The 9th International Conference on ITS Telecommunications has been held on October 20-22, 2009 in LILLE (North of France) at Nouveau Siècle congress centre. After Phuket in Thailand, the conference is returning to France. It has bring together engineers and scientists in the field of New Technology of Information and Communication for Intelligent Transport Systems (ITS).

The reduction of the impact of transport on the physical, social and human environment is a key challenge for sustainable mobility and development of urban areas. NTIC for ITS will contribute to the deployment of solutions that aim to optimize the use of existing infrastructures (road, rail, maritime, river), to enhance safety and security, to reduce operating and maintenance costs and to offer new services to customers and staff developing inter modal behaviour. ITST conference and its topics are attracting more and more players, academic and industrial researchers, working on Intelligent Transport Systems (ITS). The theme of this year event is “NTIC for ITS to increase traffic efficiency and optimize transport modes under safety and security constraints”.

Works accepted for presentations and poster sessions of ITST 2009 are balanced from the academic and industrial research community around the world. There are keynote speakers and invited papers for plenary sessions. Technical exhibitions and demonstrations has been held in the conference centre. Technical visits will be proposed. We have also invited recognized experts from the ITS community as keynote speakers and presenters to enrich the participation.
COMMITTEE

GENERAL CHAIR

Chair: BOURGEOIS Guy, INRETS, General Manager (France)
Co-Chairs: SHOZO Komaki, Osaka University (Japan)

STEERING COMMITTEE

HORICHI Hiroki, KDDI (Japan)
ITAMI Makoto, Tokyo University of Science (Japan)
LANDRAC Gabrielle, Institut Telecom/Telecom Bretagne (France)
LENARDI Massimiliano, Hitachi Europe SAS (France)
LU YiLong, NTU (Singapore)
NAJARIAN Paul, Dep. of Commerce, NTIA/OSM (USA)
WEN Guangjun, UESTC (China)
PRATHOMBUT Passakon, NECTEC (Thailand)

INTERNATIONAL ADVISORY BOARD

AMARGER Stéphane, Hitachi Europe SAS (France)
CHOMETTE André, Institut Telecom/Telecom Bretagne (France)
FINGER Ulrich, Institut EURECOM (France)
FUJISE Masayuki, NICT (Japan)
IMA Hideki, Chuo University (Japan)
KAWASHIMA Hironao, Keio University (Japan)
LYE Kin Mun, I2R (Singapore)
MEDEVIELLE Jean-Pierre, INRETS (France)
OHMORI Shingo, NICT (Japan)
THAJCHAYAPONG Pairash, NSTDA (Thailand)

TECHNICAL PROGRAM COMMITTEE

Chair: BERBINEAU Marion, INRETS (France)
Co-Chair: ITAMI Makoto, Tokyo University of Science (Japan)
WEN Guangjun, UESTC (China)

Members:
AGUADO Marina, Basque University Bilbao (Spain)
BONNIN Jean-Marie, Telecom Bretagne (France)
CANAVERO Flavio, Politecnico di Torino (Italy)
CHEN Ling-Jyh, Academia Sinica (Taiwan)
CHOADIT Asawatkul, CU (Thailand)
CHINRUNGRUENG Jatuporn, NECTEC (Thailand)
CHOI Wan, KAIST (Korea)
DENIAU Virginie, INRETS (France)
ERNST Thierry, INRIA (France)
FICHEUX Serge, UTAC (France)
FESTAG Andreas, NEC Europe SAS (Germany)
GARNIER Christelle, Telecom Lille1 (France)
GILLIERON Pierre-Yves, EPFL (Switzerland)
GRANSART Christophe, INRETS (France)
HORICHI Hiroki, KDDI (Japan)
KAZI M. Ahmed, AIT (Thailand)
KOSCH Timo, ICN (Germany)
KRAIRIKSH Monai, KMITL (Thailand)
KUNG ANTONIO TRIALOG (France)
LE DEVENDEC Marc, ERO (Denmark)
LE ROUX Yvon-Marie, Telecom Bretagne (France)
LIENARD MARTINE IEMN (France)
LIN LAN, Hitachi Europe SAS (France)
MARAIS Juliette, INRETS (France)
MEYER ZU Hörste Michael, DLR (Germany)
PARK Jong-hyun, ETRI (Korea)
PENSIER Guillaume, CETMEF (France)
RIVENCQ Atika, IEMN (France)
SHAO Zhenhai, Futronic Tech. PTE LTD (Singapore)
SHANKAR Jaya P, I2R (Singapore)
TAK Sungwoo, Pusan National University (Korea)
TATKEU Charles, INRETS (France)
TOMOYUKI Yashiro, Chiba Institute of Technolog, (Japan)
VAN DEN Abeele Didier, ALSTOM-TIS (France)
VAUZELLE Rodolphe, XLI-M-SIC (France)
WENDONG Xiao, I2R (Singapore)
ZHANG Yan, Simula Research Lab (Norway)

