboring channel (one with center frequency 2 MHz and the other at 3 MHz) for $f_d = 0.175$. As expected, visual inspection of the curve shows that both channels are very well
hardware (with a fast AD converter) for our Bluetooth re-
model has been built. In this model, Bluetooth pack-
verts the incoming GFSK signal into a Non-Return-to-
Zero (NRZ) signal. As we want to use the HiperLAN/2
ulator
performance of the algorithms, a Bluetooth simulation
Rate (BER) of
different demodulation algorithms. First this paper will
per symbol.

demodulator can be split up into two parts, the
for Bluetooth GFSK signals. In order to evaluate the
we aim to use the (digital) HiperLAN/2 hardware for (a
algorithm for Bluetooth signals. In literature we found an SNR value
GFSK signals. In literature [5] we found an SNR value
value of $T_s$. Continuous FSK uses an Voltage-Controlled
implementation no phase shifts occur between bit transi-
tions, which explains the name continuous phase FSK.

Fig. 1 shows a GFSK modulator. First the bits are con-
applied signal with value $-1$ and a '1' by a signal with value
is the frequency deviation, $f_d$, defined as:

The power spectrum of a Bluetooth signal at 2 MHz with an
may vary between 0.140 and
power spectrum of a Bluetooth signal at 2 MHz with an
may vary between 0.140 and

Fig. 3. Power spectrum of a GFSK signal

Fig. 4. Power spectrum of two neighboring GFSK channels

The demodulation part of digital communication signals
be divided into two parts:

- demodulator
- decision block

The demodulation function converts the incoming GFSK
signal into a NRZ signal. This can been seen as the digi-
tal equivalent of an analog demodulator. The second part,
the decision block determines which bit was transmitted.

III. DEMODULATION OF GFSK SIGNALS

The demodulation part of digital communication signals
can be divided into two parts:

- demodulator
- decision block

The demodulation function converts the incoming GFSK
signal into a NRZ signal. This can been seen as the digi-
tal equivalent of an analog demodulator. The second part,
the decision block determines which bit was transmitted.

A. Demodulation algorithms

As FM signals cannot be demodulated directly [10], sev-
eral types of indirect FSK demodulation methods exists
[11]:

- FM-to-AM conversion, also called FM discriminator
- Phase-shift discrimination
- Zero-crossing detection
- Frequency feedback

According to [12], the FM-to-AM conversion or FM dis-
mcriminator allows the implementation for low-cost radio
units, which is essential for Bluetooth units. It seems
therefore appropriate to research the "cheapest" demod-
ulator algorithm first. The second type of method we
investigated is the phase-shift discrimination method.
Both methods will be described shortly. The two other methods, zero-crossing detection and frequency feedback, have not been researched.

**A.1 FM discriminator.** Goal of the FM discriminator method is to translate a frequency shift into an amplitude change. A possible implementation is to use a time-delayed version of the incoming low Intermediate Frequency (IF) signal, see Fig. 5. This time-delayed signal is multiplied with the original, (not time-delayed) signal. The output of the FM-to-AM-conversion block with time delay \( \tau \) depends on phase \( \phi(\tau) \), which is the phase difference between the original and time-delayed signal:

\[
V_{out} = A(t) \cos(2\pi f + \theta) \cos(2\pi f + \theta + \phi(\tau))
\]

\[
= \frac{1}{2} A(t) \cos(\phi(\tau)) + \cos(4\pi f + 2\theta + \phi(\tau))[V]
\]

where \( A(t) \) is the amplitude, \( f \) the frequency of the incoming signal, \( \theta \) the initial phase and \( \phi(\tau) \) the phase shift caused by the time delay.

If a low-pass filter is used after the FM-to-AM-conversion block, the second term is assumed to be eliminated. So the output depends solely on \( \tau \). The time delay \( \tau \) is chosen in such a manner that it will produce for the central frequency, \( f_c \), a phase shift of \( \pi/2 \), so \( V_{out} = 0 \). If \( f_1 = f_c + f_d \) is being transmitted the phase shift will be more than \( \pi/2 \) and \( V_{out} \) will be negative. For \( f_0 = f_c - f_d \) the corresponding output will be positive. In order to retrieve the original bit sequence an inverter has to be placed after the FM discriminator.

For Bluetooth signals the modulation index may vary between 0.28 and 0.35 [2]. For our experiments we use the middle of the two values: \( h = 0.315 \). The frequency deviation \( f_d \) (Eq. (1)) equals 0.1575. As we want a phase shift of \( \pi/2 \) for \( f_c \), the time delay must be:

\[
\tau = \frac{1}{4f_c}
\]

Furthermore, for a digital implementation with sample frequency of 80 MHz, we want \( \tau \) to be an exact multiply of the sample time (in this case \( \frac{1}{80} \) μs). There are several possible values for the time delay \( \tau \). In general a larger time delay has a larger amplitude at the output. On the other hand a larger time delay degrades the performance of the FM discriminator because the relation between the signal and the time-delayed signal becomes less.

In this paper we chose the time delay to be 10 sample times (\( = 10 \times \frac{1}{80} = 0.125 \) μs). The center frequency should, in this case, be \( \frac{1}{1575} \) = 2 MHz.

**A.2 Phase-shift discrimination.** The phase-shift discrimination is, according to [10], a better demodulation method than the FM discriminator method because this method utilizes only the phase of the signal, the amplitude is not used. In the previous method however amplitude variations of the incoming GFSK signal are directly passed through the output (Eq. (2)). A limiter could be used to overcome this problem.

Fig. 6 shows a phase-shift discriminator. The first step is to down convert the incoming IF signal (Eq. (4)) to a complex Base Band (BB) signal (Eq. (5)).

\[
s(t) = A(t) \cos(2\pi f_c + \Delta\omega \int_{-\infty}^{t} m(\tau) d\tau) + n_1(t)
\]

where \( A(t) \) is the amplitude, \( f_c \) the carrier frequency, \( \Delta\omega \) the deviation constant, \( m(t) \) the Gaussian filtered message bit at time \( t \) and \( n(t) \) noise.

\[
s'(t) = A(t) \cos(\Delta\omega \int_{-\infty}^{t} m(\tau) d\tau) + n'(t)
\]

The two paths, In-phase (I) and Quadrature (Q) path, are low-passed filtered to eliminate the high frequency products caused by mixing. Then the phase is extracted by the arctan block (Eq. (6)). In order to retrieve the NRZ signal, the output of the arctan block has to be differentiated (Eq. (7)).

\[
s''(t) = \Delta\omega \int_{-\infty}^{t} m(\tau) d\tau + n''(t)
\]

\[
s'''(t) = m(t) + n'''(t)
\]

**B. Decision algorithms**

This section describes two decision algorithms that have been investigated:
- the integrate-and-dump (IaD) algorithm
- the decision feed-forward and feed-back (DFF-DFE) algorithm (non-adaptive)

The first algorithm, the integrate-and-dump algorithm sums all samples of one bit period and decides on the output of the sum whether the incoming bit is an '0' or '1'. So the algorithm does not take into account the influence of the Gaussian filter.

The second algorithm is more advanced and eliminates the influence of the Gaussian filter. For signals in the 800 MHz - 6 GHz band the maximum delay spread is 120 ns [13]. The Gaussian filter has an impulse response of about 3 bit times (= 3000 ns). So the dominant distortion is caused by the Gaussian filter. We assume that multipath fading can be neglected. Therefore we can use the shape of the Gaussian filter to calculate and correct the influence of the previous detected bit on the samples of the current bit. Furthermore we can estimate the value of the future bit and calculate its influence on the samples of the current bit. After correction we can use the shape of the Gaussian filter for a matched filter to achieve best performance. This algorithm can be represented by the following pseudo code:

```
EstimateNextBit();
for i = 1 to NrSamplesPerBit
    CorrectedCurrentSample = CurrentSample;
    - InfluencePreviousBit = InfluenceNextBit;
end;
CurrentBit = MatchedFilter(CurrentSamples);
```

IV. SIMULATION MODEL

This section discusses the Bluetooth simulation model we used to evaluate the different GFSK demodulation algorithms. Whereas our previous simulation model [14] was transmitting bits continuously, this model is packet based. Fig. 7 shows the top view of the simulation model.

The transmitter creates so-called DH5 packets that are the longest packets in Bluetooth [2] (with an payload of 2712 bits). Then, the packet is transmitted according the Bluetooth specs using a carrier frequency of 2 MHz.

To get realistic performances we assumed that the Bluetooth signal was sampled with a sample rate of 80 MHz. Noise is added and the distorted signal is filtered by a 512-taps Finite Impulse Response (FIR) bandpass filter which has a 1 MHz bandwidth with center frequency of 2 MHz. The relation between SNR and $\frac{E_b}{N_0}$ (bit energy divided by the power spectral density) is:

$$SNR = \frac{E_b R}{N_0 B} = \frac{E_b}{N_0}$$

where SNR is Signal-to-Noise Ratio, $E_b$ the bit energy, $N_0$ the power spectral density, $R$ the bit rate and $B$ the bandwidth.

So the $SNR$ is, in our case, equal to $\frac{E_b}{N_0}$ because the bandwidth $B$ is equal to $\frac{1}{T}$. In our simulation model we measured the bit energy by integrating the power spectral density function (of 800000 points) from 1.5 to 2.5 MHz. The measured energy was 0.249 $\mu$J. In our simulations we used this value for calculating the noise floor at a particular SNR.

In the receiver (Fig. 8) the signal is demodulated by the FM or phase-shift discriminator. After demodulation the signal is down-converted with an factor $n$ (in our simulations 80 and 10). So if $n = 80$, there is 1 sample per symbol and if $n = 10$, there are 8 samples per symbol. Finally, the decision block determines which bit was transmitted and the BER can be calculated. Bit synchronization is achieved by correlating the synchronization word of the packet with the incoming sample stream.

In our model the receiver receives 80 Mega Samples Per Second (MSPS) and after the demodulator this data stream has to be down sampled to 1 MSPS or 8 MSPS. However, down sampling is a time-variant process. In our simulation we choose an zero offset in down sampling. In the case of 8 MSPS this means that the samples 1, 11, 21, 31, . . . are taken of our simulation stream. This is a worst case scenario because the DFF-DFE algorithm assumes an symmetric sampled NRZ signal. For our 1 MSPS bitstream we used the samples 31, 111, 191, . . . (derived from the 8 MSPS data stream), which is also not the optimal sample moment (40, 120, 200, . . .).

V. RESULTS

To evaluate the different demodulation algorithms and decision algorithms we performed the following tests:

1 sample per symbol:
- FM discr. and the IaD algorithm
- FM discr. and the DFF-DFE algorithm
- Phase-shift discr. and the IaD algorithm
• Phase-shift discr. and the DFF-DFE algorithm

8 samples per symbol:
• FM discr. and the IaD algorithm
• FM discr. and the DFF-DFE algorithm
• Phase-shift discr. and the IaD algorithm
• Phase-shift discr. and the DFF-DFE algorithm

Fig. 9 shows the results of the performance of the different algorithms for 1 sample per symbol and Fig. 10 for 8 samples per symbol. For each \( \frac{E}{N_0} \) value we have simulated 199 packet (which is equal to 539688 bits).

In Fig. 9 we see that there is a negligible difference between the IaD algorithm and DFF-DFE algorithm. For both demodulation algorithms the IaD and DFF-DFE algorithm achieve the same performance. Probably the DFF-DFE algorithm has too little information to perform better than a simple IaD algorithm. On the other hand we see that the choice between the two demodulation algorithms matters. The performance of the phase-shift discriminator algorithm is compared with the FM discriminator algorithm about 1.5 dB better. To achieve a BER of 0.1% the phase shift discriminator needs about 15.2 dB and the FM discriminator needs about 16.7 dB.

In Fig. 10 the results are quite different. In this picture, for 8 samples per bit, the choice of the decision algorithm has most influence on the performance. The IaD algorithm performs about 1.5 dB worse compared with the DFF-DFE algorithm. The Inter-Symbol Interference (ISI) caused by the Gaussian filter has especially, at the samples at the border of the symbol period, large influence. So it is assumed that the performance of the IaD algorithm is affected by this ISI. For both demodulation algorithms, the performance of the IaD algorithm is about equal to the performance of the FM discriminator in Fig. 9. However the performance of the DFF-DFE algorithm is much better. The best performance is achieved with the phase-shift discriminator algorithm, about 14.8 dB is required for a BER of 0.1%. Especially the performance of the FM discriminator algorithm with DFF-DFE algorithm has improved significantly. So it looks like the DFF-DFE algorithm can compensate for the performance loss caused by the FM discriminator algorithm.

VI. CONCLUSIONS

In this paper we have analyzed two implementations of an FSK demodulation algorithm, the FM discriminator algorithm and the phase-shift discriminator algorithm, for the use in Bluetooth systems. Furthermore we have analyzed two decision algorithms, an integrate-and-dump algorithm and a non-adaptive decision feed-forward and feed-back (DFF-DFE) algorithm. The performance, achieved by all of these algorithms, is much better than the 21 dB found in literature [5].

Two scenarios were investigated, a scenario in which 1 sample per symbol was used and another in which 8 samples per symbol were used. The best combination of demodulation and decision algorithms requires a SNR of about 14.8 dB for a BER of \( 10^{-3} \). The best performance is achieved when the phase-shift discriminator algorithm is used in combination with the DFF-DFE algorithm with 8 samples per bit. A performance loss of about 0.5 dB occurs when only 1 sample per bit is used.

The FM discriminator algorithm on the other hand has in the case of 1 sample per bit, a 1.5 dB performance loss compared with the phase shift discriminator. In this case about 16.7 dB is required for a BER of 0.1%. However in the case of 8 samples per bit and the DFF-DFE algorithm, the performance difference between the FM and phase-shift discriminator has almost vanished (about 0.1 dB difference left).

In the case of 8 samples per bit the integrate-and-dump algorithm performs less compared with the 1 sample per bit scenario. It is assumed that the performance of the integrate-and-dump algorithm degrades, because the Gaussian filter alters the shape of the NRZ signal significantly (especially the samples at the border of the symbol period).

Not only performance counts, the complexity of the algorithm is also an important issue. The phase-shift discriminator, for example, is more complex, due to the arctan function than the FM discriminator. For the decision algorithms the same applies, the DFF-DFE algorithm is more complex than the integrate-and-dump algorithm. Furthermore more samples per symbol require also more processing power. So there exists a trade-off between demodulation/decision algorithms and the channel selection. Simple Bluetooth demodulators will require more SNR i.e. channel selection than more complex demodulators.

For FSK signals with a modulation index, \( h = 0.3 \), in an Additive White Gaussian Noise (AWGN) channel, the required SNR for a BER of 0.1% is about 12.5 dB [8] (plotted in Fig. 9 and Fig. 10). In this paper the best combination of demodulation and decision algorithm requires about 14.8 dB. It is assumed that the Gaussian filter causes this degradation in performance. For example the Gaussian filter lowers the output value of for example the ‘1’ in the ‘010’-sequence. So the amplitude of the output signal becomes lower. However the bit energy remains the same, the Gaussian filter only spreads the bit energy over multiple bit times. So the performance loss of the Gaussian filter could be compensated if advanced algorithms are used, such as multi-bit detection. Also it is expected that adaptive Decision Feedback Equalizers [15] and matched filters with a tapped delay line [16] could perform better than the algorithms investigated in this paper.

For further research we investigate the following questions: What is the influence of the pre-detection filter? What is the performance gain, if demodulating algo-
gorithms are used, which use more than 1 symbol period for making a decision which symbol has been received? What is the performance of adaptive demodulation algorithms?

ACKNOWLEDGMENTS

This research is carried out in cooperation with the Integrated Circuit Design Laboratory of the University of Twente. This research is supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW.

REFERENCES

[14] R. Schiphorst, F.W. Hoeksema, and C.H. Slump. Channel selection requirements for Bluetooth re-
Fig. 9. Performance of the different algorithms for 1 sample per bit

Fig. 10. Performance of the different algorithms for 8 samples per bit
BENEFITS AND LIMITS OF PARAMETERIZED CHANNEL CODING
FOR SOFTWARE RADIO

Arnulf-Ragnar Rhiemeier

Institut für Nachrichtentechnik
Universität Karlsruhe (TH), Germany
arm.rhiemeier@int.uni-karlsruhe.de

ABSTRACT — In the first part of this paper a parameterized signal processing structure for channel coding is introduced. It is capable of fulfilling the coding requirements of several wireless communication systems - GSM, TETRA, TETRAPOL, UTRA FDD and UTRA TDD - and different transmission modes therein. The channel encoder solution is based on the established approach of parameterized software radio. A theoretical review and the example of an RSC encoder serve best to illustrate this approach. The generation of both turbo and convolutional codes are addressed. A particular tail-biting code, employed in TETRAPOL, is discussed in more detail, notably with respect to its block MLSE decoder. Design parameters for an efficient decoder implementation are derived from simulations of a TETRAPOL voice connection. In the final part of this paper the flexibility of parameterized decoding is reassessed in view of the large variety of coding schemes and of practical implementation issues. Benefits and limits of the parameterized approach are identified. An outlook on the continuing line of research concludes the paper.

1. BACKGROUND

Today’s mobile communication standards like GSM reflect the successful transition from analog to all-digital voice communication. The prospering of short message services in GSM together with the popularity of Internet connectivity have drawn much attention to wireless data communications in recent years, so next generation terminal and network equipment is being designed to respond to these emerging needs of private users.

Meanwhile, professional users would like to catch up with the development. In fact, advantages of the digital world would have been appreciated in the professional mobile radio (PMR) sector earlier than in the private, but surprisingly enough digital systems came up only recently, and the most prominent competing systems in Europe are TETRA [1] and TETRAPOL [2]. Both offer advanced features such as end-to-end encryption, group call and direct mode operation between handsets. Also, regular network operators try to invade the PMR sector [3][4] by offering their so-called virtual private networks (VPN).

Under these circumstances interoperability between 2G, PMR and prospective 3G systems is more than desirable. The work presented in this paper is based on the understanding of software radio as an enabling technology for manufacturing multi-mode terminals. A study of 2G and 3G standards had already been carried out [5] before. However, GSM has entered its phase 2+ in the meantime, and the two UTRA modes have been harmonized in 1999 [6]. The inclusion of PMR systems is a completely new technical thread.

2. REVIEW

Parameterized software radio claims to identify common aspects of mobile communication standards and arranges its signal processing procedures in a way to cover all standards under consideration while using a single processing chain. This chain consists of a series of signal processing blocks, each containing a piece of code by which to accomplish a certain function. Early during the design process this function may have been identified as universal to all considered standards. Should the function not be represented in a specific standard or transmission mode input data is being passed through the block unaltered, either explicitly by a code part realizing a simple write-through operation or implicitly by giving the block a neutral kind of parameterization. In the latter case the block executes its generic code, yet not affecting the original input signal.

3. CHANNEL CODING

Under the conditions of parameterized software radio one important assumption is that all known wireless standards use more or less the same kind of
base band signal processing, possibly under different names and abbreviations, but regardless of inventor, company, or other interests. This is because the limiting factor in the transmission chain is the mobile channel with its adverse properties. Therefore we find approximately the same kind of channel coding measures in GSM, TETRA, TETRAPOL and UTRA. Additionally, the UTRA modes have been harmonized, and there is no difference between FDD and TDD, for example, in turbo coding.

