EURASIP Journal on Applied Signal Processing 2001:3, 163–168

A High Performance Pocket-Size System for Evaluations in Acoustic Signal Processing

Uwe Rass

Siemens Audiologische Technik GmbH, Gebbertstraße 125, 91058 Erlangen, Germany
Email: Uwe.Rass@med.siemens.de

Gerhard H. Steeger

Georg-Simon-Ohm-Fachhochschule, P.O. Box 210320, 90121 Nüremberg, Germany
Email: Gerhard.Steeger@fh-nuernberg.de

Received 15 May 2001 and in revised form 10 August 2001

Custom-made hardware is attractive for sophisticated signal processing in wearable electroacoustic devices, but has a high initial cost overhead. Thus, signal processing algorithms should be tested thoroughly in real application environments by potential end users prior to the hardware implementation. In addition, the algorithms should be easily alterable during this test phase. A wearable system which meets these requirements has been developed and built. The system is based on the high performance signal processor Motorola DSP56309. This device also includes high quality stereo analog-to-digital-(ADC)- and digital-to-analog-(DAC)-converters with 20 bit word length each. The available dynamic range exceeds 88 dB. The input and output gains can be adjusted by digitally controlled potentiometers. The housing of the unit is small enough to carry it in a pocket (dimensions $150 \times 80 \times 25$ mm). Software tools have been developed to ease the development of new algorithms. A set of configurable Assembler code modules implements all hardware dependent software routines and gives easy access to the peripherals and interfaces. A comfortable fitting interface allows easy control of the signal processing unit from a PC, even by assistant personnel. The device has proven to be a helpful means for development and field evaluations of advanced new hearing aid algorithms, within interdisciplinary research projects. Now it is offered to the scientific community.

Keywords and phrases: acoustic signal processing, digital signal processor, DSP assembler.

1. INTRODUCTION

Digital signal processing is applied in many electroacoustics apparatus (e.g., telephones, recorders for CDs and DATs, equalizers, hearing aids). Though digital signal processors (DSPs) nowadays provide much more processing power than their predecessors some few years ago, specialized hardware solutions are in many applications still as attractive as ever. Reasons for this attractiveness are: high processing speed, minimized power consumption, and low costs in mass production. However, every re-design of a custom-made integrated circuit (IC) is a cost and time consuming undertaking and should, therefore, be avoided as far as possible. That is, the algorithm to be realized ought to be tested thoroughly in computer simulations prior to implementing it on a specific IC chip. However, many acoustical signal processing algorithms can be reliably judged only by critical audition of human subjects. Thus, the simulations should be run in real time, with realistic input signals. On the other hand, it should be possible to alter the algorithm easily, in order to optimize it with low time expenditure. Since many electroacoustic products are wearable and used in very complex sound fields (e.g., conversation in a street with fluctuating traffic noises), the simulation device should be wearable nearly as freely as the intended hardwired end product.

The optimal solution for such an evaluation equipment is a fast low-power DSP, supplemented by stereo analog-to-digital (ADC), digital-to-analog converters (DAC), pre- and power amplifiers as well as user controls. Clearly, the quality of ADC and DAC has to be high, to ensure that actually the processing method is judged by the subjects but not the technical limitations of the device. The computational speed of the signal processor should be enough to test innovative, high sophisticated algorithms. On the other hand, the power consumption should be kept low to reduce the size and the weight of the devices and to enable an acceptable wearing time without re-charging the accumulators.

Simulation devices which meet these requirements on a state-of-the-art technological basis are developed and built at the Fachhochschule Nuernberg since 1994. These devices were supplemented with acoustical front-ends needed for the

evaluation of hearing aid algorithms and, therefore, named *DASi* (digital auditory signal processor). Three generations have been built so far and applied in a German research project as well as in the European research projects HEARDIP (HEaring Aid Research with Digital Intelligent Processing) and SPACE (Signal Processing for Auditory Communication in noisy Environments). The first generation unit, DASi-1, was still a 5 Volt device [1]. In DASi-2, delivered since 1997, all components are operated with supply voltages of 3.3 Volt or lower. Thus, volume and weight could be approximately halved while providing the same processing speed [2]. The most recent development, DASi-3, enables a five-fold processing speed compared to DASi-2 while maintaining the mechanical dimensions unchanged. Its hardware is described in Section 2.