ORGANIZING COMMITTEE

Chair: JOIGNAUX Guy, INRETS (France)

Secretaries:
BRUNIN Sandrine, ERT (France)
DAVOUST Corinne, INRETS (France)

Members:
BOURBOTTE Daniel, INRETS (France)
COCHERIL Yann, INRETS (France)
COUDOUX François-Xavier, IEMN (France)
DAYOUB Iyad, IEMN (France)
FUJITA Seitaro, NICT, Paris office (Japan)
GAILLOT Davy, IEMN (France)
GHYS Jean-Pierre, INRETS (France)
MASSON Emilie, INRETS (France)
NAKAGAWA Fumihiko, YPR (Japan)
NOYER Jean-Charles, LASL-ULCO (France)
RIOULT Jean, INRETS (France)
SIMON Eric, IEMN (France)
WAHL Martine, INRETS (France)
FPGA-based Vehicular Channel Emulator for Evaluation of IEEE 802.11p Transceivers

T. M. Fernández-Caramés, M. González-López, L. Castedo
Dep. of Electronics and Systems
University of A Coruña
A Coruña, Spain 15071
Email: {tmfernandez, mgonzalezlopez, luis}@udc.es

Abstract—We present a hardware vehicular channel emulator aimed at assessing the performance of IEEE 802.11p transceivers in realistic scenarios. Contrarily to commercial emulators, which are usually very expensive and not quite flexible, we have developed an FPGA-based solution that is cheap, flexible and reconfigurable. In this paper we outline its design, implementation and basic operation. To show its capabilities, we present two examples of performance evaluation: we measure the bit error rate and packet error rate of an IEEE 802.11p system when performing transmissions between two vehicles that are approaching at high speeds in a highway and in an urban canyon.

I. INTRODUCTION

In recent years, wireless communications between moving vehicles (Vehicle-To-Vehicle, VTV) and from vehicles to infrastructure (Roadside-to-Vehicle, RTV) have received a great deal of attention due to the increasing demand of applications, mainly aimed at providing safety and commercial services.

The IEEE 802.11p wireless standard is one of the proposals that defines a suitable architecture for operating in vehicular environments. It is based on the specifications given in [1], that describe medium access control (MAC) and physical (PHY) layers very similar to those used in the wireless local area network standard IEEE 802.11a [2]. Slight changes are included in order to allow working in high delay spread scenarios, typically present in urban canyons.

The main advantage of IEEE 802.11p over IEEE 802.11a when overcoming the effects of vehicular channels, is its bandwidth. The 20 MHz bandwidth used in IEEE 802.11a is reduced to only 10 MHz in IEEE 802.11p. Thus, the OFDM (Orthogonal Frequency Division Multiplexing) symbols are longer in the time domain and the system can deal with large delay spreads, being able to avoid ISI (Inter Symbol Interference). Moreover, other improvements have been made in the IEEE 802.11p MAC and PHY layers to address vehicular communications and to increase the range with respect to that of IEEE 802.11a.

To assess the performance of an IEEE 802.11p transceiver, it is very important to carry out an evaluation in realistic situations. The tests can be performed directly in a vehicle, driving through different environments, but this is a time-consuming task and the experiment can be affected by unintended side effects that may be uncontrollable. It is more convenient to use a hardware channel emulator and measure the performance inside a testing lab with a testbed platform like [3]–[5]. However, commercial channel emulators are usually very expensive and do not offer enough flexibility when configuring the wireless channel parameters.