Any discussion of the UTRA physical layer goes well beyond the scope of this paper. Spreading and despreading, for example, belong to the most prominent issues when integrating both UTRA FDD and TDD [7] with GSM and PMR systems. Turbo coding, however, has an impact on the data link layer, so this aspect of UTRA will naturally appear again in later sections of this work.

Concatenated coding and unequal error protection are well known from GSM [8]. Usually, a block code is used as an outer code to detect transmission errors in a class 1a info bit block. In addition, an error-correcting inner code protects the entire class 1. Class 2 bits remain unencoded. TETRA does approximately the same, only it protects its class 1a info bits by two block codes, a CRC plus parity check code. Coding of a TETRA full-rate speech channel includes a fixed rate matching (RM) scheme, whereas rate matching in UTRA may result in almost arbitrary puncturing or repetition, according to [9][10].

Figure 1 shows our channel coding structure for GSM, TETRA, TETRAPOL, and UTRA modes. We perceive three branches: two identical branches A and B for flexible concatenated coding, and a lower branch C for unencoded classes. Multiplexers (MUX) and demultiplexers (DMUX) distribute and recollect blocks of data. The A and B branches are connected between block (CRC, PAR) and turbo coding (TURBO), a feature to support high data rate frame coding in TETRAPOL [2]. Every single processing block contains a generic code, conforming to the paradigm described in section 2.

It is very likely that any one mode out of a set of supported transmission modes does not use all coding blocks of figure 1. In our implementation, if an entire branch remains unused, all its constituting processing functions are switched to an idle mode. Thus, they are not called during execution of the coding procedure, and they do not waste processing power, neither in the simulation nor in a potential implementation. This marks an essential difference to the original concept of section 2. The idle mode is a new functional features of the structure in figure 1. Understanding its implications paves the way for modular software radio, to be seen as a supplement to the strictly parameterized version.

Before continuing with this discussion let us go deeper into channel coding and analyze an RSC encoder, which had originally been programmed to generate UTRA turbo codes.

Figure 2 shows our generic RSC encoder. At a first glance it is just an ordinary RSC encoder, but indeed it is implemented as a generic C code where memory length, the number of feedforward generator polynomials, their actual values and the feedback polynomial are taken as parameters. Furthermore, the feedback generator polynomial as well as the systematic feedforward path may be rendered idle. Upon initialization the shift register memory is filled with init bits which are taken from an external port, much in the same way as the info bits.

When rendering both the systematic and feedback path idle the structure incorporates an ordinary convolutional encoder. Uncoded throughput can be
achieved when switching all of the above to idle mode except for the systematic feedforward path.

![Diagram of UTRA turbo encoder]

Figure 3: UTRA turbo encoder

Figure 3 shows how RSC encoders are employed for UTRA turbo coding. The modular approach continues to have an impact here as well: the UTRA turbo encoder transforms into a efficient convolutional encoder as soon as we render the entire lower data path idle, including the interleaver (INTL) and the lower RSC encoder.

4. TAIL-BITING CODES

If we draw the RSC encoder init bits from the end of an info bit block instead of initializing in the zero state we recognize that the structure of figure 2 effectively generates a tail-biting code. Figure 4 shows the TETRAPOL encoder and our ring interpretation of the info bit block. It is essential to notice that the initialization and termination states are identical, yet unknown. Keeping this representation in mind let us now turn to the decoder design for tail-biting codes.

![Diagram of TETRAPOL tail-biting encoder]

Figure 4: TETRAPOL tail-biting encoder

5. BLOCK MLSE DECODER

A first decoding idea was to apply a trellis search to the received code block and modify the existing block MLSE decoder so as to prefer only trellis paths with identical initialization and termination state. However, modifying an existing decoder plus introducing another piece of code to cover just another particularity of a new communication standard is not only time-consuming and error-prone, but it is also a step away from the described software radio paradigm, which postulates to identify and implement generic rather than specialized communication functions. Therefore we have reused the existing MLSE decoder in a different way.

Figure 4 shows that the info block structure is circular, and so is the resulting code block. We simply repeat a received code block several times, thus creating a new, synthetic code frame. Then we apply block MLSE decoding to this code frame but consider only part of the decoder bit decisions for any subsequent receiver signal processing. Figure 5 illustrates this decoding idea for the example of a code rate $R = 0.5$ tail-biting code.

![Diagram of raw decoding scheme]

Figure 5: raw decoding scheme

The different stages are: 1) code block input, 2) $N$-fold repetition of received code block, 3) MLSE decoding of the synthetic code frame, 4) extraction of one particular info block, 5) info block output. The key idea in the suggested decoding scheme is that path metrics accumulate during a trellis search over the entire synthetic code frame. As a consequence, the larger the considered chainback memory the better the correct info bit path is distinguished from others.

Figure 6 shows simulation results from a TETRA-POL voice transmission over an AWGN channel and decoding according to figure 5. The three curves represent BER estimates, plotted over the $E_b/N_0$ available at the receive filter sampler output.

The worst BER performance is observed if the received code block is not repeated at all, because the the decoder does not take into account any knowledge about initialization and termination. The same curve results if $N$ copies of the original code block make up a synthetic code frame, but bit decisions are taken from the latest decoder output block.
Figure 6: TETRAPOL, class 1 residual BER

Much better BER performance results from bit decisions taken from the earliest decoder output block. The best performance is achieved from any intermediate output block, regardless of N, if N ≥ 3. The improvement between taking the latest block and only the second block is enormous. Therefore we have studied the degree of improvement, depending on a tail offset which is smaller than the length of one info block. Also, we can try to minimize the trellis search effort by shortening the synthetic code frame. Its length need not necessarily be an integer multiple of the code block length. Figure 7 shows the corresponding decoding scheme. The length of the tail offset, LTO, and the length of the synthetic code frame, LCF, are the two design parameters of the modified scheme.

1) code block input
2) repetition
3) MLSE decoding
4a) extraction of info block
4b) ordering of info block
5) info block output

Figure 7: modified decoding scheme

Stage 4 is now split into two parts, 4a to extract a block of info bits from the decoder output, and 4b to shift this block cyclically, thus restoring the original order of info bits. Figure 8 shows the BER improvement as the tail offset LTO increases.

Four curves are plotted for different SNR. It is remarkable that there is no more improvement as soon as LTO reaches 14 bits. Taking into account that the TETRAPOL class 1 block is 26 bits we may develop this result into a general design rule: choose...

\[ L_{TO} = \frac{L_{IB}}{2} + 1 \]  \hspace{1cm} (1)

...where LIB is the length of the info block.

The curves of figure 8 have been obtained with a constant LCF = 4 · LCB. Now we can reduce the decoding effort by minimizing LCF. Figure 9 shows the achievable BERs at different SNR, depending on the synthetic code frame length.

The abscissa spans the range of \( L_{CF,min} = 80 \) to \( L_{CF,max} = 104 \), or in general terms:

\[ L_{CF,min} = \frac{L_{TO} + L_{IB}}{R} \]  \hspace{1cm} (2)

\[ L_{CF,max} = \frac{2 \cdot L_{IB}}{R} \]  \hspace{1cm} (3)
Beyond \( L_{CF, max} \) there is no further improvement in BER. The curves of figure 9 indicate that the decoding loss is minimal as \( L_{CF} \) approaches \( L_{CF, min} \). In fact, the loss is always less than or equal to the difference between the two lower curves of figure 6.

To sum up the experimental part let us propose our design rule for the two parameters \( L_TO \) and \( L_CF \):

\[
L_{TO} = \frac{L_{IB}}{2} + 1 \quad (4)
\]

\[
L_{CF} \leq L_{CF, max} \quad (5)
\]

This way we have successfully included the decoding of the TETRAPOL tail-biting code into our software radio receiver, in fact without reengineering the original MLSE decoder. The following part of the paper takes us back to the discussion of section 3 and to the decoding of all possibilities originating from the structure of figure 3.

6. DISCUSSION

We have seen that convolutional codes for GSM, TETRA and TETRAPOL – despite of peculiarities such as tail-biting – do represent a subset of turbo codes. Regarding the issue of a common decoder we can argue that a suitable MAP algorithm, designed for UTRA turbo decoding, can as well be used for decoding convolutional codes [11]. The idea has been proposed and implemented before [12], however, for zero-state terminating codes only. Now, there are some arguments against continuing with this generalizing approach and against modifying a given code over and over again:

1. When using a generalized turbo version instead of a more specialized Viterbi decoder the gain is only minimal.
2. When using a generalized turbo version the computational expense for processing a convolutional code is disproportionate to the decoding gain.
3. Writing a generalized turbo version entails awkward reengineering of existing software.
4. Every time there is a major change in the software radio system (introduction of a novel coding scheme, addition of a new standard etc.) the reengineering situation would occur again.

Therefore we have solved the channel decoding problem in a rather modular way (see figure 10). The input and output switches select one of the signal processing functions while the other one is forced into idle mode. The activated decoder may be well-adapted to the required task, so its operation results in optimum runtime. Switching between modules as well as signal path redirection is no issue, since we are in software radio anyway. Admittedly, some memory is wasted for keeping idle code in the device. However, the required amount of program memory is small compared to data memory, and memory comes at virtually no cost today.

7. GENERALIZATION

When generalizing these considerations we may conclude in the following way: parameterization of base band signal processing functions is a strong approach to software radio, because it features fast reconfiguration capability and leads to practical implementations. Its benefits have been shown in a number of publications (see [12] and references therein), and it has allowed to include tail-biting codes, as described in this paper.

Parameterization allows to build communication systems with flexible components, but flexibility is restricted to a predefined set of transmission modes. As soon as we leave this set we may encounter a new type of problem – a new code class, for example – so that existing signal processing functions are likely to fail, whether parameterized or not. In order to enhance legacy functions their source code must be accessible, and often enough it is not because it is intellectual property of a company or because the original software development team has been split up, or new tools have been introduced in the meantime.

Given these practical design issues we suggest to consider modular software radio as a supplement to parameterization. We have exemplified the modular approach in this paper: in the channel coding structure with its idle branches, and in the switchable decoder of figure 10.

8. CONCLUSION

In this paper we have thoroughly discussed parameterized channel coding for software radio, and a tail-biting code for TETRAPOL in particular. An
existing parameterized Viterbi algorithm has been used for decoding both zero-state terminated and tail-biting codes, in fact without modifying the original trellis search function. Reasonable choices for two parameters have been derived from experimental data. Finally, we have commented on options for the design of a generalized decoder for both turbo and convolutional codes. Our generalization leads to the conclusion that sooner or later in the lifetime of a software radio parameterization has to be complemented by a modular approach. We have successfully employed the modular approach in our simulation setup.

To a certain extent parameterized software radio implies that all existing parts of a structure be used for one purpose or another [12]. All signal processing functions are assumed to execute some generic code part, come what may.

We believe that we must drop this assumption for our future work on software radio. As a consequence, modular software radio will potentially link multi-thread processing and scheduling theory for real-time systems to communications.

9. ACKNOWLEDGEMENTS

The author would like to thank Piotr Rykaczewski for his help in the practical implementation of the modular signal processing structures depicted in figures 2.3 and 10. Furthermore, Ralf Machauer and Henrik Schober deserve credit for many fruitful discussions on decoders and algorithms.

10. REFERENCES

[9] “ETSI TS 125 212: UMTS; Multiplexing and channel coding (FDD).” V4.0.0 (2000-12).
[10] “ETSI TS 125 222: UMTS; Multiplexing and channel coding (TDD).” V4.0.0 (2001-03).
Resource Management in Software Radio Architectures

Gyula Rábai, Sándor Imre, József Kovács, Péter Kacsuk

Parallel and Distributed Systems Laboratory
MTA SZTAKI, Computer and Automation Research Institute
H1518 Budapest, P.O. Box 63, Hungary
rabai@hit.bme.hu, imre@hit.bme.hu, kacsuk@sztaki.hu, smith@sztaki.hu

Abstract

As the number of radio services are growing, and the complexity and sophistication of them advances, the demand for resource management also greatens. In the proposed presentation the discussion is about resource management at the physical layer in reconfigurable radio (software radio) architectures. The described architecture was developed in the frames of the European 5th Framework CAST (Configurable Radio with Advances Software Technology) project. The presentation points out the latest achievements and results concerning this work.

In the CAST project resource management is handled by a resource controller software module. The task of this module is to dynamically set up and maintain processing chains in order to provide optimal performance. The managed processing chains consist of functions, which are placed onto reconfigurable hardware objects, such as DSP’s and FPGA’s. The functions are connected to each other in a way where the output of a function is fed as input to the next function in the chain.

The functions, hardware objects and processing chains are handled at an abstract level by the resource controller. They are represented by JAVA classes which form the generic building blocks of a JAVA abstraction layer. Since the abstraction layer is capable of describing different hardware objects and different configurations, the developed resource controller architecture and the proposed management algorithms are fit for different software radio systems. It is also ready for future reconfigurable hardware devices.

The resource controller software itself is a collection of algorithms, methods and intelligence which help to build and maintain a suitable configuration for the hardware. The tasks taken care of include allocation of hardware resources, building and braking up processing chains and handling of active and passive configuration. During the reconfiguration process the decisions are made based on different aspects which can vary from system to system. The functions used by the resource controller software work with the classes provided by the JAVA abstraction layer and are implemented in JAVA.

The introduced approach to resource management provides a possible solution for those who would like to integrate reconfiguration capability to their system at the hardware layer.

I. Introduction

The term Software Radio [1],[2] defines a system, whose behavior can be redefined by software. Such a system often serves as a platform for multi channel data processing applications. These applications are used for communication purposes and are called services. Each service consists of one or more processing chains which operate simultaneously. These chains are built from processing functions for example: Voice encoder, Convolutional coder, DES encryptor, etc. The processing functions in a chain are connected to each other sequentially in a way where the output of a function is fed as input to the next function in the chain. In order for these functions to operate they are configured on different hardware devices such as DSPs, ASICs, ASSPs and FPGAs.

The task of resource management is to dynamically set up processing chains while maintaining flexible and efficient hardware configuration. This process involves intelligent mechanisms for locating suitable hardware capacity, choosing a configuration out of possible choices and breaking up configured, but not used services. The desired functionality can be achieved by a resource controller software module, often referred to as IRC (Intelligent Reconfiguration Controller) [1] The main goal of this software module is to manage the underlying reconfigurable subsystems in
order to provide high utilization over the physical resources.

This paper presents a possible implementation of the resource controller software module. This implementation is introduced using object-oriented approach in JAVA and it uses the functionality of an ORACLE database server through a JDBC connection.

II. Architecture

While constructing the resource controller, we have to deal with a physical layer, that consist of reconfigurable hardware devices such as DSPs, ASICs, ASSPs and FPGAs. These devices can be configured for different tasks and they can operate simultaneously. One device can often run more than one application (function) at the same time. Constructing a good configuration for this hardware is hard for two reasons. First, the different devices are suitable for different tasks. For example programmable ASICs are appropriate for front/end filtering tasks and DSPs could be used for channel modem and baseband signal processing tasks. [3] Second, in different scenarios different configuration algorithms should be used to satisfy the different aspects.

To be able to handle these issues and to develop a lasting architecture for the resource controller we have to treat the hardware at an abstract level. This means, we assume that a universal configurable hardware is available, which provides a way for all configured functions to be connected regardless of their location in the hardware. We assume that the sequence of the configuration requests towards the hardware is irrelevant, thus the same configuration is provided if we switch the order in which we place the functions on the hardware.

This approach can be applied with 3 layers of abstraction. (Fig.1.)

The first layer is the „Hardware Abstraction Layer”, the second is the „Function Abstraction Layer” and the third is the chain abstraction layer.

The hardware abstraction layer consists of a set of JAVA classes and a database table. The different JAVA classes represent the different reconfigurable devices and the records in the database table represent the available resources in the system. When a resource is needed a new instance of the corresponding JAVA class is instantiated and it is bound to a hardware address stored in the database table.

The hardware resource and its configuration can be managed through the appropriate instance of the JAVA class. When the instance is destroyed, the resource looses its configuration and is placed back to the list of free resources.

The function abstraction layer implements the set of building blocks needed to construct processing chains. The functions operate on allocated hardware objects. When a function is placed on a hardware object we are talking about configured function. The method of placing a function on a hardware objects is often referred to as configuring a function on a hardware objects. Each function is represented by a JAVA class. This class has standard methods used for configuration, activation, deactivation and information retrieval purposes. When the function class is initiated it is not yet configured on a hardware device. In fact a function can sometimes be configured on different types of hardware resources, because it has more than one binary implementations. The configuration can be done by calling an appropriate method of the hardware class and passing the unconfigured instance of the function class to this method.

It is important to mention that a function should always be placed on a device, before it is used. Unless a function is configured on a device it cannot be connected to other functions and cannot form the part of a processing chain. Processing chains contain a set of configured functions attached to each other.

The chain abstraction layer consists of one JAVA class, called the chain class, and a few database tables. The class is capable of handling all the different chain types that are used in the different services. When a new chain is built a new instance of the class is created by calling its constructor method. The
constructor method requires a database identifier that identifies the chain type that should be constructed. Based on this identifier the resource manager determines in which order, and what type of functions are needed to construct the chain.

The chain construction is based on information stored in the database. An instantiated chain is not yet installed on the hardware. The installation process can be started by calling a method of the chain class. This method uses appropriate procedures and pre-defined resource allocation algorithms to allocate hardware resources. After the necessary resources are allocated the installation method places the necessary functions on them and connects the functions to each other in the correct order. If this is all done successfully we say the chain is installed and ready for action.

A configured chain provides methods for maintaining active and passive chain status, which can be used for shadow switching. Shadow switching (Fig.2.) is an efficient technique to burst performance. In the software world it is called object pooling. The main idea behind it is that it is possible to activate and deactivate a chain without losing its configuration. This is very useful, because configuration is a costly procedure, and if the same chain is needed sequentially a lot of reconfiguration can be saved.

**Shadow switching**

![Shadow switching diagram](image)

**Figure 2. Shadow switching**

III. RSC Implementation

As you can see the introduced RSC architecture consists of a set of JAVA classes and a few database tables.

The database server is used for storing information that describe the system. The information retrieved from the database (available hardware addresses and chain descriptions) are not maintained by the resource controller. The database is used through a standard JDBC connection, so an appropriate server implementation can be chosen. For a mobile handset a small embedded database server would do the job and for a base station a massive Oracle server could provide the best results.

The abstraction layer classes provide the functionality of the resource controller. The implementation can be done in the latest JAVA development environment, but other development languages can also be used. A JAVA implementation would naturally provide a JAVA interface for the system that would use the RSC. Other implementations could provide the same functionality with different interfaces for example CORBA.

The implementation of these abstraction layers would bring flexible solutions that are ready for the different hardware resources and their evolving features. The flexibility can be increased by integrating the, intelligent decision making and resource management algorithms into exchangeable modules called plug-ins. These plug-ins have a well defined interface which allows them to be replaced as circumstances changes.

Each resource manager plug-in contains a resource management algorithm. When the RSC is deployed, the appropriate algorithm can be selected and installed. The selection can be made according to various aspects described later in this article. It is probable that a carefully chosen algorithm would bring satisfying results.

![Algorithm manager plugin](image)

**Figure 3. RSC plugins**

In some scenarios, such as under changing traffic patterns or adapting to different service environments it might be vital to be able to change optimization algorithms dynamically. This can be achieved by using an algorithm manager plug-in (Fig. 3.). The algorithm
manager plug-in would have the same interface as an optimization algorithm plug-in, with the difference that it would serve as a gateway between the RSC and several real algorithms. This algorithm manager plug-in would be able to monitor the environment and if the circumstances would make it necessary it would be capable of switching to a better algorithm.