To ease the algorithm development and the fitting of parameters to the individual user's need, supporting software has been developed. Since compilers for high-level programming languages (e.g., C) are not able to exploit the special hardware structure of DSPs, the resulting code is much slower than assembler code. On the other hand, assembler code programmers need an exact knowledge of the hardware on which the algorithm should work. The resulting code is hardware dependent and, therefore, not portable from one system to another.

The DASi-3 Kernel software presented in Section 3 consists of a set of assembler code modules. These modules perform all initializations and communications between the DSP and the peripherals, thereby freeing the programmer from learning the hardware structure of the evaluation device. For algorithms working in the frequency domain, an Overlap-Add processing scheme [3] has been implemented. Parameters can be easily changed by means of configuration files. Another module enables the communication between the DASi-3 and a personal computer which is connected to the serial interface. This so-called monitor program is able to change the parameters while the application algorithm is running.

Concluding remarks on the experience with clinical applications and an outlook can be found in Section 4.

2. WEARABLE SIGNAL PROCESSOR HARDWARE

The wearable signal processing device DASi-3 (see Figure 1) is based on the Motorola digital signal processor DSP56309 [4]. The arithmetic unit of this DSP works with 24-bit fixed-point numbers. DASi-3 can be equipped either with the 80 MHz version or the 100 MHz version of the DSP56309. This DSP type performs one instruction per clock cycle, that is, 80 million instructions per second (MIPS) or 100 MIPS, respectively. Note that each processor instruction originates several hardware activities, in general. This DSP includes a large on-chip memory which is very important for the performance of the system, since accesses to internal memory are at least two times faster than external memory accesses. In addition, the DSP can simultaneously fetch data and code from all three address spaces (program memory and two data memory spaces) within one instruction cycle only, when

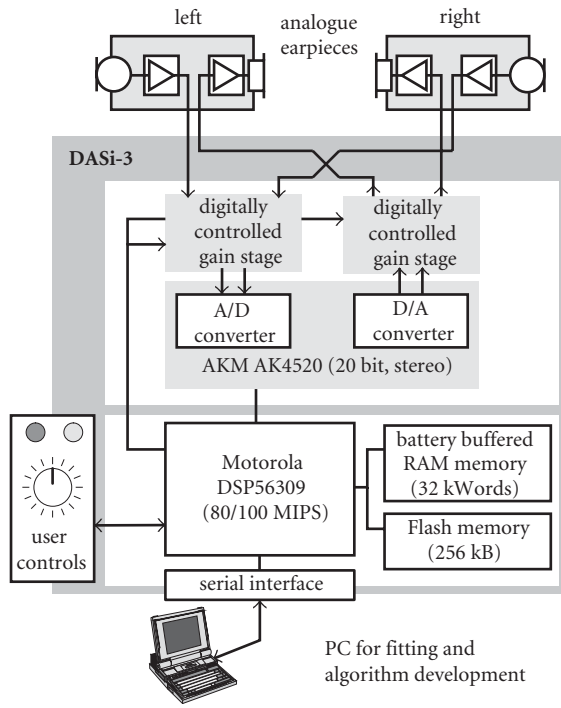


Figure 1: The evaluation device DASi-3, configured for testing of hearing aid algorithms.

on-chip memory is used. Therefore, the on-chip memory is the first choice for program code and data which are frequently needed during signal processing.

In addition to the on-chip memory, 32 kWords (24 bits each) of fast external static RAM memory have been added. The memory chips are able to hold their contents when the device is switched off, by means of a Lithium cell. This data retention feature can be used, for example, to implement an electronic log book function for each algorithm available to the user. The DASi-3 can hold up to four different algorithms or parameter sets in a 512 kByte nonvolatile Flash EPROM memory. The user selects the algorithm by means of a program switch on the device. A second switch with 16 positions can be programmed as a digital volume control.