Motivated by these observations, we have built a cheap and flexible alternative to evaluate the performance of IEEE 802.11p transceivers in vehicular environments: our emulator can be reconfigured fast and easily, providing enough flexibility to implement customized wireless channel models. To achieve these goals, we have used an FPGA (Field-Programmable Gate Array) and Xilinx System Generator, that allows us to develop the software model much faster than using hardware description languages like VHDL or Verilog.

The implemented channel models are based on [6]. There, the authors obtained channel models based on data acquired during a measuring campaign carried out in a metropolitan area at a frequency of 5.9 GHz. Six different situations are contemplated, including measures in common vehicular environments such as a highway, an urban canyon and a suburban area.

II. STATE OF THE ART

Channel emulation has been typically used when evaluating product performance in realistic situations before commercial release. With the aid of a real-time channel emulator developers avoid unintended interferences, hence the simulation environment can be controlled. Furthermore, the tiresome task of performing successive field measurements is kept to the minimum (i.e. obtaining the channel model, if there is none already available) and the rest of the experiments are carried out inside a testing lab.

There are many commercial channel emulators that are manufactured by companies such as Spirent [7], Rhode & Schwarz [8], Azimuth Systems [9] or Agilent [10]. These emulators are usually general-purposed (for instance, Spirent’s SR5500 or Rhode’s AMU200A), but there are some that are aimed at evaluating a specific technology, like Azimuth’s 400WB MIMO Channel Emulator (for IEEE 802.11n and Mobile WiMAX Multiple-Input Multiple-Output systems) or Agilent’s N5106A PXB MIMO Receiver Test (with built-in LTE and Mobile WiMAX channels).
All these channel emulators are robust and work well for most applications, but they are normally quite expensive and may not offer enough flexibility to researchers when setting channel configuration parameters. To tackle these issues, low-cost ad-hoc channel emulators have been recently proposed.

To develop a low-cost and easily-reconfigurable channel emulator different technologies can be considered. Due to real-time constraints, microcontrollers and affordable DSPs (Digital Signal Processors) are not valid. Among the rest of the semiconductor technologies, three of the them are specially suitable to implement a channel emulator: CPLDs (Complex Programmable Logic Devices), ASICs (Application-Specific Integrated Circuits) and FPGAs. Although ASICs offer superior performance, their developing time is too high for a prototype (they are mainly used for final products). Thus, the choice must be made between CPLDs and FPGAs. The latter are the ones commonly used due to two reasons:

- FPGAs own specific resources that are more adequate for implementing a channel emulator (counters, arithmetic operator blocks...).
- Although CPLDs can execute tasks at a faster rate, the maximum allowed complexity of the designs is inferior to the one offered by FPGAs.

Therefore, most of the proposed ad-hoc emulators are based on FPGA technology. Furthermore, during last years, FPGAs have become a more attractive tool to researchers: their unitary cost has been dramatically reduced and new implementation tools make easier the formerly tedious tasks of programming and design verification.

Some examples of FPGA-based channel emulators are described in [11]–[13]. In [11] an FPGA-based AWGN channel emulator is implemented. The emulator is a hardware white Gaussian noise generator based on combining the Box-Muller and Central limit theorems, and the whole model was developed in VHDL.

Similarly, in [12], the authors use a Xilinx Virtex-II Pro to implement a fading channel emulator. The fading process models use sum-of-sinusoids (SOS) algorithms that allow designing and implementing Rician and Rayleigh fading channels. The final designs use only between 2% and 5% of the FPGA slices and are able to generate 201 million 16-bit complex-valued fading samples per second.

Finally, in [13], it is presented a baseband multipath fading channel emulator implemented on a Virtex-IV using the Xilinx XtremeDSP FPGA platform. The authors implement the emulator using Matlab Simulink models and Xilinx System Generator IP blocks. The final design is limited to a two-path channel (due to the extensive use of FPGA resources), the input/output rate is set to 20 MHz and the Doppler frequency is 5 Hz. After the verification stage, the authors download the emulator to the FPGA board and test it with baseband QPSK signals, achieving results similar to those obtained through their Matlab simulations.

All the previously mentioned work is unquestionable, but their development has at least two major drawbacks. First, the use of low-level description languages such as VHDL for implementing complex behaviors can derive in slow developments. Although, in most cases, VHDL allow obtaining a resource-efficient FPGA design, programming can become a cumbersome task that may consume a large amount of time and economic resources. There are new sophisticated tools like Xilinx System Generator which make possible to work with high-level blocks that give the possibility to build complex designs easier and faster.