Keeping these strategies in mind a well performing resource manager can be constructed. The plug-able algorithm technology and the JDBC database interface helps to keep vital parts of the RSC fit for the job.

IV. Algorithm selection

Once the resource controller is built, there are still challenges that should be faced concerning the different plug-able algorithms. Since these algorithms control the behavior of the resource controller, selecting the right one is a very important issue. A bad selection could cause problems while a good one could boost system performance or reach the desired results.

One might think that the problem of selecting a suitable algorithm could be avoided if a good all-purpose algorithm is introduced. Unfortunately creating a resource management algorithm that fits all needs is impossible. Aspects can vary from environment to environment and sometimes what is an advantage in one place can be a disadvantage in another. What we should do is select a set of aspects that are important in our environment and choose or create an algorithm that best satisfies these aspects.

For example if the resource controller is deployed on a mobile handset, the low power consumption would be of greater importance than fault tolerance, while on a base station this would be the other way around, so on a mobile handset we would choose the hottest first algorithm and on a base station we would use the coldest first algorithm. These algorithms are described later in this article.

Let’s take a look at some of the aspects that could be taken into account and the algorithms that would satisfy these aspects.

V. Aspects

Maximum lifetime could be an important aspect in the system. If resource allocation leads to wear and tear, the frequently allocated resources could experience a lot of wear and tear and could fail to operate sooner than expected. In this case an algorithm for even utilization is needed. This algorithm would extend the lifetime of the system, since it would make sure that resources are used at pretty much the same frequency.

Even utilization is also a key issue in Fault tolerance. If we spread our services evenly over a large field of hardware, when part of the hardware fails, only a few services will be effected. There are two algorithms that accomplishes even utilization. One of them is called Coldest First [3], another is called Blind Monkey.

While the Coldest First or Random Pick algorithms create even utilization they can waste valuable resources. If some tasks can only be served by these valuable resources the performance of the system is limited because of a badly chosen algorithm. In these cases Maximum utilization is a very important aspect. Maximum utilization should be taken into account when only a limited number of resources are available and a large number of services should be served. In this case a number should be assigned to the configurable devices that would describe the usefulness of the device. When a new chain needs to be installed the least valuable devices should be reserved. A way to determine the number that describes how valuable a resource is, is introduced in the Maximum Capacity algorithm.

In mobile handsets Power consumption is one of the most important aspects. If a resource is not in use it can be powered down, thus saving energy. Since a processor can often serve more than one tasks, when new functionality is needed it can be placed onto resources already powered up. In these cases the main goal of resource management is to only utilize the minimum number of hardware. An algorithm that would do the job is called Hottest First [3]. It is described in the following section as well as the other algorithms already mentioned.

VI. Algorithms

The Maximum Capacity algorithm requires a lot of calculation. The goal of this algorithm is to find the least valuable hardware object in the system. We consider a device more valuable if it is suitable for more than one task, we need it often and the system is lack of this resource. All these aspects are summed up in one
number. This number is recalculated for each resource after every configuration. If the least valuable resource is used all the time, the probability that more services can be served is higher.

Hottest First is a LIFO type allocation. It’s goal is to allocate the resource last released on the next resource allocation request. To implement this algorithm in the resource controller, you either need timestamps or a LIFO stack. If you use a stack, all the free resources are kept in stack. An allocation request is serviced by popping a free resource from the stack. When a resource is freed, it is pushed back on the top of the stack.

Coldest First is a FIFO type allocation. The resource controller (RSC) keeps the free resources in a queue or maintains a timestamp for each resource. When a resource is needed the resource not allocated for maximum time is selected first. If the implementation is done with a queue a resource allocation request is serviced by removing a resource from the head of the queue. A freed resource is returned to the free list by adding it to the tail of the queue.

[3]

Another inconvenience can come if more then one aspects should be taken into account when the optimization algorithm is chosen. in this case more than one algorithm should be used at the same time, which means that a lot of calculation is needed to determine which resources are best fit. At this point the resource management process itself can be too lengthy, what could lead to a lot overhead.

VII. Bottlenecks

While construction these algorithms and the above architecture, we assumed that the resource controller software has a way of operating without interfering with the configurable hardware resources. This can be achieved by placing the functionality of the RSC on a separate service processor. If such a processor is not available then distributed resource management should be used. Discussing distributed resource management is beyond the scope of this article. Finding a good place for the resource manager could be a hard task.

While construction these algorithms and the above architecture, we assumed that the resource controller software has a way of operating without interfering with the configurable hardware resources. This can be achieved by placing the functionality of the RSC on a separate service processor. If such a processor is not available then distributed resource management should be used. Discussing distributed resource management is beyond the scope of this article. Finding a good place for the resource manager could be a hard task.

VII. Bottlenecks

While construction these algorithms and the above architecture, we assumed that the resource controller software has a way of operating without interfering with the configurable hardware resources. This can be achieved by placing the functionality of the RSC on a separate service processor. If such a processor is not available then distributed resource management should be used. Discussing distributed resource management is beyond the scope of this article. Finding a good place for the resource manager could be a hard task.

VII. Bottlenecks

While construction these algorithms and the above architecture, we assumed that the resource controller software has a way of operating without interfering with the configurable hardware resources. This can be achieved by placing the functionality of the RSC on a separate service processor. If such a processor is not available then distributed resource management should be used. Discussing distributed resource management is beyond the scope of this article. Finding a good place for the resource manager could be a hard task.

VII. Bottlenecks

While construction these algorithms and the above architecture, we assumed that the resource controller software has a way of operating without interfering with the configurable hardware resources. This can be achieved by placing the functionality of the RSC on a separate service processor. If such a processor is not available then distributed resource management should be used. Discussing distributed resource management is beyond the scope of this article. Finding a good place for the resource manager could be a hard task.

VIII. Summary

This paper presents a resource controller architecture and resource management algorithms. The introduced technology improves hardware utilization and overall performance and is capable of handling future systems. A prototyping system is currently developed.

It is important to mention that software radio services most of the time have real time functionality. According to this all parts of the system must be defined with exact timing specification for each stage. The introduced optimization algorithms might not guarantee results in an exact amount of time, thus might not be suitable for all real time purposes.

Acknowledgement

This document is based on work carried out in the EU-sponsored collaborative research project CAST. Nevertheless, only the authors are responsible for the views expressed here.

References


RECONFIGURATION IN SOFTWARE RADIO SYSTEMS

Apostolos A. Kountouris, Christophe Moy

Mitsubishi Electric, ITE-TCL
Rennes, FRANCE

Abstract—Reconfiguration, dynamic or static, partial or complete is an essential part of software radio technology. Thanks to it, systems can be designed for change and evolution. In a sense "change" becomes part of the mainstream system operation. In this paper issues relating to the required device-level support will be covered. Device level support implies appropriate hardware and software architectures as well as design approaches accounting from one hand for the supported reconfiguration scenarios and from the other for the device specific constraints. We have opted for a customized and thus lightweight component-based approach using as guidelines a typical software-upgrade scenario for "bug-fixing" and device performance enhancement.

I. INTRODUCTION

Reconfiguration, dynamic/static, partial/complete is an essential part of software radio technology [1]. Thanks to it systems are designed for change and evolution. In other words "change" becomes part of the mainstream system operation. Recent work in European Union R&D projects (i.e. the TRUST, CAST projects), the SDR Forum and lately WWRF, clearly shows that the concept of reconfiguration especially in the context of mobile cellular networks is a complicated business. Reconfiguration still raises questions on the required system-level support both at the reconfigured devices and at the network side [2].

Over the past decade previous work has concretely demonstrated the technical feasibility of the Software Defined Radio approach in the design of radio communications equipment. The project SpeakEasy was a turning point. This previous research and experimentation has resulted in a deep understanding of the SWR technology and its potential applications. Initially effort has mainly focused on issues relating to the area of increasingly software implementation. The advantages of this approach are numerous. In addition, the flexibility of increasingly software implementations offers potential advantages especially when the radio equipment is considered as part of a network as is the case in cellular mobile radio networks. This potential can be concretely exploited through equipment reconfiguration. In the evolution path towards and beyond 4G this potential flexibility can be useful in many technically challenging as well as commercially attractive use scenarios. Moving towards more cognitive (i.e. intelligent) radios [3] the WWRF vision indicates that in 4G SWR technology will play a key role. This is because in future networks (or network of networks) access transparency for the user, service quality and network management optimization will necessitate to consider reconfiguration as part of the mainstream system operation.

In this paper we shall attempt to analyze and discuss the software radio issues relating to reconfiguration and more specifically the required device-level support. The rest of the paper is organized as follows. First a brief overview work related to reconfiguration will be given. Next, the hardware platform we use for experimentation and prototyping will be described. On top of this reconfigurable architecture a component based design approach is employed to develop the needed reconfiguration mechanisms. Our approach can be tailored to handle various reconfiguration scenarios necessitating more or less network involvement. Several case studies undertaken by our laboratory will also be described. Finally, some conclusions will be drawn.

II. PREVIOUS WORK AND A PLAUSIBLE ROADMAP

In this section a rapid overview of software radio literature related to reconfiguration is given. Space limitations do not permit to be more exhaustive in our review.

A. Previous Work

[4] gives an overview of Japanese R&D in SWR. For reconfiguration the main application targets is multi-mode and multi-service operation as well as remote upgrades for performance enhancement and bug-fixing. Depending on the reconfiguration type, reconfigurability has to be designed in both the physical and higher layers. The interest in Japan for SWR stems from the assessment that in the 4G era 3G will coexist with 4G to ensure coverage in urban and rural areas respectively. Users will require greater transparency for access as well as greater service integration; this is the single terminal trend. In addition the different backbone networks should be able to cooperate. The final goal is, thanks to the reconfiguration capabilities, to offer the user what is called in the paper best communication; best in terms of quality, price, coverage.

In [5] a scheme for a parameter controlled reconfiguration and a prototype is presented. This scheme targets multi-mode and by extension multi-service terminal operation based on a common hardware platform. The baseband

119
functional blocks in the transceiver chain are created so that common aspects in the different modes are factored by functional block parametrization. Some parameters control the specific ways that blocks can be connected or by-passed. In a sense the software architecture is static since the software for all modes is resident at all times. The mode of operation is selected by downloading the specific parameter values for each air-interface. Though flexibility is constrained, this approach by virtue of its simplicity is robust, reliable and can provide for fast mode switching.

In [6], [7] the terminal design approach of project CAST is presented; it intends to give terminals more flexibility by making future extensions possible. In this approach the decisions on the system flexibility are transferred from design-time to run-time. Two elements are basic. First, a modeling of both hardwired and software functions as well as logical and physical connections by means of object oriented techniques (UML); second, hardware and system support for the dynamic instantiation, by means of a resource controller (RSC), of signal processing chains on the available hardware. The proposed target architecture combines Java technology to DSPs and FPGAs and hopes to provide for partial and complete system reconfiguration.

In [8] Moessner et al. describe a complete network-wide architecture framework to support three typical reconfiguration scenarios in a mobile cellular network. These scenarios are terminal boot, multi-mode operation and software upgrades. This architecture aims to support partial or full reconfiguration of all protocol layers as needed, control and management of the reconfiguration process, and finally, control and monitoring of the network nodes in respect to their configuration that may change over time. The following elements are described: (i) the terminal software architecture, (ii) the network entities for reconfiguration control. A CORBA based solution is suggested for realizing a configuration software bus within the terminal to connect the two terminal functional parts i.e. configuration and radio related parts.

In [9] the approach taken by the EU project TRUST is presented and a thorough analysis of the reconfiguration problem is given helping to grasp its high complexity.

Finally, in [10], [11] both papers present algorithms and techniques for the "blind" identification of the air-interface standard/modulation by a receiver. Such schemes will permit the terminals to become more intelligent and thus more independent so remove from the network both the responsibility and the workload for reconfiguration (i.e. processing and messaging).

B. A Plausible Deployment Roadmap

From the above discussion the complexity of the general reconfiguration problem becomes evident. Different scenarios necessitate different types of reconfiguration, par-

tial or total, static or dynamic, with or without network implication. Furthermore, the way reconfiguration capabilities will be deployed in the future is not yet completely known. More experimentation is needed to help standard bodies, regulation authorities and business actors, define some kind of deployment roadmap. Past experience shows that technologies evolve from simple towards more complex applications and on a need basis. The scenarios on software upgrades for bug-fixing and performance enhancement as well as algorithm dynamic change (i.e. algorithm diversity) within a single mode of operation will be deployed first. These schemes permit download and reconfiguration signaling through logical/physical channels existing within the mode of operation. Next will come simple robust schemes for multi-mode/multi-service operation without or with minimal network implication. Alternative uplink-air-interfaces could be used whenever a mode of operation disposes only of a downlink, e.g. DAB. During this period device reconfiguration mechanisms will mature, a higher reliability of the reconfiguration processes will be attained and regulation issues will become more clear. At the same time the move towards 4G will advance network interoperability. This fact will push forward software radio applications for dynamic mode switching under network control. This will enable dynamic spectrum and network resource management, more intelligent air-interface selection for "best" communication and service integration. Progress in the domain identification algorithms will contribute in making the reconfigurable radios more independent and will help to lower the impact of reconfiguration on the network.

In any case the first step will be to design devices (both terminals and basestations) to support the required reconfiguration scenarios. As [6], [8] and [9] show, this implies new approaches in hardware organization, software architecture and reconfiguration interfaces.

III. A SWR Experimentation Platform

The hardware platform used for our experimentation in software radios and reconfiguration of the radio operation is shown in FIG. 1. A quad TIC6201 DSP processing board interfaces to the analog world through mezzanine cards for A/D, D/A conversion. These cards include by-passable digital frequency translation components for down/up-conversion (DDC/DUC). Most of the operation parameters are under software control through well defined APIs and hardware interfaces. The platform provides of a fast ethernet connection and has a bi-directional R/T data streaming interface as well as a separate interface more appropriate for control signaling. Through these interfaces remote hosts can interact with the platform both for development and in the context of demonstration applications. In our demonstrations the MATLAB environment running on a host is connected to the DSP platform.
for R/T application data visualization and application control. This setup effectively permits to study and demonstrate different types of reconfiguration scenarios and the interactions of the various entities in a cellular network when air-interface reconfiguration occurs.

As it will be explained later on, the platform resources are represented in the software domain by software abstractions acting as components. In this way application software and platform hardware are modeled in a uniform manner. Though we currently use available technology (processors, A/D/A converters) we concentrate in working out solutions to system-level problems anticipating the rapid evolution in these technologies.

IV. COMPONENT BASED DESIGN

In this section a component based approach will be discussed. Components have been used in the software engineering community for quite sometime. A component based approach is to be considered as an extension and complement to the object oriented approach. Components are a design approach to enforce and achieve reuse (Meyer [12]). In a wider sense components are also the means to achieve system extensibility and evolutivity after deployment (Szyperski [12]). For a system, extensibility refers to the addition to new elements to the existing ones and evolutivity to the replacement of old ones by new ones. These are precisely the goals of reconfigurability and reconfiguration in software radios. More details on components can be found in [12], [13].

A. Components, Composition and Configurations

A component can be defined as a completely encapsulated behavior representing a unit of change. Change may occur when the system is operating (hot change) or when it is stopped. Change has to be supported by some means of dynamic linking and late binding. Though basic object oriented design concepts (e.g. encapsulation/information hiding) apply to component design as well, a component is not necessarily a class, it can be a collection of tightly coupled peer classes. With adequate rules, discipline and some basic system support, C language can be used to build components though using some OO language will certainly help. Programming practices allowed by many OO languages have to be avoided (here is where discipline is capital). Examples are the use of global variables (compromise encapsulation) and inheritance (compromises extensibility and evolutivity).

Components imply composition which is a recursive operation. Through composition more complex components are built from simpler ones. The full system may be considered as the top-level component. Here configuration enters into play. Webster's defines configuration as: "...something (as a figure, contour, pattern, or apparatus) that results from a particular arrangement of parts or components...". For our purposes we interpret this definition to say that a configuration gives a static view of the system's structural and functional aspects which define the system operation. Structural aspects relate to the interconnection of the various system components.

Re-configuration is the process of changing a system's configuration by modifying either its structure or the functional aspect of one or more of its components or by changing both structure and function at the same time. As already stated this is what components offer: extensibility and evolutivity.

At a first time in our work we consider the simple case that structure (component interfaces and interconnections) will not change and that change will concern the algorithms that implement the component behaviors. Extensions to handle more general reconfiguration cases are envisioned.

B. Configuration and Configuration Data

We make the distinction between configuration and configuration data. A configuration describes a system state of affairs while the configuration data is a machine representation of that state. Assuming that a configuration change does not modify the system structure, i.e. how the system components are interconnected, the configuration data may consist of a combination of the following elements:

- parameter values for each hardware and software component with some degree of generality in its design;
- state variable initial values for each software component whose process is not stateless;
- binary code data representing the implementation of software component functions and binary data bit-streams representing hardware component implementations on reconfigurable hardware (e.g. SRAM-based FPGA); these account for the reconfigurability offered at the chip level;

When also structure is allowed to change via reconfiguration, the configuration data must also include a machine representation of component interconnection information as well as execution scheduling information.

Configuration data stored using an explicit storage format
form configuration records that can be further structured as a configuration database. The term database implies some form of indexing. In our case the index key consists of a configuration identifier unique for each configuration. It should be noted that unique component identification is a feature of current component frameworks. The supported configurations is the set of all configurations for which there is a record in the configuration database. Thanks to this database, references can be obtained to the entire data collection or individual parts of it making it easy to access either the entire system configuration information or the configuration information of specific subsystems, or individual components. This is the work of the configuration manager described next.

C. Reference Architecture

Our reference architecture is shown in FIG. 2. It is generic enough to represent the basic reconfiguration architecture elements and their interaction. It also provides for future evolution by representing the cases that reconfiguration control and reconfiguration data will be distributed across a network. The reconfigurable transceiver software consists mainly of the following software entities: the transceiver (TRx) that implements the signal processing tasks and the configuration manager (CMan) that is responsible for the static or dynamic configuration (and reconfiguration) of the system.

![FIG. 2: Reference architecture for reconfigurable radio devices](Image)

Of importance are the control interfaces (I/F) presented to the configuration manager (CMan) by the software and hardware (A/D, RF) components allowing CMan to control their configuration related aspects. The CMan is a hierarchical entity distributed in the system. In an a sense reconfiguration is a recursive process starting at the top-level component and propagating down to the leaf components. Each component is responsible for its own re-configuration and is sensitive to only a part of the configuration data. For on-line re-configuration, special sequencing and synchronization is needed to control how configuration data propagates through the system and thus preserve consistency.

Finally, depending on the case, the reconfiguration controller (RC_ctrl) and the configuration store (CSt) may be distributed entities in the sense that processing intelligence and the associated data may be distributed across several physical entities. Consequently specific communication paths are implied. For instance this may be the case in a cellular network where terminal function depends on decisions taken at the network side, i.e. the device is under network control.

In this case these distributed entities have a remote and a local part in respect to the terminal equipment. The local part acts as a proxy for the remote part. When the device, from the standpoint of (re-)configuration, is completely independent the remote parts disappear. In this case the reconfiguration decision making and deployment are performed locally using locally stored configuration data.