The interface to the analog input and output signals consists of a 20 bit stereo analog-to-digital-(ADC)- and digital-to-analog-(DAC)-converter chip and digitally controlled gain stages. Therefore, binaural signal processing schemes can be evaluated, too. The combined ADC/DAC chip AK4520 of Asahi Kasei Microsystems offers a dynamic range of 88 dB(A) at the input and 94 dB(A) at the output. The DSP is able to program the input or output gains, respectively, in a range of 63 dB in 1 dB steps. A socket which accepts an 8 pin connector is mounted on the frontplate (see Figure 2). It enables connection of electroacoustical front-end and back-end devices which are necessary for a specific application, including power supply for them. As mentioned above, the first application of the DASi units was hearing aid algorithm research. For this purpose, commercial analog hear-

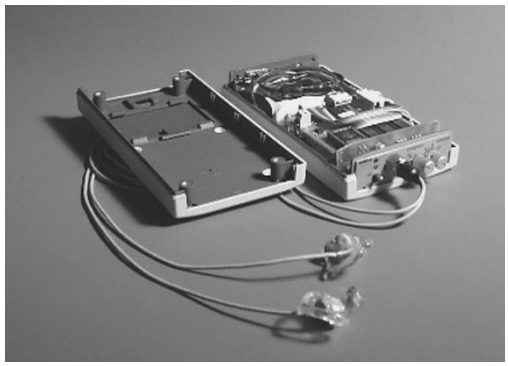


Figure 2: DASi-3 (housing opened) with analogue earpieces. Dimensions: 150 × 80 × 25 mm (when closed), weight: 350 g including accumulator cells.

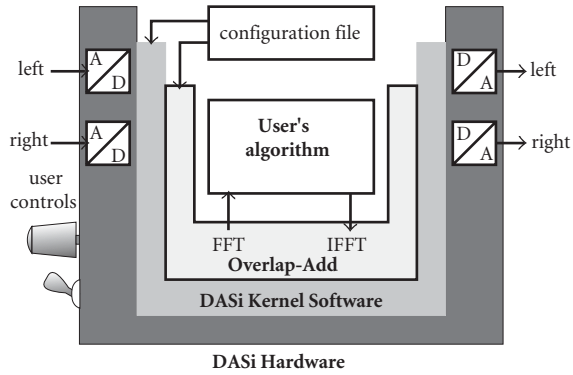


Figure 3: The DASi software as a multilayer shell concept.

ing aid housing, circuits, microphones, and earphones are used to pick up the acoustic signal at the patient's ear and to reproduce the processed signal in the ear canal. The earpieces and the signal processing part are connected over a highly flexible 3-wire, double shielded cable, which also supplies the remote components with power from the main battery (see Figure 2).

The complete system runs with a power supply of 3.1 Volt delivered by an accumulator of 3 Nickel Metal Hydrid (NiMH) cells. The nominal accumulator voltage is 3.6 V. Maximum current drain is 180 mA, when the DSP works with full load at 80 MHz clock frequency and approximately 200 mA when a 100 MHz DSP works with full load. With this worst case power supply current, the units will work for approximately 10 hours at 80 MHz or 9 hours at 100 MHz, respectively. However, in applications which are not time-critical, the DSP can partly enter stand by mode within each sample interval, thereby reducing the overall power consumption. In addition to the basic current consumption of 66 mA, the DASi-3 draws 1.2 mA per MIPS from the accumulator. This number was measured with typical memory access rate and IO activity.

The housing is small enough to carry it in a pocket (see Figure 2).