The second problem is related to the use of high-level tools. These tools allow fast prototyping but they usually generate non-optimized large designs that may not fit into the FPGA. For instance, in [13] the authors only download a two-path channel emulator due to the lack of available hardware resources. Hence, for large designs, optimizations must be made.

In this paper, we tackle these issues: we use System Generator to develop the channel emulator faster than using VHDL, and then we optimize our design in order to be able to implement a twelve-path channel emulator, even leaving space for future extensions.

III. IMPLEMENTED VEHICULAR CHANNELS

The implemented channel models are based on the excellent work in [6]. There, the authors present channel models for six different environments:

- **VTV-Expressway Oncoming**: two oncoming vehicles approach at high speeds (87.5 mph, 140 km/h) in a highway without middle wall.
- **VTV-Urban Canyon Oncoming**: two oncoming vehicles approach at 72 mph (115 km/h) in an urban canyon.
- **RTV-Suburban Street**: a vehicle at 72 mph exchanges messages with an antenna placed next to a suburban street intersection.
- **RTV-Expressway**: communications are performed between an antenna on a pole off the side of an expressway and a vehicle at 87.5 mph.
- **VTV-Expressway Same Direction With Wall**: two vehicles at 87.5 mph communicate when travelling on the same lane in a highway with middle wall.
- **RTV-Urban Canyon**: an antenna mounted on a pole near an urban intersection communicates with a vehicle at 72 mph.

Although the measurement campaign was performed at lower speeds (65 mph (104 km/h) in highways and 20 mph to 30 mph (32-48 km/h) for surface streets), the presented models have scaled Doppler frequencies to be consistent with vehicle speeds of 87.5 mph and 72 mph. In addition to the high Doppler effect suffered in each channel, it is noticeable the large delay spreads (see Table I), which can be up to 701 ns.

IV. THEORETICAL MODEL

For generating each channel coefficient of the $i$-th path at time instant $t$, we used the following model:
### Table I

**Main parameters of the vehicular channel emulator**

<table>
<thead>
<tr>
<th>Vehicular Channel</th>
<th>Number of paths</th>
<th>Delay spread (ns)</th>
<th>Coefficient generation rate [Effective rate] (Hz)</th>
<th>Interpolation rate</th>
<th>Occupied slices (%)</th>
<th>Output rate [Effective rate] (Hz)</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressway Oncoming</td>
<td>11</td>
<td>302</td>
<td>3484 [4000]</td>
<td>x2500</td>
<td>71%</td>
<td>71%</td>
<td>2.813</td>
</tr>
<tr>
<td>Urban Canyon</td>
<td>12</td>
<td>501</td>
<td>2194.6 [2500]</td>
<td>x4000</td>
<td>79%</td>
<td>79%</td>
<td>2.813</td>
</tr>
<tr>
<td>Expressway</td>
<td>12</td>
<td>401</td>
<td>2168 [2500]</td>
<td>x4000</td>
<td>79%</td>
<td>79%</td>
<td>2.813</td>
</tr>
<tr>
<td>Urban Canyon Oncoming</td>
<td>12</td>
<td>401</td>
<td>3314 [4000]</td>
<td>x2500</td>
<td>80%</td>
<td>80%</td>
<td>2.813</td>
</tr>
<tr>
<td>Suburban Street</td>
<td>12</td>
<td>700</td>
<td>1988 [2000]</td>
<td>x5000</td>
<td>80%</td>
<td>80%</td>
<td>2.813</td>
</tr>
<tr>
<td>Express. Same Direction With Wall</td>
<td>12</td>
<td>701</td>
<td>3170 [4000]</td>
<td>x2500</td>
<td>80%</td>
<td>80%</td>
<td>2.813</td>
</tr>
</tbody>
</table>

\[
h(i, t) = \sqrt{K_i P_i / (K_i + 1)} \bar{h}(i, t) + \sqrt{P_i / (K_i + 1)} h_w(i, t)\]

where:
- \( K_i \): Rice factor of the \( i \)-th path.
- \( P_i \): power of the path.
- \( h_w(i, t) \): represents the contribution due to the non-line-of-sight (NLOS) component. It is a variable that follows a complex Gaussian distribution with mean zero and variance one.
- \( \bar{h}(i, t) \): contribution of the line-of-sight (LOS) component. It is determined by:
  \[
  \bar{h}(i, t) = e^{j(2\pi f_{D,i} \cos(\theta_i) t + \phi_i)}
  \]

where:
- \( f_{D,i} \): maximum Doppler spread.
- \( \theta_i \): angle of arrival of the \( i \)-th path.
- \( \phi_i \): phase of the LOS component of the path.