D. The Configuration Cache

At this point it is useful to describe into more detail how the configuration data is organized and as well as the mechanisms that govern access to this data. The configuration store, CSt, defines the following: how configurations are stored, where configurations are stored, and how the configuration data are accessed.

We chose to organize the CST as a 3-level cache (i.e. a configuration cache). The first level (L-CSt1, L standing for local) corresponds to configurations stored in execution memory (processor internal memory). These configurations are in a sense pre-installed and ready for execution after some initialization. Switching between such configurations does not impose significant overhead delay since it only necessitates diffusing parameter values to the concerned components and resolving pointer references of software component functions. The second level (L-CSt2) corresponds to configurations stored in secondary (processor external) memory. Switching to these configurations necessitates first bringing the configuration data into execution memory (the 1-st level) using some data transfer mechanism (e.g. background DMA). The transferred data replaces some other 1-st level configuration. Switching then continues as previously described.

The third level (R-CSt, R stands for remote) corresponds to configurations stored at some remote site. Such configurations can be transferred either directly to the 1-st level if a reconfiguration was requested or to the 2-nd level if only an update of the locally stored configurations is requested. This transfer requires the establishment of a communication link (wireless or not) based on some transfer protocol that guarantees error-free data delivery.

E. Implementation Aspects

Existing component infrastructures, like for example Java and Java Beans, due to their genericity are quite heavy-weight; they offer a lot more capabilities than we actually need at the expense of system resources. For implementa-
tion we were inspired by the ideas developed by Stewart in [14] namely the port based object (PBO) abstraction. A port based object is a component combining the following elements: (i) a context independent I/O interface (ports), (ii) an object which is viewed as an abstract data type offering encapsulation and (iii) a process that implements the component behavior. This process is represented by a finite state machine and according to its state the appropriate object methods are called. Replacement independence at the algorithm level is the byproduct of object encapsulation and the context independent I/O interface. In addition to the PBO abstraction the port-based framework offers a basic and uniform execution environment for both preemptive and non-preemptive implementations.

The implementation approach of [14] being specific to a domain i.e. reconfigurable robots, uses a constrained type of components (port-based objects) and so it has certain restrictions in terms of the reconfiguration capabilities sought in software radios. Changing the configuration constants of generic components readily implements the parameter-controlled type of reconfiguration. In addition, replacement independence at the algorithm level allows for software upgrades and bug-fixing scenarios provided that the upgrades influence only the component internals and they do not have any type of structural impact to the rest of the system.

Structural impact can be as simple as requiring an extra input port for the new algorithm to work, or as complex as adding/removing components requiring both component interconnection and execution scheduling modifications. The port-based implementation model can be easily extended to cope with such structural impact inherent in other software radio reconfiguration scenarios. One extension consists in implementing I/O port interfaces as objects where ports can be dynamically added/removed and connections between old components and new ones can be dynamically created. A second extension represents the component scheduler as an object where processes can be added and removed dynamically. Hence, in a system constructed as a hierarchy of components, the system scheduler is viewed as a hierarchy of component schedulers.

Currently we stick to the constrained implementation whose basic premises are defined in [14] by adding the needed support for on-the-fly reconfiguration while keeping the needed infrastructure overhead low. For this we have opted for a C based implementation without any RTOS support. Components can be software or hardware. Hardware components are represented in the software domain by means of abstraction components providing a componentized interface to the actual hardware.

V. CASE STUDIES

In the past two years, using the software radio platform described earlier on, we carried out in our laboratory several experiments covering increasingly software receiver implementations for various standards. Lately we shifted our focus on reconfiguration applications.

Our first experimentation consisted in building a full simplified UMTS-FDD downlink (from IF to BB) including the RAKE receiver and turbo decoder blocks. All functions including carrier recovery, timing adjustment and frequency translation from IF to BB, were implemented in software. This case study served mainly to benchmark the capabilities of our platform on a demanding air-interface.

The second experiment consisted in implementing entirely in software the modem for various mobile communication standards namely GMSK, 3pi/8 offset PSK, QPSK and Frequency Hopping-FSK for GSM, EDGE, UMTS and Bluetooth respectively. The goal was to demonstrate the single platform implementation of a wide panel of modulation schemes for standards that will most probably be present in the multi-mode terminals and basestations of the future. In this experiment reconfiguration granularity is coarse and the reconfiguration process consists of a network initiated switch command triggering complete reconfiguration by paging-in the required application binary file (image) from external processor memory using DMA or the host disk via an ethernet connection. More details and results can be found in [15].

A third experiment, partly conducted as student internship projects, consisted in fully implementing in software receivers for broadcast AM/FM. In the FM case three different demodulation algorithms were implemented. These alternatives permitted to automatically switch from one algorithm to another based on a simple SNR based criterion. This demonstrates the possibilities of trading-off processor cycles (thus power) for better performance under varying reception conditions. In addition changing demodulation algorithms necessitates principally the reconfiguration of the demodulator component block which in turns necessitates the reconfiguration of hardware components by changing their parameter values. For instance, reconfiguring from a real to a complex IQ demodulation scheme necessitates reconfiguring the Rx module from real to complex operation, changing the sampling frequency and the data format. This experiment revealed not only the potential of algorithm diversity in service quality enhancement but also the subtleties of configuration dependencies and consistency. Reconfiguring a single component while preserving operation consistency necessitates reconfiguring other system components.

Finally, the "bug-fixing" scenario was further studied in the case of an EDGE receiver. The sampling time adjustment function was implemented entirely in the software domain by means of an interpolation process combining pulse shaping and polyphase filtering. This function was componentized for replacement independence. System software provided for remote binary code downloading from a remote host via a TCP/IP connection. A bug was
simulated in the timing adjustment function and was subsequently corrected by downloading only the binary code corresponding to this function. Complete knowledge of the target system memory map dispensed us with the need for dynamic linking facilities. Reconfiguration was either interactive (remote operation and maintenance) or automatic after stopping the system. We also tested the case of reconfiguration without stopping the system by background downloading into memory locations provided for this effect. Interference with the system was minimal and the transition (switch) to the new operation was seamless. This experiment increased the interest for reconfiguration scenarios where transition from a current configuration to a new one could be implemented incrementally.

An interesting concept relevant to software radios is the concept of algorithm diversity. An example is given by Laster in [16] for the case of GMSK demodulation for which different algorithms exist. Instead of using a unique demodulation algorithm yielding good performance on average, thanks to software radio and reconfiguration more flexible schemes can be envisioned. In the future we shall experiment with this concept because it corresponds to one of the interesting and short term applications of software radio reconfiguration in a cellular network.

VI. Conclusions

Reconfiguration in software radios may have different manifestations depending on the targeted use-case scenarios. In its most general form it is a quite complex problem. To navigate through the problem space we had to define typical use-case scenarios and a plausible road-map for the deployment of reconfiguration capabilities. We chose to first address the issues relating to the design of radio equipment supporting reconfiguration. Such platform support is a basic point and of great interest from the standpoint of equipment manufacturers. A component based design approach for the representation of hardware platform capabilities and software architecture is estimated as necessary.

We hope in the future to be able to treat within a more formal framework issues relating to the compositional aspects of component based architectures. Finally, an important open issue is the required network-wide architecture and simultaneous management of reconfiguration and inter-network handoff. Resolving such questions will necessitate close collaboration between all implicated actors, manufacturers, operators, regulators and standard bodies. Experimentation will provide valuable feedback for standardization.

VII. Acknowledgments

We would like to thank Trium R&D for supporting our investigation of reconfiguration aspects of software radio.

VIII. References

ABSTRACT
Software definable radio offers a wide range of new possibilities for mobile communications and mobile networking. At the same time it provides a variety of challenges in regard to stability and reliability of reconfiguration procedures as well as from the regulatory side. This paper briefly outlines the various challenges, justifies the need for a reconfiguration management structure for Software Radios and presents the Reconfiguration Management Architecture (RMA) and the functionality and mechanisms of its Configuration Management Module. Finally, the initiation sequence of a reconfiguration procedure, in our RMA model, is described.

I. INTRODUCTION
"Software Radio Terminals can be described as hardware platforms configurable, by the means of software, to various/any radio access technology" [1]. Such terminals have to be capable to support different degrees of reconfigurability; this may range from parameter based radio definition (as described in [2]) to exchange of complete software modules (like the implementation of a digital baseband, etc.). Thereby, the number of functional modules that may need to be reconfigured within a terminal depends on one side on the programmable radio platform, but also on the specifications of the initial and the target access schemes. Differences between two configurations may range from, comparatively, minor changes in the RF part (e.g. the adaptation between different frequency bands, like GSM 900/1800) to a complete reconfiguration from one access scheme to another (e.g. GSM to IS-95). Accordingly, reconfiguration procedures may be, comparatively simple like the alteration of one particular parameter/module, or very complex, e.g. when all entities within a terminal, including all protocol stack levels from application to physical layer (i.e. the radio processing platform) are being reconfigured.

Adaptation and reconfigurability from one to another existing radio access standard will be the initial application of Software Radios whilst future use may even encompass the dynamic, application specific allocation of spectrum and access scheme [3]. However, such a rigorous type of reconfigurability, will require that mechanisms to ensure the (technical and regulatory) reliability are developed and in place. Eventually, reconfigurable communication equipment may require the introduction of additional signalling channels as well as expanded network infrastructure to support, manage and control reconfiguration procedures within the terminal [4]. Major aim of such a management scheme must be to ensure standard compliance of current and intended configurations of software definable radio equipment, a responsible network authority (e.g. mobile an fixed network provider/operator) will need control over any reconfiguration procedure of Software Radio Terminals requesting reconfiguration. The future application and circulation of such terminals may be restricted if the regulatory problems, like ensuring standard compliance, are not being overcome. The Reconfiguration Management Architecture (RMA) [4], described in section II, provides the means to manage and control terminal reconfiguration. The internal operations of the RMA, implementing a functional plane that deals with
system wide reconfiguration management and its application within reconfigurable mobile communication networks, are also described. The main objective of the RMA can be summarised as to prevent unsolicited radio access scheme configurations in Software Radio Equipment.

The structure of the RMA and in particular the functionality of the ‘Configuration Manager’ module, as core part of the RMA (see Figure 1), are explained in the following sections.

II. RECONFIGURATION MANAGEMENT ARCHITECTURE (RMA) – reconfiguration management plane

The RMA deals with those signalling and download issues and procedures that are related to terminal reconfiguration; it provides the mechanisms to manage and control soft terminal reconfiguration and, most significantly, ensures that the terminal complies to the radio standards at any time during and after the reconfiguration process.

The RMA implements a logical, network and terminal based, distributed ‘management and control’ structure within reconfigurable mobile communication networks. This additional signalling system controls and manages configurations of nodes within the networks, it enables and supports - security, reliability, standards compliance, as well as trusted and secure download of software.

With its inherent mechanisms, the RMA offers a solution for the control of software configurations; this may help to remove the regulatory reservations about the circulation of software definable radio equipment. It uses key guarded mechanisms for a complete reconfiguration process; the first of them is the tag-file mechanism (a tag-file contains the ‘blueprint’ of the intended new radio configuration) see [5]. The second one is a ‘rule-handling’, whereby rules can be set to determine details of an intended configuration (i.e. using manufacturer, software/service/network provider and user information), see [6].

The actual radio module implementations are undertaken through ‘Radio Module Controllers’ (RMC), see [5] and a reconfiguration validation procedure (i.e. verification and test of the tag-file), to approve an intended configuration, is performed in the network part of the RMA.

The structure of the RMA is based on a complete distribution, this ranges through all functional levels of the architecture; whereby on the system level, the functionality is shared between network and terminal, whilst on the module level, the functionality is shared between various functional entities

(see [4, 5, 6]). On the system level, there are three different parts each of them contains several functional modules (see Figure 1). These system level parts are A) Configuration Control Part (CCP) - the CCP provides the means to host the authority responsible to control reconfiguration of terminals. Tasks of the CCP include:

- Download, provision and negotiation of software,
- Performing virtual configuration for tag-file evaluation,
- Compliance with the standard requirements,
- Constant overseeing of network nodes,
- Registration of the current/new configuration;

The second system level part of the RMA is the B) Configuration Management Part (CMP) - tasks of the CMP include the procurement of configuration software; handling of configuration rules; generation and compilation of tag-files; implementation of new configurations and finally reconfiguration related signalling. The C) Radio Module Part (RMP) - it is a combination of modules representing the radio part. The RMP represents the actual hardware platform and may be based on an agglomeration of many different types of processors, a possible format of such a SDR platform can be seen in [7].

Communication/signalling between the various parts of the RMA relies on a set of protocols and open interfaces between the various functional blocks.

III. RECONFIGURATION PROCEDURES AND SCENARIOS – control and management

The ‘Configuration Manager (CM)’ module is at the centre of the RMAs Configuration Management Part. The mechanisms implemented in the CM manage reconfiguration processes and conduct all interactions between the various modules within the RMA but it also manages the signalling between CMP and CCP.

The interfaces between the different functional modules
and the Configuration Manager hide the implementation details of each functional block and are defined in IDL. As previously stated, the whole architecture is based on a distributed structure; this scheme stretches even into the internal mechanisms of the Configuration Manager, where a main process controls the functions of other internal processes (see figure 2).

Internally, the CM consists of two independent ‘child’ processes and two associated message handlers. As shown in figure 2, the CM module itself has one message entry and two exit points enabling the communication between its internal functions and connected modules. One of the child processes performs an internal check of any anticipated/intended configuration. The other one triggers and manages the Dispatch Message Queue (DMQ). This queue posts the outgoing signalling messages to the other functional modules within the RMA and, in particular, to the RMCs. The DMQ is internally organised with two ‘IDispatch’ connection points – one of these points has a direct connection to the RMCs, whilst the second point acts as external connection to the other RMA modules within the terminal. This structure enables the complete independent handling of signalling messages, i.e. if, for example, if the CM receives a message triggering an CM-internal event, the ‘Main Messages Handler’ (MMH) will accept and process it, whilst a message that needs to be executed outside the CM and requires to be delivered to other external modules, will be dispatched by the Message Distribution Handler (MDH) and will be posted to the DMQ.

Dispatch via the MMH and MDH mirrors the two different types of messages handled in the CM: a) ‘notification’ and b)‘dispatch’ messages. Notification messages are processed by the MMH and do not issue messages to modules outside the CM, whilst dispatch messages are passed through the MDH to the DMQ, which distributes them to the specified connection point (see figure 2). This approach implements the decoupled/distributed structure of the CM module. This distribution of processing load between the internal entities decreases the complexity of the single function blocks within the CM, introduces more flexibility and decreases the individual message processing time.

The CM controls every configuration or reconfiguration process within the terminal; to pursue an actual reconfiguration process it has to control the following:

- To request a virtual configuration procedure;
- Continuous monitoring and state control;
- Termination and removal of internal and external software components;

To implement these tasks, the CM communicates, using various message sequences, with the other parts and modules within the RMA, these sequences include:

- Request and procurement of software, via the Local SoftWare Repository (LSWR);
- Request and procurement of configuration rules via the Configuration Rule Handler (CRH);
- Creation of new tag-files in the Tag-File Handler (TFH);
- Establishment and termination of secure connections, by the Security Manager (SECMAN);
- Procurement of a ‘module map’, from the comparison block within the TFH, to determine the necessary changes to the ‘current’ configuration [4];
- Installation of radio modules to RMC;
- Request procedure whether the user authorises a reconfiguration;
- Registration of the new configuration to TFH and AcA;

Figure 3  Configuration Manager Boot Sequence

The Boot-Sequence

To complete a boot sequence, the configuration manager performs a number of activities, as shown in figure 3. The boot process starts the RMA’s main processes and each of the modules then instantiates its child processes, initialises the internal interfaces and creates the messages handlers and a parameter matrix, specifying the system limits for each of the radio modules. Once all modules in the CMP of the RMA are activated, the CM initiates the installation of the radio modules, see [5]. After completion of this installation, i.e. the configuration of the terminals configurable radio part is complete, the CM starts to monitor the events and activities of the RMP and AcA. The CM and waits for any incoming message or event notification. In case
a message requests the closing down of the terminal, the CM co-ordinates the termination of all initialised modules and terminates the active radio implementation-objects within the RMP and the control/management-objects of the CMP. Finally, the CM closes down its own instances, its child and main processes.

**Reconfiguration Control**

Part of the responsibilities of the CM is to monitor, via the RMCs, the behaviour/use of the installed radio module(s) and to control different communication related parameters (e.g. QoS, Bandwidth, etc.).

An example for this, i.e. the implementation sequence, is depicted in figure 4. This sequence shows the communication between the radio module controller, the Configuration Manager and the Radio Module (n.b. figure 4 also shows the internal elements of the RMC). Once instantiated, the RMC continuously monitors specified (radio-subsystem) parameters, the internal (RMC) state machine continuously receives information (i.e. the specified parameters) from its associated radio module. The parameters received are evaluated, whether or not they fit into defined limits (i.e. the ranges for these parameters are defined within the tag-files, see [4]). The RMC internal state machine created (during its instantiation) a parameter matrix; during run-time the RMC periodically compares the incoming parameters with the entries in this matrix and forwards the results of this comparison to the Configuration Manager. If the parameters monitored do not satisfy the requirements, the RMC state machine will issue a message to the Configuration Manager (n.b. the message may also be passed to the user to inform him/her about the problem).

**Figure 4** Management and Control of Reconfiguration Request

The CM then may decide whether a reconfiguration procedure needs to be undertaken or not based on the tag-file stipulations. If the user precludes reconfiguration (i.e. set in the policy or via a message notification, see [5]), the internal timer of the RMC is simply reset and the monitoring continues until the next time the user/CM receives a message about changing parameters/conditions. In case the user/CM agrees to the request for reconfiguration, the CM may trigger the complete reconfiguration procedure, see [5]. Once a reconfiguration is triggered, the Configuration Manager informs the RMC using a message that contains information about the exact part of the radio module that needs to be reconfigured. In case no parameter matrix exists within the tag-file (or none has been included by the manufacturer or software vendor), or if the radio module is unable to supply the necessary feedback/parameters, the only way to trigger a reconfiguration process is a direct command from the user interface. Whereby the user may initiate reconfiguration sequences via a ‘reconfiguration request’ issued to the Configuration Manager. Once such a request is ‘submitted’, and if the module to be replaced is determined, a module reconfiguration sequence will be performed. In case, the module requested to be replaced cannot be located/discovered, a full terminal reconfiguration may be initialised (i.e. this however requires the permission of the user/request initiator of the new configuration. N.b. even though this reconfiguration algorithm is ‘terminal initiated’, it requires supportive services (SW download, configuration validation, etc.) from the network (i.e. ‘terminal initiated-network supported reconfiguration’).

The other case (i.e. network initiated-terminal supported reconfiguration) occurs when the network issues a request and the terminal is required to reconfigure parts or the complete of the RMP. This latter type of request is issued in the case that the network provider had made alterations to their software, like module updates (e.g. new software releases) or bug fixes.

A network initiated reconfiguration process (independent whether its a partial or complete reconfiguration) starts with a request (issued by the AcA server) for establishment of a secure connection. The AcA server sends this request to the (RMPs) security manager (SECMAN) within the terminal. Once the SECMAN has established this connection (between terminal and the requesting AcA server), the AcA server initiates, via the secure connection, the start of the actual reconfiguration procedure. This ‘initiate_reconfiguration’ message specifies the type of the reconfiguration required and passes details such as software & rules required and directions about the tag-file creation (i.e. the tag-file may also be supplied by the network).