3. DASI-3 KERNEL SOFTWARE

3.1. Overview

The DASi Kernel software eases the development of efficient algorithms in Motorola DSP Assembler language for the wearable signal processor devices DASi-2 and DASi-3. Since today's high-level programming language compilers are not able to produce optimized code for signal processors, the full processing speed can only be reached by writing the algorithms in Assembler. For example, the C compiler available for the Motorola DSPs uses only one data memory space. Therefore, only a single data move can be performed per instruction cycle, although the DSP architecture is able to move two data words in parallel to the operations of the arithmetic logic unit. The execution speed advantage of assembler coded algorithms must be paid with a loss of portability of the code. It is not possible to re-compile the algorithm for another signal processor family, perhaps of another vendor. Portability can only be reached within one DSP family of a single company. Motorola claims upward source code compatibility for their DSP families. Some few changes are in fact necessary to run algorithms written for older DSPs on the latest family members, but the share of these changes relative to the whole program code is very low.

Another drawback of the Assembler language programming is the higher programming expense. Detailed knowledge of the hardware is necessary to write, for example, interrupt routines for the ADC/DAC control or to configure the signal processor interfaces. All these functions which are needed in every algorithm were integrated in the DASi Kernel. This software interface hides the hardware details of the target system and provides tools to access the peripherals. To keep the flexibility, the Kernel is configurable to adapt the functions to the applications. The parameters can be set by means of ASCII-readable configuration files. The Kernel software currently is available for the DASi-2 and the DASi-3 devices, but it is possible to support other target systems based on the Motorola DSP560xx or DSP563xx families. The functions covered by these Assembler code modules include the initialization of the hardware, the interface control, communication with PC fitting programs and an Overlap-Add processing scheme (see Figure 3).

The standardized software interface minimizes the effort of porting the algorithms written for the older DASi-2 units to newer ones. From the programmer's point of view, the subroutine calls stay the same whether the algorithm is assembled for the DASi-2 or the DASi-3. The Kernel is not an equivalent of personal computer operating systems. Unlike an operating system, the Kernel is not a stand-alone executable program. It is configured and linked to the user code, resulting in a single executable signal processor program. To change the Kernel parameters, the algorithm has to be re-assembled. The term *configurable library* would describe the Kernel best. Depending on the system on which the algorithm should be executed, the correct Kernel has to be linked to the algorithm

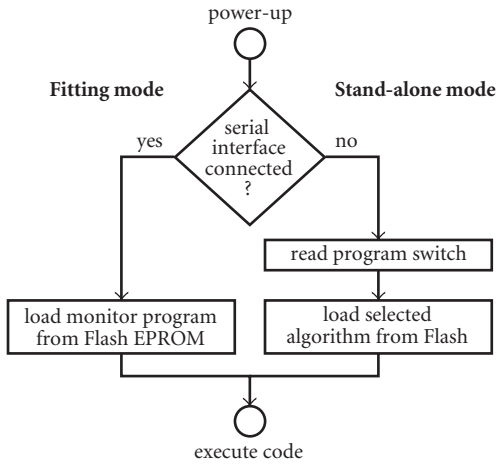


Figure 4: Power-up boot sequence.

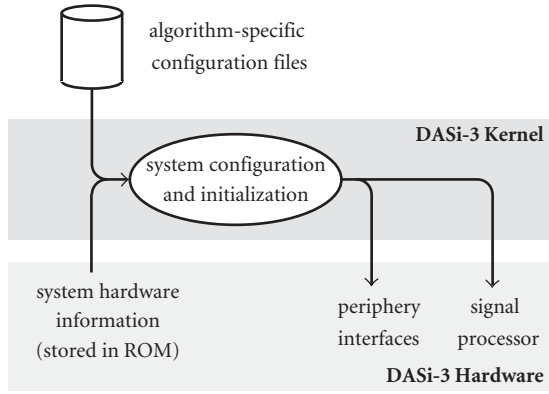


Figure 5: System configuration and initialization.

code. The programmer needs almost no knowledge of hardware details to write algorithms, that is, knowledge of the signal processor's Assembler language is sufficient to write an optimally fast executable program. If the user algorithm is not time critical, assembler programming can even be completely avoided. The user's algorithm then can be written in a high-level language (e.g., C) and compiled.