V. Description of the emulator

A. Hardware and software

The vehicular channel emulator was implemented on an FPGA using Nallatech’s BenADD-IV development kit, which has the following features:

- It contains a Virtex IV (XC4VSX35-10FF668) that allows using Xtreme-DSP slices of up to 400 MHz.
- It has 2 14-bit ADCs able to work up to 105 MS/s and 2 14-bit DACs that can run up to 160 MS/s.
- Dedicated internal clock up to 105 MHz, although it allows using an external clock.
- 4 MB of 166MHz ZBT-RAM.
- Possibility to connect the kit using a PC (via the PCI bus) or in stand-alone mode.

To decrease the amount of time required to implement the channel model on the FPGA, we decided to use Xilinx System Generator for DSP because it makes easier to design and program the FPGA. It uses libraries of high-level blocks and can interact with Matlab and Simulink. Moreover, another advantage of this software is its ability to exchange data between a design running in the FPGA and a software implementation that is executed on a PC. In fact, for our tests (Section VI) we have implemented in Matlab and Simulink an IEEE 802.11p transceiver that interacts with the vehicular emulator, which is running on an FPGA.

B. FPGA Design Overview

Fig. 1 shows a general view of the hardware design. Notice that the shown design is optimized for a specific channel (VTV-Expressway Oncoming) in order to minimize the amount of required hardware.

The design can be divided into different parts that carry out five different major tasks: generation and acquisition of the channel parameters, LOS and NLOS coefficient generation, Doppler filtering, interpolation and FIR filtering.

- The generation of the configuration parameters of the vehicular channel is performed offline to reduce the amount of required FPGA resources and due to the fact that these parameters remain constant throughout the emulation. The parameters are stored into registers that will be read by the FPGA-based emulator. Notice
that every channel has its own peculiarities and all the parameters equal to zero can be removed from the design to save hardware. For example, all the channels but the VTV-Expressway Same Direction With Wall have only one Rician component, hence in these channels we need just one register to store each of the LOS parameters mentioned in Section IV.

• To generate the NLOS coefficients we have used System Generator’s White Gaussian Noise Generator block, that generates samples that follow a Gaussian distribution with mean zero and variance one. Because the models have twelve complex paths, we need to generate twenty-four of these coefficients, that will be filtered depending on the Doppler suffered by each path. Fig. 2 shows the way we reduce the amount of FPGA resources consumed by the twenty-four output Gaussian noise coefficient generator: instead of using twenty-four independent generators we multiplex in time the coefficients, that are generated by only one generator.

• To constitute the NLOS components the white Gaussian noise samples have to be filtered depending on each path’s Doppler spectrum, which is determined by a fading spectral shape, a frequency shift and a maximum Doppler shift. Four different spectral shapes are contemplated: round, flat, classic 3 dB and classic 6 dB. Fig. 3 shows the blocks that allow applying the Doppler spectrum. To reduce by half the required hardware, we take advantage of the fact that the real and the imaginary parts of the filter have to be used twice for each path to perform the complex FIR filtering.

• LOS Doppler is also taken into account in the vehicular channels and must be applied to each Rician path as shown in (2). To achieve this, we use System Generator’s DDS (Direct Digital Synthesizer) block that generates a sine and a cosine with the required phase and frequency parameters. Since the angle of arrival of the LOS component has not been considered in [6], we always set its value to zero.

• Fig. 4 show how the LOS and NLOS components are generated and added in the FPGA model, as described...
in (1). Next, the coefficients must adapt their rate to the rate of the incoming signal (i.e. the signal from the IEEE 802.11p transmitter) that arrives at 10 MHz, while the coefficients are generated at a much lower rate that depends on the maximum Doppler shift. Indeed, in a specific vehicular channel the implicit sample rate is twice the maximum Doppler shift of all paths. In the implemented vehicular channel models this rate fluctuates between 1988 Hz and 3484 Hz (see Table I). To avoid designing a complex resampling stage, instead of using the original coefficient generation rate, we use an effective sample rate that is equal to the nearest superior integer divider of 10 MHz. Thus, we only need an interpolator. The coefficients of each path are processed by linear interpolators like the one shown in Fig. 5.

- Finally, the signal from the IEEE 802.11p transmitter is filtered with the interpolated coefficients. Fig. 6 depicts a general view of the developed complex FIR filter (Fig. 6 shows twenty-two paths instead of twenty-four because it is optimized for VTV-Expressway Oncoming, like Fig. 1). In the diagram two groups of blocks can be distinguished: blocks aimed at delaying the incoming signal and blocks for applying the coefficients to the delayed signals.

VI. EXPERIMENTS

To evaluate the developed vehicular channel emulator, we tested it with a software implementation of the PHY layer of the IEEE 802.11p standard, whose main features are shown in Table II. We have taken advantage of Xilinx Xtreme DSP software kit and we have performed measurements using the co-simulation mode: the transmitter and the receiver are implemented in Matlab and Simulink, but the channel emulator runs on the FPGA.

Fig. 7 and Fig. 8 depict, respectively, the bit error rate (BER) and the packet error rate (PER) for the vehicular channels VTV-Expressway Oncoming and RTV-Urban Canyon. A rate 1/2 FEC was used, each of the carriers was QPSK modulated, and a maximum of 10,000 64-byte data packets were averaged (the simulation stops for each \( E_b/N_0 \) value when 100 erroneous packets are detected). The receiver assumed perfect synchronization, pilot-aided channel estimation was performed and a minimum mean square error (MMSE) linear equalizer was applied.

It is important to note that [1] states that a vehicular communication system must be capable of transferring messages to and from vehicles at speeds of 120 mph (193 km/h).
TABLE II
MAIN FEATURES OF THE IMPLEMENTED IEEE 802.11p TRANSCEIVER

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Available Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Modulation</td>
<td>BPSK, QPSK, 16-QAM, 64-QAM</td>
</tr>
<tr>
<td>Code rate</td>
<td>1/2, 1/3, 3/4</td>
</tr>
<tr>
<td># subcarriers</td>
<td>48 data + 4 pilots</td>
</tr>
<tr>
<td>OFDM symbol duration</td>
<td>8µs</td>
</tr>
<tr>
<td>Guard time</td>
<td>1.6µs</td>
</tr>
<tr>
<td>FFT period</td>
<td>6.4µs</td>
</tr>
<tr>
<td>Total bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Subcarrier spacing</td>
<td>0.15625 MHz</td>
</tr>
</tbody>
</table>

Fig. 7. BER/PER performance in VTV-Expressway Oncoming.

with a PER of less than 10% for 64-byte data packets. Although the assumed speeds of VTV-Expressway Oncoming and RTV-Urban Canyon channels are 87.5 mph and 72 mph respectively, it can be seen that the 10% PER threshold is achieved with roughly 15 dB and 10 dB of $E_b/N_0$, what is a reference of the system performance.

Notice that in a real transceiver there are additional sources of error (i.e. RF impairments, synchronization errors, channel estimation mismatches...). Our channel emulator can be an extraordinarily useful tool for transceiver designers, since it allows decoupling easily the different sources of error.

VII. CONCLUSION

We have presented a flexible, reconfigurable and cost-effective solution for real-time emulation of vehicular wireless channels. Our emulator is based on FPGA technology and rapid prototyping software tools. After describing the theoretical model, we have outlined the emulator design and its basic operation. We have also detailed some of the optimizations we have done to save hardware resources. Finally, we have shown a couple of examples of performance evaluation of an IEEE 802.11p transceiver over high-speed scenarios.

Fig. 8. BER/PER performance in RTV-Urban Canyon.

ACKNOWLEDGMENT

This work has been supported by Xunta de Galicia through contract Ref. PGIDIT061IC10501PR and by Ministerio de Educación y Ciencia of Spain and FEDER funds of the E.U. under grants TEC2007-68020-C04-01 (MultiMIMO project) and CSD2008-00010 (COMONSENS project).

REFERENCES