The SECMAN receives and decodes the message and forwards it then, for further processing, to the Configuration Manager.
From this point, the configuration manager pursues the afore-described reconfiguration sequence, see figure 3. The complete reconfiguration management architecture and the accompanying reconfiguration sequence (as described in this section) have been modelled on a cluster of Unix workstations and are being ported to an actual wireless network test-bed. Aim of this implementation is to evaluate and quantify the performance of the reconfiguration management architecture (RMA).

IV. SUMMARY

Situations or scenarios requiring a terminal or network initiated reconfiguration of a SDR terminal, may occur at any time. The actual need to reconfigure a mobile terminal may arise because of changing network conditions or new requirements of applications/users, this includes varying bandwidth, changing QoS demands, etc. Therefore, to gain the flexibility reconfigurability offers to software definable/programmable terminals, it is crucial that a reconfiguration managing entity is in place. The Reconfiguration Management Architecture (RMA) described in section II, provides a system-wide framework to deliver this functionality, whilst the Configuration Manager (as part of the CMP, see section III) implements all procedures and reconfiguration sequences within the terminal and ensures reliable and stable reconfiguration of SDR terminals.

ACKNOWLEDGMENTS

The work presented in this paper has been a part of Core 2 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com, whose funding support is gratefully acknowledged. More detailed technical reports on this research are available to Industrial Members of Mobile VCE.

REFERENCES

CONFORMANCE AND INTEROPERABILITY - TESTING STRATEGIES FOR SOFTWARE DOMINATED DIGITAL MOBILE COMMUNICATION TERMINALS

R. Schmale
Nokia Mobile Phones, Germany

ABSTRACT

The role of digital mobile communication terminals is changing. New sophisticated features, colour displays, increased memory, multimedia and high-speed packet communication capabilities are being introduced. The classic mobile phone is changing into a personal communication device offering new innovative mobile data services to the user. Software has become the key enabler for leveraging the opportunities these new technologies bring. However, this will introduce new challenges to handset design and testing activities. This article focuses on the evolution of the testing methodology that is used in the mobile telecommunications industry to demonstrate compliance with a standard and interoperability of applications.

INTRODUCTION

In 1980, there were many different incompatible analog systems for mobile telephony used worldwide. The increasing need for comfortable services led to the development of several standards for digital mobile phone systems.

In 1991, the first call on a commercial Global System for Mobile Communications (GSM) network was made. The first handheld terminal was introduced in 1992. Since that time, GSM has been a big success not only from a commercial but also from technology point of view. The main ideas of the software radio concept [1] are reality in today's design of GSM mobile phones. The importance of the hardware design has decreased.

Direct conversion of the radio frequency to baseband signals and their digital signal processing is state of the art. Moreover, the main functions of the protocol stack are now implemented in software and executed on microprocessors.

The focus of the design activities has been shifting from hardware to software development. This evolution is illustrated in Figure 1 that shows the amount of memory, which is available in GSM terminals [2].

Together with the memory size, the processing power installed in mobile handsets has increased accordingly, see Figure 2 [2].

At the same time the testing part of the design...
work rapidly gained importance and, with the introduction of new features (like GPRS), there is an increased need for testing solutions that support both conformance to standards and functional testing. For example over the past years the number of test cases needed to demonstrate compliance with the GSM standard has increased as shown in Figure 3. Obviously a standard with a growing number of features and complexity results in a rapidly growing effort for conformance testing.

The evolution of mobile communication systems leads now to third generation terminals that will provide global multimedia access to the mobile user. Distributed software applications will dominate the service architecture.

However the success of such new complex services depends very much on the possibility to ensure interoperability between applications. This article examines the increased importance of interoperability between distributed applications and their effects on testing methodology.

Starting with the experience from GSM conformance testing a strategy for testing software dominated mobile communication terminals is outlined. The strategy tries to combine the traditional conformance testing approach of the telecommunications industry with new requirements emerging from the mobile Internet.

CONFORMANCE TESTING

Traditionally the telecommunications industry operates in a highly regulated world. This is reflected in the methods that are used to test radio telecommunication devices. Testing and product certification means verification that an implementation supports a standard and meets specified conformance requirements. In concrete terms this results in conformance testing against validated test equipment using a set of well defined test cases.

This concept worked very well and is one factor for the global success of GSM. Rigorous conformance testing of GSM terminals today ensures seamless roaming of mobiles in networks all over the world. Mobile terminals from different manufacturers interoperate with network equipment from various vendors.

The OSI Model

Conformance testing means testing the extent to which an implementation satisfies all relevant specified conformance requirements, e.g. requirements that are put forward in the base standards or specifications. Moreover conformance testing means black box testing of functional properties on an implementation. It tests observable behaviour. The tester performs a request and observes the response from the Implementation Under Test (IUT).

Using the Open System Interconnection (OSI) representation of a protocol this general principal of conformance testing is shown in Figure 4. This ideal tester accesses the IUT at the so-called Point of Control and Observation (PCO). The PCO is the interface to a particular layer of the IUT. The PCO is identical to the Service Access Points (SAP) of the OSI model. Via the PCO the tester sends and receives service primitives (SP) of the protocol layer under test.

However in practice it is unlikely that the SAP can be accessed with the upper/lower tester interface as shown in Figure 4. Because of that conformance testing of today's mobile telecommunication systems relies mainly on the indirect access to the SAP. A reference protocol stack is implemented in the tester. The tester can access the SAPs of its internal reference protocol stack. Based on the dataflow between the reference protocol layers the behaviour of the IUT is tested. Of course this holds true only in case the reference stack in the tester implements the protocol correctly. Therefore the success of this test method is depending on the quality of the reference stack implementation. A huge effort by the test equipment manufacturer
and conformance test houses is needed to validate the reference stack against the core standard. The principal of a reference protocol stack tester is illustrated in Figure 5.

Most existing conformance testers for digital mobile communication systems make use of this principal up to the layer 3 (Network Layer) of the OSI model. Today's commercially available conformance test systems do not provide possibilities to do structured conformance testing on application level. Another disadvantage is that during the test execution an operator is necessary to control the requests and responses of the IUT. There is usually no standardised test control interface and the only possibility to control the IUT during the test cases is the Man Machine Interface (MMI). This results in long and expensive testing time.

**BEYOND CONFORMANCE TESTING**

Primarily conformance testing determines the conformance of an implementation to a protocol standard. Conformance testing, no matter how rigorous, does not provide fail-safe assurance that a given protocol implementation will interwork with other implementations, despite their both having successfully undergone testing. In particular conformance testing cannot detect incompatible selections of optional (changeable by the user) protocol parameters. Especially on application level this is a problem. Because of that the conformance testing methodology is evolving to include protocol profiles. In addition interoperability testing is used for 'inter-product' testing.

**Protocol Profile Conformance Testing**

A protocol profile consists of a set of one ore more base standards and the identification of chosen classes, subsets, options and parameters of those base standards [3]. Profiles may be standardised like e.g. for Bluetooth. Furthermore it is necessary to create a profile conformance test specification and the corresponding abstract test suite. The conformance of the profile is then demonstrated by executing a suite of structured test cases. Due to the nature of profiles, testing of protocols within a profile is usually done protocol by protocol, working upwards from the bottom [3].

Summing up protocol profile testing extends the classic conformance testing approach by addressing the relationship between different sets of base standards (protocols) used together to accomplish a particular function.

**Interoperability Testing**

In contrast, interoperability testing determines the ability of implementations to interwork in an operational environment. It uncovers incompatibilities even when both implementations conform to the same base standard. Interoperability testing aims to increase the probability that real system interworking or
interoperability will be attained in normal use cases. The big disadvantage of interoperability testing is that each implementation must be tested against every other product with which it will interact. For the consumer market with its huge number of different (e.g. Bluetooth) devices it is obvious that this is not possible in a structured way.

The relative priority and advantages of conformance testing and interoperability testing is a controversial issue. In the past the telecommunications industry has relied on conformance testing. On the other hand the IT and related software industry favoured interoperability testing. Conformance to a standard is of less importance to this portion of the OSI market. Particularly the end user market places a higher value on demonstrable interoperability than on conformance.

**FUTURE TESTING STRATEGY**

From testing point of view the challenge is now to effectively ensure conformance to a standard and at the same time end-to-end interoperability of distributed applications.

A possible tester architecture, which tries to fulfil both requirements, is shown in Figure 6.

**Lower Layers**

The lower layers (OSI layers 1 to 4) of the protocol are tested according the classic conformance testing approach using a reference protocol stack in the tester. The RF front-end of the tester is using software radio architecture [4]. This makes it possible for the tester to reconfigure the air interface functionality to different protocol standards. In addition to that the tester accesses the IUT at an upper interface using a Test Control Interface (TCI). The specification of the TCI must be part of new core standards. For example the TCI is already part of the Bluetooth standard [5]. In the 3rd generation ETSI standard the TCI is defined as terminal logical interface [6]. By defining a TCI the tester does not need to be adapted to the different implementations of the core standard. Moreover conformance test time can be reduced because there is no need for an operator to control the IUT. The testing of the lower layers can mainly run automatically under full control of the tester. The test suite itself consists of test cases that are written in a formal notation. For example The Tree and Tabular Combined Notation, TTCN [7] can be used. If in addition the core standard is described in a formal Specification and Description Language (SDL) it is possible to generate the test cases.
directly from the core protocol specification. Besides this saves a lot of time during validation of the test cases. It is then the core standard that defines the quality of the reference protocol stack not the test equipment manufacturer. Furthermore it is possible to run TTCN test suites hardware independent on test equipment from different vendors.

**Higher Layers**

Future mobile communication terminals will offer integrated native applications (e.g. e-mail, SMS) as well as the possibility to download additional ones. Middleware that provides open Application Programming Interfaces (APIs) is a key element to enable the creation of these distributed applications [8].

Because of that the tester integrates reference standard applications (e.g. SMS, http, imap) that are used for protocol profile conformance testing of the IUT. The more complex and distributed applications are handled by the tester via a connection to the Internet. External certification bodies (e.g. Open Group [9]) are providing certification test suites on their servers that are accessed by the IUT via the tester. The tester simply provides the bearer for these tests. For example this principal can already be used today to demonstrate conformance to the Wireless Application Protocol (WAP) [10]. In future service like XHTML, SyncML, Multi Media Messaging (MMS) and Java applications will be tested accordingly. One problem remains that conformance requirements for these services are often related to the display of the IUT. Because of that there is still a need for an operator to make the final verdict if an application test case has shown the result in the correct way on the display of the IUT. This makes automated testing of the application layer still expensive and time consuming. In the longer term we can probably expect test equipment that supports advanced image-processing capabilities to evaluate the contents of the IUT's display. Moreover a standardised TCI on application level could be useful. During test case execution for higher layers it is at the moment necessary to manually change the protocol/application settings at the IUT. This could be automated with the help of the TCI.

**CONCLUSION**

The commercial success of the third generation of mobile communication systems largely depends on the end user accepting new value added services. The potential of application download to a handset as a means of creating revenues for the network operator is beginning to be realised. Effective and secure handling of distributed applications will be crucial in the future soft terminal world.

From testing point of view the challenge is to handle the growing complexity of the new mobile communication standards and services. The classic conformance testing methodology of the telecommunication industry needs to be improved. Conformance and interoperability must be combined effectively. New protocol standards development, design specifications and testing should be done using formal methods as far as possible. With increasing cost and time-to-market pressure the future testing strategy will rely on automated testing of the lower protocol layers using a TCI. The application layers will be tested on-line via remote access to test suite servers. Applying formal test methods on application level will significantly reduce the effort needed for interoperability testing.

**REFERENCES**


[6] ETSI TS 134 109 V4.1.0 (2001-06): UMTS; Terminal logical interface; Special conformance testing functions


Supporting programmable handoff architectures in wireless access networks through coordination contracts
Satish Jamadagni, P S Subramanian
{satishj, pss}@sasken.com
Sasken Communication technologies Ltd
#139/25, Amar Jyoti Layout,
Ring road, Domlur P.O,
Bangalore – 560071,
India

Abstract
Future wireless access networks will be built on a foundation of open programmable networking allowing for the dynamic deployment of new mobile and wireless services. These networks will have to support multiple handoff types in order to support inter-system roaming. Customizing handoff control and mobility management in this manner calls for advances in software and networking technologies in order to respond to specific radio, mobility and service quality requirements of future wireless Internet service providers.

Software defined radios allow for such programmable radio networks or base stations. In software radios physical layers with wide-band tunable front-ends can be configured to represent any physical data or control channel. The control plane in SDRs addresses the problem of making handoff programmable for the introduction of new services. In SDRs it is necessary that the creation, deployment and management of new wireless and mobile network services should be automated using programmable networking techniques. In this paper we discuss using coordination contracts [1], which allows for superposing of constraints on the behavior of handoff support in SDR base stations. We discuss only programmable handoff support in software defined access networks.

Introduction
Future wireless access networks should be built on a foundation of open programmable networking allowing for the dynamic deployment of new mobile and wireless services. Customizing handoff control and mobility management in this manner calls for advances in software and networking technologies in order to respond to specific radio, mobility and service quality requirements of future wireless Internet service providers.

Coordination contracts – a brief introduction
In this section we describe “coordination contracts” [1] and describe modeling a
programmable handoff architecture using the same. In general terms, a coordination contract is a connection that is established between a group of objects (participants), where rules and constraints are superposed on the behavior of the participants, which determines a specific form of interaction. The way such an interaction is established between the partners is more powerful than what can be achieved within OO languages because it relies on the mechanism of superposition as developed for parallel and distributed system design [1]. When a call is made from a client object to a supplier object, the contract “intercepts” the call and superposes whatever forms of behavior it prescribes. In order to provide the required levels of pluggability, neither the client, nor any other object in the system, needs to know what kind of coordination is being superposed. A coordination contract is defined as follows:

```
contract class <name>
participants <list of partners>
constraints <the invariant the partners should satisfy>
attributes <private to the contract attributes>
operations <private to the contract operations>
coordination <interaction with partners>
end class
```

where each interaction under a coordination rule is of the form:
```
<name> when <trigger>
  with <condition>
  do <set of actions>
```

The condition under “when” establishes the trigger of the interaction. The trigger can be a condition on the state of the participants, a request for a particular service, or an event on one of the participants. The “do” clause identifies the reactions to be performed, usually in terms of actions of the partners and some of the contract’s own actions.

The “do” clause identifies the reactions to be performed, usually in terms of actions of the partners and some of the contract’s own actions. When the trigger corresponds to the calling of an operation, three types of actions may be superposed on the execution of the operation:

1. **before** action: to be performed before the operation
2. **replace** action: to be performed instead of the operation (alternative)
3. **after** action: to be performed after the operation

In the case in which an object participates in multiple contracts with the same trigger, the sequence of execution for the different clauses is: all the “before”, one “replace”, all the “afters”. The semantics of contracts allow for only one “replace” clause to be executed. The actions that are executed as part of the “do” clause are called the synchronization set associated with the trigger. The semantics of contracts requires that this set be executed atomically, guarded by the conjunction of the guards of the individual actions together with the conditions included in the “with” clause. Therefore, the “with” clause puts further constraints on the execution of the actions involved in the interaction. If any condition under the “with” clause is not satisfied, an exception is thrown and none of the actions in the synchronization set is executed.
Programmable handoff support in access networks

One of the first steps in realizing a soft base station is supporting programmable handover for inter technology or inter system handover support. Lower layer programmability is not of much concern to us in this work. In this paper we concentrate on supporting multi handoff support in order to support inter technology handover in a wireless access network.

Programmable access networks allow different styles of handoff control (e.g., mobile controlled, mobile assisted and network controlled handoff) to seamlessly share the same mobility management services offered. However, seamless integration of handoff control systems with mobility management services is difficult to realize. Typical programmable handoff architectures should support profiling, composition and deployment of programmable handoff services and components. A multi handoff support access network comprises a service creation environment and a binding model. The binding model describes how distributed objects can be combined to form programmable handoff services on-demand. The service creation environment allows composition of handoff services taking into account user, radio and environmental factors.

For supporting multi handoff styles the binding mechanism should comprises a handoff control model and a mobility management model [2]. The underlying wireless access system is assumed to be a software radio model. Service controllers realize each model separately. In this manner, each model can use different binding mechanisms. In our framework we use coordination contracts to achieve better binding and support change in handoff support. Figure-1 depicts a multi handoff support base station with support for GPRS, Mobile IP and cellular IP handoff styles at layer 3.
Decomposing the handoff process for multi handoff support

To support multiple handoff styles we decompose the handoff process into generic and specific mobility management objects. Handoff control and mobility management functionalities are identified. The following handoff control services are identified:

*Detection algorithms*, which determine the most suitable access points that a mobile device should be attached to.

*Measurement systems*, which create and update hand-off detection state. By handoff detection state we mean the data used by detection algorithms to make decisions about handoff.

*Beaconing systems*, which assist in the process of measuring wireless channel quality for handoff decision.

The following mobility management functions are identified:

*Session rerouting mechanisms*, which control the data-path in access networks in order to forward data to/from mobile devices through new points of attachment. Re-routing services may include admission control and QOS adaptation for the management of wireless bandwidth resources.

*Wireless transport objects*, which interact with the physical and data link layers in mobile devices and access points to transfer active sessions between different wire-less channels. A channel change may be realized through a new time slot, frequency band, code word or logical identifier.

*Mobile registration*, which is associated with the state information a mobile device exchanges with an access...
network when changing points of attachment.

*Mobility state*, which is a collection of a mobile device’s connectivity, addressing and routing information, bandwidth and name-space allocations and user preferences.

A set of generic interfaces is identified which provide handoff execution bindings as illustrated in figure 1. These interfaces separate handoff control from mobility management functions. These interfaces include:

**Handoff** methods, which map down to mobility management services that execute handoff (e.g., register the care-of address of a mobile device with its home agent).

**Pre-bind** methods, which initiate ‘priming actions’ at candidate access points associated with a mobile device (e.g., start buffering packets, etc.).

**Configure** methods, which bind new signaling systems or delete existing ones.

**Coordinating heterogeneous handoff support objects**
Multi handoff support requires a decomposition of handoff process into objects, which are composed to form a specific handoff service. In a conventional handoff design all the objects involved in the service creation process are pre-declared. Such an approach works well for small access networks and simple handoff services but does not scale well. In this paper we have shown a decomposition of handoff process as discussed in the previous section. The ‘binding rules’ are captured in the generic interfaces defining specific handoff control and mobility management architecture. Multi handoff support represents the instantiation of a set of binding rules or contracts over some binding data.

Software contracts are used to enable inter-system handoff between different types of wireless access networks. The basic idea behind realizing inter-system handoff is that the same detection mechanisms operating in mobile devices and access networks can interface with multiple types of mobility management architectures that operate in heterogeneous access networks. Handoff control systems issue a number of generic service requests through the handoff execution interface, which mobility management systems execute.

![Contracts defined over the handoff components](image-url)
according to their own implementation. For example, a generic ‘pre-bind’
method call to a candidate access point would be executed by establishing a
signaling channel in a Cellular IP architecture or by joining a multicast
group specific to a mobile device and buffering packets in a Mobile IP
architecture.