The functionality of the DASi-3 Kernel is slightly enhanced compared to the DASi-2 version. But all the features of the DASi-2 Kernel are also included in the DASi-3 version to fulfill the portability requirement. In the following, the DASi-3 Kernel functions are described in more detail.

3.2. System initialization

When the DASi unit is powered on, the signal processor's built-in boot loader executes a secondary boot routine read from the Flash EPROM. This code detects whether the DASi is connected to a PC's serial interface or whether it should run in stand-alone mode (see Figure 4). The former mode is called *fitting mode*, since this mode is typically used to fit the parameters of the algorithm to the wearer's individual needs, before she/he leaves the laboratory. In the hearing aid application mentioned above, for example, the parameters are fitted to the individual hearing deficiency of the wearer. A comfortable fitting program (Communication Aid Fitting Environment—CAFE—) for this purpose has been written and runs under Microsoft Windows®. If the serial interface adapter is found, the unit enters the fitting mode by loading the monitor program. The DASi-3 waits for commands sent by the fitting program running in the PC. With these commands, DASi-3 memory locations can be read or written and algorithms can be downloaded and executed. The monitor program which is permanently stored in the Flash EPROM handles the communication with the PC and interprets the command sequences. The programming of the Flash EPROM is also implemented as a part of the monitor program. If the unit runs in stand-alone mode, the user selected algorithm is loaded from the Flash EPROM and executed. Up to 4 different algorithms can be stored in the DASi-3 Flash memory. Beside the code, also data arrays are initialized by the boot loader. The algorithm storage format enables an arbitrary code and data placement in the whole memory space of the DASi.

A special section of the Flash EPROM called the System Information Block (SIB) contains information about the hardware, like maximum DSP clock frequency, sample rate, and so forth, (see Table 1). This data is programmed during the production of the unit. Based on this information, the Kernel software is able to derive the correct settings for the clock dividers used by the on-chip periphery, like timers, serial interface baud rate, and so forth, (see Figure 5). Therefore, algorithm code is independent of the processor speed. An algorithm which was compiled for a DASi-3 with an 80 MHz DSP also runs on a 100 MHz version without modification. The DSP core clock is always set to the maximum. Further, the SIB contains the memory chip access times for correctly setting the wait states. Type and amount of Flash EPROM memory can be determined also from the SIB for selecting the correct programming algorithm. A serial number helps to identify the unit by the fitting program. The algorithm can determine the hardware version from the SIB during run time. This can be used to adapt the system configuration to different hardware versions without re-compiling the code (convenient in field tests with several different devices). Some additional initialization parameters are contained in the algorithm specific configuration files (see Figure 5), for example, timer intervals. These parameters are converted to appropriate register words during the assembly of the algorithm.

3.3. Interface control

The analogue interface of the DASi-3 consists of a stereo ADC/DAC chip and of digitally controlled potentiometers. The Kernel software is responsible for configuring the corresponding signal processor interfaces and interrupts. Also the interrupt service routines are integrated in the Kernel. Incoming samples are collected and stored in memory. The size of the input sample blocks is configurable including a block size of 1 (sample-by-sample processing). When the complete sample block has been received, the Kernel calls the signal processing routine. This could be user-provided code or a built-in function like the analysis part of the

Table 1: Contents of the system information block (SIB).

Label	Bytes	Description
serial	2	DASi unit serial number
rev	2	hardware revision
swrev	2	system software revision
dsp	1	type of DSP (e.g., DSP56309)
dspclock	1	max. DSP clock frequency
qclock	1	quartz crystal frequency
ramaccs	1	RAM chip access time
flashaccs	1	Flash EPROM chip access time
flashtype	1	type of Flash EPROM chip
fs	2	sampling frequency

Overlap-Add processing. The output samples are read from a buffer in memory and transmitted to the DAC also under Kernel control.

During system initialization, the digitally controlled potentiometers are set to their initial attenuation values, which are stored in the algorithm executable file. (The fitting program directly changes the original algorithm executable file.) The setting of the potentiometers, that is, the input and output gain of the device, can easily be changed by calling a Kernel subroutine.