In our architecture “contracts” control the handoff execution process. Handoff
contracts invoke mobility management services in an order that is specific to the
handoff style being programmed. Mobility management services (i.e.,
session rerouting, wireless transport, mobile registration and mobility state
management services) are invoked as part of the handoff execution process.

For example, in a forward mobile controlled handoff, a contract would
invoke a radio link transfer service before session rerouting. In the
backward, mobile assisted handoff the order of this execution would be
reversed. Each handoff style uses a separate contract.

Handoff contracts also ‘translate’ the handoff execution interface to the
interfaces supported by specific mobility management architectures. In this
manner, contracts hide the heterogeneity of mobility management architectures
enabling inter-system handoffs. An example of a contract class can be
defined over a configure object and the specific Mobility support style objects as
shown in the figure-3 below.

![Figure-3 Contracts defined between the configure object and the handoff specific objects](image)

For instance, $GPRS\_Configure$ could be the definition of a contract that is
superposed on the Configure operation in order to support a GPRS handoff
method. The $GPRS\_Configure$ contract binds a mobile node to a GPRS signaling
system. Naturally, additional operations or actions may be required based on
more “low-level” design decisions. The GPRS configure contract is illustrated
below. The notation used is as described
in the ATX software available at [5].

**contract class**

```plaintext
GPRS_Configure

constants gprs_Buffer : Boolean // Indication for packet buffering in GPRS service

participants x: Configure;

operations
GPRS_Service_Ref_Signaling(int);
```
// other operations

**coordination**

when *->>x.Configure(Handoff_Type) AND NETWORK.bearer_type:="GPRS_Service";

with Handoff_Type !=NULL;

before ...

replace ...

//possibly replace the whole Configure() by an operation defined in a contract.

Another example of a contract class defined for the handoff execution interface “Pre-bind”

**contract class** Pre-Bind

**participants** x: Pre-bind; y: Mobile_IP;

**constraints**
x.Mobile_IP:=y.Packet_Buffer_Request;

**coordination**

when *->>y.FinishHandoffRegister();

**after**
x.Initiate_Packet_Buffer (time);

**end class**

Results and Conclusions

In this paper we have designed and implemented a simple programmable handoff architecture for multi handoff support in access networks using coordination contracts. Such a design at the access network side supports inter-system or inter-technology handoff.

Earlier work on programmable handoff support [2] has used profile-scripting language to create or modify handoff services. Service controllers compile profiling scripts, resolve object bindings, and create handoff control and mobility management systems. In such implementations service creation requires a comprehensive architectural model for binding distributed objects to create new services. However, it is now widely accepted that, although OO techniques such as inheritance and clientship make it easier to build systems, their support for evolution in general, and the ability of systems to exhibit the agility required by a soft radio handoff support architecture is quite limited.

In this paper, we argue that using the coordination contract modelling primitive for superposing handoff coordination mechanisms over existing specific handoff objects can be applied to support multi handoff styles to achieve increased flexibility and agility in reacting to change.

**References**


AN EFFICIENT QoS NEGOTIATION PROTOCOL FOR SOFTWARE DOWNLOAD CHANNEL ESTABLISHMENT IN SOFTWARE DEFINED RADIO

KRISHNA KISHORE K R,
Sasken Communication Technologies Ltd,
139/25, Amar Jyoti Layout, Domlur,
Bangalore-560071 INDIA
kishore@sasken.com

13th February 2002

Abstract

Software download is the process of introducing new program code or other data to a Software Defined Radio (SDR) terminal with the intention to modify its operation and/or performance characteristics. A major portion of the signaling involved in download is due to the QoS negotiation while establishing a download channel. Traditional methods rely on a series of request and retrials for QoS negotiation. Owing to the time criticality of the download process, such schemes may not be suitable in the SDR context. In this paper, an enhanced, generic mechanism for QoS negotiation is suggested, which has special significance while establishing a download channel. The scheme can be fitted well in the download framework proposed in [1]. A numerical comparison of the proposed scheme with respect to the existing schemes is also provided.

1 INTRODUCTION

SDR is described as a combination of hardware and software technologies that allow flexible, adaptable wireless networks and user terminals to be developed. By composing the right hardware with the right software, wireless user terminals, applications and services can be modified dynamically via software downloads [6].

SDR plays a prominent role in future heterogeneous wireless networks. For manufacturers, SDR provides the ability to incorporate enhancements or service improvements; for operators, the ability to react dynamically to variations in traffic demand; for end users, capabilities such as enhanced roaming, selection of most attractive network (meeting the user's preferences in terms of cost, quality, offered applications etc), and over-the-air download of application software as and when required [5].

Software download is the process of introducing new program code or other data to a SDR terminal with the intention to modify its operation and/or performance characteristics [1]. The software download mechanism can either be out-of-call or in-call (dynamic). In dynamic download, software components are downloaded (typically OTA) during a call. Dynamic download is an obvious choice for a mobile SDR terminal, however, the process is highly time critical.

A major portion of the signaling involved in download is due to the QoS negotiation while establishing

Figure 1: Scenario 1 - QoS negotiation mechanism (REQUEST-ACCEPT)
a download channel. Typical methods of QoS negotiation [2] consist of sending a request message (which specifies the desired QoS) and waiting for accept/reject messages. When a "reject" message is received, the terminal retransmits the request and waits for the response. This process continues till a maximum count is reached, after which the attempt is aborted. This maximum count is decided based on the criticality of the session that is being established. Such a scheme is diagrammatically explained in Fig 1 and Fig 2.

In this paper, an enhanced, generic mechanism for QoS negotiation is suggested, which has special significance while establishing a download channel. The scheme can be fitted well in the download framework proposed in [1]. A numerical comparison of the proposed scheme with respect to the existing schemes is also provided.

This paper is organized as follows. In the next section, the explanation of proposed QoS negotiation for SDR download channel establishment is given. The simulation and experimental results are explained in section 3.

2 QoS NEGOTIATION IN SDR

In this work, we assume the system model as proposed in [7] which consists of (among other entities) a Software Download Module (SDM) and Mode Negotiation and Switching Module (MNSM). The MNSM continuously estimates the best mode to be chosen under the current environment, and initiates the SDM to download the required software components for a mode switch, if necessary. The MNSM expects the time required for a software download as one of the parameters for its internal logic. The entire sequence of decision, download and re-configuration has to happen within ‘t’ time units, as shown in FIG 3, where "Region 1" and "Region 2" are assumed to be in different modes.

The download server may be located at a far away place (need not be geographically, but in terms of the number of hops). Unlike the WWW download schemes, the MNSM process has to be seamless and hence it does not have enough time at its liberty to wait for the outcome of a download attempt to decide on further course of action. In the normal schemes for QoS negotiation, the request message consists of the QoS desired for the download session. If the requested QoS cannot be offered by the server at that point of time, a reject message is sent. The terminal waits for a (typically random) period of time before making another attempt. This goes on and finally the attempt is aborted. Clearly, this approach will be a major overhead here.

It would be a better option, if some more information can be sent to the MNSM which will help it in making a better decision in case of a reject message. The suggested scheme is that the server can maintain an estimate of the resources which are available (in terms of number of sessions, buffer availability etc.) and calculate the approximate duration at which the
A request can be serviced. This information can be passed to the terminal along with the reject message. This would not only help to prevent the terminal from making unnecessary attempts, but also provide the MNSM with further options such as

- **Wait for the specified time, say ‘x’**
  If the terminal can wait for ‘x’ time units, it resends the DOWNLOAD_REQUEST message for the same component after this much time. Of course, this is possible only if
  
  \[ x \leq t \]

  This scenario is shown in FIG 4. A further optimization is possible since the server already has information about the request, using which it can pro-actively send the requested component after ‘x’ time units without waiting for a new request.

- **Choose alternate component**
  If it is not possible to wait for ‘x’ time units, the extra information which was received along with the reject message can be used by the MNSM to carry out another estimation cycle and find out the next best component. Alternately, the Bandwidth Management Module (BMM) within the SDM can re-plan the sequence of download depending on the information received. This scenario is shown in FIG 5.

## 3 EXPERIMENTAL RESULTS

In this section, the simulation results of the proposed mechanism are presented. As an example, we use the number of active download sessions as the connection admission control logic at the server.

The arrival process of new download sessions is assumed to be Poisson in nature [4]. The active period for a download session is based on the statistics related to the module size, number of components etc. as per [4]. The server maintains an online estimate of the active period. When the number of active downloads at the server crosses N (where N is empirically determined), a new request is rejected. The server also sends average active period which it has estimated, along with the reject message.

On the terminal side, the active period information, say T, can be used suitably to decide when to attempt next. We consider two possible ways of using this data. In the first case, the terminal waits for T time units and resends the request. It is also possible for the terminal to treat the successive T values and make some predictions regarding the probability distribution of active periods at the server. This may lead to better utilization of the time available. We have considered in the second case that the terminal has roughly estimated the active period duration to be uniformly distributed between 0 and T.

The simulations were carried out using ns-2 (version 2.1b7a) [8] on Linux machine. The results of the experiment is shown in FIG 6 in which we plot the average number of retrials for a download session establishment as a function of active period duration, T (in time units). As expected, the number of retrials decreases in the two schemes given above. Also, ra-
domination of the retrials (as in the second scheme) has the effect of reducing the number of retrials further, however, at higher values of T. This can be attributed to the fact that for a good estimate of the distribution function, sufficient number of samples is necessary (Note: Here, a reject message is treated as a "sample") which is not available at low values of T. Also, note that the average delay for a successful establishment is proportional to the average number of retrials.

4 CONCLUSION

This paper proposes an efficient QoS negotiation mechanism for SDR terminals. The performance of this scheme as compared to the traditional mechanisms has been studied through simulation. The simulation results show significant improvement in terms of download channel establishment delay. Some enhancements to the basic scheme have also been suggested.

5 ACKNOWLEDGEMENTS

My sincere thanks to all those who directly and indirectly helped me while writing this paper. Specially, I should thank Umesh M N for his valuable guidance in ns-2 simulations.

References

Software Download Management for Cell Broadcast Channels in WCDMA
Eiman Mohyeldin, Jijun Luo and Markus Dillinger
SIEMENS AG, Werner v. Siemens Ring 20
D-85630 Grasbrunn/Munich, Germany
E-mail: eiman.mohyeldin@icn.siemens.de
Tel/Fax: ++49-89-722-58679/26478

Abstract — This paper presents the management of software downloads in cellular networks using broadcast channels that is suitable for terminal mass upgrades and thereby minimizing interference on regular traffic (point-to-point traffic) in sector cells. Novel approaches for coordinating cell broadcasts and random cell broadcasts approach are introduced and compared based on load analysis characterised by the interference rise. Furthermore, a packet indexing scheme for handling efficiently the fragmented software downloads resulted from handovers is introduced. With an intensive simulation campaign undertaken in a macro environment of WCDMA, quantitative results prove that the circular cell broadcast and proposed packet-indexing approaches have less impact on the regular traffic and improve the performance with respect to fragmented downloads.

I. INTRODUCTION

Software Defined Radio (SDR) is expected to play an important role in the development of future wireless communication systems, as the emerging IP based network and several third generation radio access standards [1]. The presented management strategy is based on the estimated traffic types and their volume to allow for adaptive Radio Resources Management algorithms, e.g. in reconfigurable terminals or base stations. Reconfiguration of a terminal involves the download for core software and applications that are not already on the terminal. The kind of download can range from simple parameters to actual executable program code. The downloaded software can be received in different formats such as hardware-specified binary codes, high-level software to be interpreted, software objects and mobile agents.

Numbers of publications have tackled the topic of software download [2][3], but how to organize the software in a current mobile system, and how to design the system with respect to lower intercell and intracell interference using common channels is still a challenging topic to consider. The software download process organized via the cell broadcasts/multicast channels in W-CDMA cellular networks is considered in this paper. A novel Software Download (SD) management with/including a packet-indexing scheme that enhances the efficiency of Fragmented SD resulted from handover are described.

II. CELL GROUPING BROADCAST STRATEGY

II.1 Design Principle for Software download Channel

The probability of using broadcast channel for downloading software to mobile terminal/User equipment (UE) is rather high, when many users want to receive the same information or mass upgrade of terminals takes place. Broadcast channels are not fast power controlled and increase the interference in surrounding cells. If the broadcast is used in all cells, then additional decrease of the cell capacity takes place due to the cell coupling in CDMA system with frequency reuse factor of one. Cell coupling means that an increase of transmission power in one cell will lead to a transmission power increase in neighbor cells due to the need for maintaining the SIR target for services. The following figure illustrates the design principles for selecting the appropriate channel for download.

Figure 1, Illustration of Choosing between Common channel and dedicated channel.

In Figure 1, as the number of user increases (shown in X-axis) the system load for downloads, i.e. Transmission power of Node B, increases exponentially. The load created by a broadcast channel is constant. The intersection point between these two curves is the threshold of choosing a broadcast channel or to carry out the downloads with dedicated channels. The shaded side of the intersection point is the area, where a broadcast channel should be selected for the download.

In the next subsection, a specific download management for all involved cells in a geographic area that will minimize the additional interference
due to software download when using cell broadcasts is described.

IIII Coordinated Download Grouping Strategy

An efficient cell download procedure can be organized in circular manner, as shown in Figure 2. The idea is to apply parallel downloads in cells with a low cell coupling, e.g. cells, which their 120-degree sector antennas are oriented towards one geographic direction.

After the download in the Nr. 1 direction is completed, the cell broadcasted in Nr. 2 directions is started and then completed in the sectors indexed by 3.

Figure 2, Simultaneous Downloading in Cell with Low Cell Coupling

This procedure is repeated over a certain time. If the download is not completed due to terminal mobility and the network stops downloading via the common channel, then a dedicated channel for download completion is need to be established.

In general, the network must be aware about the cell coupling factors that must stored in a database of the RNC. Only cells with low coupling as shown in Figure 2 belong to a download group. Similarly, download groups in micro cells and other morphologies can be defined.

In the following we describe mathematically the idea and verify it by system level simulations. In general we must introduce a split for the complete process into different time phases and cells. We propose to organize the download process into time division manner based on cell groups. A key parameter is defined as the cell-grouping factor $\gamma$, which defines the group size as the number of cells in the subgroups. In Figure 2, cells with the same orientation are organized into one group. The software download’s process time is divided equally into the accumulation of three parts $T_1, T_2,$ and $T_3$, where the cell-grouping factor $\gamma = 3$. i.e., $T = \sum_{T} T_1 + \sum_{T} T_2 + \sum_{T} T_3 + \ldots$. The download process located in cells which are labeled with number 1 will not interfere the rest, which are labeled with 2 and 3. The value of $\gamma$ and $T_i, i=1,2,\ldots,\gamma$ are designed according to the number of downloading users in the cells, the downloading traffic types, i.e., the QoS requirement, throughput requirement, real time limitation issues.

IIII User Group Classification

Different traffic types are present in a network, namely, download traffic (point-to-multi-point), voice traffic and data traffic (point-to-point). We partition the users according to their different traffic types, so that different radio resource management approaches can be separately designed in each group. In this paper, for each site $s$, which is defined as a Node B controlled area, download user group $G_{sd}$ and speech user group $G_{ss}$ are established. The significant differences between these two different group users are shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1, Comparison Between $G_{sd}$ and $G_{ss}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Group</td>
</tr>
<tr>
<td>Channel Capacity</td>
</tr>
<tr>
<td>Power Control</td>
</tr>
<tr>
<td>Handover</td>
</tr>
</tbody>
</table>

IIIV Interference to the Download Class

In user class $G_{sd}$, due to the nature of big amount of download data, it is obviously reasonable to use common downlink channels for downloads. However, power control management cannot be applied in a common channel; therefore, in order to guarantee correct download process, the download power is determined by the weakest admitted download user after their random attempt, which is usually at the cell borders. Introducing an interference evaluation model is used for the investigation of the gain for interference reduction. The complete set of Node B involved in the download process is denoted as $\mathbf{B}$, which is composed of $\gamma$ subsets, $\mathbf{B}_i, i=1,2,\ldots,\gamma$, where $i$ is the index indicating the grouped cells. In the cell-grouping download case, the interference in the same traffic group is described as the following equation:

$$I_{sd} = \sum_{\mathbf{B}_i} P_i \cdot \partial(D_{sd}) - P_{sd} \cdot \partial(D_{sd}) \bigg|_{\mathbf{A} \in \mathbf{B}_i}$$  \hspace{1cm} (1)

Where $P_i$ is the transmission power of each base station defined in the complete set $\mathbf{B}$; the number of interfering resources is determined by $\mathbf{B}_i$ in each download process. The user index $u \in G_{sd}$ and set $\mathbf{B}$ are basic input parameters for the admission control function $\mathbf{A}$, which gives the affiliated base station of the software download user; the transmission power from the base station is attenuated by the distance
between base station and terminal, the slow fading and antenna gain.

II.V Party Effect to the Regular Users

Similar to the same user class interference modeling, the inter-class interference model can also be established. Namely user $v \in G_{ss}$ suffers interference not only from the regular traffic but also the software download traffic. The regular interference is out of interest of this paper, which can be found in [6][7][8]. In case of the regular user is affiliated to the base station that is in the current download group, the interference from download traffic is:

$$I_{sd} = \sum_{\nu \in \mathbb{R}} P_{\nu} \cdot \partial(D_{\nu})$$

Where, the orthogonality thanks to the spreading gain cannot be ideal due to the multipath problem, as is modeled by the orthogonality factor $\delta$. In case, the regular user affiliated base station is not in download group, the interference is modeled as:

$$I_{vd} = \sum_{\nu \in \mathbb{R}} P_{\nu} \cdot \partial(D_{\nu})$$

The amount of interference from the download traffic is added to the regular interference as the whole interference the user suffers.

II.VI Probability of Finished Download in Single Cell

Probability of finished download $p_f$ for a fixed time instance is a variable depends on the vehicle speed $v$, the common channel capacity $C_d$, for the download process and the amount of data to be downloaded.

Lemma II.I: The finished download probability is identical to the average of intersections of cell shape with maximum mobility distance with original cell over $(0, 2\pi)$ range.

We define the finished probability in one direction $\alpha$ as $p_{f\alpha}$. Suppose the service region is homogeneous according to the mobility model, i.e. circular service region is supposed. In this case, $p_{f\alpha}$ is constant with respect to a variable $\alpha$. The finished download probability is identical to:

$$p_f = \int_{0}^{2\pi} f_{\alpha} \cdot p_{f\alpha} \cdot d\alpha = p_{f\mu} \cdot \frac{2\pi}{2\pi} = p_{f\mu}$$

(4)

For a fixed vehicular movement direction, the finished download probability is identical to the intersection between the service region and the assisting circle whose center point is defined by the distance the mobiles has moved during the download time, as depicted in Figure 3.

Figure 3, Two-Dimension Homogenous Area

If $|v| T_d \leq r$, as shown in Figure 3, the following equation set determines the finished download process:

$$T_d = \frac{A_f}{C_d}$$

$$\theta = 2 \cdot \arccos \left( \frac{V |v| T_d}{r} \right)$$

$$b = 2 \cdot r \cdot \sin \left( \frac{\theta}{2} \right)$$

$$A_f = r^2 \cdot \theta - b \cdot \sqrt{r^2 - \left( \frac{b}{2} \right)^2}$$

$$p_{f\mu} = \frac{A_f}{A}$$

Where $|v|$ is the absolute vehicle speed and $T_d$ is the download time identical to the size of file to be download divided by the download channel capacity.