The user controls of the DASi-3 (program switch and volume control) are observed periodically by the signal processor. Thus, the Kernel software reacts immediately on the wearer's commands (changing loudness or loading a completely different algorithm). The setting of the volume switch is translated into an attenuation value by a look-up table. The contents of this table can be changed by means of the fitting program. The programmer can also provide own code to realize other functions connected to the settings of the user controls.

The asynchronous serial interface of the DASi which is used for communication with the PC is under Kernel control, too. The corresponding interrupt service routine collects incoming bytes and interprets the data. The commands coded in this data are executed in the background, that is, while the DSP is waiting for the next sample. Therefore, parameter changes initiated by the fitting program are heard immediately, too.

3.4. Overlap-Add module

Since modern DSPs perform the fast Fourier transform (FFT) and its inverse (IFFT) very fast, it is attractive to realize acoustical signal processing functions in the frequency domain. The Overlap-Add module in the DASi Kernel software realizes the whole algorithm of Allen and Rabiner [3], except the modification of the short-time spectra in the frequency domain. The user has to program not more than this (i.e., his own) modification, the input data window function and has to choose the FFT length and the overlap of succeeding input data frames. As an extension to the processing scheme described in [3], a second window function was introduced between the IFFT

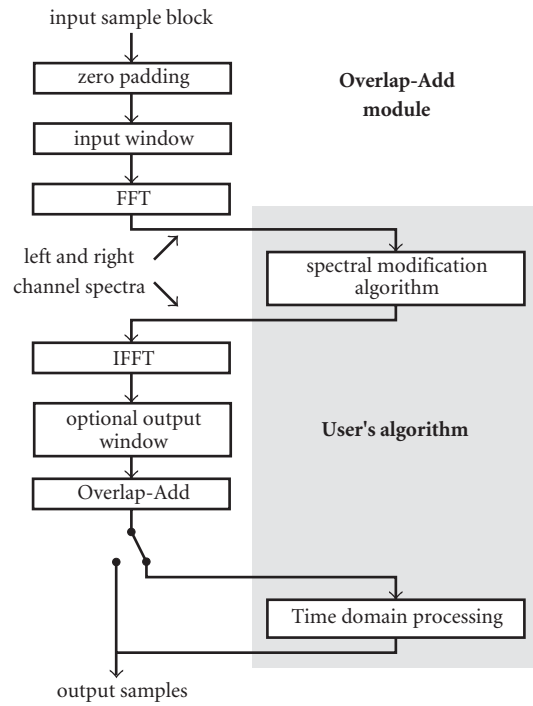


Figure 6: Integration of a user provided algorithm into the Kernel software.

and the overlapping of output blocks (see Figure 6). This window reduces time-domain aliasing errors to some extent [5]. A user who does not want this second windowing can eliminate its effect by defining a rectangular window function. Fast convolution (applying a time-invariant FIR filter by FFT and IFFT; e.g., [6]) is also possible, simply by choosing all windows as rectangular functions and setting the input frame overlap to zero.

Parts of the user's algorithm which do not work frequency specific can also be realized in the time domain. This is useful, for example, to implement a peak clipping before the output samples are transferred to the DAC. The Kernel supports this post processing (cf. Figure 6). The switch in Figure 6 symbolizes a configuration file parameter.

The length of the data window, the FFT length and the amount of overlap can be chosen in an ASCII-format configuration file. The frequently used Hann window coefficients are stored as default values, but can be replaced easily.

3.5. Supplementary functions

As already mentioned in Section 2, the battery buffered RAM memory can be used to realize a log book function, documenting the wearer's use of different algorithms. Basic functions for this purpose are provided by the Kernel, for example, measuring the working time of each algorithm and counting how often the wearer switches between algorithms. The user simply has to set the corresponding configuration parameter to enable this function in the DASi Kernel.

Another auxiliary function observes the accumulator voltage. When the voltage drops below a certain threshold,

the green LED on the DASi front plate will go off. The unit works for approximately one hour more before the accumulators have to be re-charged.

During the algorithm development, it is helpful to know the current DSP processing load. A digital output line of the DASi can be configured to monitor the work and idle states of the signal processor by switching the line high and low, respectively. The current DSP load can be calculated from the duty cycle of this signal, measured by means of an oscilloscope.

4. APPLICATION EXAMPLE, OUTLOOK

As already mentioned in Section 1, the DASi devices are applied in the European compound research project SPACE. New algorithms for future digital hearing aids are developed by several research institutes and initially tested on few subjects in the laboratory. Then, these algorithms are implemented on DASi. This enables exhaustive field testing in the everyday life of hearing impaired persons, organized and evaluated by several ear-nose-throat (ENT) clinics, engaged in the SPACE project. In addition, hearing protectors with high noise attenuation but maximum transparency for speech signals are developed. By means of DASi, their functionality can easily be tested at noisy industrial working places.

It can be expected that in the near future improved DSPs will be available from the microelectronics industry. That is, these DSPs perform more instructions per second and, in addition, are able to run at lower supply voltages. Current consumption will not rise as far as the number of MIPS increases, or can even be held constant. These improvements will enable the development of more powerful DASi devices, perhaps even in smaller housing.

REFERENCES

[1] U. Rass and G. H. Steeger, "Ein tragbares Signalprozessorsystem zur Evaluierung digitaler Hörgeräte-Algorithmen," *Audiologische Akustik*, vol. 34, pp. 126–132, 1995 (in German).

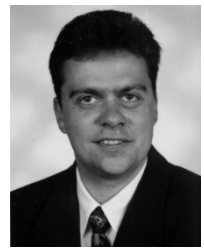
[2] U. Rass and G. H. Steeger, "Evaluation of digital hearing aid algorithms on wearable signal processor systems," in *Proc. 8th European Signal Processing Conf. (EUSIPCO)*, G. Ramponi et al., Eds., Edizioni LINT, Trieste, Italy, 1996, pp. 475–478.

[3] J. Allen and L. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," in *Proc. IEEE*, 1977, vol. 65, pp. 1558–1564.

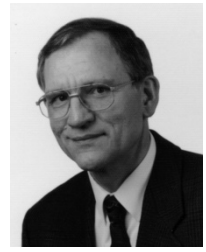
[4] Motorola, "Dsp56303: 24-bit digital signal processor user's manual," 1998, http://www.mot.com/SPS/DSP/.

[5] V. Hohmann, T. Wittkop, and B. Kollmeier, "Personal communication," Department of Physics, University of Oldenburg, Germany, 1994.

[6] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.



Uwe Rass studied electrical engineering, focusing especially on digital signal processing and circuit design at the University of Erlangen-Nüremberg, Germany. He received the Dipl.-Ing. degree in 1993. From 1994 to 2000 he was a research assistant at the University of Applied Sciences in Nüremberg working on algorithms and prototypes for digital hearing aids. Since July 2000 he is with the R&D department of the Siemens Audiological Engineering Group.



Gerhard H. Steeger received the Ing. grad. degree from the Ohm-Polytechnikum, Nüremberg, Germany, in 1969, and the Dipl.-Ing. and Dr.-Ing. degrees from the University of Erlangen-Nüremberg in 1974 and 1979, respectively, all in Electrical Engineering. From 1974 to 1980 he was a research assistant at the University of Erlangen-Nüremberg, working on measurement methods for neuroelectric signals. From 1980 to 1988 he was a R&D scientist at the Audiological Department of the Medical Engineering Group of Siemens AG, Erlangen, Germany, working on devices for diagnostic and therapy of hearing loss. Since 1988 he is a professor for Electronics, Signal Processing, and Electroacoustics at the Georg-Simon-Ohm-Fachhochschule (University of Applied Sciences), Nüremberg. He is a member of EURASIP, ASA, and EAA, and a Senior Member of IEEE.