Theorem II.I: A two-dimension download area defined service region results in a nonlinear finished download probability.

Based on the analysis of Lemma II.I, the non-linearity can be easily proven. For a single cell case, $p_f$ is identical to the handover probability as shown in the following figure, which is identical to the theoretical approach. The simulation parameters for these results are list in Table 2.
As shown in Figure 4, the longer the download size is, the higher the probability of finished download is. Due to the introduction of circular broadcasting channels, much lower successful software rate will result. Therefore, protocol managing software download repetition and re-assembling segmentation due to break of SD is of great need.

III. PACKET INDEXING APPROACH

To improve the efficiency in fragmented software download scenarios due to e.g. handovers in wireless networks, packet-numbering scheme with system control information carrying the software download (SD) status is introduced.

In order to explain the approach clearer, some terms are defined first:

- A service is one general term defined for the UE network communication. The network offers a service to the UE, e.g. a broadcast service, a dedicated service, etc.
- A session is the type of data, e.g. one software module, one news package, etc, which is broadcasted in the network. The download session length can be defined.
- A packet is fixed a mount of data segmented from the session, each packet has its own index number in a session.

III.1 Introduction of software download mode and software download channels

It is necessary to introduce a software download mode due to extraordinary actions required in the software download procedures. The UE should decode the received packet if there is duplicated reception, reassemble the received packets, and report to the RNC. About the information transmitted in uplink, the CPCH channel (uplink common packet channel) can be used. The reason to choose the uplink packet channel is that there is no high data rate requirement, and it saves the spreading codes. So, spectrum efficiency is saved.

The software download link cannot always be maintained in a common manner in the CDMA based system due to the interference reason. After a certain threshold, the mobile station should inform the network the finished download in order to be successfully switched back to the dedicated channel.

As shown in Figure 5, the software download mode listening to the broadcast channel is established as one further mode for UE. It can be triggered by the detected download paging information (download status information) carried by a control channel. Once the download is finished, the terminal returns back to the idle mode. In the download mode, if a call is received, it is up to RRC in the RNC to interrupt temporarily the current download process and to switch to the connected mode. In case the connected mode is finished and the download is still not finished, the UE should jump back to download mode. If the download process through a broadcast channel is terminated according to principle shown, a connected mode should be established in order to complete download.

The software download control channel (SCC) broadcasts the current status periodically. If the current status indicates an ongoing broadcast, the mobile user registered in this traffic class or willing to listen to this class will decode the software download data channel (SDC) and carries through. In addition to have a higher probability to enhance the software download, a time shift between the control channel (SCC) and the broadcast data channel (SDC) should be also introduced. The following figure illustrates the structure of both channels.

The “repetition indication field” carries the current repetition number \( r \) of the session. It indicates the detailed current status of SD. It starts from the total repetition number \( R \) decided by the download management in the network. When one iteration of download session is finished, the value of \( r \) is decreased by one. The UE reads the \( r \)-value and estimate the amount of remaining packets to be downloaded. If \( r \) reaches 0, it indicates that SDC is
released/terminated and if the UE still have not finished the download, it should starts to establish a regular connection in order to finish the SD.

The SDC channel carries the packet indexes, session indexes, session status field, medium access control information and the download data, where the packet index and session index are of interest in this paper. The frame structure containing the session and packet index is shown in the figure below.

![Frame Structure](image)

The number shown in Figure 8 is seen as an absolute session index, the reason is to avoid the overlaps if two packets with the same packet number but belonging to different sessions are received. The protocol stacks in RNC and UE should support this approach. The Radio network controller (RNC) has the responsibility to organize the labeling for download traffic; the UE reassembles the received packets into a complete session. In case the download phase by means of repetition is finished and the user has not completed the download, he should apply for the dedicated radio resource. The download session can be received during different cell affiliations, i.e. handover. During lifetime of a broadcast session, fragment of the session are received.

IV. SIMULATION MODEL AND RESULTS

In order to test the feasibility of the SD management schemes, a system level simulation is launched.

IV.I Simulation Parameters

The related simulation parameters are shown in Table 2.

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel profile</td>
<td>UMTS 30.03 Vehicular</td>
</tr>
<tr>
<td>Network size for statistics</td>
<td>9 sites (27 cells)</td>
</tr>
<tr>
<td>Cell radius</td>
<td>2 km</td>
</tr>
<tr>
<td>Orthogonality factor in downlink</td>
<td>0.4</td>
</tr>
<tr>
<td>Mobile station speed</td>
<td>120 km/h</td>
</tr>
<tr>
<td>Downlink broadcast channel power</td>
<td>6 dB</td>
</tr>
<tr>
<td>Power control step size (Common Channels)</td>
<td>0 dB</td>
</tr>
<tr>
<td>Power control step size (Dedicate Channels)</td>
<td>1 dB</td>
</tr>
<tr>
<td>Active set size (Download Users)</td>
<td>1 cell</td>
</tr>
<tr>
<td>Active set size (Regular Users)</td>
<td>3 cell</td>
</tr>
<tr>
<td>Handover margin</td>
<td>4 dB</td>
</tr>
<tr>
<td>Inclusion hysteresis</td>
<td>4 dB</td>
</tr>
<tr>
<td>Replacement hysteresis</td>
<td>1 dB</td>
</tr>
<tr>
<td>Thermal noise power</td>
<td>-103dBm</td>
</tr>
<tr>
<td>Cell Grouping Factor</td>
<td>3</td>
</tr>
<tr>
<td>$T_i$</td>
<td>100ms</td>
</tr>
</tbody>
</table>
Download Traffic Model | Negative Exponential (average 100s)  
Lognormal Fading Variance | 7 dB  
Mobility model | Uniformly distribution over \(2\pi\)  
Packet Size | 300 bits  
Download Grouping Time | 100 ms  
Common Channel Capacity | 15 kbps

### IV.II Interference to Download Group

The system level simulation’s results shown Figure 9, illustrate the interference distribution resulted from the two approaches (cell grouping and no cell grouping). For instance, in the case that there are 20 users downloading software in each cell, it can be seen that the cell-grouping approach gives 7.5 dB gain at the 90% point. e.g. 90% users suffer 7.5 dB less interference by using the cell-grouping approach. This brings evident gain for the dedicated connections when they necessarily exist in the network.

![Figure 9](image)

**Figure 9, Comparison between the two approaches**

The comparison of interference to regular user is shown in Figure 10 below. It can be seen that the cell grouping approach can be designed independent on the user density. However, this only can be concluded if the handover process is not considered. Simulations show that the cell grouping approach always give much better gain than the no cell grouping case.

![Figure 10](image)

**Figure 10, Comparison between Cell Grouping and no Cell Grouping case**

It can be seen that due to power control function in the regular traffic group, the gain of cell grouping algorithm gives 4 dB gain, where 3 dB relatively less gain resulted from the gain brought be the power control function.

### IV.III Comparison of performance of finished download using Packet Indexing scheme

In the circular cell broadcast the handover process is tackled by introducing the packet-indexing scheme, results have shown that repetition increases the download probability and Packet indexing enhances the download. The performance is shown in table below.

<table>
<thead>
<tr>
<th>File Size (Kbytes/s)</th>
<th>Unfinished Download Probability after first iteration</th>
<th>Unfinished Download Probability after second iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9.71e-4</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0.0011</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0.0012</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0.0013</td>
<td>0.8e-6</td>
</tr>
<tr>
<td>500</td>
<td>0.0015</td>
<td>1.1e-6</td>
</tr>
</tbody>
</table>

### V. CONCLUSIONS AND OUTLOOK

This paper gives an introduction of cell groups for broadcast download in CDMA systems, which brings more interference-limited problem. Up to 7 dB gain can be reach in non-power controlled channels and 4 dB gain can be obtained in power controlled channels. However, it is evident that the cell-grouping strategy still brings added value to system with frequency reuse factor bigger than one. The downloading cell group size depends on the download profile, requirement, average size of download, and real time requirement. A good system gain can be found in heterogeneous traffic type i.e. the coexisting of voice service, data service embedded in dedicated channel and download in
broadcast/multicast channels. It is suitable to be applied in broadcast/multicast downloading case.

Furthermore the Packet Indexing scheme resolves the problem of unfinished download due to handover and radio bearer service selection. In case the bearer service has been changed for the UE, e.g. due to interference reason, and the download channel is changed from broadcast channel to dedicated channel, the UE still can apply for the missing packets to complete the unfinished download. A table of missing packet is sent to the network after the broadcast session. It also introduces a frame structure for the software download channels, i.e. software download control channel (SCC) in parallel with data channel (SDC).

Although circular broadcast software download approach highly reduce the interference in an interference limited cellular networks, it extends the software download time therefore the probability of successfully finished download decreases. Thanks to SD repetition in broadcast channel and packet indexing approach as radio resources control and link control protocols greatly enhancing the SD performance.

However the compromise between the interference reduction and efficient SD according to end user QoS requirements is still an interesting question to be answered.

References

[9]. ETSI TR 101 112, UMTS 30.03 version 3.2.0, April, 1998
Radio Resource Management Schemes Supporting Reconfigurable Terminals

J. Luo, M. Dillinger and E. Mohyeldin
SIEMENS AG, Werner v. Siemens Ring 20
D- 85630 Grasbrunn/Munich, Germany
E-mail: jesse.luo@icn.siemens.de
Tel/Fax: ++49-89-722-47452/26478

ABSTRACT

Future radio network must support emerging reconfigurable terminals to enhance the vitality of future mobile business. In general reconfigurable terminals are seen as the enabler for interworking among heterogeneous networks. We present a framework for joint radio resource management scheme supporting software download and regular traffic in networks, whereas the co-operation among co-existing sub-networks and interworking among different control layers are described. For terminal mass upgrades or content delivery to many terminals, a cell broadcast strategy suitable in cellular networks is introduced and some results are presented. We illustrate the impact of push services for software downloads over cell broadcasts in WCDMA networks. In the end, conclusions and future research topics are presented.

I. CHARACTERISTICS OF FUTURE NETWORK AND SERVICES

The future radio network meets challenges of high quality of service requirement by supporting high mobility and throughput for multimedia services. Besides user, service and network profiles, which must be taken into account for decision-making processes, the need for defining suitable terminal profiles is mandatory for reconfigurable terminals. Mobile users expect services, which will not only depend on a traditional single traffic type, but multiple traffic types even supported by simultaneous connections from different network types, e.g. cellular, ad-hoc, etc. On the other hand, the freedom of co-existing heterogeneous networks raises further questions of how to manage the traffic in networks in an efficient way. In the system management point of view, a simple physical layer processing restricted system cannot meet the co-existing multiple radio access technologies. A programmable environment, flexible and scalable network and traffic management is of high interests and expectations to emerge. It should support the complete network with convergence towards an IP based network and ubiquitous, seamless access among 2G, 3G, broadband and broadcast wireless access schemes, augmented by ad-hoc networks schemes and short-range connectivity between intelligent communication applications. On the other hand, we will have access to a variety of mobile terminals with a wide range of display sizes and capabilities. A mobile user might use handset access regular voice services in high mobility environment, however, s/he might use a laptop in local range environment. The ability offered by the emerging 3G WCDMA system, 384kbps for wide-area coverage and 2 Mbps for local-area coverage for a single band can be obtained. Because of the physical characteristics of cellular radio networks and requirement of users, the data rate of an ongoing radio links will also vary, complicating the joint radio resource management over heterogeneous networks covering cellular and decentralized networks.

I.I Networks Supporting Reconfigurable Terminals

The existing and future technologies must meet future network requirement, e.g. reconfigurable radio (terminals) [1][2], terminal capability improvement, upgrading the architecture of IP based radio network, dynamic radio resource management over heterogeneous networks, etc. System beyond 3G will take into account the interworking of heterogeneous networks that exist in the same operating area. It is expected that reconfiguration technologies will play an important role in the development of future wireless communication systems, as the emerging IP based network and several third generation radio access standards. The Software Defined Radio (SDR) concept has been an active research topic for many years encouraged mainly by the potential it offers to provide universal multi-mode terminal functionality within a single reconfigurable platform (baseband, RF front-end of the SDR terminals and protocols). The immense plethora of currently standardized Radio Access Technologies (RAT), both in 2G and 3G systems, forces the industry and academia to stress research in this field in order to utilize the potential available networks to the maximum efficiency. But SDR does also address reconfiguration of all layers. This trend is led by market requirements, especially user needs to roam seamlessly across network boundaries getting access to services anytime, anywhere with no concern of the underlying technology changes queried by the services in use [3][4].

I.II Basic Requirements on Reconfigurable Terminals

The majority of the published papers on reconfigurable terminal and network technology were originally focused on the so-called software defined radio aspects related to the digital front-end of the reconfigurable terminals and functionalities of the radio interface. The assumption of future reconfigurable terminals should be
able to compare all the available RATs at any time and use that one that fulfils a number of quality of service criteria as well as the user preferences. As 3GPP documentation [5] describes, there are defined four types of possible multi-mode 3G terminals according to their capability of concurrent reception and monitoring more than one mode. Among the four types, the higher the index number is, the more complex and more expensive the terminal is. A reconfigurable terminal should be at least equivalent to the third type of 3GPP terminal, i.e. one Tx and several Rx chains per terminal to allow faster scanning. The relevant research work with respects to the terminal restrictions is carried out. Technology and system/network researches identify the system support concepts, enabling technologies and standardization required realizing the scenarios, and through subjective evaluation, system modeling and simulation evaluate the feasibility of the proposed solutions. It is important for reconfigurable systems, that the challenges of spectrum sharing techniques, joint radio resource management techniques, mode identification/switching, reconfiguration management and software download (SD) management must be solved.

I.III Overview on Network Architecture

Reconfigurable terminals and the local reconfiguration procedures must be supported by network architectures and their user and control plane functions. In addition to the multiplicity of air interfaces in future systems, reconfigurable terminals must be managed by the network infrastructure in different air interface cases. One instance of centralized network architecture is based on a network-centric architecture (in Figure 1) involving the association of Home Reconfiguration Manager (HRM), Serving Reconfiguration Manager (SRM) and Proxy Reconfiguration Manager (PRM) [2]. This architecture is useful for cellular networks and provides a centralized software distribution [6]. A decentralized distribution from terminal to terminal suitable in ad-hoc or hybrid (interworking between cellular and ad-hoc) networks can also be similarly established [7].

Interactions between terminal and network are crucial, as the available bandwidth on the wireless link is a limited resource that should be used for user data rather than negotiations. Furthermore, resources on the terminal itself are usually also limited. In order to relieve the terminal from the burden of frequent interactions with network entities, information from the network could be generally obtained via the PRM, which is located in the radio access network. It serves as a proxy instance for negotiations with other network entities, in particular the proxies enhancing software download, i.e., the SRM and the HRM.

II. ARCHITECTURE AND FUNCTIONS OF JOINT RADIO RESOURCE MANAGEMENT

In order to fully exploit the flexibility provided by the reconfigurable terminals and networks capabilities, further research on advanced RRM, combining both classical RRM and advanced spectrum management needs to be investigated. This aims at providing efficient solutions for RRM in a composite radio environment, supporting multiple RATs in different network topologies (hierarchical, decentralized) and moreover being potentially managed by the same or different operators. This includes spectrum management for asymmetric regular traffic, measurement and criteria for inter-system (vertical) handovers, design of potential collaborative RRM schemes considering solutions of spectrum sharing between operators, and flexible spectrum allocation in a context of re-configurable equipment and self-organizing networks.

The presented management architecture and strategy is based on the assumption of co-existing different RATs with different profiles. The estimated traffic types and their volume are useful in dynamic usage of a fixed radio resource for a sub-system. The load information and traffic information are required to be shared by cooperating networks. The interworking of the cooperative system is illustrated in Figure 2. Each radio access network needs an efficient interworking between traffic volume, measurement (prediction) function, traffic scheduler, load control unit and admission control function. The Traffic Estimation module (TREST) in each system informs the administrative entity Session/Call Admission Control (SAC) on the predicted traffic and planned traffic information to update the priority information for each connection and the admission decision within the sub-network. The priority information is an input vector for the scheduling algorithm in lower layer. The load balancing between software download traffic using broadcast/multicast channels and regular traffic over heterogeneous networks should be performed by a centric intelligent entity.

Apart from the previous overview of near-term joint radio resource management algorithms based on fixed spectrum pre-assignment, a full benefit of SDR technology can be reached by employing new spectrum engineering techniques. The spectrum allocation can evolve from the current static one to the complete dynamic and flexible spectrum allocation; i.e., dynamic allocation and sharing of spectrum between radio access technologies and operators.

II.1 Interworking between Different Sub-Networks

The interworking between different RATs requires new protocols defined for convergence reasons. It should offer IP packet based convergence sublayers to networks to guarantee QoS. Due to the heterogeneity of coexisting different networks many different policies are conceivable for joint management functions, in particular when considering legacy and new network types. Systems in different generations are equipped with different functionalities, protocols and management requirements. For future terminals having simultaneous connections to different RATs is one
possible operation mode. In general, loose up to tight
coupling schemes between different network types must
be considered for such multiple connections. For a
possible tight coupling between UMTS subsystem and
wireless LAN, one must consider the restrictions in each
sub-systems, e.g. the transport block size and minimum
transmission time interval for each are differently
defined according to the specifications. Tight coupling
allows joint scheduling of traffic streams between
involved networks and terminals. A joint interworking
must also take into account the user, service, network
and terminal profiles, which consists of static and
dynamic features.

A two-stage admission control approach is shown in the
following sub-chapter (Figure 3). It is introduced to
define the handling of network operation related with
the static and dynamic network, service, terminal and
user profiles. The first stage handles the static, whilst
the second stage handles the dynamic features, e.g.
current QoS of each user. In a cooperative environment
with different coexisting subsystems, the joint radio
resource management working with different scales will
enhance the utilization of resources. The spectrum can
be shared by a dynamic allocation scheme; the load can
be balanced by a joint admission control and load
control scheme; even lower layer resource management
function e.g. power control, resource scheduling (RS)
algorithms can jointly contribute the spectrum
efficiency.

The conventional admission control is designed for each
access system working independently among coexisting
access systems and RATs. In the heterogeneous
cooperative environment, a joint Session/call Admission
Control must be defined. The neighbor RAT system
load is taken into account by the Joint Session
Admission Control (JOSAC) as shown in Figure 2. The
traffic stream can be routed alternatively through the
cooperating sub-systems according to the restriction and
the advantage of each. E.g., the wide coverage of
universal cellular system, e.g. GSM, UMTS; whereas
the high transmission rate can be obtained though
wireless LAN. With the information of estimated load
in all the sub-networks (dynamic network profile), the
Joint Load Control entity (JOLDC) located together
with JOSAC will distribute the traffic based on the
characteristic of the co-existing RATs (static and
dynamic network profile), the QoS requirements for the
service and the number of applicants for the software
download service to determine the software download
strategy, i.e., which RAT and channels with committed
capacity should be selected. The Joint Resource
Scheduler (JOSCH) is important for terminals having
simultaneous connections to different networks, i.e.
when a tight coupling between networks is foreseen.
JOSCH is responsible to schedule traffic streams being
split over more than one RATs. It helps to optimize
utilization of radio resources in the whole system. It also
synchronizes the stream being split, e.g. video stream
with basic layer and enhancement layer being
transmitted over different air interfaces individually or
separated main object and inline objects of HTTP
service belonging to the same session, etc.

The SAC entity in each sub-network consists of
Software Download Traffic Control entity (SDSAC)
and Regular Traffic Control entity (RESAC). Based on
the time schedule and QoS requirement of the SD traffic
the SDSAC assigns an optimal broadcast or multicast
bearer service to distribute the download to the
terminals. The load generated by this push service
traffic and regular traffic should be jointly balanced by
SDSAC and RESAC by taking an appropriate SD
strategy to determine the needed common channel
capacity defined by the JOSAC.

In case the single traffic is split into two streams carried
by two RATs in tight coupled manner, the joint
admission control should previously give the admission
to the connections in both systems, in order to allow the
tight scheduling algorithm work jointly. In this tighter
coupled case, queues for each sub-systems will be filled
up by amount of data coming from the same traffic
stream.

II.II Interworking among Different Radio
Resource Management Layers

Because of the requirement for heterogeneous services,
the queuing of data stream, priority arrangement should
be executed by TRSCH. Some important input
information required by admission control should also
be offered by TRSCH. The trade off between
maximising the utilisation of the system and reducing
rate of dropping or negatively affecting the QoS demand
of users should be considered to design an admission
control algorithm. The interworking between admission
control and RS are shown in Figure 3. The incoming
traffic in side of the systems is divided into different
traffic types after the first stage, the traffic types with
different QoS requirements is attached, i.e. the real time
requirement, the throughput requirement, etc. The
location of users is part of the dynamic terminal profile
information to decide on the bearer services to be
assigned. Therefore, SAC should select the transmission
physical mode of the bearer service or drop the
application in case the network cannot provide the
requested service. Based on the chosen static service
and network profile in the first stage (JOSAC), the first
stage of the admission control assigns a certain range
for the weights defined for the service types based on
the network, terminal and user profiles, which are
offered to the second stage in the SAC. In case the
incoming calls ask for service beyond the static
restrictions, the JOSAC should reject them immediately.
The tight-coupled traffic stream over two RATs should
be scheduled by JOSCH which works between JOSAC
and SAC. The split traffic after JOSCH should be
forwarded to individual SAC in each RAT defined sub-
network with delay bounds. With the offered control
information from JOSCH, SAC should map the split
traffic into conventional traffic type with a concrete
priority weight (PW).
As introduced in Figure 3, the QoS class dimension (Y) of the scheduling allocates resources for different QoS classes. The cost function that the admission control is based on requires the priority weight (PW) for each traffic type. In this case, the cost function parameter will be harmonized with the Y-dimension scheduling algorithm. The principle of generic process sharing algorithm (GPS) [8] can be applied to guarantee the resource for the sessions with different priority; i.e., a minimum service rate can be obtained as the ratio between the committed weight to the sum of all. In addition, the PW can also be applied into the calculation of allocated resource for the committed service type according to the weights assigned for this type. The weighting vector can be handled by the network operator to offer dedicated radio resource to particular users. Number of transport channels are mapped after the scheduler, where the common/shared channels carrying software download traffic and dedicated channels are included. Different multiple access schemes represented by transport channels are also shown in Figure 3.

II.III Software Download Strategies

As explained in the previous chapter, the software download management is one important issue for SDR system. The impact of mass download to the regular traffic in wireless network is an important issue, which is tightly related with the peer to peer (P2P) QoS. For a soft blocking problem restricted system, the impact between SDR related traffic and regular traffic can be modeled by additional interference to evaluate the impact of the chosen software download strategy. In general we must consider point-to-point and point-to-multipoint software downloads. Point-to-multipoint can be carried out by a shared or common channel in a network. For many users a broadcast channel with a certain data rate is of particular interest here for the cell grouping strategy discussed in the next chapter. The software can be downloaded from a PRM in the network, or be retrieved from another terminal (decentralized download schemes), or from local databases in the terminal, i.e. libraries, which may contain the required module from a previous reconfiguration. It is also important that the detailed SD status should be negotiated between SD module and the PRM module.

The download channel utilization and download interference reduction are of particular interest, when choosing a download strategy [9]. The probability of using broadcast channels for downloading software to mobile terminal (MT) is rather high, when many users want to receive the same information or mass upgrade of terminals is taken place. Broadcast channels are not fast power controlled and increase the interference in surrounding cells. If the broadcast with a certain data rate is used in all cells, then the cell capacity is further decreased due to the cell coupling in CDMA system with frequency reuse one. Cell coupling means that an increase of transmission power in one cell will lead to an increase of transmission power in neighbor cells due to the need for maintaining the target SIR for services. To minimize the additional interference due to software download when using e.g. cell broadcasts a specific download management for all involved cells in a geographic area must be applied. The idea of spreading downloading interference into time division by cell grouping strategy will highly decrease the interference to the regular traffic and the download traffic. Also parallel downloads for different cell groups can be applied. In the other words, in cells with a low cell coupling factor, e.g. cells in a macro environment, which their 120 degree sector antennas are oriented towards one geographic direction can be clustered into one download group [10][11]. It can be proven that certain gain has been obtained by the cell-grouping strategy. A higher download probability can be achieved due to repeating download sessions [11].

III. CONCLUSIONS AND OUTLOOK

An overview on the aspects for joint radio resource management considering also download aspects for reconfigurable terminals is provided in this article. Also, architecture in the network supporting reconfiguration and download for terminals are reflected. For the future research topics on this domain, two main directions are of interest.

For the first direction, number of research items are listed as the following: procedures for terminal reconfiguration; network architectures and functions for reconfiguration management; software download and P2P interaction in cellular and ad-hoc networks supporting QoS for software download based on IP-based transport; mobility management; terminal-centric and network-centric reconfiguration supervision; flexible adaptive protocols and negotiation protocols; wireless middleware and infrastructure services to support reconfigurability and peer to peer applications; terminal reconfiguration concepts; modeling of push services; traffic models for software download based on different session layers as email, WAP, HTTP, SOAP to include the protocol investigations; secure software download including authentication, capability exchange and integrity assurance.

The joint RRM is the second direction which aims at providing efficient solutions for RRM in a composite radio environment, supporting multiple RATs in different network topologies and, moreover, being potentially managed by the same or different operators. The main RRM research topics are the following: RRM for asymmetric regular traffic; research on vertical handover procedures based on the network architectures and mobility management scheme; Inter-system handovers measurement and criteria, considering cellular (UMTS/FDD, UMTS/TDD, GSM, unlicensed TDD), WLAN (Hiperlan/2), PAN (Bluetooth) RATs, and use of multicast and broadcast in cellular systems; design of potential collaborative RRM schemes considering also solutions of spectrum sharing between operators; flexible spectrum allocation in a context of...
re-configurable equipment and self-organizing networks; joint design of RRM schemes over different management layers; the vertical depth of management layer penetration for co-existing different technologies; interworking between heterogeneous networks with considerations of cooperative and non-cooperative operators.

ACKNOWLEDGEMENT

This work has been partly performed in the framework of the IST project IST-1999-12070 TRUST, which is partly funded by the European Union. The authors would like to acknowledge some contributions of their colleagues from Siemens AG, France Télécom – Research & Development, Centre Suisse d’Electronique et de Microtechnique S.A., King’s College London, Motorola Ltd., Panasonic European Laboratories GmbH, Telefonica Investigacion Y Desarrollo S.A. Unipersonal, Toshiba Research Europe Ltd., TTI Norte S.L., University of Bristol, University of Southampton.

References


Figure 1, Network-Centric Architecture
Figure 2, Joint Radio Resource Management Architecture

Figure 3, Two-Stage Admission Control and Resource Scheduling
A smart space enabled SDR environment for enterprise and home networks
Satish Jamadagni, Sanjay Jamindar
(satishj, sanjam@sasken.com)
Sasken Communication technologies Ltd
#139/25, Amar Jyoti Layout, Ring road,
Domlur P.O, Bangalore – 560071, India

Abstract
Recently there has been an increase in the number of wireless access technologies for the enterprise and home environments like WLAN, Bluetooth, IRDA, diffuse optical wireless etc. It will not be surprising if large enterprises evolve their own indoor access standards in the future driven by security or any other specific concerns. As communication and computing technology continues to become increasingly pervasive and ubiquitous, we also envision that enterprise or home environments will be embedded by a variety of sensors able to sense and support our daily activities. Such environments are termed as smart or aware environments [1] with sensors aiding in the capture of computing or communication context of a device or a person. In this paper we outline an architecture suggesting how such environments help achieve soft radio functionalities in the face of availability of multiple modes and multiple bands in an enterprise scenario. We discuss the nature of the emerging smart or aware environments and discuss their interface with the radio access link layer to enable “soft radioness” in enterprise environments.

Introduction
A Software Defined Radio (SDR) is a multi-band, multi-mode radio with dynamic capability defined through software in all layers of the protocol stack including the physical layer. Recent evolution of 3rd Generation telecommunication networks has brought to focus the needs for reconfigurable or soft radio terminals. Also the recent proliferation of low range, high bandwidth wireless enterprise technologies like Bluetooth, Wireless LAN technologies, Optical wireless access systems and emerging enterprise access technologies like ultra wideband technologies demands appropriate interoperable technologies to be in place. The more immediate benefits of SDR technology lies in providing interoperability solutions to the number of indoor or enterprise wireless access technologies emerging in the market today. The interoperability demands are due to the fact that these technologies provide greatly varying service experiences and in varied geographic localities. Enterprise wireless access type availability can vary based on a number of parameters like security (point to point optical is preferred), presence of ambient light and excessive shadowing (RF is preferred) etc.

Enterprise wide inter-technology switching and inter-technology handoff support will be a critical element of any indoor wireless access deployment. SDRs will provide the much-needed support for the co-existence of these technologies. The availability of a sensing and signal-understanding infrastructure or smart spaces in enterprise networks makes the realization of software defined radio technology that much more simpler.

What are software radios?
The term “Software radio” is not precisely defined. In this paper we understand software radio as being a reconfigurable platform, which can be adapted to specific requirements ex. different standards or services by simply downloading the specific software. Software defined radios can also be visualized as enabling seamless mobility across heterogeneous access networks. With the emerging all IP wireless networks, reconfiguration in SDRs will primarily focus on lower layer reconfiguration. The term software radios is some time used inter-changeably with multi mode radios as they are emerging within the 3G standardization [4] efforts. To facilitate seamless mobility,
Software defined radios are expected to provide inter working mechanisms between various wireless technologies. It would not be inappropriate to term software defined radios as enablers for Ubiquitous communication. The immediate need for ubiquitous communications is far greater in enterprise networks than in the wide area. This makes the exploration of methods to achieve software-defined radios within enterprise networks an immediate challenge. The SDR challenge is more relevant to the enterprise space with the emergence of a host of wireless access technologies and standards specifically for the enterprise space. In this paper we discuss achieving software defined radio objectives with the help of smart spaces [1]. The integration of smart spaces and wireless protocols can enhance upper layer (Link layer and above) intelligence and aid in lower layer reconfiguration.

**Smart and Aware Environments**

Smart spaces or smart environments [1] visualize environments that can sense our activities with the help of a variety of low cost sensors. Many articles [5] [6] [7] discuss the importance of a sensing and signal-understanding infrastructure that leads to awareness of what is happening in an environment and how it can best be supported. Such an infrastructure supports a variety of low-end sensing allowing for detailed interpretation, modeling, and recognition from sensed information. Smart spaces also attempt to compose extremely small, low-power computing and wireless communications devices available in an enterprise environment into sophisticated, ad hoc and cooperative computational and communications structures. A significant aspect of an aware environment is to developing techniques for processing and analyzing the captured sensory data to provide meaningful interpretations. This is significant for the specific purpose of supporting SDR functions in an enterprise.

A network of sensors that is configured with a network of processing devices can yield a rich multi-modal stream of sensory data. Sensory data is, in turn, analyzed to determine the specifics of the environment and provide context for interaction between collocated and distributed users and environments. Such analysis is also useful to determine what is happening in an environment, so that it is supported effectively. By the definition of smart spaces we are specifically interested in high-end interpretations that support SDR functionalities. In the following sections, we discuss how such sensory data can be interfaced with protocol elements to achieve “soft radioness” in enterprise networks.

Typical sensors in our definition of a smart space cover **Optical Sensors** for ambient light measurements, **Identity sensors** for user identification, **Location sensors** for object or user tracking, **Activity, behavior and Pose Tracking sensors** for recognition of short-term or long term activities of a user or an object, **Audio sensors** for noise levels or speech recognition applications.

A typical smart space is shown in figure-1 consisting of a layer of sensors and a management layer. The sensor layer embeds a variety of sensors, the management layer embeds the interpretation and modeling mechanism. The smart space management entity maintains “complete context” information about every node in its coverage area. The context information covers sensory data like location, activity in terms of applications or usage of a device, behavior information etc. This context information is used to allow for sophisticated functions like:
Location sensor enabled “Neighbor discovery”: Nodes can query for location and neighbor information or emit beacons and establish the set of directly reachable nodes. Nodes can then exchange information (such as ID, type, capabilities, interests etc) to determine which links to establish and keep alive.

Self organization and federations: Nodes need to self organize into a network by forming hierarchical clustered “federations”. Federations provide the basic network-level infrastructure over which routing and QoS can occur. Federation topology will have to adapt to the dynamic behavior of mobile nodes and environment conditions as seen by the smart space.

Resource discovery: Communicating nodes may need a variety of resources to operate, such as content servers, caches, and alternate access network availability in a region. The smart space management layer can notify mobile nodes about available resources that will facilitate its functioning.

User centric communications and session protocols: Communications should be based more on user context (behavior, activity, pose etc) and not on physical nodes or locations alone. Thus protocols, resource location, and agents are needed to allow users and applications to communicate with one another by name, with content encoding and location handled transparently. A session is a collection of application associations, and may involve a variety of network flows and connections through a set of federations. Sensory data can be interfaced with session protocols to handle dynamic behavior, including join/leave and merge/split of individual multipoint multiflow sessions.

Reflective base stations / access points for enterprise SDR realization

Future mobile communication devices need to be “reflective” in order to make decisions about a mode to use given an environment context. A high degree of environment awareness is necessary for such decisions. The environment sensing is supported by the smart space. In existing mobile devices sensing takes the form of received signal strength measurement to support mobile assisted hand over. In emerging SDR devices such sensing efforts across the spectrum might prove to be extremely costly and as such relying on a smart space for such measurements is a viable option. Emerging wireless technologies like diffuse optical wireless technology will have to rely on external measurement of parameters like shadow and ambient light intensity, which affect its link performance. As such a fusion of smart space sensing ability with wireless communication protocol entities looks inevitable.
Mode switch decisions will be greatly enhanced with the availability of smart environments. The smart space can force mode switch decisions or can just facilitate such a switch. In this paper we discuss both cases. We also discuss link layer interfaces with the smart space for sensory data use by a protocol entity.

We have implemented an access point controlled reflective handoff, which is assisted by a smart environment. We consider access points of a host of wireless communication technologies like wireless radio access points, wireless diffuse infrared access points etc. Access points transmit beacons that additionally carry globally unique identifiers designating specific access networks. Mobile nodes transmit unique QoS requirement identifiers that can be mapped across these heterogeneous networks within an enterprise network. The sensor network data is used to achieve automatic (no user intervention) mode control and reconfiguration.

A reflective detection algorithm resident at the access network entity uses access network identifiers to determine whether a mobile device is likely to move to the coverage area of a new access network. Every access point is aware of the coverage areas of each access point as well as access points that implement or support other access technologies. The smart space management layer is also aware of these access points. The reflective detection algorithm can also reside at the smart space management layer.

Each mobile device maintains a local cache of access system modules. Access system modules are collections of objects supporting specific access point signaling services in mobile devices. Before a mobile device performs a handoff to a new access network, it checks whether a access module associated with the new candidate access network is cached. If an access module is not cached it is dynamically loaded through a download process. Access points support module loaders as application layer entities.
An access system module is loaded from the old access network. A two-way handshake mechanism is used for loading access modules. Such handshake mechanisms are loaded before reflective handoff management protocol stacks are invoked at mobile devices. Two distinct types of access points support reflective handoff in our test-bed. Diffuse infrared access points and a RF based WLAN access point.

We have used a Mobile IP based test-bed. Mobile IP [8] delivers fast handoff control in datagram oriented access networks. In addition, Mobile IP supports per-mobile host state, paging, routing and handoff control in a set of access networks that are interconnected to the Internet through gateways. Mobile IP signaling modules use the IP protocol to communicate with access networks. Both WLAN and diffuse infrared access points support Mobile IP in our test-bed.

In our implementation the shadow and ambient light calculations achieved by sensors of a smart space help in decisions on switching to an RF mode from an Infrared (IR) mode or vice versa. Security considerations in specified locations can force a terminal to switch to a point-to-point IR mode. Bandwidth considerations can force a diffuse optical mode on a set of terminals. A mobile terminal using some debugging applications on a device can switch to a WLAN mode without loosing its connectivity to other services like some notification services. Reflective mode switch support through smart space assisted measurements helps provide a user with consistent QoS feel.

In the figure above the smart space location sensors and light intensity measurement sensors are used to achieve a handoff between an RF mode a diffuse infrared mode. Under shadow or intense ambient light conditions a diffuse infrared mode might suffer link degradations. Under such conditions an RF mode would be preferable. Under conditions when diffuse optical wireless systems can be used they are preferred due to the very high bandwidth (QoS) supported. The smart space measurements on light intensity across a floor help achieve such mode decisions and a better QoS feel for the user. Figure–4 shows typical shadow regions where diffuse infrared mode usage would lead to degradation of services. A smart space assisted reflective access point will notify the mobile terminal of the availability of alternate access technologies as well as the difference in QoS offered by the other technology in a given location area.
Subscribing to sensing services
In order to utilize the smart environment capabilities a mobile device or the communication access point must be able to query a smart space management entity for service and subscribe for a set of services. A context transfer mechanism between the communication access node and the Smart Space Management Entity (SSME) is defined once such a subscription handshake is complete. Sensor notification should be across the stack instead of any one-entry point and this is reflected in our implementation. The mobile IP entity can directly subscribe to location [7] notifications from the smart space entity. Shadow notifications can be directly handled by the link layer etc. Neighbor discovery notifications from the smart space can be handled directly by a service discovery protocol entity. Applications could query a smart space management entity for user behavior attributes for specific application use. A mobility management entity can query the smart space management entity for alternate mode availability in a given location region. Location tracking information could be used to enable applications or disable a set of services to a mobile node. Our implementation is restricted to achieving handoff support between RF and infrared access points through light intensity measurements by the smart space measurement sensors. Such a light intensity notification from a smart space entity is handled by the diffuse infrared link layer to achieve better power control.

Concluding remarks
In this paper we implement a smart or an aware environment that can sense locations and light intensity. This information is used to achieve a switch between an RF mode and an optical wireless mode. It is argued that a smart space can be used to provide for soft radioness in enterprise and home network environments. In this paper we provide a simple smart space architecture that can support SDR functionality in indoor environments. We present a specific reflective mode switch support between an optical and RF wireless technologies.

Smart spaces can also support selective location and context based download of modes. In enterprises requiring high security, secret modes might be forced onto mobile terminals in specified locations. Support for such modes might not be available in other locations. Such locations should also provide for the dynamic download of the pertinent modes and a later uninstallation of such modes once a user is leaving such a location.

The notion of enterprise wide smart spaces can be extended to the wide area. With advances in wide area location tracking [7] and other technologies it is possible to define a wide area smart space to aid SDR deployment in the wide area.
References:


