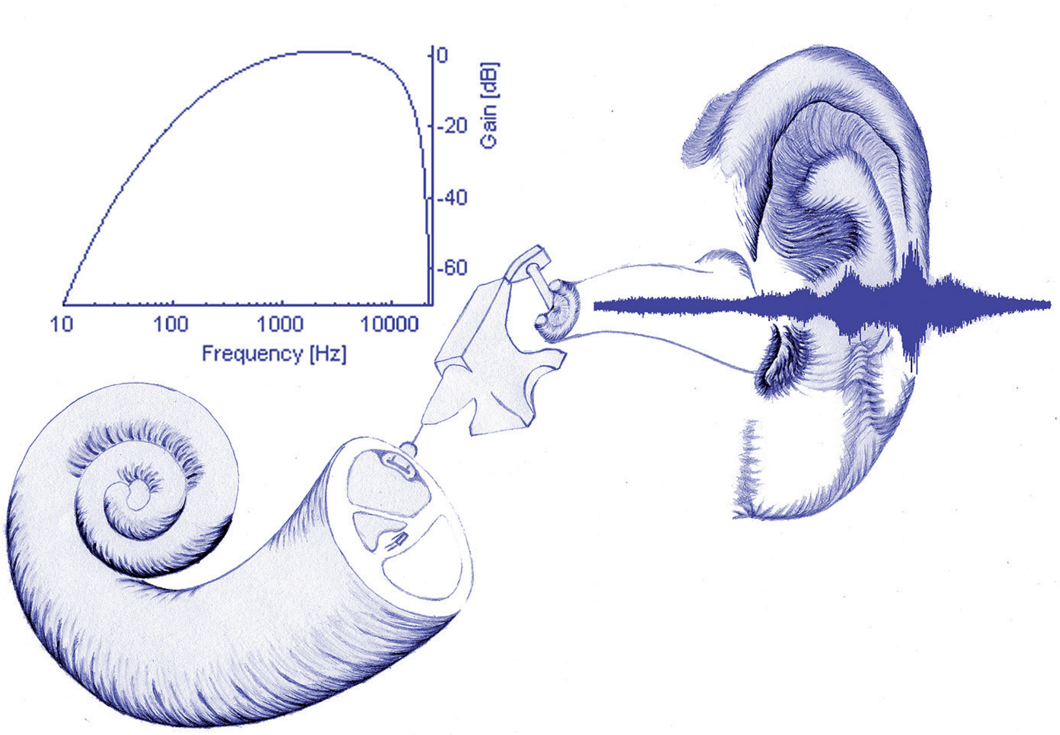
JYVÄSKYLÄ LICENTIATE THESES IN COMPUTING

17

Mikko Lahtinen

Digital Processing of Environmental Noise Samples



JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ LICENTIATE THESES IN COMPUTING 17

Mikko Lahtinen

Digital Processing of Environmental Noise Samples

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2014

Digital Processing of Environmental Noise Samples

JYVÄSKYLÄ LICENTIATE THESES IN COMPUTING 17

Mikko Lahtinen

Digital Processing of Environmental Noise Samples

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2014

Editor
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä

URN:ISBN:978-951-39-5758-2
ISBN 978-951-39-5758-2 (PDF)

ISBN 978-951-39-5757-5 (nid.)
ISSN 1795-9713

Jyväskylä University Printing House, Jyväskylä 2014

Dedicated to my beloved wife, Sinikka

on 26th anniversary of our wedding

12 / 12 / 2013

ABSTRACT

Lahtinen, Mikko
Digital Processing of Environmental Noise Samples
Jyväskylä: University of Jyväskylä, 2014, 176 p.
(Jyväskylä Licentiate Theses in Computing
ISSN 1795-9713; 17)
ISBN 978-951-39-5757-5 (nid.)
ISBN 978-951-39-5758-2 (PDF)
Finnish summary

Principal focus of this thesis is to implement A-weighting on an embedded systems platform. A-weighting origins from a difference in frequency response between human auditory system and measurement microphones. This difference requires a correction to be applied in environmental noise measurements before evaluating environmental noise induced risks on human well-being and health. The correction on the frequency response of measurement microphones is achieved by the use of A-weighting filters. After exploring physiological features of human peripheral auditory system leading to the A-weighting specification we analyze the A-weighting filter characteristics. By MATLAB simulations a digital design alternative of an A-weighting filter suitable for implementation on an ARM processor architecture is developed. Finally, this digital A-weighting design is realized using ARM assembly language. This work aims secondarily on examining possibilities to develop a distributed monitoring system for environmental noise assessment. Physiological symptoms induced by environmental noise tend to develop into health problems not until several years exposure to environmental noise. Therefore, a long-term monitoring system would be a valuable tool in environmental noise assessment and health risk analysis. Development of a wireless sensor network for long-term monitoring of environmental noise and follow-up of its effects on human's physiological responses would require an effort of multidisciplinary nature. As a third objective this thesis presents background information for the different aspects relevant to this kind of large cooperation project. The A-weighting filter implementation developed in this work could be applied as one core component in a distributed long-term monitoring system of environmental noise.

Keywords: environmental noise, human auditory system, A-weighting filter, embedded systems, ARM architecture, ARM assembly language, fixed-point implementation

Author Mikko Lahtinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

E-mail: mikko.lahtinen@ecuras.fi

Supervisor Programmes Manager Jani Kurhinen
Avance Executive Education
Jyväskylä University School of Business and Economics
Finland

Examiners Professor Ismo Hakala
Head of IT Unit
Kokkola University Consortium Chydenius
Finland

Professor Tarmo Lipping
Director of Pori Campus
Tampere University of Technology
Finland

ACKNOWLEDGEMENTS

I would like to express my gratitude to all of those people who have helped me to reach this goal. Above all, I gratefully acknowledge the help of the following persons:

Jarkko Vuori for making me realize I must start from the beginning by learning the basics of digital signal processing first.

Antti Niemi for teaching me those basics and providing me with practical insight into the theory of DSP.

Jani Kurhinen for arranging an embedded systems platform for me to realize the A-weighting filter algorithm and for listening patiently to my long explanations of those problems I encountered along the way with MATLAB simulations, assembly implementation on ARM and preparation of a manuscript for this thesis.

Ilkka Niskanen for borrowing me measurement equipment with which experimental measurements of this research were conducted.

Eki Lehto for providing me with a possibility to gain hands-on experience of environmental noise measurements in real-life context.

Vagan Terziyan for accepting us to take part in his TEKES funded Ubiware project with the Dimeca plan.

Nicholas Dimakides for his academic writing module 'Cohesion and Coherence' which has helped me to increase the readability of this thesis.

Antti Pirhonen for taking the time to read through one of the latest manuscript versions and for presenting valuable comments and offering me justified and constructive criticism of the complete manuscript.

My wife Sinikka, our son Antti and our daughter Anu for showing me understanding, support and love even if the originally intended one year period of this project was prolonged with another two.

Last and most of all, I want to thank God who has given me faith to believe it is possible.

ACRONYMS

ADC	Analog to Digital Converter
ARM	Advanced Risc Machines
BT	Bluetooth
CPU	Central Processing Unit
DSP	Digital Signal Processing
DSSS	Direct Sequence Spread Spectrum standard
END	Environmental Noise Directive
FFH	Fast Frequency Hopping technique
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum standard
GIS	Geographic Information System
GPS	Global Positioning System
GSN	Global Sensor Network
HW	Hardware
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIR	Infinite Impulse Response
IPv6	Internet Protocol, version 6
ISA	Instruction Set Architecture
ISM	Industrial, Scientific and Medical frequency band
LAeq	Equivalent Continuous SP level
LLF	SP level with Linear frequency and Fast time weighting
LLFMax	Maximum **LLF** value within a measurement period
LLFMin	Minimum **LLF** value within a measurement period
LLN	Low-power and Lossy Network
LLpk	Highest Peak value for Linearly weighted signal
LPF	Low Pass Filter
LSB	Least Significant Bit
MAC	Medium Access Control
MCU	Microcontroller Unit
MLA	Multiply Accumulate Instruction

MSB	Most Significant Bit
MW	Middleware
NOISE	Noise Observation and Information Service for Europe
OS	Operating System
PCB	Printed Circuit Board
PHY	Physical layer of the network stack
PIC	Peripheral Interface Controller
PSN	Participatory Sensor Network
Q13	Fixed Point Representation with 13 Magnitude Bits
Q14	Fixed Point Representation with 14 Magnitude Bits
Q15	Fixed Point Representation with 15 Magnitude Bits
QFSK	Quadrature Frequency Shift Keying
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
RSSI	Received Signal Strength Indicator
RX	Receiver
SI	International System of Units
SLM	Sound Level Meter
SOA	Service Oriented Architecture
SOS	Second Order Section
SP	Sound Pressure
SPL	Sound Pressure Level
SQNR	Signal to Quantization Noise Ratio
SW	Software
SWE	Sensor Web Enablement
TX	Transmitter
WBAN	Wireless Body Area Network
WiFi	Wireless Fidelity
WSN	Wireless sensor network
6LoWBAN	Low-power WBAN utilizing IPv6 protocol

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

During millions of years sounds have been important for humans. Hearing has played a vital role in the survival of prehistoric generations. Without the ability to hear sounds generated by animals it would have been more difficult to cope with everyday life. It was essential to be able to locate places where birds and other catch resided. The first estimate of their location could be made based on voices of those animals. Together with visual information it was possible to pinpoint the final position of the quarry. So one benefit of hearing was to assist in finding food for eating. Another use for this sense was to prevent the opposite – that is, to fall prey and be eaten. It was vital to hear the warning sounds that indicated the approach of a predator. Otherwise one might not have had enough time to run away.

Earlier in the history of humankind the sounds originating from nature have been dominating the soundscape observed by humans. Human auditory system has adapted to this natural soundscape and is alert to even very weak sound signals. In the modern society the soundscape 'looks' completely different compared to the natural soundscape. In the urban soundscape we do not have any more long periods of silence, but there is constantly a certain extent of background noise present. Also noise peaks on top of the background are often more intense and more frequent compared to the ones in the natural soundscape. These changes have created a situation where there is much more burden on human auditory system. One physiological feature – or lack of it – makes this even worse. In the visual sensing system we have eyelids, which we are able to close in case too much visual information is entering our eyes. In the auditory system, however, this is not possible. The auditory channels are open all the time; even while we are sleeping and do not register sound signals consciously.

Nowadays we therefore have an increasing mismatch between the incredible sensitivity of human auditory sense organs and the intense urban noise background overloading our auditory system. This mismatch is the origin of constantly growing problems created by environmental noise within urban areas. People are becoming more and more aware of these problems and also authorities start to realize severity of this situation. For example recent EU legislation is setting guidelines for environmental noise levels and this increases the necessity of environmental noise assessment as well. Already year 2007 big European cities were obligated to report yearly averages of environmental noise levels to the European Commission for strategic noise mapping.

This licentiate thesis serves several purposes. Primary focus of this work is in digital design and implementation of an A-weighting filter on an embedded systems platform. Secondary goal is to present background information for future application development challenges. A final vision of applications in the future include a distributed long-term monitoring system for environmental noise. Development of this kind of application would require a large interdisciplinary cooperation project to be undertaken. Therefore, we present background information widely from multiple disciplines in hope that all cooperating partners could have basic understanding of different aspects related to this target application. The organization of this thesis is as outlined in the following paragraphs.

Environmental noise is a widespread problem affecting quality of life of millions of people. This wicked problem gave an initial spark for the research conducted in this thesis. Hence, Chapter 1 begins by defining what is included in the category of environmental noise. We then continue by stating some legislative actions taken against environmental noise. Related to distributed monitoring systems in general we give a couple of examples in Section 1.3 for monitoring of simple environmental parameters. The purpose of this section is to show that a vast amount of building blocks already exist for wireless sensor networks and distributed monitoring. The next section points out a couple of shortcomings faced while assessing environmental noise related risks on human health with traditional sound level meters. Section 1.4 presents also three methods other research groups have developed to address these shortcomings. In the last section of Chapter 1 we introduce different tasks related to environmental noise measurements of which one is selected to be solved in this thesis. The scope of this research is narrowed down to be digital implementation of A-weighting on an embedded systems platform.

Chapter 2 concentrates on describing how human's peripheral auditory system processes environmental noise signals. The main emphasis is on frequency discrimination, because the specific focus of this research is in frequency weighting of measured signals. Based on earlier research we explain the origin for a frequency weighting known as A-weighting. This weighting is the most commonly applied correction in standard environmental noise measurements. For this reason knowledge of its origin is essential. De facto usage of A-weighting in environmental noise inventories led us to select its implementation for the main focus of this research.

In Chapter 3 we present an overview for hardware and software components used in a typical environmental noise measurement. We start by describing what kind of physical observables needs to be measured. The topic of Section 3.2 is operating principles of measurement microphones used to transduce physical pressure signals into electric signals. Following this an example for a traditional and widely used sound level meter (B&K 2238) is given in Section 3.3. Another purpose of Chapter 3 is to present those experimental measurements conducted to acquire environmental noise samples for our research. Section 3.4 outlines an experimental setup where B&K 2238 was utilized to collect basic environmental noise indicators as well as to provide an analog environmental noise signal to be digitized and stored for further development steps of this work. This chapter concludes by showing results obtained from the traditional part of our measurement setup to provide the reader with a brief insight on typical environmental noise indicators.

In Chapter 4 one essential prerequisite for reliable environmental noise measurements – namely calibration – is discussed. In order to achieve measurement results, which are comparable with other measurements, measurement equipment must be adjusted properly. Frequency and sound pressure calibrations demanded in standard environmental noise measurements are the topics in Sections 4.2 and 4.3, respectively. Before these calibration types we take up an additional procedure (i.e., synchronization of time axes) specific to the experimental setup of our research.

First part of this thesis in Chapters 1–4 explores environmental noise and human hearing related background information from which the interest for this research has aroused. Chapter 5 acts as a transition into the second part of the present work where we concentrate on the principal topic of this thesis – that is, digital design and implementation of A-weighting. Traditional environmental noise measurement equipment (like B&K 2238) implement frequency weightings with filters designed according to analog specifications. In Chapter 5 we first give specifications for A-, B- and C-weighting filters in analog domain. We then describe the steps required to transform these analog filter designs into the digital domain. Finally, we go through this transformation step by step for one of the above mentioned frequency weighting filters.

In Chapter 6 the A-weighting filter is simulated bearing in mind the final application field defined in earlier chapters. Characteristics of this digital filter design are analyzed while reducing the accuracy of filter coefficients. Especially the issues of stability of the filter design and conformance to A-weighting filter specifications are the main focus of this chapter. MATLAB is utilized in simulations to analyze A-weighting filter's transfer function properties of different designs. The objective of these MATLAB simulations is to find a sensible implementation alternative for an embedded systems platform which provides only a limited amount of computational resources.

Development for the actual research topic continues in Chapter 7. This development is carried out in time domain using a sequence of samples digitized from one of the experimental measurement signals. Based on the simulation results obtained in the previous chapter we develop a practical software algorithm for A-weighting. This algorithm must be suitable for an embedded systems platform in order to be applicable in the target application (i.e., distributed long-term monitoring system of environmental noise). We start by going through limitations in computational resources posed by embedded systems platforms. We then present workarounds and compromises that must be applied to achieve a realizable implementation. For example we select a suitable representation for digitized samples data to be used in filtering. Finally, we describe – in a general level – the A-weighting algorithm which can be implemented using assembly language on an embedded ARM processor architecture.

Chapter 8 is devoted to verification of the ARM implementation of A-weighting. We first present some statistical parameters describing characteristics of an input signal measured for this purpose. The implementation challenges are revisited and the chosen solution is examined in more detail. Implications of input samples and filter coefficient quantization are reviewed. We discuss possibilities to estimate how sensitive our ARM implementation is in regard to reducing coefficient accuracy in A-weighting algorithm. Measures to estimate the impact of this reduction are introduced. We then continue to verify this ARM implementation by going through the steps executed in the core of our assembly code. We devote two sections to estimate the consequences implied by the chosen implementation methods on the probability of overflows and the amount of arithmetic errors. Section 8.5 concludes Chapter 8 by presenting a verification example using digitized samples from one of our experimental measurements. This example compares ARM filtered outputs with MATLAB generated references in order to evaluate the accuracy of our A-weighting realization.

Finally in Chapter 9 we discuss strengths and weaknesses of the implemented A-weighting filter. Advantages and disadvantages of the chosen implementation method as well as a couple of rejected alternatives are examined. We present ideas for future development of frequency weighting filters on embedded systems platforms. Usability of the implemented A-weighting filter in measurements of environmental noise indicators is discussed. This chapter also considers limitations of A-weighting, which should be taken into account while applying this de facto weighting in environmental noise assessment. Some remarks are made on the pioneering work others have performed on environmental noise assessment and noise mapping using wireless and participatory sensor networks. We conclude this thesis by presenting future visions for distributed long-term monitoring of environmental noise.

1.1 Environmental Noise

Several types of environmental noise exist. A generalized division of environmental noise could be as follows:

- industrial noise
- construction noise
- transportation noise
- leisure noise

This thesis concentrates primarily on transportation noise. Noise types in the category of transportation noise include road traffic, railway and aviation noise. Road traffic is one of the fastest growing sources for environmental noise. For example in Finland three million passenger cars, 375 000 vans, 128 000 trucks and 14 000 buses were registered in the stock of vehicles at the end year 2012 [TRA12]. An increase in the number of vehicles in the aforementioned four groups from year 2000 has been 43 %, 58 %, 97 % and 51 %, respectively. An elevenfold increase can be observed in the two first mentioned categories during half a century long period starting on year 1962. Compared to big international metropolitan areas density of vehicles in Finland is still minimal, but even here road traffic noise is already recognized as a severe environmental problem. In Finland 85 – 90 % of citizens exposed to environmental noise are suffering of road traffic noise [Sip07, Saa13]. Currently, there are one billion cars worldwide and over 220 000 new ones are produced every day [UN12]. Moreover, it is alarming that the growth in the number of motorized vehicles exceeds even the growth rate of the world population.

Road traffic noise is mainly composed of two components, namely engine noise and tire noise. For the engine noise a rather well defined frequency peak is found at low frequencies. The highest energy concentration of road traffic noise falls in the frequency range between 50 Hz and 100 Hz. These frequency components origin from combustion engines of vehicles. Suppose a four-stroke internal combustion engine with four cylinders is running at a constant speed of 3000 rpm. In this case the frequency of engine revolutions becomes 50 Hz and during each revolution half of the cylinders are firing the fuel injected in them [Alb10]. Therefore, the firing frequency of this engine equals 100 Hz (i.e., twice the frequency of revolutions). Combustion processes initiated by these firings are the cause for a frequency peak at 100 Hz in this specific engine setup. In real traffic the workload of an engine affects the noise emission. The workload itself depends on driving conditions (e.g., uphill versus downhill, acceleration versus braking). While driving speed is below 35 – 55 km/h engine noise dominates noise emissions of light and heavy vehicles, respectively [Amu05].

The increase of speed enhances the portion of another main component in road traffic noise. For example measurement results for a passenger car the speed of which is 59 km/h reveal that the low frequency (in this case 95 Hz) engine noise is 84 dB. With this speed higher frequency tire noise between 800 Hz and 1100 Hz is at a level of 90 – 99 dB [Yan11]. If all engines and all tires would have exactly equal characteristics the frequency spectrum would only contain two distinct frequency peaks. In practice the spread in engine and tire parameters as well as alterations in driving conditions introduce broadening of frequency peaks in the spectrum. Modest broadening for the engine noise peak and more spread for noise emitted by tires is observed. The latter spread results from a wide variety of parameter values characterizing the type of tires (size, tread pattern, material, etc) as well as variations in the properties of road surfaces and differences in rolling speed of tires. Hence, the noise emission produced by tires shows a more widespread frequency spectrum (cf. Figure 18).

Propagation of vehicular noise from streets and roads toward people exposed on it depends also on a number of different parameters. Atmospheric conditions contribute strongly on the outdoor sound propagation. Most important atmospheric parameters are wind direction, relative humidity and temperature gradients. Other factors of influence are structure of the terrain, type of vegetation as well as edges and corners of buildings. The multitude of the parameters affecting the outdoor sound propagation complicate the task of producing reliable predictions of long-term environmental noise exposure on inhabitants.

An additional problem with road traffic noise is the lack of monitoring tools because of a widespread geographical area on which people are exposed to it. Aviation noise, for example, is easier to monitor because problems caused by this type of noise are confined to a smaller area where airplanes are at low altitudes around airports. In this case already a dozen of strategically placed noise monitoring stations are able to provide a good impression of the influence of aircraft on environmental noise levels along the routes of arrivals and departures (see e.g, a WebTrak system around Helsinki Airport[1]).

1.2 Environmental Noise Legislation

Earlier it has been common to concentrate on physical damage caused by for example occupational noise. Already for several decades it has been known that high levels of

[1] http://webtrak.bksv.com/hel

industrial noise can induce permanent damage on employees' hearing organs [Møl77]. To prevent these irreversible physiological changes legislation has been enacted to govern working conditions on factories and other noisy working places. Originally action values on levels of 85 and 90 dB were set for daily personal noise exposure [EEC86]. Later these threshold values were reevaluated in another directive by the European Parliament and the Council of the European Union [EPC03]. National legislation has followed this European directive and adopted corresponding laws and practice for example in United Kingdom [UKsi05] and Finland [VNa06]. As a result of this legislative work guidelines exist which instruct employers to make personal hearing protectors available when the specific occupational noise level (i.e., $L_{EX,\ 8h}$ = 80 dB) is exceeded. Likewise another action value ($L_{EX,\ 8h}$ = 85 dB) demands employees to use the provided protectors while working on areas which possess occupational noise levels in excess of the latter level. Moreover, this upper exposure action value requires employers to mark the excess areas with appropriate warning signs, restrict access to these areas as well as strive to implement noise abatement actions on these areas. Consequently, thus far the primary purpose of occupational noise legislation has been to protect employees from severe physical damage (hearing loss etc.).

More recently increased attention has been paid on other noise induced effects besides of the extreme cases of permanent damage mentioned above. Much more frequent effects include different kinds of inconvenience induced by environmental noise. These adverse effects are in the main focus for example in a publication commissioned by the Ministry of the Environment in Finland [Jau07]. According to this publication environmental noise disturbs many physiological functions. For example sleep disturbances of varying severity are typical, performance of cognitive tasks may be reduced and comprehension of speech affected. Several research reports (e.g., [Pir10], [Hal12], and [Han13]) support the findings described in the aforementioned publication. In these cases humans are not injured permanently. Long-lasting exposure, however, can increase the risk to catch an illness of a more permanent nature (e.g., cardiovascular disease, dysfunction of respiratory system or skeletal defects). Hence, exposure to lower levels of environmental noise can also be harmful even if it would not cause instant drastic damage to humans' health.

Because of all the reasons mentioned above, legislative initiatives and action plans for protection of inhabitants from environmental noise exposure of a lesser extent have been created by national and international authorities during the early years of this millennium. Included among others are an Environmental Noise Directive (END) proposed by the European Commission [EPC02], an Environment and Health Indicator System (EHIS) proposed by the World Health Organization [WHO04a] and a National Noise Abatement Program in Finland by the Ministry of the Environment [YM04]. The extension of the environmental noise problem is revealed in these reports. In Finland almost one million inhabitants are living in areas where the recommended maximum environmental noise level (L_{Aeq}= 55 dB) is exceeded. In Europe approximately 120 – 170 million people (i.e., about 25 – 30 % of the population) are exposed to equivalent continuous noise levels higher than the maximum recommended value during the daytime. Exposure to transportation noise may have both social and economic impacts on people's life [Jon12]. Social and economic costs due to transportation noise include for example health care costs, loss of productivity and reduced market value of houses. These external costs have been estimated by European Commission, OECD and ECMT (European Conference of Ministers of Transport) to account for 0.2 – 0.5 % of the gross domestic product (GDP). By extrapolating from the GDP data of the second quarter of 2013 released by Eurostat [ENR13] a total cost of 26 – 65 billion euros will arise in the EU-27 countries for the year 2013. This is a rough estimate because a lot of uncertainties exist in monetary valuation methods of these external costs [Qui94, And13].

Environmental noise experts participating in the WHO working group for EHIS proposed adoption of six environmental noise policy related indicators. Only two of them were finally recommended to be included in the European Community Health Indicators (ECHI) system [WHO04b]. Others were either left out or they required further development to justify their feasibility. Even if the discarded indicators were considered as very valuable in regard of noise policy evaluation the bad quality of data available and gaps in information for generating these indicators were seen as a problem. Public dissemination of traffic noise information and research knowledge was regarded as an important task in raising awareness of the environmental noise problem and supporting policy-making in order to reduce transport related noise emissions. One requirement identified by WHO and UNECE (United Nations Economic Commission for Europe) to reach this goal was developing more advanced information systems to monitor traffic noise and share assessment results [ECE09]. Currently, after 14 years of planning and development, the ECHI system provides altogether 88 health indicators. Still, environmental noise related indicators are missing from this system (which was renamed European Core Health Indicators in May 2013)[2].

Based on the National Noise Abatement Program the Government of Finland adopted a resolution on noise abatement [VNp07]. This Government resolution sets a goal to reduce the number of citizens exposed to unacceptable levels of environmental noise by 20 % compared to the situation on 2003. A schedule for an action plan proposed to achieve the desired reductions extends to year 2020. The noise abatement program stresses the importance for monitoring of the environmental noise situation. In order to be able to plan necessary noise abatement tasks as well as to estimate the effectiveness of fulfilled implementations monitoring plays a vital role. The program report also states that currently prerequisite for monitoring of environmental noise situation does not exist. One intention assigned for an evaluation period of the program during 2004 – 2012 was to estimate the effects of implemented noise abatement actions on the number of exposed people. However, the success in meeting noise assessment goals set in the action plan was difficult to measure. Actions urged for the remedy of this problem included development of systematic acquisition methods for environmental noise data as well as elevation of the quality of research by encouraging research groups on larger cooperative projects.

One tool developed for monitoring the status and success of noise abatement work in Finland is a national information system for noise abatement (MTTJ[3]). This kind of centralized monitoring is essential in order to be able to implement required reductions in the number of exposed citizens. Currently, MTTJ is not yet comprehensive

[2] http://ec.europa.eu/health/indicators/indicators/

[3] http://www.ymparisto.fi/fi-FI/Kartat_ja_tilastot/Tietojarjestelmat/Meluntorjunnan_tietojarjestelma__ohjeita(24050)

enough to provide adequate information which could be disseminated to all relevant interest groups (e.g., local authorities working with urban planning and land use, practitioners involved in noise assessment and abatement). Improvements are required to the current version of the system before MTTJ could serve as a centralized source of information to support national noise abatement efforts. Monitoring of qualitative and quantitative improvements achieved by implemented noise abatement actions is not possible without a better data acquisition system. Besides of enhanced data acquisition also processing of aggregated data must be organized more effectively in order to guarantee the success of the noise abatement efforts [Saa13].

Similar legislative goals and actions plans for environmental noise assessment and abatement are in effect in other Scandinavian countries as well. As a result of these legislative demands, for example in Denmark a nationwide GIS-based system is made available to visualize environmental noise levels around large cities and major roads [4].

The Environmental Noise Directive requires yearly averages of environmental noise levels to be reported to the European Commission. This obligation of the END directive was activated for large European urban areas (with more than 250000 inhabitants) already year 2007. Smaller urban areas which have a population in excess of 100000 must start supplying these yearly noise reports in 2012. From the environmental noise data reported by the EU member states a European noise database is compiled by the European Environment Agency (EEA). An accompanying web service to promote dissemination of noise information to European citizens also exists. This NOISE web service[5] was introduced to Finnish environmental noise authorities, policy makers and other professionals during biennial National Noise Abatement Days 2011 by an environmental scientist Núria Blanes [Bla11]. The EU demanded strategic noise maps are visualized in this web service. It is also intended that the EEA NOISE database could serve as a data source for the European Environment and Health Information System (ENHIS) in order to derive an indicator called 'Population exposure to noise from different sources' [WHO11]. Both temporal and spatial challenges still exist in comprehensiveness of the NOISE maps. These challenges are discussed more in Section 9.4.

1.3 Environmental Monitoring

In the past ten years monitoring of environmental parameters has been increasing rapidly following the advances in new techniques for wireless communication, physical sensor miniaturization, low power embedded platforms etc. New inventions in these research fields facilitate scientists' and engineers' work in development of wireless sensor networks (WSN) for environmental issues. Thus far WSN implementations have been introduced in monitoring of slowly varying environmental parameters. Typical objectives in these projects have been monitoring of temperature and humidity, tracking of animal mobility and activity as well as monitoring of geophysical events. In this section we first give examples for a few environmental monitoring systems which are already in operation or have been in testing for a couple of years. We then continue be presenting hardware and software components which have evolved from the pioneering work of research groups in the field of environmental monitoring.

[4] http://miljoegis.mim.dk/?profile=noise

[5] Noise Observation and Information Service for Europe, http://noise.eionet.europa.eu/viewer.html

1.3.1 Monitoring System Examples

Utilization of WSNs started from monitoring of natural environments. This subsection presents examples of environmental monitoring systems which fall into three typical categories. First of these categories include systems targeting in monitoring humidity on air or water content below the ground level. In rainforests illumination and humidity conditions may vary considerably on the vertical level due to exceptional height of vegetation. By installing wireless humidity sensors on different heights on trees it has been possible to monitor how the relative humidity alternates during the day and night time [Cul04]. Another example of WSN implementations intended to measurements of simple environmental parameters is a soil moisture monitoring system [Can12]. In order to collect data for soil moisture maps a WSN system consisting of a central network node and 12 measurement stations was installed into an area of 100 km^2 to measure ground humidity from four different types of soil. Besides of a capacitive humidity sensor the measurement stations were equipped with digital thermometers as well. These measurement stations were implemented using an 8-bit microcontroller (PIC16F876 from Microchip) which included a built-in 10-bit ADC. An external real-time clock was utilized to wake up the microcontroller periodically to conduct soil moisture measurements. The humidity probe needs to measure soil moisture values only a couple of times during one hour. Therefore the supply voltage to it can be disconnected most of the time to reduce power consumption. A typical wake-up period to start a new measurement is once in every ten minutes resulting in six soil moisture values per hour. A central node of the network requests soil moisture values measured by the end nodes for example once an hour. With these arrangements the sensor nodes can stay in a power saving mode 96 % of the time and achieve an average current drain of 14 mA.

Another category of examples introduces systems developed for animal tracking, wildlife monitoring or habitat monitoring. In a ZebraNet application GPS equipped wireless sensors are utilized to track movements of a zebra population. By observing the locations visited by zebras in their search for nourishment the geographical evolution of the population can be traced [Liu04]. Densely packet electronics is found in an activity monitoring system intended for zoo animals. A 14*14*12 mm^3 monitoring node includes a temperature sensor, an accelerometer, a wireless 2.4 GHz transceiver and an MSP430 microcontroller. All of the required features are integrated in a stack of three small printed circuit boards. Each of the layers in the stack implement one of the required functionalities – that is, sensing, wireless communication and processing. The accelerometer records the movements of animals thus revealing the status of their physical activity [Bra07]. A third example of animal tracking is a wireless sensor system for badger monitoring, which was composed of four components, namely RFID tags, detection nodes, environmental sensor nodes, and an Internet gateway [Dyo12]. The RFID tags were attached on badgers to monitor their movements. The detection nodes were installed near locations which were known to be populated by badgers. These nodes were able to detect badger RFID tags from a distance of 30 m. Besides of an RFID receiver the detection nodes included a Tmote Sky mote to collect and transfer badger data further towards the Internet gateway. A set of ten environmental sensor nodes tracked climate parameters around places where the most active badger mobility was anticipated. The hardware platform for these sensor nodes was also Tmote Sky which was enhanced with two additional temperature and capacitive humidity sensors. The temperature and relative humidity values were measured once in every five minutes. Temporarily a fifth component was present in the monitoring system while researchers observing the behavior of badgers acted as mobile sinks in the data transfer network. The power consumption of different types of nodes in this monitoring system varied considerably. A single 3V, 620 mAh battery was estimated to sustain the RFID tag operation about two years with a transmit rate of twice a second. The RFID reader, on the contrary, was the most power hungry component of the monitoring system and it had to be powered from a 12V, 18Ah battery. Therefore an adaptive method was utilized to find out suitable active and sleep durations for the RFID reader. Using this adaptive method to switch the detection nodes ON and OFF it was possible to obtain a high detection efficiency (89 %) of badgers, but still preserve energy by spending long periods of time in sleep (95 %). Experience gained during the deployment of the first version of this monitoring system was utilized to design a refined and more useful new version. In the second version of the system for example the power consumption and start-up time of the RFID reader were reduces to one tenth and one hundredth of the original, respectively. Another interest in this project – besides of the design and development – was to estimate the requirements for a long-term field deployment and management of a wireless monitoring system.

Our third category of natural environment monitoring relates to geophysical structures like volcanoes, mountains, atmosphere, oceans and glaciers. From this class we take as an example a WSN system developed to monitor evolution of glaciers. The research goal of a Glacsweb application is to track ice movements and relate them with ice-sediment boundary conditions. In this WSN implementation temperature, pressure and orientation sensors are embedded in sensor nodes buried in 50–80 meters depth beneath the surface of a glacier [Mar04]. The original Glacsweb deployment in Norway relied on two different wireless technologies to transfer sensor data. Sub-glacial sensor nodes uses a 433 MHz radio link transmit their observations through the ice to a base station on top of the glacier. From there a 466 MHz wireless radio modem communicated the sensor data tagged with GPS information to a relay station which acted as an Internet gateway and transferred sensor data to a database server. In this version of Glacsweb the sensor nodes woke up once in four hours to read and store their sensor values. Data communication between the Glacsweb components was executed once per day. An enhanced version of this monitoring system was developed in 2009 [Mar09]. The main difference in comparison to the original implementation was the replacement of peripheral interface controllers on the base station with a combination of an MSP430 mixed-signal microcontroller (from Texas Instruments) and an ARM-processor based board (a Gumstix Connex). The sensor nodes still retained their PIC implementation where the microcontroller was duty-cycled under the control of a Real-Time Clock. The PIC microcontroller in turn supervised power delivery to sensors and wireless radio on the sensor nodes. In these sensor nodes the sampling rate was increased to once per hour and the wireless radio was switched on for one minute every day to enable data transmission. With this kind of duty-cycling configuration the sensor nodes consumed only 550 µWh per day. The GPS module included in the base station was recording five minutes once a week. In this version a fixed reference station for differential GPS was introduced one kilometer away from the base station. The base station communicated sensor data from the glacier in Iceland to England via a GPRS link. The more powerful ARM-processor on the Gumstix PCB was waked up for this purpose only once a day to keep the power consumption of the base station down. The power consumption has been one of the major concerns in the development of Glacsweb, but still an additional solar panel and a wind generator are required for the base station besides of a 36 Ah lead acid battery [Mar11]. Despite of these extra power sources and a massive battery an adaptive power management must be applied which selects between four different power states to set a suitable operational mode for the components of Glacsweb. Power ratings for these operational modes on the base station have been as follows:

- 7 μW while the μC was in a sleep mode and the ARM CPU was OFF
- 52 mW while the μC was reading sensors and the ARM CPU was OFF
- 900 mW while both the μC and the ARM CPU were ON
- 4.5 W if the dGPS module was also active

In the 2012 deployment of Glacsweb the principal wireless link for measurement data transfer was changed [Mar12]. In this latest version a 2.4 GHz wireless link is used to transmit base station data first via the reference station to a nearby building. From there data are transferred via Internet to Southampton where remote database servers used for permanent storage are situated. The GPRS link utilized in the earlier Glacsweb deployments is still reserved as a backup communication method to add redundancy and increase fault-tolerance to the system. Sampling rate of the dGPS receiver is increased considerably compared to earlier deployments. Now the GPS readings are checked once in fifteen seconds instead of once a week as is was in the 2009 version. In addition to sensors mentioned in the original 2004 version this new deployment uses also geophones to detect seismic events. A major difference in the Glacsweb sensing hardware in comparison to earlier versions is that now also the sub-glacial sensor nodes are based on ARM-processors.

1.3.2 Monitoring System Components

A common denominator for all of the above mentioned wireless sensor network (WSN) applications is the fact that the input rate for the parameter logging is considerably low. Only a handful of simple parameters needs to be collected and digitized with a slow rate. Therefore, the demand for computational resources is very modest and can easily be fulfilled. Another advantage of the slow sampling rate for the input parameters is long inactive periods in processing and communication performed by these WSN systems. All of these features make it possible to switch this kind of systems into different types of sleep modes for majority of their operating time. Several experiments – both measurements and simulations – have been conducted to examine the effect of sleep modes on energy consumption (e.g., μSleep on ARM architecture [Bra04], controlled greedy sleep algorithm [Sim07], adaptive staggered sleep protocol [Ana09], adaptive low-power sleep mode [Jur10], synchronous sleep and wake scheme [Gao11]). By applying efficient sleeping algorithms WSN systems monitoring slowly varying environmental parameters can achieve low power consumption and be battery operated. These energy-aware arrangements make them capable for long-term monitoring without human intervention.

A huge amount of effort has been put in the research and development of WSN systems during the past several years. Both hardware platforms and software solutions to be used as building blocks for distributed wireless monitoring of environmental parameters have evolved as a result of this work. Examples of WSN hardware platforms include Medusa-MK2 sensor node developed within Center for Embedded Networked Sensing at University of California (Los Angeles) [Sav02], diverse versions of Mote platforms originating from the work initiated by University of California (Berkeley) (Mica / Mica II / MicaZ / Telos / TelosB motes [Hil02, Pol05, MEM11a, MEM11b], Intel Mote / Mote 2 / Imote 2 [Nac05, CRO07, Nac08], Tmote Sky that was used in several WSN implementations during years 2006 and 2012 [MOT06, Wer06, Gun10, Abb12]), MIMOSA (Microsystems platform for Mobile Services and Applications) [Jan08], EcoSpire [Che09], CHILImodule [CHI10], SunSPOT [SUN10], TinyNode 584 [TIN11], IRIS [MEM11c], Lotus [MEM11d], Gumstix [Baj11], Arduino hardware platform [Fis12] and AmICA [Wil12].

Wireless communication in WSN systems can be implemented utilizing several frequency range alternatives. Most commonly a physical layer of the network stack complies with the IEEE 802.15.4 standard [Zhe06, IEE11]. According to this standard 27 frequency channels are available from three license-free ISM (Industrial, Scientific and Medical) bands. One of the most popular ISM bands is a 2.4 GHz frequency range within which a wireless radio can operate on 16 different carrier frequencies. For example 14 out of those 20 WSN hardware platforms mentioned in the previous paragraph implement the physical layer using the 2.4 GHz ISM band. For application development targeting this frequency band communication chips and development tools exist from several manufacturers. Some of these wireless radio frequency chips include also a built-in microcontroller unit (MCU) thus providing components for two basic functions required from a wireless sensor node – that is, for processing and communication. One manufacturer of this kind of integrated RF-MCU chips is Nordic Semiconductor with its nRF24L-series System-On-Chip components for 2.4 GHz applications accompanied with development tools[6]. Also Texas Instruments' CC2430 and CC2530[7] SoC solutions belong to this category. Quite a few commercial applications are already operating on the 2.4 GHz band. Many of them fall in the category of wireless body area networks (WBAN) [IEE12a] like wristop training assistants equipped with additional wireless sensors by Suunto, Polar Electro and Garmin.

Popularity of the 2.4 GHz ISM band provides an advantage of a broad installation base with example implementations as well as matured development tools. On the other hand, however, it also creates some disadvantages. Already back in several years the unlicensed 2.4 GHz frequency band has been quite crowded [Rak07]. Several wireless communication protocol standards like ZigBee [Whe07], Wi-Fi [IEE12b], Bluetooth Low Energy (formerly known as Wibree) [Mac12] and 6LoWPAN [Yib11] can utilize this frequency band. No strict regulation inside the allowed frequency range (i.e., 2400 – 2483.5 MHz) is applied by any authority. For this reason, interference between data transmission of applications using different wireless protocol standards is inevitable

[6] http://www.nordicsemi.com/eng/Products/2.4GHz-RF

[7] http://www.ti.com/product/cc2530

and calls for methods to detect and counteract the problem [Her12]. To alleviate radio spectrum overcrowding and mitigate the adverse effects of cross technology interference between applications' communication channels different workarounds have been invented [Ree12]. As a result of this development methods like frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS) are available. These two spread spectrum techniques are applied for example in the IEEE 802.15.1 [IEE05] and IEEE 802.15.4g [IEE12c] networks, respectively. In the former technique spreading is achieved by hopping pseudo-randomly within at most 79 frequency channels on the physical layer of the 2.4 GHz band. An algorithm example of DSSS implementation for a multi-rate and multi-regional offset quadrature frequency shift keying (MR-O-QFSK) PHY layer can be found in the IEEE 802.4g standard. In addition to these two basic spread spectrum techniques also combinations of them exit like a hybrid DS/FFH method which combines DSSS with fast frequency hopping (FFH) [Ola12].

Another problem of the 2.4 GHz ISM band regarding environmental monitoring by resource constraint WSN platforms is a higher transmit power requirement for a specific communication range. This demand is a consequence of the Friis transmission equation which states that the received RF power is proportional to the square of the wavelength applied in wireless transmission [Mas10]. By solving this equation for the transmission distance and expressing the range R as a function of the carrier frequency f we obtain

$$R = \frac{c}{4\pi f} \cdot \sqrt{G_T G_R \frac{P_T}{P_R}} \tag{1.1}$$

where c is the speed of light, G_T is the gain of the transmitter antenna, G_R is the gain of the receiver antenna, P_T is the transmitted power, and P_R is the received power.

Let us make a same kind of simplifying assumption as in [Ans12] for equal gains of TX/RX antennas for two pairs of wireless radios operating at different frequencies. Moreover, assume that the transmit power budget of both radios is the same and we want to know the achievable range when also the received power is fixed at a specific level. For this hypothetical setup we can conclude that a five times longer transmission range could be achieved by replacing a 2.4 GHz frequency with a 433 MHz frequency. In this respect it would be beneficial to utilize lower frequencies for wireless communication on WSN nodes that only possess limited power resources. One example where lower frequency data communication for an environmental monitoring system has been tested is TUTWSN [Suh06]. A group at Tampere University of Technology (Tampere, Finland) has selected a license-free 433 MHz frequency band for signaling on wireless radio links between the nodes in their WSN implementation. The group has managed to verify that even in demanding Nordic winter conditions a battery powered WSN system utilizing a lower frequency radio can reach a transmission range of a couple of hundred meters in average and up to 500 meters at maximum. This TUTWSN monitoring system is a low power implementation which is able to operate six to eight months with a 3000 mAh battery pack [Suh12].

On the software domain research has yielded diverse development tools for distributed WSN implementations. A wide range of operating systems have been made available for development of WSN systems. The most widely used operating systems on microcontroller based WSNs are TinyOS [Lev05], Contiki [Dun04] and Mantis OS [Bha05]. All of these WSN operating systems apply a low-level programming model according to a classification presented in [Sug08]. Several surveys explore and summarize the main features, strengths, weaknesses and challenges relates to the OS support on WSN platforms (see e.g., [Dwi09], [Don10], [Far11]). Real-time performance of these sensornet operating systems is one issue that has received increasing attention. Especially the original event-driven TinyOS, where high-level tasks could not preempt each other, has been enhanced with real-time extensions (e.g., TOS-Threads [Klu09], priority-based scheduling algorithm for TinyOS [Lei11], TOS-PRO (TinyOS PRemptive Original) [Lin12]).

The three most common WSN operating systems mentioned above are ported to sensor nodes equipped with Atmel Atmega128-series (TinyOS, Mantis OS) and Texas Instruments MSP430 microcontrollers (TinyOS, Contiki, Mantis OS) as well as certain ARM-processors (TinyOS). An additional operating system targeted at WSN systems is Enix OS, which is especially developed for Intel 8051 microcontroller based implementations [Che10]. Besides of these processor specific OS implementations an attempt has been made to develop a higher level operating system which would provide network level abstraction instead of hardware level abstraction. This operating system (RIOT) is primarily intended for Internet of Things (IoT) applications in which sensor networks, however, are an essential component [Bac12]. Therefore, RIOT could be applied in the development of WSN systems as well.

In the early days of WSN research the requirements of computational resources could be fulfilled by simple low-power microcontrollers. Instead of microcontrollers the power consumption of WSN nodes was dominated by wireless radio transceivers. Therefore, efficient access and utilization of RF hardware on physical layer was imperative. To address this issue several protocols have been implemented for wireless medium access control. These protocols include B-MAC (Berkeley Media Access Control) based on Low Power Listening method of wireless radio duty cycling [Pol04], X-MAC utilizing short preambles and early acknowledgments to reduce the power consumption [Bue06], LL-MAC (Low Latency MAC) in which the energy consumed is controlled by adjusting the frequency of control information packets [Mar08], CMAC (Convergent MAC) which combines broadcast and unicast transmissions to achieve a convergent packet forwarding mechanism [Liu09], TUTWSN MAC where low-energy operation is obtained by applying accurate synchronization and predefined channel access schedule thus avoiding need for carrier sense [Koh09], and A-MAC relying on automatic hardware acknowledgments in synchronization [Dut10]. The primary concern of these link-layer protocols has been energy-efficiency. A growing number of WSN deployments need to pay attention in end-to-end delay of sensed data transmissions. Therefore, issues related to delay guarantee of data delivery are becoming increasingly important [Dou13]. Besides of the actual link-layer protocol another essential consideration is the quality of the physical wireless link over which the protocol will be transmitted. The selection of reliable communication channels from interference-free wireless links is reducing the need for retransmissions and thus decreases data transfer delays. Therefore, channel selection mechanisms are required

[Hän11a]. Earlier research exploring link layer protocols' influence on the power consumption was mainly concentrating on battery-powered WSN nodes. More recently MAC protocols have been implemented also for WSN systems, where the nodes utilize energy-harvesting as their power source. This kind of systems pose different requirements on the design and analysis of MAC protocols, which must be specifically customized for energy-harvesting WSN deployments [Ian12].

Acquisition of sensed data from sensor nodes is one of the primary tasks of WSN systems. In order to facilitate this goal a rich set of data collection protocols have been implemented. Some examples of these protocols are Dozer (an ultra-low power data gathering protocol) [Bur07], CTP (Collection Tree Protocol) [Gna09], BCP (Backpressure Collection Protocol) [Moe10], DISSense (an enhanced version of CTP customized to support adaptive duty-cycling) [Col11], Transmission Scheduling Protocol for Data-Gathering WSNs [Gao12], BFC (Broadcast-Free Collection protocol) [Puc12], TDL (Token-based Data Leaching) and SPC (Single-Packet Collect) [Mik13]. The data collection protocol alone is not enough to tell how effectively a WSN implementation performs in transferring sensor data. An accompanied characteristics affecting the data collection performance is a network topology. A specific data collection protocol may perform differently depending on positioning of sensor nodes in the WSN system, dynamical properties of wireless links between the nodes and the location of the sink node(s) [Puc11].

One group of protocols closely related to the network topology are those dealing with routing of data packets from sensor nodes via intermediate nodes to the data sink. Most commonly routing in WSNs is of type many-to-one, but in certain implementations many-to-many or one-to-many data flows can also be applied. Due to the very nature of sensor networks available paths from data source(s) to data destination(s) may be unreliable and unpredictable. Thus, routing must adjust dynamically to those changes that occur in the sensor network topology. An example of this kind of adaptive routing service is presented in [Bou11]. The objective of their dynamic routing mechanism is to proactively predict the behavior of an unreliable multi-hop sensor network in order to forecast which links should be avoided in the future. Another dynamic routing protocol is SYNC2SINK in which routing decisions are based on link parameters (e.g., RSSI and hop-count to sink) extracted from sensor network's synchronization messages [Kiv11a]. By a proper choice of RSSI thresholds the routing algorithm is able to select sensible multi-hop routes which guarantee an optimal packet receive ratio from sensor nodes to the sink. An example of a routing approach in which sensor data origins from multiple sources and also multiple data destinations exit is MUSTER (MUltiSource MUltiSink Trees for Energy-efficient Routing) [Mot11]. A configuration of this kind requires special considerations for routing in order not to waste scarce resources of those intermediate nodes forwarding packets in the network. MUSTER aims in merging sensor data from different sources into a single data stream as early as possible. By transmitting data from multiple sources inside one stream and splitting this stream to multiple destinations only as late as possible MUSTER manages to reduce the number of nodes involved in sensor data transfer. This solution helps to decrease the total energy required for routing and to prolong the lifetime of the network. Overall, in WSN implementations energy is one of the main concerns which needs attention. Hence, this aspect is addressed also in other routing protocols like RPL (Routing Protocol for Low-power and Lossy networks) [Win12]. In outdoor deployments the wireless link quality may deteriorate temporarily for example due to atmospheric conditions. These kind of transient variations in the link quality affect the topology of the sensor network. In RPL, however, a routing topology will not be updated to follow the changes of the physical topology unless some sensor data needs to be transferred. As a result of validating the routing topology only while the demand arises RPL is able to reduce power consumption. Besides of energy minimization other objectives (e.g., low latency, low hop-count, high throughput, high link quality) may be used in optimizing routes on RPL enabled WSN nodes [Vas12]. The classification of WSN protocols into categories of data collection versus routing is not unambiguous. For example CTP is defined both as a data collection and routing protocol [Gna09].

Communication protocols on the transport layer aim to provide a reliable transmission for sensed data from data sources to the sink node. Basic components required for reliable transport of sensed data include packet loss detection, error recovery methods, congestion detection and transfer rate control. Reliable transport protocols can be divided in two main categories. First type of these transport protocols rely on an end-to-end error recovery scheme. In this mechanism all the above mentioned components are taken care by the sink. Upon detecting a lost packet the sink generates a negative acknowledgment message to initiate retransmission from the original source node. Also congestion of the network is monitored by the sink which controls the transfer rate of sensed data if it detects congestion. Another category of reliable transport protocols are those based on hop-by-hop retransmissions. One of the main differences between the end-to-end and the hop-by-hop mechanisms is that in the latter forwarding nodes are able to cache data packets. In case a forwarding node detects a loss of a packet (e.g., indicated by a missing sequence number) it can request a retransmission from its predecessor in the data stream. Because of this hop-by-hop protocols are more energy-efficient in comparison to end-to-end mechanisms which require a lost packet to travel the complete chain from the original source node via all forwarding nodes to the sink. A review of more than a dozen reliable transport protocols is found in [Aya11]. Some classification criteria for transport protocols are defined in [Sri11] which presents additional protocols utilizing congestion control without reliability guarantee.

According to [Sri11] Quality of Service issues of WSN systems are one research area requiring more attention. This aspect of sensor network design on several protocol stack levels is addressed in [Suh12]. Instead of focusing only on a single parameter of the network performance a set of QoS metrics parameters are defined. Besides of those parameters mentioned in conjunction with the RPL routing protocol (i.e., link quality, throughput, latency) additional QoS metrics like reliability, availability, network and node lifetime, node density and communication range are proposed in this framework. By selecting a set of suitable QoS metrics an application developer can define a QoS profile to be used in evaluation, comparison and verification of different WSN configuration alternatives from the QoS perspective.

Division of the protocol stack into separate layers is not always easy to establish, because dependencies exist between the layers. For example routing decisions may be tightly coupled with the link layer characteristics [DiM12]. Therefore, some research groups have developed communication protocols which do not split the network stack into several layers or all layers are embedded inside a single bundle. Instead of a multilayer stack these implementations glue an application directly to the operating system (or wireless radio driver) or alternatively split application data transfers and network management operations into separate modules (see e.g., cross-layer management entity [Hak06], IRIS [Cam12] or LWB (Low-Power Wireless Bus) [Fer12]).

A traditional strategy in protocol design is adopted by efforts to implement a complete IPv6 protocol stack for WSN operating systems [Ko11]. Running this kind of standardized protocol stack benefits in better interoperability of WSN nodes built using components from different manufactures. Integration of new hardware to a WSN system becomes more straightforward without a need to port implementation specific protocol layers to new HW platforms. Internet Engineering Task Force has proposed several standards to assist in development of IPv6 protocol stack implementations for low-power and lossy networks (LLN). One of the first IETF proposals [Mon07] defined IPv6 frame and header formats as well as methods for delivering these frames through a link-layer mesh network. Multi-hop routing in this context requires among others an adaptation layer with fragmentation and reassembling to adapt to the limited packet size available on LLNs. An implementation example following this IETF proposal is presented in [Lem10]. Development of the described NanoStack implementation was started as an open-source initiative and it has evolved later to a commercial product. Version 2.0 of NanoStack is compliant with IEEE 802.15.4 and IEEE 802.15 4g standards providing libraries for integrated RF-MCU components operating on the 2.4 GHz and sub-GHz frequency bands[8]. Even if the original IETF IPv6 standard was intended for IEEE 802.15.4 networks the proposed 6LoWPAN concept can be implemented on top of other MAC / PHY layers as well. An example of this kind of an effort is the work in progress to build IPv6 networks utilizing Bluetooth Low Energy as a wireless radio platform [Nie13]. In order to guarantee that WSN applications are able to apply IPv6 protocol in sensor data transmission special attention needs to be paid for instance on packet header compression [Hui11] and neighbor discovery [She12]. Both of these extra requirements are a consequence of the physical constraints on WSN nodes which can only provide a limited bandwidth, a reduced maximum transfer unit size and low power resources for communication with neighboring nodes. In conclusion, we can state that IPv6 is an option for WSN system integration if it can be implemented without an excessive usage of scarce WSN node resources.

Another way to provide a more unified interface for WSN applications to access the underlying hardware and protocol stack is to utilize a middleware layer. Already in the early days of WSN research first middleware (MW) software systems like Impala [Liu03]were developed. This lightweight MW implementation provided for example a possibility to dynamically update software modules on sensor nodes. The Impala MW architecture, however, was tightly customized to a specific WSN system (i.e., ZebraNet described earlier in this section) and resembled also partly a WSN OS. In [Wan08] an explanatory description exists of principal features and components included in typical middleware implementations. A reference model is also presented with which different MW implementations can be classified and compared. During the first decade of WSN development the research efforts concentrated heavily on sensor layer topics (HW, OS, SW, etc.) and wireless link related issues (as can be seen from the references on six previous pages). More recently the abstraction level in development has been increasing

[8] http://www.sensinode.com/EN/products/software.html

resulting in several groups focusing their work on the middleware layer on top of the sensor layer and network protocol stack. Some of these research groups envision a global utilization of wireless sensor networks and lean towards Internet of Things and Web of Things type solutions. Examples of recent MW developments include a Java based middleware SAL (Sensor Abstraction Layer) used in SEMAT SNMS 2.0 (a Sensor Network Management System of the Smart Environmental Measurements and Analysis Technologies project) [Lee11], a new generation of Sensor Web Enablement (SWE 2.0) [Brö11], SensorsMW based on Service-Oriented Architecture (SOA) [Ana12], a five-layer component-based SOA approach for environmental monitoring networks [Dau12], and a Global Sensor Network (GSN) MW [Per12]. Even if vast amounts of progress in development and standardization has been achieved there still remains challenges for the MW research (e.g., how to enable automatic integration of new sensors to application layer programs without a need for manual coding of adapters or wrappers for the added sensors).

Already at the development phase and still during the deployment of a wireless sensor network specialized tools are needed to assist in realizing reliable and robust systems. Even if the layout of sensor nodes and the sink remains more or less stable in the majority of WSN implementations dynamic properties of wireless links may vary thus affecting the topology, operation and performance of these systems. Therefore, tools and methods to collect, store, monitor, analyze and visualize parameters related to wireless sensor networks' dynamics have been developed. In [Liu10] WSNs are diagnosed using an almost passive method in which only marginal amounts of additional data is attached to normal sensor data frames. This is achieved by a packet marking scheme that adds diagnostic information only occasionally into data packets transferred in the network. Another diagnostics method is the use of neighbortables [Hak12]. These tables can be generated locally on sensor nodes by extracting network related parameters from synchronization and management messages passed within the network. Diagnostic data stored in the neighbortables include e.g., radio link quality, battery levels, number of received packets, neighbor lists, and neighbors' hop counts to sink. An accompanied tool called CiNetView enables a user to visualize the current structure and state of the WSN system as well as to present statistical and diagnostic parameters of wireless link dynamics [Hak10a]. One more analysis tool called MNT (Multi-hop Network Tomography) to explore long-term performance of data packet delivery in real-world WSN implementations is described in [Kel12]. The MNT algorithm aims in inferring per-hop arrival order and arrival time for sensor data packets from information available on the sink. The observed packet order and timing information is used to reconstruct the actual packet paths and to monitor the flow of sensor data packets in a WSN. The aforementioned analysis solutions rely on in-network diagnostics. In these implementations the diagnosed WSN participates in the monitoring process itself. An analysis approach in [Hän11b] differs in that a passive observer is added inside a WSN to capture ongoing traffic events without participating actively in the operation of the diagnosed sensornet. The passive observer in this approach is an external sniffer device which is capable of monitoring wireless traffic on a sensor network and timestamping captured packets with an accuracy of 16 μs. Diagnostics data obtained include similar parameters as stored in the neighbortables. From the sniffer device the captured diagnostics information can be transferred to network analysis software for further analysis. This network analyzer includes diagnostics tools for different phases of WSN lifecycle. A bit different features are available for designers, users and maintainers of WSN systems during the development, deployment and maintenance phases.

1.4 Environmental Noise Monitoring

Currently most environmental noise inventories are carried out by using integrating sound level meters. Usually these stand-alone meters are used for short-term measurements on one place at a time. An increasing demand, however, exists for long-term inventories lasting several weeks or months. Instead of monitoring environmental noise just in one measurement location it is necessary to measure noise simultaneously in several places, which may even locate geographically far away from each other. Integrating sound level meters are not suitable for implementing this kind of distributed monitoring systems for two reasons. Firstly, the setup, monitoring and analysis of measurements using these individual unconnected tools require too much manpower. Secondly, integrating sound level meters are too expensive to be kept in multiple measurement locations for long periods of time. An example of these limitations is clearly seen in the assessment method used to prepare a Large Analysis and Review of European housing and health Status [Bon07]. The LARES survey was conducted in eight European cities which were scattered around an area of more than 550000 km^2. Due to the reasons given above environmental noise could not be measured, but the assessment method relied solely on subjective estimates of interviewed residents. In another survey environmental noise induced risk on myocardial infarction was evaluated [Sør12]. Estimates of environmental noise exposure were based on residential address history of inhabitants. During an eight year long follow-up period the noise exposure was estimated four times. These estimates were calculated for the facade of the buildings outside of the dwellings by noise modeling software. This noise exposure estimation method relying on modeling includes several sources of uncertainties. Evaluation of environmental noise induced health risks would be more reliable with a suitable distributed monitoring system based on measurements of actual environmental noise levels.

Several attempts have been made in the course of the last five years to address the above mentioned problems. Three major categories can be identified in these attempts, namely wireless sensor networks, participatory sensor networks and external acoustic sensor boards. A couple of typical examples for each of these categories are given in the following subsections.

1.4.1 Wireless Sensor Networks

As explained in Section 1.3 wireless sensor networks have been applied in monitoring of a diversity of simple environmental parameters. One of the first WSN experiments related to monitoring of environmental noise was reported by researchers from ETH Zürich, Switzerland and Sapienza University of Rome, Italy [San08]. Their main interest was to find out how well wireless sensor networks are suited for environmental noise monitoring. Especially they wanted to explore the possibilities of mote class sensor nodes which had been the most common platform in monitoring simple environmental parameters. As opposed to these simple parameters monitoring of environmental noise requires considerable amounts of processing power from the monitoring system. In order to test the feasibility of motes for this purpose they implemented an Environmental Noise Watcher (Ennowa) application on two mote class platforms (i.e. Tmote Sky and Tmote invent). As expected those resource constraint microprocessor based sensor motes had difficulties to cope with the computational demands posed by the Ennowa application. For example the Tmote invent prototyping platform was in principle capable of sampling an omnidirectional electret microphone with a sampling rate of 32 kHz, which would have been more or less adequate. In practice, however, Ennowa experiments had to be conducted applying an 8 kHz sampling rate. Another limitation due to constraints in computational resources was that A-weighting had to be excluded completely in the Ennowa experiments. As a result of this eight Ennowa nodes were monitoring unweighted sound pressure levels (Leq,1s) and communicating acquired noise levels to a central sink. The WSN sensor nodes and the sink were only concentrating on the noise level data. Timestamping, processing and noise map generation was dedicated to a more powerful computer capable of running MATLAB. For this computer the Ennowa group developed a tool called tinyLAB to remotely control and monitor their WSN system as well as to analyze and visualize the data collected from the sensor nodes. Besides of the above mentioned limitations Ennowa experiments revealed also that calibration of sensor nodes is an open issue which must be solved before WSNs can be utilized in environmental noise monitoring.

Another WSN system targeted for environmental noise monitoring is CiNet, which is designed, implemented and tested at University Consortium Chydenius in Kokkola, Finland [Kiv11b]. CiNet sensor node hardware is built around an 8-bit ATmega128L microcontroller manufactured by Atmel. Environmental noise signals are digitized with a sampling rate of 33 kHz on CiNet sensor nodes using a 10-bit ADC built-in to the Atmel microcontroller. Special arrangements are required to obtain a wide enough dynamic range for the measured signals while using this internal ADC. Therefore the output signal from an omnidirectional electret microphone (Monacor MCE-400) is routed via two alternative signal paths and passed through an analog A-weighting network to the ADC input. By switching automatically between these two signal routes the CiNet sensor node can select different gains and manages to increase the dynamic range of the digitized signal by more than 30 dB. This arrangement extends the measurement range up to 100 dB.

A periodic scheduling of sampling and communication tasks is implemented utilizing a cycle time of five seconds. In each cycle three phases can be identified, namely synchronization, sampling and data transfer. Every scheduling period starts with the SYNC phase during which clocks of all CiNet nodes are synchronized with a clock running on a CiNet sink node. This synchronization enables to implement timestamping for the sensed data in CiNet. Timestamping of measurement data already on the sensor node level is important for synchronization of environmental noise events detected by individual WSN nodes. This timestamping implemented in CiNet is an advantage compared to Ennowa where the WSN nodes lack the timestamping feature.

The second phase of the periodic task scheduling is sampling of environmental noise signals. During this four second long sampling phase the microphone signal is sampled continuously and the digitized noise samples are stored in an internal 4 kB SRAM. This continuous sampling prevents CiNet sensor nodes to apply sleep modes to reduce the power consumption. By default CiNet sensor nodes apply 'Fast' time weighting on the digitized environmental noise samples. The acquisition software, however, is written so that the time weighting can easily be reconfigured to 'Impulse' or 'Slow' by modifying one acquisition parameter.

Finally one second is allocated for the third phase of the task scheduling during which A-weighted sound pressure levels are transferred from the sensor nodes to the sink node. This data transfer phase relies on multi-hop communication where intermediate CiNet nodes are relaying data from other nodes towards the sink node which resides at the root of the network tree topology. Routing for the measurement data transfer to the sink is established already in the SYNC phase. Because possible routes to the sink are checked in conjunction with every SYNC phase measurement data can be rerouted dynamically in case a malfunction prevents a certain CiNet node from acting as a link in the data transfer. Physically the network part of the CiNet platform is implemented using a wireless 2.4 GHz radio link (Texas Instruments CC2420) for which a cross-layer protocol stack has been written.

An additional strength of the CiNet implementation is that the developers have paid attention on calibration issues of the sensor nodes. The calibration is conducted by comparing digitized A-weighted output voltages of CiNet sensor nodes to reference sound pressure levels obtained by a Class 2 SLM (Pulsar Model 94). One second long RMS averages of the ADC outputs from four different CiNet nodes were compared against LAeq,1s values from the reference SLM. The comparison of responses on 22 separate sound pressure levels in the range 32 – 95 dB was used to derive an empirical fitting equation for the calibration. The correlation coefficient between this fitting curve and the prototype sensor nodes' ADC outputs was very good (0.99493). An advantage of the applied calibration method is the use of over twenty calibration points instead of only one SPL (i.e. 93.8 dB) as is commonly the case in SLM calibrations. A minor drawback of the described calibration scheme is the use of a Class 2 SLM, but it is easy to overcome this defect by replacing the SLM with another that meets the Class 1 specifications. As a matter of fact an earlier article from these authors [Hak10b] reports a calibration by a Class 1 SLM (CESVA SC-20c integrating-averaging sound level meter). The developers of the CiNet platform have also taken into account the drift in the sensor node component values due to outdoor deployment conditions. To compensate this drift they have worked out a field calibration procedure which can be run before outdoor measurements to finetune the calibration coefficients of their WNSN nodes.

Outdoor experiments have been conducted to test the accuracy of this wireless noise sensor network (WNSN) in a realistic environmental noise assessment setup. In Nordic winter conditions traffic noise from a highway was measured with six CiNet sensor nodes. Comparison with a Class 2 SLM confirms an average error of less than +-2.1 dB in assessment results between the tested CiNet nodes and the reference SLM. The observed inaccuracy is partly due to the fact that CiNet sensor nodes are occupied 20 % of the total time in non-sensing tasks (i.e. data communication and time synchronization).

1.4.2 Participatory Sensor Networks

Thus far three methods have been mentioned for environmental noise assessment and monitoring, namely

- measurements conducted using unconnected integrating SLMs
- generation of noise maps by mathematical modeling
- utilization of wireless sensor networks

Recently a fourth method involving mobile phone users has gained a lot of attention. It has been argued that this method could provide much better areal coverage compared to WSN systems [Chr11]. Several experiments have been reported to generate noise maps by applying smart mobile phones to sense environmental noise (see e.g. [Hua10] or [Kan10]). This method is referred to using a variety of terms, such as collaborative sensing, public sensing, people-centric urban sensing, collaborative mobile phone sensing, participatory sensing, participatory noise mapping, public participatory mobile phone sensing [Rua11], [Sil12], [Kan13], [Kha13]. General idea behind participatory noise sensing is that a smart mobile phone is acting as a noise sensor contributing data to a collective noise map which is generated and made available to general public by an Internet server. In participatory sensing systems mobile phone users are deliberately participating in environmental noise sensing. Because ordinary citizens are both data providers and data consumers in collaborative noise mapping these systems are also called end-to-end participatory urban noise mapping systems. Prerequisite for using mobile phones as environmental noise sensors was explored already earlier by the Ennowa group [San09]. Experiments conducted by them pointed out several aspects which require attention for a successful application development. These development challenges include for example individual calibration of mobile phones, disabling of noise cancellation and bypassing low-pass filtering.

An example of participatory sensor network (PSN) approach applied to traffic noise mapping is an Ear-Phone mobile sensing application [Ran10]. It was developed for monitoring of ambient roadside noise as a collaboration between University of South Wales (Sydney, Australia), Portland State University (Oregon, United States) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO ICT, Australia). Ear-Phone developers criticized EU compiled strategic noise maps of being updated all too infrequently. According to their argument update interval of noise maps generated by PSN systems could be more frequent – typically days instead of five years as in case of strategic noise maps. Ear-Phone is in a sense a WSN system in which mobile phones are acting as wireless sensor nodes. Participatory mobile phone sensing, however, differs from actual WSN implementations in that the sensing nodes are mobile instead of being statically installed like the nodes typically in WSN systems.

Ear-Phone architecture consists of mobile phones and a central server. Ear-Phone is implemented using Java application threads on Nokia N95 and HP iPAQ mobile platforms. These smartphones running an Ear-Phone application comprise a wireless sensor node called MobSLM. In Ear-Phone the mobile phone is responsible for calculating A-weighted equivalent sound pressure levels. To facilitate this a 10th-order digital filter has been designed which approximates the IEC651 Class 2 A-weighting specification within the frequency range 0 – 8 kHz. Using a built-in microphone the MobSLM samples ambient traffic noise signals, weights them with the digital A-weighting filter and computes one second long LAeq,1s noise levels. The MobSLM stores these measured traffic noise levels locally on data records which also include a timestamp and GPS data for the current measurement location. The collected data records are kept on MobSLM's local memory until the phone detects connectivity to an available wireless link through which it transmits these records to the central Ear-Phone server.

What comes to the accuracy of Ear-Phone it has been observed that uncalibrated LAeq,1s noise levels differ from those measured by a reference Class 2 SLM roughly by 10 dB. Part of this difference is likely to be due to the approximation of A-weighting in the Ear-Phone implementation. A correction of the MobSLM measured noise levels by the calibration offset mostly compensates for the observed discrepancy. Using calibrated MobSLM phones an accuracy of +-2.7 dB compared to the reference SLM levels has been verified in the experiments conducted by the Ear-Phone developers. The accuracy of Ear-Phone measurements varies depending on the position in which the mobile phone is kept. Outdoor test results show an accuracy within 2.7 dB, 3.4 dB or 4.1 dB for different phone positions, namely the phone held in the palm, inside a shirt pocket or on the waist and finally inside a backpack, respectively.

Mobile users participating in collaborative noise mapping execute random walk and therefore measurements are taken randomly in space and time. From these incomplete and random samples the central server belonging to the Ear-Phone architecture must try to generate noise exposure maps for an urban area. A problem arises of how representative noise maps can be produced from this randomly aggregated data. Ear-Phone attacks this problem with a so called compressive sensing approach [Ran11].

An urban area to be mapped is divided into grid elements using two different granularities. While MobSLM users are collecting LAeq, 1s noise level readings a measurement grid of size 10m * 10m is used. During the reconstruction phase on the Ear-Phone server traffic noise samples measured within a larger 30m * 30m area are aggregated into one reconstruction grid element in a community noise map. In case measurement data for every measurement grid element inside a reconstruction grid element would be available for every second a long-term LAeq could be calculated from a complete samples set. Bearing in mind the very nature of participatory sensing an unknown amount of measurement grid elements will always lack data either in respect of spatial or temporal dimension. Therefore reconstruction of long-term noise exposure maps is always based on incomplete samples set. The Ear-Phone developers, however, claim that by compressing data both in spatial (i.e. from multiple measurement grid elements) and temporal (i.e. LAeq,1s readings from a period of one or several hours) sense as well as from multiple participants an adequate reconstruction accuracy can be obtained. This claim is supported by the evidence they have gained through outdoor experiments and simulations. In a specific roadside experiment they discovered that already five participatory users walking along a certain route are able to supply adequate amounts of data to successfully run compressive noise map reconstruction algorithm. Besides of this outdoor experiment Ear-Phone developers have simulated generation of noise exposure maps with different parameters. These simulations suggest that adequate accuracy in creation of the noise maps can be achieved even if 40 % of the samples belonging to a reconstruction grid element would be missing.

Another participatory noise mapping solution is NoiseTube [Mai10]. The development of it started at the Vrije Universiteit Brussel (Brussels, Belgium) in cooperation with the Sony Computer Science Lab (Paris, France). NoiseTube platform consists of a smart phone supporting the Java J2ME with certain extensions providing facilities for GPS localization and enabling recording of raw audio signals in suitable format (e.g. PCM, WAVE, AIFF). Maximum sampling rate available for digitization of these signals is 48 kHz. Another component of NoiseTube is a real-time signal processing application executed on the smart phone to calculate environmental noise indicators locally. Wireless communication capabilities of smartphones are utilized to transfer calculated noise pollution levels to a web-based community memory server.

Besides of hosting the community memory for environmental noise indicators the web server provides tools for presenting and post-processing mobile phone sensed data stored in this memory. Among these tools reverse-geocoding is utilized to automatically convert GPS coordinates embedded in the measurement data into semantic annotations (e.g. name of a city or street) in the post-processing phase on the community memory server. Also the possibility to aggregate all sensed data within a confined region (e.g. a street segment) into a single SPL value helps to create illustrative noise maps (i.e. aggregated noise maps). Creation of these aggregated noise maps resembles the compressive sensing approach to noise mapping introduced by the Ear-Phone group.

Already other contributions presented in this section have identified the calibration as a major issue while monitoring environmental noise with WSN or PSN systems. Likewise the NoiseTube group shares this concern and pays extensive attention on evaluation of possibilities to calibrate mobile phones in a proper way [DHo13]. As an additional bonus this evaluation explores also frequency response aspects of environmental noise measurement channels on mobile phones. The frequency response of a subset of tested phones is compared with each other as well as against a PC based laboratory reference system. In an anechoic room a frequency scan using pure tone frequencies from 27 consecutive 1/3 octave bands within a frequency range 50 – 20000 Hz is utilized. In order to find out the frequency response of the mobile phones and the reference system these frequency scans were run through seven times with different output sound pressure levels starting at 60 dB and increasing in 5 dB steps. Unfortunately the frequency response results were set aside and a frequency dependent calibration option was abandoned. Instead, a frequency independent calibration method was chosen which concentrated only on deviations of broadband sound pressure levels between the mobile phones and the laboratory reference system. Calibration points for this approach were determined using a white noise acoustic signal and a wider SPL range of 30 – 105 dB. This time the complete set of ten test phones (all of the same model from a specific manufacturer) were checked to obtain individual calibration coefficients for them. Applying this white noise calibration method the original nonlinear SPL response of these phones could be corrected close to an ideal linear response. Between the 16 actual calibration points measured for each phone a piecewise linear approximation was applied to calibrate the intermediate SPLs. In the CiNet WSN system mentioned earlier the developers managed to work out a logarithmic calibration curve to link the ADC output voltages with the sound pressure levels.

Following the calibration some field measurements were conducted. One of the mobile phones equipped with the NoiseTube software was compared with a Class 2 SLM (CEM-DT8852). During a test walk in an urban outdoor environment the measured noise levels differ less than 0.2 dB. This accuracy applies to the total 81 minute walk as well as for a shorter 25 minute period at the beginning of the field test. Wind induced errors, however, are much bigger, because the mobile phone microphone was not equipped with a windshield. In the worst case wind was shown to cause almost a 10 dB error on an LAeq,4min value along the test route.

It was also observed that GPS coordinates for a specific urban location given by mobile phones in repeated measurements varied in average by roughly 2.5 meters while the maximum difference was about 8 meters. This knowledge acquired for the precision of GPS coordinates was used while selecting a suitable grid size for the noise maps in participatory measurement campaigns. Two campaigns were run of which the latter (called Phase 2) resembled more closely to a situation that might be relevant in real-life deployment of NoiseTube. The intention of the campaign was to assess noise in the Linkeroever area on the left bank of the river Scheldt in the city of Antwerp, Belgium. Ten volunteers used NoiseTube phones for five days to take one second long measurements roughly one hour daily while walking freely inside a predefined noise mapping area. In one minute NoiseTube phones can collect 30 measurements. During this five day experimental campaign more that 84 000 LAeq,1s values were acquired from an area of one square kilometer. Phase 2 results are compared with an EU compiled strategic noise map showing Lden. In the analysis of the NoiseTube phase 2 campaign the grid size had to be increased from 20m * 20m to 40m * 40m in order to obtain statistically significant data for the grid elements. Those grid elements possessing at least 50 measurements were regarded to have adequate statistics for noise mapping. A daytime participatory noise map agrees rather well with the simulation based strategic noise map showing Lden estimates. Participatory noise maps created for this Ademloos (i.e., name of the voluntary group) noise sensing campaign can be accessed online[9].

One benefit of this type of participatory urban sensing is the elevated awareness of citizens regarding the environmental noise problem. Mobile phone sensing of environmental noise can act as a personal noise dose meter besides of supplying data for collective noise mapping. For example the NoiseTube application is able to warn the user of the noise annoyance level using a color coded indicator – green, yellow or red to be interpreted as no risk, be careful and risky, respectively.

An important aspect to be considered while applying mobile phone sensing to traffic noise assessment is privacy of mobile users participating in generation of collective noise maps. NoiseTubePrime [Dro12], a privacy-preserving extension of the original NoiseTube, addresses this concern. NoiseTubePrime relies on agent based technology in which software agents executed in a cloud service process encrypted data received from individual mobile phone users who participate in environmental noise measurements. These software agents accompanied with public key cryptography enable to aggregate participatory noise maps still preserving participating users' privacy.

Some experiments exist where more mobile components have been added to a PSN system. An example of these tests is a VANET Noise Monitoring System (VNMS) [Rad12], which is a collaboration between Technical University of Cluj-Napoca, Romania and Polytech Nantes, France and École Polytechnique, France. Also in the VNMS

[9] Ademloos participatory noise maps, http://www.brussense.be/experiments/linkeroever

system mobile phones act as traffic noise sensors. What differentiates VNMS from PSN is that a Vehicle Ad-Hoc Network (VANET) is used as a temporary sensor data storage and data carrier. One of the vehicles in VANET possesses the role of a sink node and provides a gateway to the Internet. Compared to other mobile phone sensing approaches one advantage in VNMS is that a lightweight application on mobile sensing devices will suffice, because part of the wireless network administration tasks (e.g. routing decisions or packet forwarding) can be delegated to VANET nodes.

1.4.3 External Acoustic Sensor Boards

Resource constraints on microcontroller based mote class WSN platforms make it hard to sustain high enough sampling rates (~ 32 – 41 kHz), which are required in environmental noise monitoring. Therefore some research groups have implemented separate external acquisition and signal processing modules to reduce computational load on wireless sensor nodes and mobile phones. Already 2008 the Ennowa group reported that they tested an external noise level meter together with their WSN system [Fil08]. This dedicated sensor board was intended for sampling and processing of acoustics signals (e.g. filtering and weighting). The external noise level meter was able remove the most demanding tasks from a Tmote node, which only had to fetch the calculated noise level values with a much lower rate compared to the sampling rate of the acoustic signals.

To overcome problems experienced with built-in microphones of mobile phones a research group at Vanderbilt University (Nashville, USA) has specifically designed an external microphone board for mobile phone sensing applications [Völ11]. An example application described in their research article is acoustic source localization, but this smart acoustic sensor board hardware could be used for environmental noise monitoring purposes as well. Equipped with a powerful ARM-Cortex M3 embedded processor the sensor board is capable of continuously monitoring environmental noise using a sampling rate of a few hundred kilosamples per second. This sampling rate exceeds the requirements posed by environmental noise monitoring, but is an indication of computational resources provided. In this respect the small external sensor board (which measures 3cm by 7cm) is superior compared to the microcontrollers in WSN and PSN implementations mentioned earlier in this section. On the downside, of course, is the power consumption. The current implementation is equipped with a 1000 mAh Li-ion battery which can supply the operation of the external sensor board only for eight hours.

Third example of tailor-made environmental sensing platforms is TECO EnvBoard designed at Karlsruhe Institute of Technology, Germany [Bud12]. The TECO EnvBoard is packed with a rich set of different sensors for monitoring environmental parameters including noise. One of the built-in sensors is a Panasonic omnidirectional electret microphone (WM-61A). The board is equipped with additional sensors (i.e. motion / vibration / orientation) which provide information of the user activity (walking, cycling, etc.) and the position of the board. These additional sensors make it possible to add context-awareness features into the conducted measurements. The TECO EnvBoard is composed of commercial of the shelf (COTS) components and built around an 8-bit Atmel ATmega 2561 microcontroller using a C/C++ based open-source

Arduino language and Arduino development environment. Wireless connectivity is provided by a class 2 bluetooth transceiver (Bluegiga WT12), which enables the EnvBoard to communicate with mobile user's smartphone. This wireless communication channel provides an alternative route to store measurement data besides of an internal Micro SD card. A difference in comparison to mobile phone sensing platforms is that the TECO EnvBoard is a custom board dedicated to environmental monitoring only instead of being a general mobile communication device like smartphones used for example in NoiseTube.

1.5 Background and Focus of this Research

Increasing awareness of environmental noise problems has initiated an intense search for more accurate and reliable methods for measuring and predicting the risks of noise pollution on inhabitants. Besides of those groups mentioned in the previous section also a small Central Finland based company (Hansa Ecuras Oy) has started an R&D project to alleviate shortcomings in the current environmental noise assessment methods. The goal of Hansa Ecuras' Dimeca project is to develop a distributed measurement system for environmental noise monitoring which enables centralized analysis of measurement results.

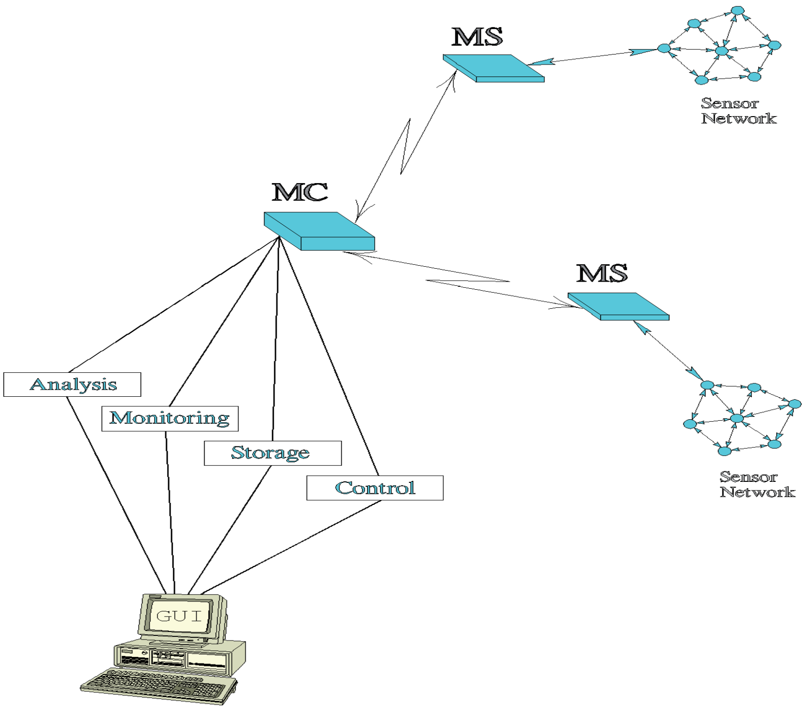


FIGURE 1 Architectural overview of Dimeca

As depicted in Figure 1 the Dimeca (Distributed Measurements with Centralized Analysis) architecture will consist of three main components. Measurement Stations (MS) are distributed across places were monitoring of environmental noise is desired. The purpose of a measurement station is to register, digitize and preprocess signals detected by a sensor network. Besides of acoustic transducers also sensors recording physical parameters affecting the propagation of sound (temperature, humidity etc.) are to be connected to each measurement station. A Measurement Center (MC) will be implemented to provide users with tools for setting up, monitoring and analyzing measurements in a centralized manner. Report generation for long-term inventories (e.g. yearly average noise levels) will be one of these tools. A third component of the Dimeca architecture is going to be a wireless communication link connecting measurement stations to the measurement center. Via this link users operating on the measurement center can control their measurements and collect results from remote locations. One design goal of Dimeca is to develop a distributed architecture with good scalability. Therefore, each measurement station will possess enough processing power to enable generation of basic environmental noise indicators already at the measurement station level. This kind of preprocessing on the front-end reduces the load of the measurement center and results in better scalability of this distributed monitoring system.

Signal conditioning of measured environmental noise signals in traditional sound level meters differs from the approach intended in Dimeca. Traditionally, the steps taken in processing of microphone's analog signals can be depicted as follows.

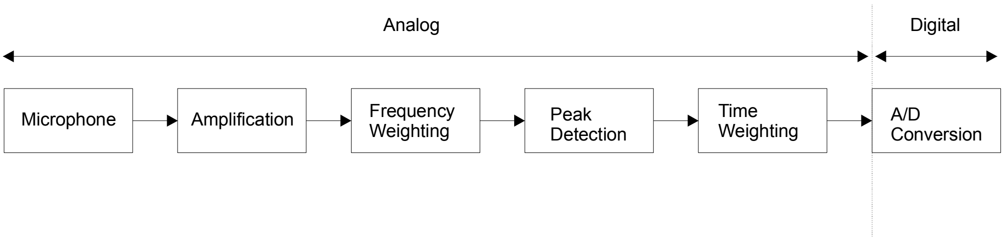


FIGURE 2 Traditional signal conditioning flow of environmental noise signals

The measurement signal is already frequency and time weighted while entering the analog to digital conversion stage in traditional sound level meters. Digital processing approach of Dimeca sets a goal to digitize the measured environmental noise signals in as early stage as possible. The signal conditioning path looks therefore like the one in Figure 3.

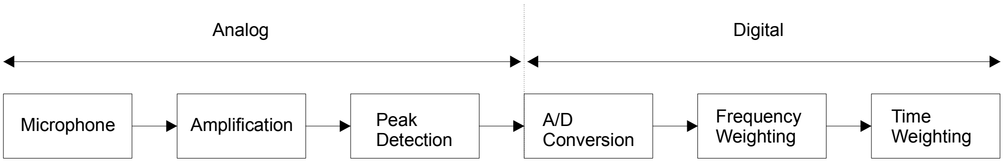


FIGURE 3 Digital approach to conditioning of environmental noise signals

The research in this thesis concentrates on one of those tasks that must be performed digitally by measurement stations. The digital subsystem on a measurement station receives as an input a stream of digitized samples representing sound pressure variations detected by acoustic transducers. The output data expected after digital preprocessing are various environmental noise indicators. These indicators include:

- effective (RMS) sound pressure
- sound pressure level (L_p)
- minimum / maximum L_p (L_{min} / L_{max})
- fast A-weighted L_{min} / L_{max} (L_{AFmin} / L_{AFmax})
- peak value for L_p (L_{peak})
- C-weighted peak value (L_{Cpeak})
- equivalent continuous A-weighted L_p (L_{Aeq})

An established standard procedure involved in the calculation of several of the above mentioned indicators is frequency weighting. The origin of this lies in the fact that the frequency response of human ear differs from that of a microphone. Multiple weighting curves have been invented to compensate this difference and to bring measurement results closer to human perception of sound. The most commonly used of these is a so called A-weighting which attenuates low frequencies (~20 Hz) by 50 dB, keeps middle frequencies (~1-4 kHz) unchanged and applies a smaller attenuation (~15 dB) on high frequencies (~16 kHz).

Traditionally A-weighting is implemented in the analog domain as depicted in Figure 2. For example the CiNet platform relies on analog components in implementation of A-weighting. In Dimeca the intention is to realize A-weighting digitally. Therefore this work, which provides background research to assist in the development of Dimeca, aims in finding a suitable digital implementation for A-weighting on an embedded systems platform.

This research is related to earlier work of others presented in the previous section in the following way. The Ennowa group postponed the application of A-weighting in their WSN system, because Tmote sensor nodes they experimented with were lacking the required processing power. This difficulty regarding computational resources was faced also by the CiNet group who stated explicitly that

> "... theoretically such a filter can be implemented by software
> as a digital filter. However, a digital filter involves excessive
> floating point calculation which definitely surpasses the limit
> of 8-bit ATmega128."

A digital A-weighting filter was designed for the Ear-Phone PSN application to approximate Class 2 specifications, which request only a subband (i.e., 0 – 8 kHz) of human auditory range to be covered. Participatory sensing experiments reported by the NoiseTube group indicate the use of A-weighting, but the group do not give implementation details for the applied filter. For the TECO EnvBoard no knowledge exists of any frequency weightings. The original target application for the ARM-Cortex M3 based acoustic sensor board is not related to environmental noise assessment. Therefore, A-weighting is not implemented for this external sensor board.

As a summary we can state that the development thus far has provided several solutions which do not possess enough processing power for a digital implementation of A-weighting. In addition to these systems a couple of mobile phone DSP applications exist where A-weighting is implemented with an accuracy which may not fulfill the official requirements of environmental noise assessment. Finally, we referred to one external sensor board which could offer more than adequate computational resources for the challenge, but is not customized for environmental noise monitoring. Therefore, one missing fragment that remains to be done is to implement A-weighting digitally on a powerful enough platform which can be applied in distributed monitoring of environment noise. The primary focus of this thesis is to develop this kind of filtering algorithm for the ARM processor architecture.

2 BASICS OF HUMAN HEARING

In this chapter basics for a physical and physiological structure of human ear as well as primary functionality of human hearing are explained. We want to find answers to the following questions. What is the state of a human auditory system when nothing is heard? What processes occur when a sound pressure fluctuation enters human ear? Which parameters affect the propagation of sound signals inside the ear? What is the influence of different frequencies on physiological responses?

Several models exist to describe human auditory system's actions required to decode information embedded in physical pressure fluctuations. One of them is a model for signal processing in the human auditory system [Sot05]. According to this model four steps can be identified while humans' auditory system is processing sound pressure signals. The stages involved are preprocessing of acoustic signals in the outer and middle ear, vibrations along the basilar membrane, firing of auditory nerves in hair cells and finally neural processing in the brain. In this chapter we concentrate on the two first stages mentioned in this signal processing model. These stages are the most relevant ones in regard to frequency weightings discussed more in detail later in Chapters 5 and 6.

2.1 Overview of Human Auditory System

Physically the human auditory system can be divided roughly into two parts – that is, a peripheral auditory system and a neural auditory system. A schematic division of components in the human ear belonging to the above mentioned two main blocks of the auditory system is depicted in Figure 4.

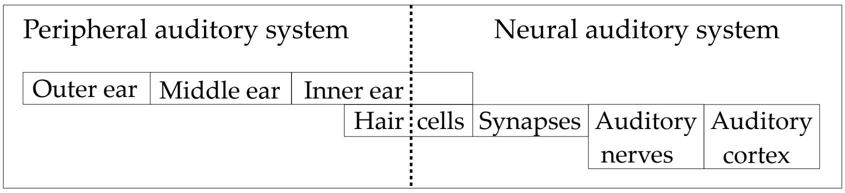


FIGURE 4 Physiological division of human auditory system

On a general level responsibilities between the peripheral and neural auditory systems can be defined as follows. The peripheral auditory system functions as a pressure transducer to transform physical sound pressure signals into electric form. As a result of this signal conditioning phase the original sound stimuli are converted to neural impulses which are transmitted further via the neural auditory system. Endpoint of this signal transmission is an auditory cortex in the human brain which weaves together a final perception of sounds.

In the peripheral auditory system there are three tightly integrated components which each have their own dedicated function. A short summary of responsibilities that are taken care of by these components is presented in the table below. More details for the tasks each component of the peripheral auditory system must deal with are given in the forthcoming sections.

TABLE 1 Main tasks of different components in peripheral auditory system

Outer ear	Focusing sound energy into the ear canal
Middle ear	Adapting differences in acoustic impedance between air and cochlear fluids
Inner ear	Resolving the frequency contents of incoming sound signals

2.2 Preprocessing of Acoustic Signals

The outer ear resembles an astronomical radio telescope which is focusing energy from weak signals to a more confined space thus increasing the energy density. The outer ear focuses incoming acoustic energy into an ear canal. Along this 2.5 cm long waveguide sound pressure signals propagate to an eardrum which separates the outer ear from the middle ear. Density variations of the air molecules in the ear canal bring the eardrum in to vibration. This leads to movements and acceleration of hearing bones in the middle ear. A hammer attached to the eardrum is a tiny bone following the vibrations of the eardrum. From the hammer these mechanical vibrations are further transferred to the inner ear via two additional bones, namely an anvil and a stirrup (see Figure 5). The final link between the middle ear and the inner ear is the stirrup which mediates the vibrations from the anvil to a base of a cochlea. The main part of the inner ear is the cochlea where the oval window is serving as a connection point for the stirrup and as an input window for the vibrational energy. Through the oval window by the stirrup footplate functioning as an actuator mechanical vibrations are mediated to a fluid filled chamber inside the cochlea.

What is the functional meaning of the above described construction in preprocessing of acoustic signals? For what do we need an additional bone complex (i.e., hammer, anvil, stirrup) in between the outer and inner ear? The ear canal is filled with air whereas fluids inside the cochlea resemble water [Kim11]. Acoustic properties of these two media differ considerably. Therefore, sound pressure waves would need to penetrate through a boundary between two acoustically different media in case these so called temporal bones would be absent from the middle ear. Only part of the incoming sound energy could be transmitted through this boundary. An essential parameter characterizing this transmission is an acoustic impedance *Z* defined as a product of the density of the medium and the speed of sound [Cer01]. For air in room temperature we obtain an acoustic impedance $Z_{air} = 1.29\ kg/m^3 * 344\ m/s = 444\ kg/m^2s$. Corresponding impedance for the cochlear fluid using acoustic properties of sea water is $Z_c = 1000\ kg/m^3 * 1520\ m/s = 1\,520\,000\ kg/m^2s$. From these values we realize that the acoustic impedance of air is only 0.03 % from that estimated for the cochlear fluid. In case the fluctuations of sound pressure would be knocking directly on the oval window only a small fraction (~ 0.1 %) of the energy would become transmitted. This transmission ratio is a rough average, because the division between reflected and transmitted waves is a more complex frequency dependent function [Goo94]. Majority of the energy would bounce back to the ear canal due to a mismatch in acoustic impedance between the receiver and the signal source (cochlear fluid vs. air). Thus, matching of acoustic impedance required to increase transmission of energy into the inner ear is provided by the temporal bones.

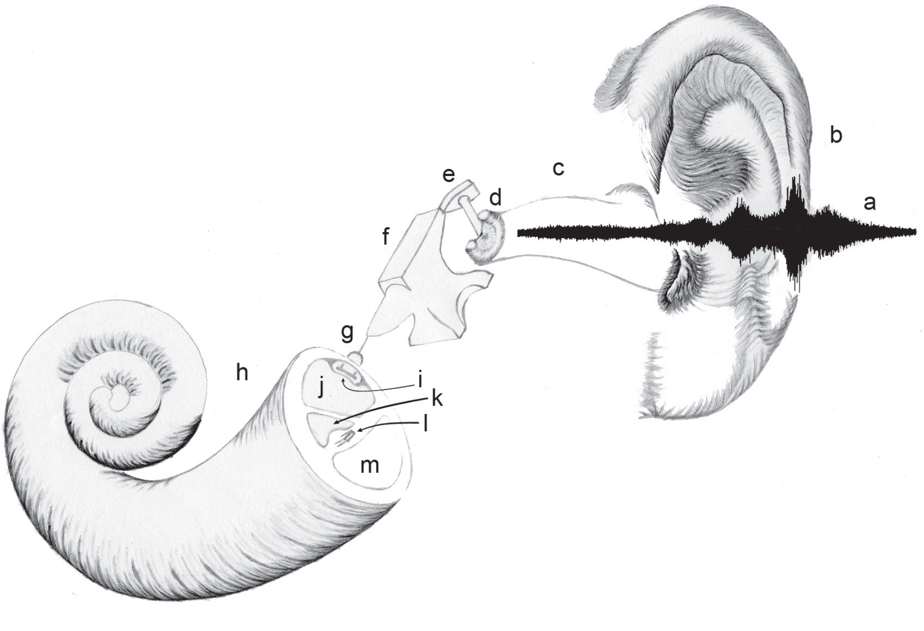


FIGURE 5 Schematic drawing for peripheral auditory system
a) Environmental noise signal, b) Outer ear, c) Ear canal, d) Eardrum, e) Hammer, f) Anvil, g) Stirrup, h) Cochlear, i) Oval window, j) Scala vestibuli, k) Scala media, l) Basilar membrane, m) Scala tympani

Why the cochlea needs to be filled with fluid? With an air filled cochlea the impedance mismatch problem would not exist at all. A fluid filled cochlea, however, provides an advantage over an air filled one. An additional amplification is possible while sound pressure signals are transmitted via the stirrup to the fluid inside the cochlea. Compressibility of water like fluid is much less than with air. Therefore, the stirrup is able to produce higher pressure changes on cochlear fluid when the stirrup is moving with a certain velocity. It has been shown by measurements that a middle ear gain from the ear canal into the cochlear fluid can be at maximum 23 dB [Aib01]. A construction of the middle ear including temporal bones together with the fluid filled cochlear is reducing the acoustic impedance to an average of $21*10^9$ kg/m^{4}s. In order to compare this with the value estimated in the previous page we must multiply it with the stirrup footplate area (3.21 mm^2), because the impedance definitions differ slightly in [Cer01] and [Aib01]. A comparable impedance obtained this way equals 67 410 kg/m^{2}s.

As a result of preprocessing in the outer and middle ear a sufficiently large portion of the acoustic energy entering the ear is focused into the ear canal and can be transmitted further to the inner ear. Furthermore, an extra amplification is achieved on those frequencies which are most important for human interest.

2.3 Signal Processing in Cochlea

In the inner ear acoustic signals are processed inside the cochlea the geometry of which is a tapered tube shaped spiral (see Figure 5). The cochlea is wider in diameter at the base near the oval window and becomes narrower at the apex. The longitudinal length inside the cochlea is 35 mm [Chh14]. Included in the cochlea there are three fluid filled cavities (scala vestibuli, scala media and scala tympani) in which the fluid is like sea water. These tube like cavities stretch through the complete two and a half rounds of the cochlear spiral.

The cochlea functions as a resonator or a frequency analyzer. The fluid inside the scala vestibuli attached to the oval window starts to vibrate with a frequency corresponding to the incoming acoustic signal. High frequencies are analyzed at the basal end and low frequencies at the apical end. Cochlear geometry plays a crucial role in preparing for frequency analysis in the inner ear. Low frequency limit of the inner ear depends on the curvature of the cochlear spiral. One way to classify the curvature is to use a ratio between the radii at the basal and apical ends of the cochlea. This ratio is shown to correlate well with the lower limit of audible frequency range [Man08]. For the human cochlea the ratio related to this curvature is 8.2 which corresponds to the low frequency limit of 31 Hz.

Due to its curvature the cochlea has a property which enables it to increase the energy density of incident pressure waves. By using a ray tracing model the phenomenon can be understood as follows. While the wave motion propagates from the base of the cochlea towards the apical end the curvature of the spiral shape interacts with it. The rays associated to the wave front are focused more and more on the outer side of the cavity in which they travel. As a consequence of this increase in energy density the wavelength of vibrations will decrease while the wave propagates towards the apical end of the cochlea [Man06]. In this way the cochlear curvature helps to decrease the wavelength associated with low frequency vibrations which are analyzed at the end of the cochlear spiral. An incident 200 Hz sound frequency was observed by von Békésy to be mapped on a wavelength of 4 mm in the human cochlea [Bek60]. In air the corresponding wavelength would be about 1.7 m. Therefore, the compression achieved inside the cochlear by its curvature must by a factor in excess of 400 for the incident 200 Hz sound wave.

2.4 Resonances on Basilar Membrane

Inside the cochlea there is a basilar membrane between two of the fluid filled cochlear chambers (i.e., scala media and scala tympani). Mechanical and elastic properties of the 32 mm long basilar membrane vary along its length. It is narrow (0.04 mm) at the base and becomes wider (0.5 mm) at the apex of the cochlea [Man08]. The stiffness of it also increases gradually towards the apical end [Nob03]. Therefore, each part of the basilar membrane responds differently on different frequency components of vibrations that excite it. The mechanoelastic properties of the transverse slices of the basilar membrane set limits on frequencies with which different parts of it can resonate. In a simplified model the basilar membrane can be regarded as a collection of vibrating strings like a stringed instrument (e.g., a harp). Depending on the frequency of the incoming sound the maximum amplitude is reached at a specific position along the length of the basilar membrane. Different parts of the basilar membrane can be thought to possess different characteristic frequencies. For young healthy people the audible frequency range is known to be from 20 Hz to 20 kHz. Each frequency within this range match up with a characteristic frequency of certain basilar membrane position.

What happens when a sound pressure signal reaches the cochlea? While the vibrations from the hearing bones enter the cochlea through the oval window cochlear fluids start to vibrate. These oscillations of fluid molecules propagate from the basal end towards the apex of the cochlea inside the scala vestibuli. Along the way the vibrations affect two other cavities and the basilar membrane between them. Depending on characteristic frequencies of basilar membrane locations they absorb energy differently from the propagating vibrations. In case the characteristic frequency differs considerably from the frequency of fluid vibrations the basilar membrane at that point is reluctant to absorb energy. Therefore, only a small fraction of vibrational energy can be absorbed by that specific position on the basilar membrane. As a result, displacements of the basilar membrane on this point possess just a small amplitude. If, however, the characteristic frequency coincides with the frequency of propagating vibrations there is a resonance and the vibrations leave their energy efficiently at that place. The maximum amplitude of displacements results on this basilar membrane location, because majority of energy is transferred there from the cochlear fluid to the basilar membrane. Due to this efficient energy transfer the propagating vibrations loose most of their energy and will cease out.

From experimental data a mapping between frequencies and basilar membrane positions can be established. Most commonly a mapping invented by Greenwood is utilized [Gre96]. According to the Greenwood approximation the frequencies and positions are related as follows.

$$f(x) = A \cdot \left(10^{\alpha \cdot x} - k\right) \tag{2.1}$$

Parameter values $\alpha = 2.1$ and $k = 1$ are suitable to approximate the response of the human basilar membrane. The position x is a proportional distance from the apex normalized with the basilar membrane length. Therefore, the parameter x equals zero at the apical end and one at the basal end. The purpose of parameter A is to enable fitting of this frequency to position mapping with experimental data. While generating a graph for mapping of frequencies on the human basilar membrane the fitting parameter was set to a value $A=185$. This choice resulted in a fit where 20 kHz and 20 Hz frequencies map to a position 1 mm from the base and 1 mm from the apex, respectively. In order to provide a more intuitive interpretation for the basilar membrane position a transformation to express distances in millimeters from the basal end was applied. Hence, in Figure 6 the value 0 mm represents the position of the oval window where the vibrations enter the cochlea.

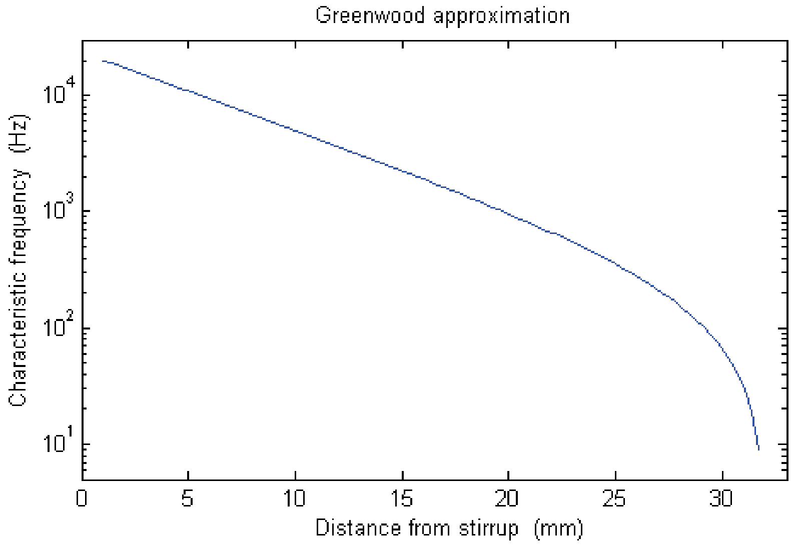


FIGURE 6. Mapping of basilar membrane positions to frequencies

The place of resonance for high frequencies is located near the oval window and the stirrup. While the frequency decreases the place of the maximum excitation moves towards the apical end of the cochlea. For example a 2 kHz frequency generates a maximum displacement roughly at the middle of the basilar membrane. Response of the basilar membrane is rather linear in logarithmic scale up to a distance of 20 mm from the oval window. After that point the cochlear curvature has an even stronger effect on compression of the frequency scale.

2.5 Disturbing the Sounds of Silence

In silence no pressure fluctuations can be detected and therefore the air around the ear is homogeneous. The concentration of air molecules is constant depending on the atmospheric pressure. These circumstances generate an equilibrium for the human ear where a constant pressure of one atmosphere (101.3 kPa) in average is surrounding the eardrum in the middle ear. The eardrum remains in a steady state and no vibrations are mediated by the temporal bones from the middle ear to the inner ear. Inside the cochlea the basilar membrane rests in peace and hair cells sense no basilar membrane induced activity. Synapses of auditory nerves do not generate impulses (besides of spontaneous firing) and auditory nerves receive no essential impulses to carry forward. Only sounds of silence reach our brain.

How the situation changes when someone suddenly starts for example to sing a third octave C note[10]? The frequency of the c''' note is close to 1 kHz. Hence, a pure tone with a 1 kHz frequency begins to interact with the molecules of air. As a result the equilibrium state is broken and the concentration of air molecules remains no longer constant in time and space. Instead, more and less condensed locations of air molecules build up. Compared to the atmospheric pressure these fluctuations are extremely small (20 μPa – 20 Pa). These local deviations from the equilibrium pressure start to propagate further away from the singer. After reaching a listener's ear and traveling through the ear canal they hit the eardrum which is forced to move from its equilibrium position. While the eardrum is pulsating along with these pressure fluctuations a hammer attached to it starts to tinkle on the anvil. Finally, the stirrup connects the mechanical vibrations from the middle ear to the inner ear.

What happens when the third octave C enters the cochlea? Due to differences in mechanical and elastic properties of the basilar membrane some parts of it are more willing to absorb energy from the incoming vibration than others. The 1 kHz frequency finds its counterpart from a position which is roughly 20 mm away from the base of the basilar membrane. That specific place is most sensitive for this kind of vibrations, because it has a characteristic frequency of 1 kHz. Hence, it resonates with the singer's c''' note and absorbs energy most efficiently from incoming vibrations. Thus, on the basilar membrane there exists one location with a maximum amplitude of vibrations for this incoming frequency.

This phenomenon is in analogy with resonating strings in a guitar. Suppose we have a guitar the tuning of which is correct. By placing a finger behind the fifth fret of the fourth string and striking it we can give energy to the string and bring it in to a vibrating state. The characteristic frequency of the fourth string pressed this way equals that of an open third string (the g' note). Therefore, the third string – which originally was at rest – starts to vibrate, because it resonates with the fourth string and absorbs energy from it. Other strings, however, stay at rest, because their characteristic frequencies do not match.

2.6 From Mechanical Vibrations into Electric Impulses

The response of the basilar membrane is still of mechanical nature. Pressure fluctuations entering the outer ear are mediated as vibrations by the middle ear into the inner ear. Vibrations of the stirrup footplate introduce mechanical displacements on the basilar membrane via a hydrodynamic coupling provided by fluids in the scala vestibuli and scala tympani. Between the basilar membrane and the scala media resides an organ of Corti which processes the mechanical vibrations further. A transduction process from mechanical domain into electric domain is accomplished by hair cells within the organ of Corti. Two types of hair cells exist (i.e. inner and outer) of which the inner hair cells are responsible for this transduction process [Hud97]. The purpose of these inner hair cells is to transform information originally coded into the pressure fluctuations

[10] Marked as c''' in Helmholtz pitch notation, http://www.dolmetsch.com/musictheory1.htm#helmholtz

from mechanical vibrations to electric impulses. We could think of the inner hair cells as motion sensors which monitor fluid movements in the scala media. Each hair cell possesses a bundle of hairs which are bent while the fluid in the scala media flows due to the organ of Corti vibrations generated by the basilar membrane. Movement of the hairs in one direction opens up Ca^{2+} and K^+ ion channels in the inner hair cell [Mam07], [Rap03]. This leads to build up of potential differences between the top and bottom parts of the inner hair cell. These membrane potentials initiate an impulse on synapses connected to this hair cell. Majority of synaptic connections from inner hair cells are to afferent nerve fibers [Moo97] which propagate the impulses further to auditory brain stem. The above mentioned transduction channels are closed when the hairs move in the opposite direction. Vibrations of the basilar membrane generate periodic back and forth movements of hair bundles attached to hair cells on a frequency dependent position of the organ of Corti. The transduction channels of these hair cells are repeatedly opened and closed. As a consequence, an impulse train is fed through the synapses to afferent nerve fibers.

We may consider the opera singer example again. While the c''' note is heard by a listener the inner hair cells associated with a basilar membrane position 20 mm away from the basal end would receive additional K^+ ions through their transduction channels. As a result, a pulse train of impulses is generated on those afferent synapses connected to these inner hair cells. Finally, the impulses are transmitted from the inner ear via auditory nerve fibers into the brain where the perception takes place. Interpretation might be: 'Karita Mattila is preparing for the Metropolitan.'

What happens if the third octave C is sang simultaneously when the G string of a guitar is played? Two different locations on the basilar membrane are absorbing energy from the combined vibration of the cochlear fluids. The two notes in this case are far enough from each other in the frequency scale. The difference of 650 Hz between the c''' and g' notes guarantees that the frequency selectivity of the basilar membrane is adequate to separate the frequencies from each other. Hence, inner hair cells attached to two distinct locations on the organ of Corti generate impulses for afferent synapses and finally the sensation is two separate tones. If these two tones, however, would be too close together the frequency resolution of the peripheral auditory system would be inadequate. Instead of two separate peaks in the displacement pattern on the basilar membrane a double peak merging the effects of the two tones would be sensed. The groups of inner hair cells involved might be so close to each other that impulses generated end up into the same bundle of auditory nerve fibers. As a result the brain might have an impossible task to interpret the impulses as coming from separate sources. Therefore, the final perception would be that only one tone was heard.

2.7 Sensitivity as a Function of Frequency

Several parameters affect the transfer of vibrational energy from hearing bones to the basilar membrane. Included in these parameters are for example hydrodynamic properties of cochlear fluids (density, viscosity, etc.). Also mechanical and elastic properties of the basilar membrane (e.g., width, thickness, mass, stiffness, positional viscosity, and shearing viscosity) influence the energy transfer. All of the aforementioned parameters together with the properties of the stirrup to the oval window connection as well as the cochlear geometry determine how effectively vibrational energy is transferred to basilar membrane for frequency analysis. A net effect of these factors can be combined into a coupling constant. The coupling constant reveals us how strong coupling exits from the middle ear to the basilar membrane for vibrations with different frequencies. A hydrodynamic model of the human inner ear postulates a shape for the coupling constant as a function of fractional distance from the oval window [Nob03]. By combining it with the Greenwood approximation the shape of the coupling constant can be depicted as a function of frequency in Figure 7.

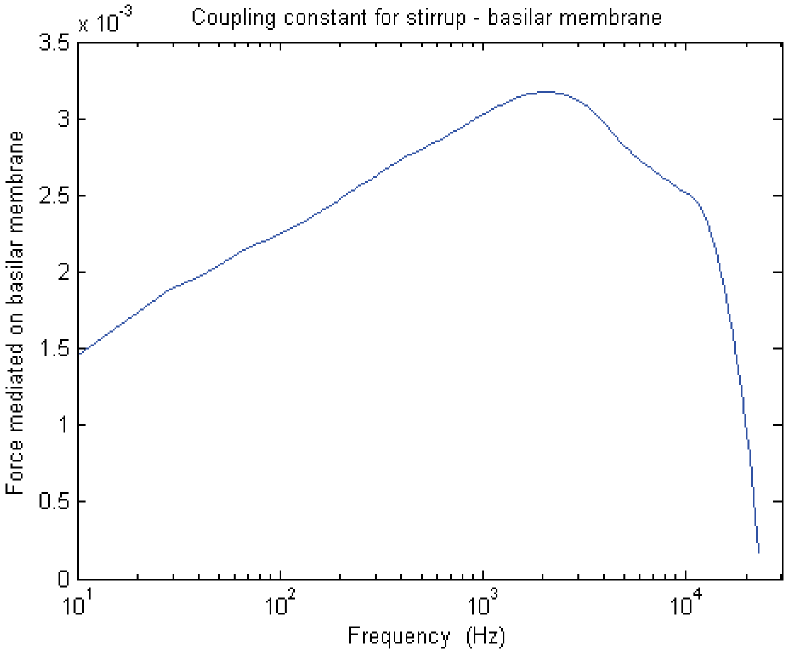


FIGURE 7 Transfer function for vibrations from stirrup to basilar membrane

As can be seen middle frequencies are transferred most effectively from the stirrup via the cochlear fluid to the basilar membrane. Therefore, auditory sensitivity for middle frequencies is better than for low or high frequencies. Consequently, a nonlinear frequency response is introduced for the perception of sounds in the human auditory system.

The fluid coupling constant for the stirrup to basilar membrane connection is related to threshold of hearing. The coupling constant describes primarily how vibrations propagate physically from the ear canal to the basilar membrane. Another method, however, can be applied in search for an answer to the question of how human perception of frequencies happens. Audiological experiments to measure the threshold of hearing have been conducted already during half a century. Several methods have been developed to fit threshold of hearing curve on experimental data. A dashed line in the figure below shows how the threshold of hearing changes as a function of frequency on average in human ear.

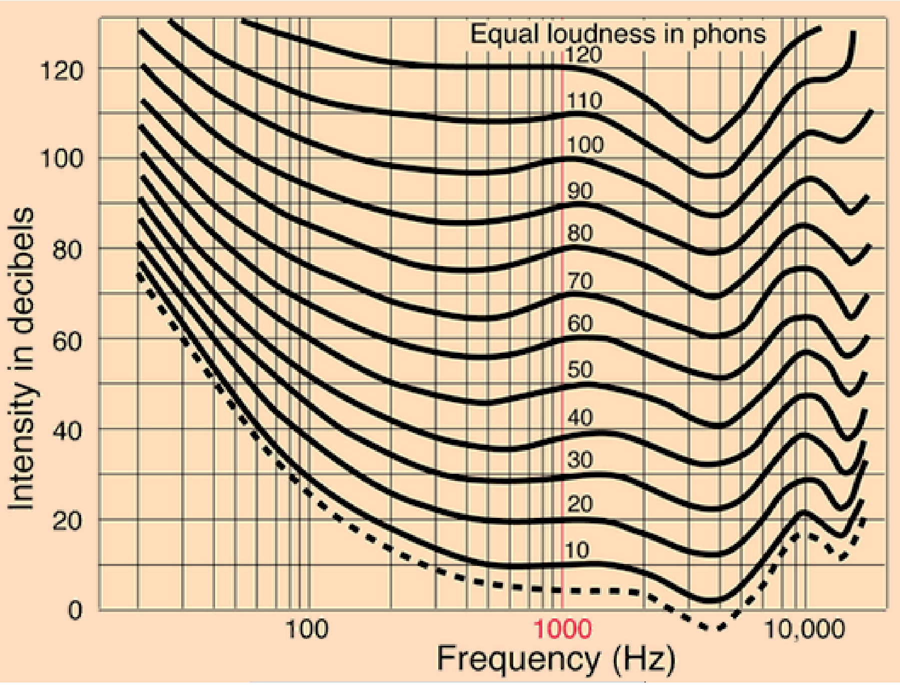


FIGURE 8 Threshold of hearing and equal loudness contours[11]
(© 2013 C.R. Nave, figure reprinted by kind permission of Dr. Rod Nave)

The cochlea is not just a passive detector for incident sound signals. For example outer hair cells can modify the amplification in the transduction process of acoustic signals. The cochlear amplification is not constant in all circumstances, but is able to actively adapt to changes in the intensity of sound. Feedback from auditory cortex can adjust the hair cells to select a higher or lower gain for transduction of vibrational information to electric impulses. This adjustment of the amplification gain is achieved physiologically by controlling the tension of tip links with which shorter hairs are connected to neighboring taller hairs [Hud97]. Variations in this amplification manifest themselves in equal-loudness contours which depend on ambient noise levels as shown in Figure 8. These equal-loudness contours are the origin of frequency weightings (A-, B-, and C-weighting), which are used commonly in environmental noise measurements. A more detailed description for the shape of these frequency weighting curves will be given in Chapter 5.

[11] http://hyperphysics.phy-astr.gsu.edu/hbase/sound/eqloud.html#c1

3 ENVIRONMENTAL NOISE MEASUREMENTS

In this chapter an overview is given for measurements of environmental noise. We start by explaining what is the physical observable to be measured and what kind of processes are generating environmental noise signals. The chapter continues by presenting measurement equipment required to transform environmental noise signals from physical domain into electric signals. We take up for example the following questions. How microphones are characterized, how they interact with a sound pressure field and how they transduce physical sound pressure fluctuations to electric signals. Which are the most important properties of microphones that affect this transduction process? What kind of hardware and software components are required from integrating sound level meters for assessment of environmental noise. Which are the corrections that need to be applied on the measured signals. After the aforementioned introduction a section for measurement setup follows which explains how experimental data was acquired for road traffic noise. Finally, some examples are given to show how measured sound pressure signals look like in the time domain, what kind of histograms are obtained for basic environmental noise indicators and how energy in road traffic noise is distributed in frequency domain.

3.1 Fundamental Measures of Noise

The physical quantity of interest in environmental noise measurements is pressure. The origin of environmental noise is in extremely small fluctuations of pressure around an equilibrium. The atmospheric pressure acts as a reference value against which small positive or negative disturbances are superimposed by environmental noise sources. The sound pressure is defined as “a dynamic variation in atmospheric pressure. The pressure at a point in space minus the static pressure at that point” [BK12]. The sound pressure is expressed in pascal which is “a unit of pressure corresponding to a force of one newton acting uniformly upon an area of one square meter. Hence, 1 Pa = 1 N/m^2 ”. In physical sense no distinction exits between the concepts of sound and environmental noise. The latter is regarded as unwanted, unpleasant or disturbing sound which may be harmful for humans' health or well-being [Hei09]. From the physical point of view, however, both of them stand for exactly the same phenomenon. One could say that sound becomes environmental noise when it has propagated through human ear, has been filtered there and finally has caused a sensation which is not a pleasant one.

What kind of processes act as sources for environmental noise? Without going deeper into signal theory we can state that most often we finally deal with non-stationary random signals while measuring environmental noise. For example, engine noise from a single vehicle on steady driving conditions will possess periodic components which would imply a deterministic process. Quite seldom, however, this steady state can be kept continuously. There may be abrupt changes for example due to a need of revving while shifting to a lower gear prior to an uphill. Hence, the environmental noise signal generated can not be predicted and it turns to a random one. Furthermore, propagation of environmental noise from noise sources to a point of measurement is affected by atmospheric conditions and diverse obstacles which add more random features to the signal. Sound pressure signals from road traffic are a superposition of emissions from several vehicles with different and constantly alternating driving conditions. Therefore, the statistical structure of road traffic noise is not kept unchanged and we are measuring non-stationary signals.

Instead of concentrating on instantaneous values of the sound pressure signal different types of time averaging are applied in environmental noise measurements. Human ear is not capable of resolving all of the fastest fluctuations of sound pressure signals. Therefore, different time averages have been developed to mimic temporal processing in human hearing. The fluctuations of the original sound pressure signal p(t) get both positive and negative values in reference to the atmospheric pressure (Figure 9). An effective sound pressure for these pressure fluctuations can be calculated as a root-mean-square average. For a time varying quantity the RMS average is defined as "a value which delivers the same amount of power on average as a constant quantity" [Ber12]. In case of sound pressure as a measured physical quantity this RMS value is obtained as a square root of an average signal power which is calculated by averaging over a time period T an integral of squared instantaneous sound pressure values p(t) as defined below

$$p_{RMS} = \sqrt{\frac{1}{T}\int_{T} p^{2}(t)\,dt} \tag{3.1}$$

Evaluation of the effective sound pressure includes an integration, which in traditional sound level meters has been performed by an analog RC integrator. In this so called time weighting a capacitor is first allowed to integrate the squared pressure signal for a period of time T, which equals the time constant RC of this analog integrator. Afterwards the voltage across the capacitor is detected to obtain a value for the integrated charge, which is proportional to the power of the input signal. Time constants typically used in these analog time weighting networks are denoted by F, S, I and stand for Fast, Slow and Impulse, respectively. Durations for the time constants F, S and I are given in the table below [Lah95]. Besides analog RMS detectors, which apply exponential time averaging on their inputs, another alternative is to evaluate effective sound pressure digitally. In this case linear averaging over an integration period twice as long compared to exponential averaging is utilized. For measurements of random noise both of these averaging methods yield equal statistical uncertainties [Jac11]. Therefore, both of these alternatives are applicable, because environmental noise entering the microphone is a random signal as stated above.

TABLE 2 Time weighting constants and equivalent integration times

	Fast	Slow	Impulse
Analog	0.125 s	1 s	0.035 s
Digital	0.250 s	2 s	0.070 s

During the time weighting process the sound pressure signal becomes also rectified because the instantaneous sound pressure is squared. Hence, the values of the effective sound pressure are always positive (Figure 9).

The incredible capability of the human auditory system to sense both very weak and very intense sounds calls for a wide dynamic range for the measurement system as well. Weakest sounds at the threshold of hearing generate only a 20 µPa sound pressure. A 20 Pa sound pressure, on the other hand, is close to the threshold of pain. The ratio between these two thresholds is 1:1000000 in sound pressure values, which must be detectable by the measurement equipment. This kind of linear range, however, is difficult to handle while presenting measurement results. Instead of a linear scale a more convenient way to deal with them is a logarithmic scale. Therefore, a concept of sound pressure level has been introduced. The sound pressure level L_p is a ratio of the measured sound power to the reference power expressed in decibel as defined in the equation below

$$L_p = 10 \cdot \log_{10}(\frac{p^2}{p_0^2}) \qquad (3.2)$$

where p is the RMS value (unless otherwise stated) of the sound pressure in pascal, and the reference value p_0 is 20 µPa for measurements in air [BK12]. In Figure 10 an example time history is shown for sound pressure levels which are calculated by using effective sound pressure values depicted in Figure 9.

Another type of time integration is needed while generating an indicator called equivalent continuous sound pressure level. In this case integration lasts a longer time compared to the RMS average in time weighting. L_{eq} values can be calculated for example to account for a second, a minute or an hour long periods. L_{eq} is defined as "an imaginary constant sound pressure which would possess equal amount of energy as the actual measured sound pressure signal" [RIO02].

$$L_{eq} = 10 \cdot \log_{10} \frac{1}{T} \int_T (\frac{p^2(t)}{p_0^2}) dt \qquad (3.3)$$

In principle the integration time T for calculating the equivalent continuous sound pressure level can be selected freely. For example the L_{eq} values depicted in Figure 10 have been calculated for 5 second long periods. As mentioned above typical integration times in environmental noise measurements last between one second and one hour. While referring to integrating sound level meters it is this integration involved in generation of L_{eq} values that is meant. Integrating SLMs are capable of computing some short-term values already online during measurements (cf. L_{Leq} in Figure 16). Later in the analysis phase measurement data can be processed again and long-term averages for L_{eq} can be generated (e.g., for 8 h, 24 h, 1 week periods). Consequently, fundamental noise measures presented in this section are the base from which deduced environmental noise indicators can be calculated.

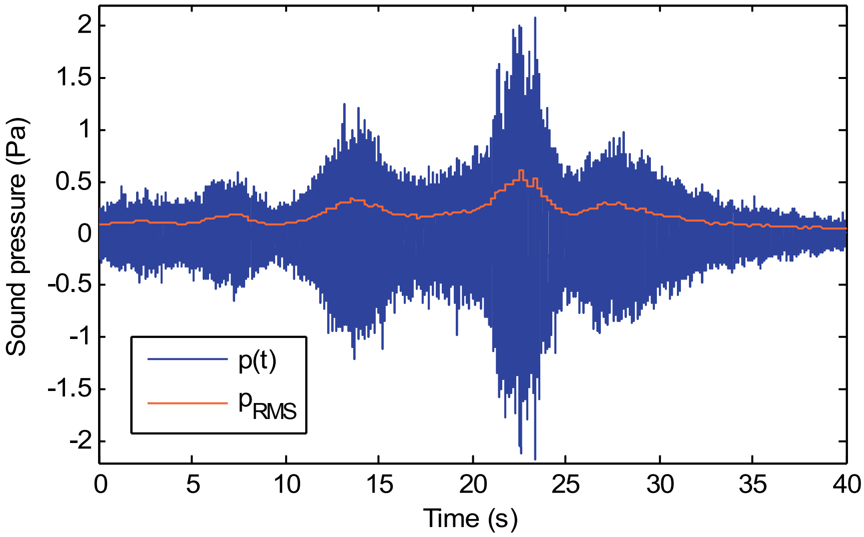


FIGURE 9 Original sound pressure signal p(t) and effective sound pressure p_{RMS}

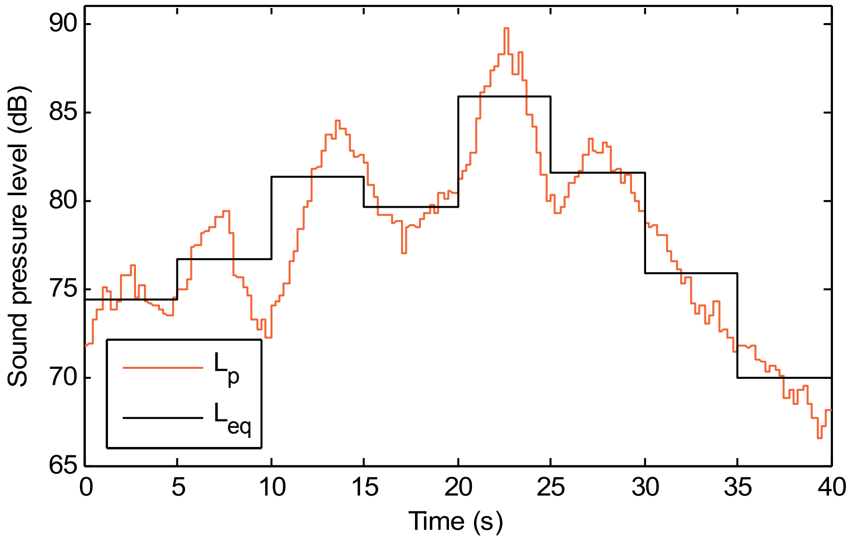


FIGURE 10 Sound pressure level (L_p) and equivalent continuous SPL (L_{eq})

3.2 Measurement Microphones

In Chapter 2 we followed sound pressure signal's propagation through the human ear until transduction to an electric signal occurred. A corresponding analysis for a microphone is presented in this section.

The interaction between a sound pressure field and a microphone starts when fluctuations of air molecules reach a microphone diaphragm, which is a thin metal or polymer foil stretched over the opening on top of a microphone cartridge. In the same way as the eardrum was moved from its equilibrium position the microphone diaphragm is deflected by the sound pressure. A higher concentration of air molecules deflects the diaphragm membrane inwards and a lower concentration allows a static pressure inside the microphone cartridge to deflect it outwards. A principal structure of a prepolarized condenser microphone is depicted in Figure 11.

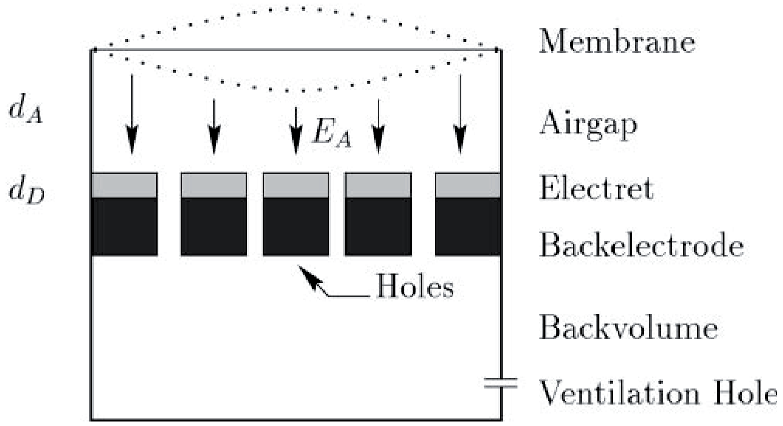


FIGURE 11 Schematic drawing of microphone structure [Amj00]

Several categories of microphones exits. Besides of traditional condenser microphones new transducer types relying on piezoelectric effect, micro structures or nanotechnology have been developed. Classification of microphone types can be based on different criteria some of which are summarized in Table 3.

TABLE 3 Microphone classification criteria

Criterion	Typical parameter alternatives
Physical size	1”, 1/2”, 1/4”,1/8”, MEMS
Diaphragm material	Steel, aluminum, parylene, polysilicon
Generation of E	External voltage, prepolarized
Application area	Laboratory tests, field measurements, consumer electronics

Microphone manufacturing process, diaphragm material choice, microphone size as well as the way to generate an electrostatic field between the diaphragm membrane and a rigid backplate electrode influence the characteristics of a microphone. Sensitivity, frequency range and noise are the most important characteristics of the microphone behavior. The physical parameters listed in Table 3 affect these characteristics. For example 1″ microphones have better sensitivity than 1/4″ microphones, but their frequency range is more limited compared to smaller diameter microphones.

How a microphone interacts with a sound pressure field? The first parameter influencing this interaction is mechanical sensitivity of the diaphragm membrane. The mechanical sensitivity is a measure of the diaphragm membrane deflection w as a function of the sound pressure p [Amj00]. By definition the mechanical sensitivity of the microphone S_M is expressed as

$$S_M = \frac{dw}{dp} \tag{3.3}$$

The extent of this deflection is affected by mechanical and geometric characteristics of the diaphragm membrane. Among others the membrane material, shape, size, density, thickness and internal stress play a role for the mechanical sensitivity. While designing microphones the selection of proper values for these characteristics depends on the intended application (e.g., accurate laboratory tests, practical field measurements, high volume consumer applications). During the microphone design phase also the intended measurement environment (free-field, pressure field, diffuse field) must be taken into account. For measurement microphones, which are our main interest in this work, thin metal foil diaphragms of circular shape are traditionally utilized. The membrane deflection w on these microphones is a parabolic function of the radial distance r from the membrane center [Sch03].

$$w(r) = w_0 \cdot \left(1 - \frac{r^2}{R^2}\right) \tag{3.4}$$

where R is the radius of the diaphragm membrane,
w_0 is the deflection at the diaphragm center,
r is in the range $-R \leqslant r \leqslant R$

The maximum deflection w_0 is proportional to the sound pressure. Higher absolute values of p deflect the diaphragm membrane more. Hence the deflection of a specific point of the membrane becomes a function of both the radial distance and the sound pressure.

While sound pressure fluctuations interact with the microphone the diaphragm membrane vibrates following the small positive and negative changes relative to the ambient atmospheric pressure. Vibrations of the diaphragm membrane are the only mechanical response of the microphone. No equivalent for hearing bones to mediate these vibrations, neither hydrodynamic coupling nor basilar membrane like resonator exist. These diaphragm membrane deflections are directly coupled from the mechanical domain to the electric domain via an electrostatic field. This coupling is dictated by another component of the microphone sensitivity, namely electrical sensitivity, which specifies how big change is generated on the microphone output voltage V by a certain change in deflection of the diaphragm membrane.

$$S_E = \frac{dV}{dw} \tag{3.5}$$

The electrical sensitivity depends mainly on the electrostatic field of the microphone. Two methods are available for generation of this electrostatic field. In a direct method an external polarization voltage is applied across the air gap between the membrane and the backplate. Instead of an external polarization voltage a dielectric coating can be grown on the backplate electrode to generate an electrostatic field for a measurement microphone. The strength of the electrostatic field depends on the height of the air gap and the charge storage capacity of the dielectric material. Already for 20 years one of the most commonly used dielectric materials in measurement microphones have been fluorocarbon electrets [Mur89]. The popularity of these electrets is based on their long-term capability to hold trapped charges. An electret layer provides a possibility to manufacture prepolarized microphones, which do not require an external bias voltage. This is a great benefit in environmental noise measurements because a typical external polarization voltage of 200 V would be hard to generate in the field.

Open-circuit sensitivity of a microphone is obtained by combining the above mentioned two components of sensitivity together. It informs how a microphone transduces sound pressure fluctuations into electrical signals.

$$S = S_M \cdot S_E = \frac{dw}{dp} \cdot \frac{dV}{dw} = \frac{dV}{dp} \tag{3.6}$$

The open-circuit sensitivity is determined by the microphone manufacturer. By attaching an electrostatic actuator on top of a microphone the response of the microphone diaphragm can be obtained. This is a so called actuator response, which is not altogether accurate in regard to field measurements. When the actuator is removed and the microphone is released to a free-field environment the measurement geometry is changed. Without a tight actuator cap around the microphone the interaction between the diaphragm and the sound pressure field is different. After a certain frequency, which is dependent on the microphone dimensions, the wavelength of the incident sound wave becomes comparable with the microphone size. Therefore, the microphone starts to interact with the sound pressure field and the frequency response of the microphone is affected. Hence, a correction must be made to the actuator response before a free-field response can be obtained. For example a free-field correction on the order of 1 – 10 dB may need to be added to the actuator response at high frequencies (5 – 20 kHz, respectively) to get a free-field response for the microphone [Gra01].

The microphone sensitivity is expressed commonly in two alternative units – that is, either in mV/Pa or dB. The first of these alternatives defines how the output voltage of a microphone changes as a function of the sound pressure deflecting the diaphragm. The second possibility to specify sensitivity values is in decibel which is calculated using 1 V/Pa as a reference value. Therefore, a common notation for this sensitivity specification is "dB re 1 V/Pa". The relation between these two sensitivity measures is as shown below [BK94]

$$S^{dB} = 20 \cdot \log_{10} \frac{S^{mV}}{1000\, mV/Pa} \tag{3.7}$$

As an example we can mention a Brüel & Kjaer 1/2″ prepolarized free-field condenser microphone 4189. In a calibration certificate of this microphone a sensitivity value of 51.3 mV/Pa is given. By using Equation (3.7) the sensitivity specification transforms to -25.8 dB re 1 V/Pa. However, we should remember that the obtained figure is the open-circuit sensitivity, which means it specifies how the microphone responds in the absence of the rest of the signal conditioning chain (e.g., a preamplifier or an amplifier). The open-circuit sensitivity is a function of frequency. For example the sensitivity values given above for the B&K 4189 are valid accurately at 250 Hz frequency. Within a certain frequency range, however, good quality measurement microphones detect and transduce all frequencies with equal sensitivity.

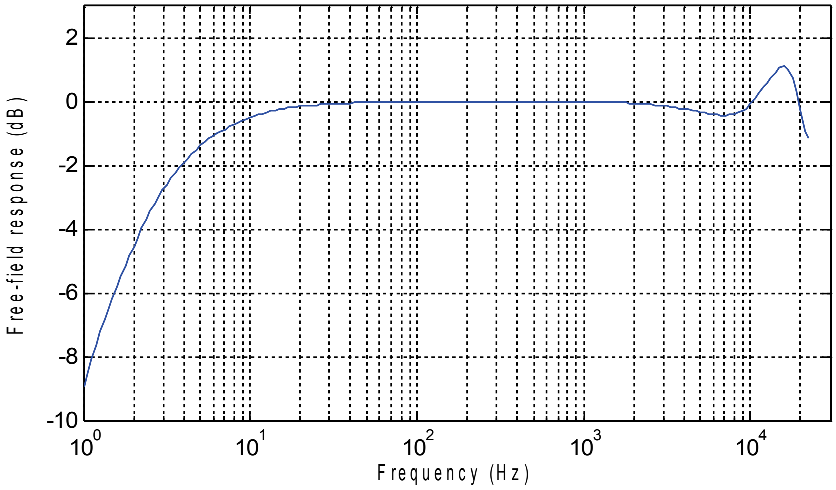


FIGURE 12 Frequency response for B&K 4189 microphone

In the above figure a free-field response is given for the B&K 4189 microphone when the sound incidence is from an angle of zero degrees. What can be observed from the response curve is that this microphone transduces linearly a frequency range which extends one decade below and above of 250 Hz (i.e., 25 – 2500 Hz). For frequencies 3 – 10 kHz only a slight deviation from linear response is detected. The frequency response stays within ±1 dB limits from about 6 Hz up to 20 kHz. On the highest operating frequency side the range is limited due to the resonance frequency of the diaphragm. For a 1/2″ microphone the diaphragm resonance occurs typically at 20 kHz [Suh10].

3.3 Integrating Sound Level Meters

Most frequently used device in environmental noise measurements is an integrating sound level meter. Typically integrating SLMs are comprised of the following components for conditioning of sound pressure signals:

- a microphone
- a preamplifier
- an amplifier
- correction filters
- frequency weighting filters
- peak / RMS detectors
- time weighting filters

A microphone detects fluctuations of the sound pressure and converts them from the physical domain into electrical signals. These weak signals are fed through a preamplifier and possibly transferred via an extension cable to an amplifier. The amplifier stage of an integrating SLM may provide an option to adjust the amplification individually for each measurement case. A dynamic range of traditional SLMs is often limited for example to 80 dB. A bias gain of 40 dB would enable a measurement range of 40 – 120 dB for environmental noise assessments. The amplified signal continues to correction filters which are able to correct distortion effects due to sound incidence angle, optional windscreen and the microphone itself. Like in all types of measurements also in this case the measurement affects the physical phenomenon. In free-field conditions only the microphone disturbs the sound pressure field slightly due to diffraction and interference between the diaphragm and the pressure field. If a manufacturer has equipped its SLM with a proper filter the influence of the microphone on the sound field can be corrected. Alternatives for a sound incidence correction are either frontal or random. The first correction is applicable in circumstances where the environmental noise origins from a well defined source and the incidence angle of incoming sound to the microphone is close to zero degrees. In a diffuse sound field where the sound enters the microphone from several different angles a selection for random sound incidence correction is the proper one. A windscreen commonly used to decrease the influence of low frequency fluctuations induced by wind will affect also to the measured environmental noise signal. Therefore, an opposite correction must be applied if the windscreen is used in a measurement.

Integrating sound level meters are equipped with time weighting networks which perform averaging for measured environmental noise signals. Three different time weighting constants are typically in use. These are Fast (F), Slow (S) and Impulse (I) as listed earlier in Table 2. The most commonly used one of these is the fast time weighting where the rise and decay times of a signal both are defined to be 125 msec. Thus the total averaging time for the fast time weighting becomes 250 msec. Besides of time weightings the measured signals are weighted also in regard to frequency. Traditionally SLMs provide A-, C- and linear weightings to adjust amplitudes of sound pressure signals' frequency contents. A more detailed discussion of frequency weightings will be given in Chapter 5.

The measurement microphone used in this work was a prepolarized 1/2″ free-field electret condenser microphone (B&K 4188). This microphone type was chosen for the following reasons. The physical size of 1/2″ provides still a good sensitivity and keeps thermal noise of the microphone on a low level. Free-field microphones are intended to be used for measurements in situations where the sound origins mainly from one direction and there are no reflecting surfaces. Prepolarized electret material makes outdoor measurements easier to conduct, because no external polarization voltage needs to be generated.

As an example of an integrating sound level meter we will take a look at a handheld sound analyzer B&K 2238 Mediator manufactured by Brüel & Kjaer. All the tasks of an integrating SLM mentioned earlier (excluding time weightings) are realized in B&K 2238 hardware by analog electronics. After transducing, amplifying, correcting, and weighting the measured signal it is converted using a 16-bit ADC. Further processing of obtained environmental noise samples is accomplished by analysis software running on SLM's digital unit for control and processing. Depending on installed software packages B&K 2238 is capable of operating in different measurement modes. A basic sound level meter (SLM) software (BZ-7126) enables it to act as a monitoring device for instantaneous values of basic environmental noise indicators. Occupational health indicators are available in this mode as well. An enhanced SLM software (BZ-7125) adds a possibility to run two RMS measurements simultaneously with different frequency weightings. By installing a logging SLM software (BZ-7124) the measured environmental noise indicator values can also be logged into files on B&K 2238's internal memory at specified regular intervals. A need for a more detailed frequency information of the assessed environmental noise situation is fulfilled by a frequency analysis software (BZ-7123) which must be accompanied by an additional hardware filter set. This combination turns B&K 2238 into an octave and 1/3 octave band analyzer [BK98a].

B&K 2238 is suitable for environmental noise assessments where instantaneous sound pressure levels are monitored on-line from sound level meter's display or environmental noise indicators are logged into the local memory of the device for further off-line analysis. The version of B&K 2238 used in this work for measuring road traffic noise had the logging SLM software installed. Measurement parameters selectable by the user of this software package include a logging period which is adjustable in a range from one second to 60 minutes [BK98b]. This kind of logging periods are available in normal logging mode of the BZ-7124 software package. Another alternative (i.e., short logging mode) fixes the logging period to 100 ms. A limitation in the short logging mode is that essentially only one environmental noise indicator can be logged. Quite typical choice in environmental noise inventories for the indicator integration period is one second. The normal logging mode in which this logging period is selectable has also another advantage: up to 12 basic environmental noise indicators are available for logging. Therefore, the normal logging mode with a one second logging period was selected for experimental measurements which are reported in Sections 3.4 and 3.5.

3.4 Measurement Setup at Kirri

Environmental noise measurements for this work were conducted along a four-lane motorway stretching north from the city of Jyväskylä in Central Finland. The measurement location was situated in a distance of about 35 meters from the outer south heading lane of the Kirri motorway and is shown in Figure 13. The outer south heading lane in the figure is the one on which a truck is seen. These experimental measurements were carried out on a cloudy day at the beginning of the wintertime while the weather conditions were as follows: temperature -3.4 °C, atmospheric pressure 100.5 kPa, relative humidity 98% and south wind 2 m/s.

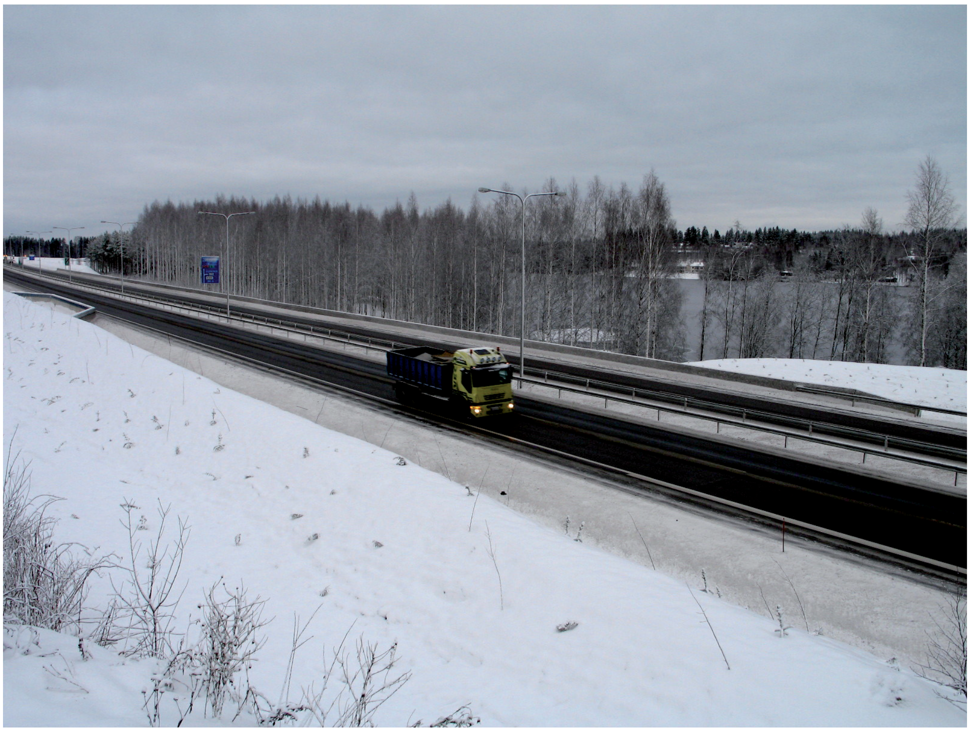


FIGURE 13 Measurement location along the Kirri motorway

Experimental setup of the measurement is depicted in Figure 14. The reasoning behind this setup was to arrange a possibility for two simultaneous measurements. Firstly, the integrating sound level meter (B&K 2238) was used to measure basic environmental noise indicators. For these measurements a noise indicator channel (the upper channel drawn inside the block diagram of B&K 2238) was utilized. It enabled us to select from three different frequency weightings (A, C, or Linear). Likewise, three alternative time weightings (Fast, Slow, or Impulse) existed. After setting up the frequency and time weighting parameters generation of equivalent continuous sound pressure levels, maximum SPL and minimum SPL values for the selected weightings was started. One more basic indicator was obtained from a peak detector channel (the middle channel), where only linear weighting was available before the peak detector. Furthermore, the peak detector channel did not apply any time weightings on the measured signal. All of the above mentioned basic environmental noise indicators were stored on a 2 MB internal memory of the sound level meter.

Secondly, a possibility for sound recording was desired for these measurements. This was provided by an audio channel (the lowest channel in the block diagram). The same analog signal, from which basic environmental noise indicators were generated, was extracted from the amplifier for this purpose. The amplifier stage of B&K 2238 was responsible for adjusting the measurement range according to our requirements. Because the dynamic range for this SLM is 80 dB the amplification gain of it had to be modified in order to shift the measurement window to a proper level. For all Kirri measurements a bias amplification of 30 dB was chosen resulting in an actual measurement range of 30 – 110 dB. The amplified analog signal was routed directly out from B&K 2238 via an AUX1 connector. To record this signal a laptop equipped with B&K sound recording software was connected to the SLM. The measured analog sound pressure signal was first fed to a sound card, which digitized it using an Avance97 sound chip. Sampling frequency for the analog to digital conversion was 44100 Hz. Digitized time histories for road traffic noise were stored in WAV-format to files on the laptop's hard disk.

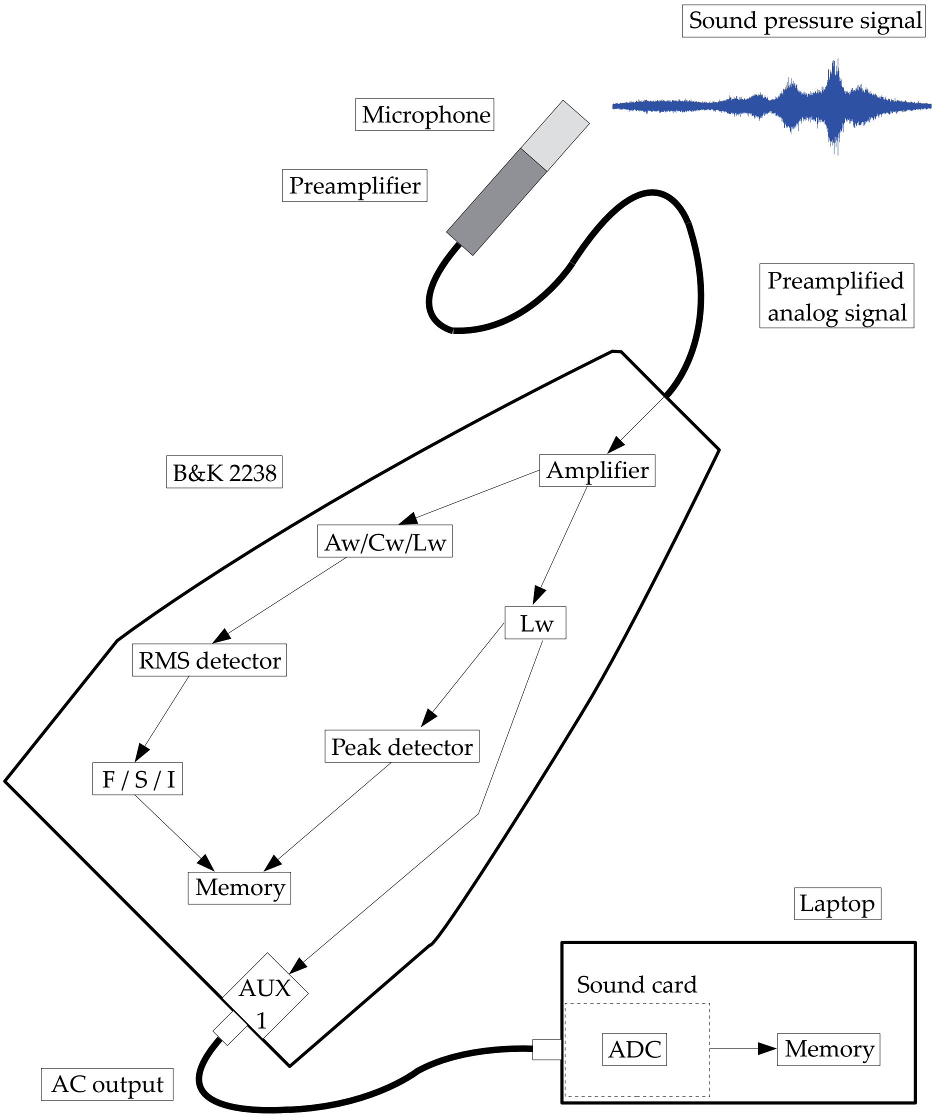


FIGURE 14 Block diagram for measurement setup at Kirri

Before starting environmental noise measurements the noise indicator channel of the sound level meter was calibrated. Prior to the calibration the SLM needs to know the level of a calibration signal to be used. For the Kirri measurements a 93.8 dB signal from a sound level calibrator B&K 4231 was available. The calibration level parameter in the SLM setup was set accordingly. During the calibration phase B&K 2238 monitors the calibration signal emitted by the sound level calibrator and compares the signal with the specified calibration level which was set by the user while initiating the calibration. Based on the comparison B&K 2238 calculates a value for the sensitivity so that a correct output value matching the user supplied calibration level results. For the Kirri measurements the sensitivity value of -31.8 dB was obtained.

3.5 Examples of Measurement Results

As a result of measurements at Kirri we obtained two sets of environmental noise data files. The first data set contains basic environmental noise indicators collected via the noise indicator and peak detector channels. Files in this data set were generated by B&K 2238, which was integrating the amplified sound pressure signal and logging the indicator values for one second long periods. The fast time weighting was applied for all indicators. Three measurements were conducted with frequency weightings A, C and linear and they had a total duration of four minutes each. Hence, all data files in the first set provide 240 values for the basic environmental noise indicators.

Digitized audio signals for all three measurements are included in the second data set. This data set was acquired concurrently with the first one. The amplified analog environmental noise signals were digitized with a 44100 Hz sampling frequency, thus producing 10584000 samples for each four minute measurement. All three files in the second data set contain digitized samples which origin from a linearly weighted signal (cf. Figure 14). In the chapters to come also these measurements are referred to as KirriA, KirriC and KirriL. It should, however, be pointed out that only files in the first data set differ in respect of frequency weighting. The KirriA, KirriC and KirriL measurement files belonging to the first data set are weighted by the SLM with A-, C- and linear weightings, respectively. Audio files in the second data set are all recorded with linear weighting even if they are referred to as KirriA, KirriC or KirriL.

All examples given in this section refer to linearly weighted measurement data, because we have not yet explained details for the frequency weightings. Let us take an audio signal from the KirriL measurement as our first example. In Figure 15 a sound pressure signal detected by the microphone can be seen. The first minute of the KirriL measurement until time 12:57 is related to the calibration of the audio channel. Details of this calibration procedure will be explained in Chapter 4. The actual environmental noise measurement thereafter shows how the sound pressure fluctuates due to road traffic noise emitted by cars and trucks passing by. The second example describes how the measured signal is processed in the noise indicator channel. By taking a look at Figure 14 again we realize that an analog signal equivalent to the time history shown in Figure 15 was fed to the noise indicator channel. After propagating through a linear frequency weighting filter the sound pressure signal entered an RMS detector, where it became rectified and averaged over the fast time weighting period. In Figure 16 the equivalent continuous sound pressure level, maximum and minimum SPL with fast time weighting as well as maximum peak values for the measured signal are shown.

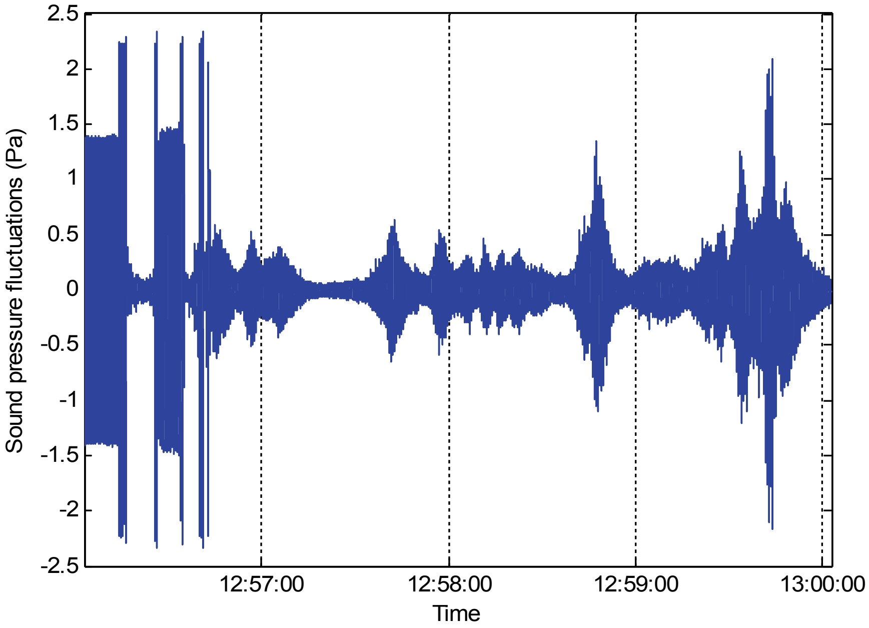


FIGURE 15 Amplified sound pressure signal from KirriL measurement

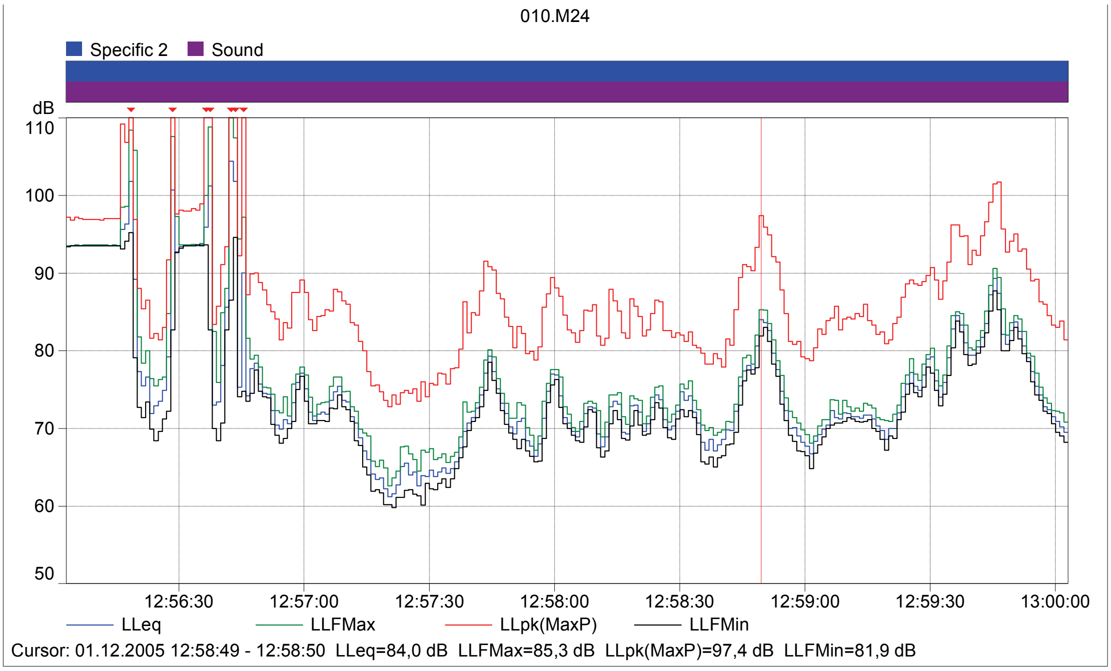


FIGURE 16 Basic environmental noise indicators for KirriL measurement

How basic environmental noise indicators shown in Figure 16 were generated by the SLM from the original sound pressure signal? For the LLeq histogram the sound pressure signal was integrated for one second long periods according to Equation 3.3. For the LLF indicators the averaging time was 125 ms. For each second of the LLFMax and LLFMin histograms a maximum and minimum value, respectively, was picked up by B&K 2238 from the LLF values produced by the RMS detector. The fourth histogram LLpk(MaxP) on Figure 16 was obtained from a peak detector channel (i.e., the middle channel in the block diagram of Figure 14). The peak detector of the SLM integrates the incoming signal only for 100 μs. Thus, it will detect also fast peaks, which are averaged out in generation of the three other basic environmental noise indicators.

Third example of results is related to percentile levels which are plotted in Figure 17 for the KirriL measurement. This figure reveals two different aspects of statistical distribution of environmental noise during the measurement. First, a level histogram shows percentages of the total time a certain equivalent continuous sound pressure level has been present. While generating this percentile level histogram the SLM divides the LLeq range in 0.5 dB wide bins. Thereafter, every time a one second LLeq value is calculated a count in the bin corresponding to this new value is incremented. As a result a distribution for LLeq levels is obtained. For example in Figure 17 the level 70 <= LLeq < 70.5 dB is encountered 4.2% of the measurement time. Second, a more useful and descriptive statistical summary is represented by a cumulative graph. From it we can easily observe that for 100 % of the total time the environmental noise level has been at least 60 dB. A cursor set at the 70 dB point in the cumulative graph shows directly that three quarters of the total measurement time the LLeq level has exceeded 70 dB. On the upper right corner of this statistical distribution plot seven commonly used percentile levels are listed in numeric form as well.

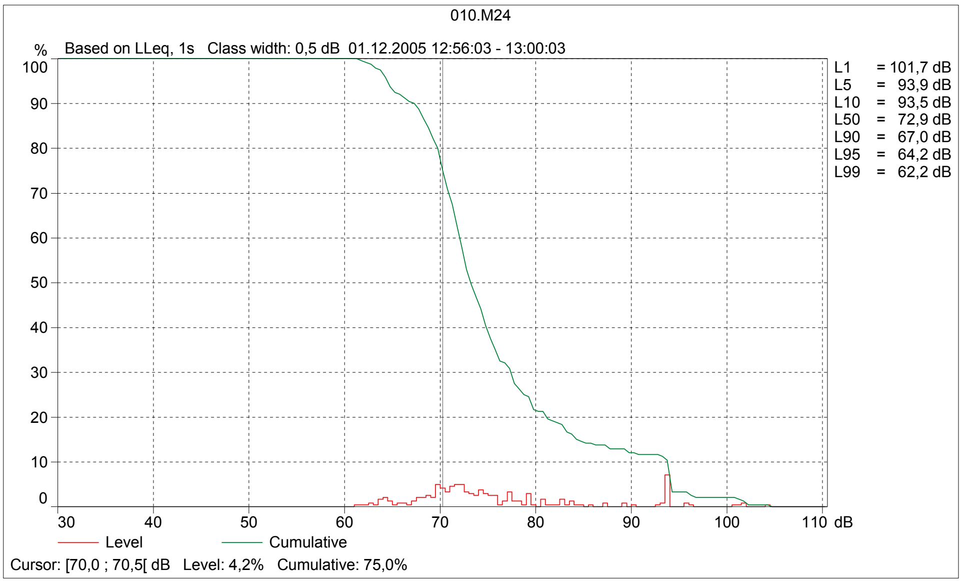


FIGURE 17 Percentile levels for KirriL measurement

In Figure 17 statistics for the complete KirriL sound pressure signal (i.e., the calibration period and the actual measurement) is included. From the cumulative percentile graph we realize that the calibration signal at 94 dB level and the level histogram bins above it create a 12% bias on the graph. This bias is due to the calibration signal itself and those transients generated by switching the calibrator OFF/ON/OFF and finally dismounting it from the microphone (cf. Figure 16). This can also be seen in the LLeq histogram which only exceeds the 94 dB limit at four points during the calibration period.

The final example of measurement results is related to frequency contents of road traffic noise. Measurement data for this example is taken from the KirriA measurement which is used also later in Chapter 8 for verification purposes. A three minute long time history of the measured instantaneous sound pressure shown in Figure 44 was converted to a frequency spectrum using a 4096 point FFT transform on MATLAB.

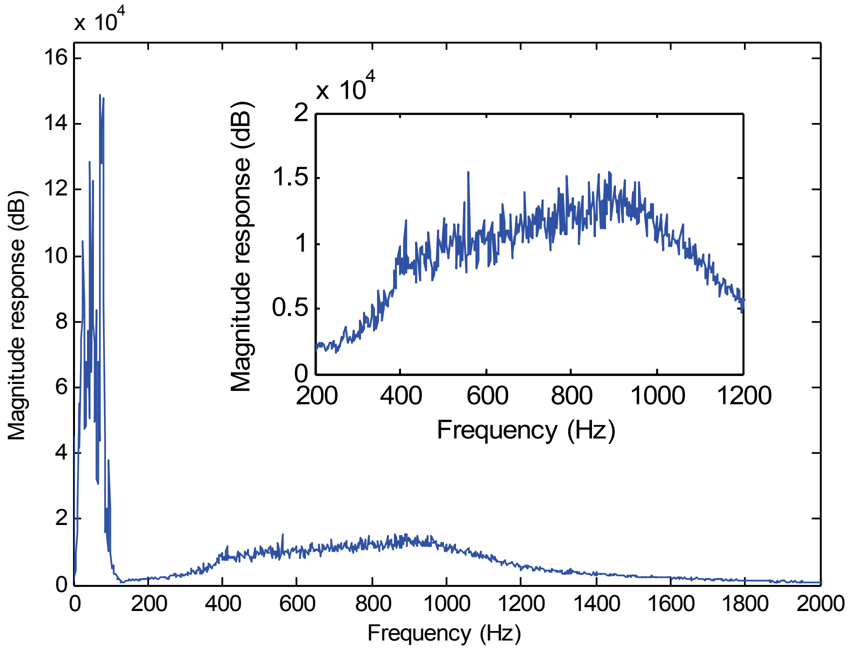


FIGURE 18 Frequency spectrum for KirriA measurement

From this frequency spectrum we are able to identify common frequency components which are present in road traffic noise. At the low frequency end in the figure 14 peaks are found below 100 Hz of which two most intensive ones are located at 73 Hz and 80 Hz. These frequencies are typical in four-stroke combustion engine noise as explained in Section 1.1. By listening an audio recording of this measurement one can identify 14 vehicles of which 7 are heavy trucks. In Figure 44 highest peaks are emitted by these trucks.

In the inset of Figure 18 another prevalent road traffic noise component, namely the tire noise, is seen more clearly. Compared to the engine noise which shows sharp frequency peaks the tire noise spreads to a broader frequency range; in this example between 400 – 1000 Hz. The shape of this tire noise spectrum agrees very well with measurements carried out in similar four-lane freeways where the speed limit has been comparable with our measurement location (80 km/h in our setup; 90 km/h on Freeways 4 and 7 in [Bur14]). In all of these three measurements the maximum intensity of the tire noise spectrum is located at about 900 – 1000 Hz.

4 CALIBRATION

In the previous chapter we took a look on the practice of environmental noise measurements. One aspect we did not handle was calibration of measurement apparatus. The fluctuations of sound pressure were originally normalized by B&K 2238 so that the measured voltages were confined within a range -1 V < U < 1 V. These floating point values as bare numbers are worthless. One essential component of the measurement chain is still missing. Without calibration the measured values are indeed floating – they do not know where a decimal point should be set. A reference is needed with which measured voltages can be related to a known scale of sound pressure levels. Where do we find this kind of reference?

International System of Units (SI) was developed to provide universal definitions for fundamental and derived physical quantities as well as their units. In environmental noise assessment the physical quantity we are primarily interested in is pressure. Which of the fundamental physical quantities are relevant for pressure measurements? The pressure is by definition a force applied on a unit area [SI06]. The latter is measured in square meters and meter is already a base unit for the fundamental physical quantity length. The force, on the other hand, is a derived quantity. It is expressed in units of newtons for which a further decomposition exits i.e., N = kgm/s^{-2}. Here we have two additional fundamental quantities involved. Hence, in sound pressure measurements we are dealing with three fundamental physical quantities; namely the length, mass and time.

Still half a century ago the international SI norm for the length was a one meter long piece of 'railway track' in Paris. If someone wanted to calibrate a length scale (s)he had to travel to Paris to perform a comparison against this platinum-iridium prototype of the meter. At present the definition of the meter is related to the speed of light in vacuum [Man12]. Therefore, also national metrology laboratories are able to realize the length calibration with a good accuracy. Hence, we do not need to travel to Paris any more to calibrate our own measure of length. Fortunately, there is no need too calibrate the units of length, mass and time separately either. What we have available at the moment are sound level calibrators which can be used to calibrate sound pressure levels directly. In some sense, however, these calibrators must still make a 'virtual tour to Paris'. In order for environmental noise measurements to be officially acceptable the measurement apparatus must have a valid and up-to-date calibration certificate. Once a year sound level meters must be sent to an accredited calibration laboratory where they are calibrated with more accurate methods. Also national central laboratories of metrology exist (e.g., MIKES[12] in Finland) which provide calibration norms for accredited national laboratories. The checkup chain continues so forth until an international reference (the 'railway track in Paris') is reached. One essential requirement for a successful process is the comparison of uncertainties in each step of this calibration chain. This process of stepwise checks is referred to as traceability [Gar12]. For example the calibration of the sound level meter B&K 2238 used at Kirri can be traced back to the B&K calibration laboratory, Danish Primary Laboratory of Acoustics and finally to an international standardization body (National Institute of Standards and Technology (NIST), USA). This kind of traceable calibration chain is a prerequisite for reliable and repeatable environmental noise measurements.

The noise indicator channel of B&K 2238 was calibrated at the beginning of Kirri measurements in a normal way. The sound level calibrator used for this purpose had undergone a traceable calibration process as described above. In the following sections we will consider the requirements for the calibration of the audio channel (see. Figure 14). Audio channel calibration is exceptional and not part of typical environmental noise assessments performed with basic sound level meters. For this work, however, it is an important additional requirement. The term calibration in the rest of this chapter has a threefold meaning. Different aspects of it are time, frequency and amplitude calibrations. We consider first the calibration of time. We then proceed on to the frequency and amplitude calibrations.

4.1 Synchronization of time axes

While measuring equivalent continuous sound pressure levels and recording corresponding audio data samples a short time shift was present in time axes between these two measurements. Due to technical limitations it was impossible to start both measurements exactly simultaneously. Logging of audio data samples could be initiated only after a short delay had elapsed following the start of the L_{eq} measurement. This resulted in a synchronization mismatch of time axes while comparing L_{eq} histograms obtained directly from B&K 2238 and those histograms calculated from digitized audio samples. Actually, the question is about synchronization of time axes between the two measurement channels (i.e., noise indicator versus audio). At the beginning of each measurement a 30 – 60 second long period of calibration sequence was generated. The structure of the calibration signal was arranged in such a way that it enabled us to synchronize axes of time histories for these two measurement channels.

Time resolution in the noise indicator channel is one second as can be detected from Figure 16. Time information is embedded in measurement data so that date and time annotations accompany every equivalent continuous sound pressure value in the noise indicator data file (see Appendix A). Times for the measurement start and stop as well as the duration of the measurement are stored also separately in the header of this data file. The audio channel has much better time resolution in comparison to the noise

[12] Center for Metrology and Accreditation, www.mikes.fi

indicator channel. The digitization with a sampling frequency of 44100 Hz produced samples which cover a period of 22.7 microseconds each. We only know the length of one sample, but the position of it on the time axis is not specified. Start times for audio channel measurements cannot be resolved from the samples data file directly. Therefore, we do not know the exact synchronization of time axes between these two measurement channels. Recording for audio data started approximately 2 – 3 seconds after the corresponding measurement in the noise indicator channel was initiated. We have, however, generated sharp edges in the calibration signal. These transition positions act as timestamps which define a common reference of time for both of the channels. Signal transitions from calibration level down to 'calibrator OFF' level can be used to adjust the audio channel time axis so that it is consistent with the noise indicator channel timing.

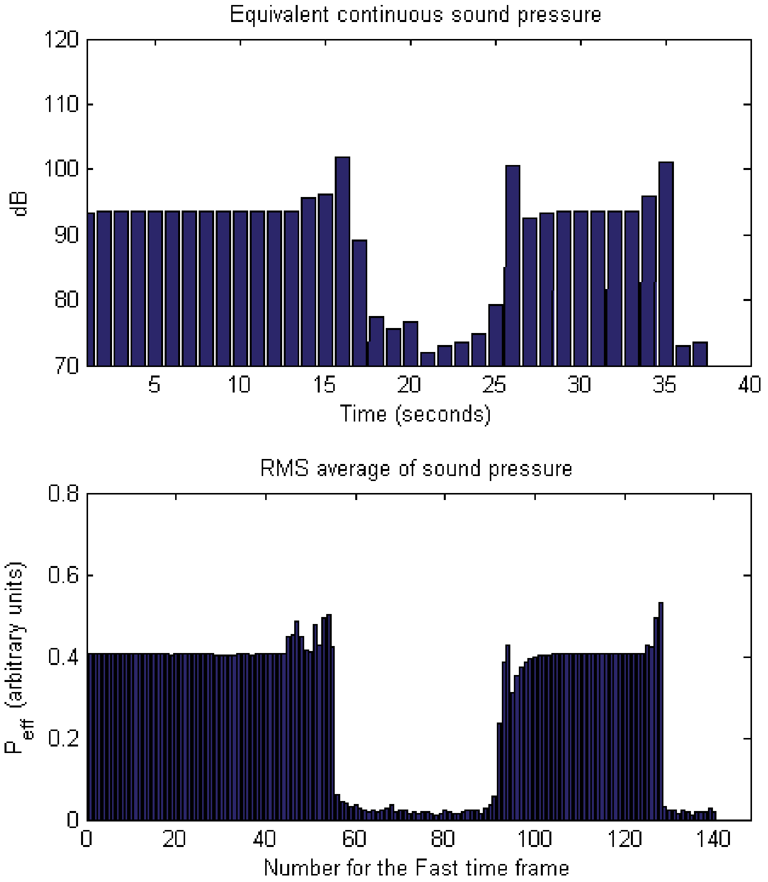


FIGURE 19 a) L_{Leq} for calibration signal of KirriL (upper)
b) p_{eff} for calibration signal of KirriL (lower)

The difference in time resolutions between the two measurement channels is substantial. For each noise indicator value we have 44100 observations in the audio channel. To make the time calibration process more straightforward we first reduce the time resolution in the audio channel. As an example let us consider the KirriL measurement where a linear frequency weighting was applied in both of the channels. By using the Fast time weighting constant (i.e., 250 ms) as a time period for integration we calculate an effective sound pressure for the calibration signal of KirriL measurement. A procedure explained in Equation 4.2 of Section 4.3 enables us to generate a plot for the p_{eff}. This RMS sound pressure is drawn in Figure 19 together with L_{Leq} graph obtained from the noise indicator channel data. In this figure $p_{eff}(t_F)$ is shown for the KirriL measurement when the Fast time frame t_F gets values $1 < t_F < 140$. Thus, 35 seconds of the calibration signal is depicted.

The highest L_{Leq} value in the upper histogram is found for the 16th second. After this the equivalent continuous sound pressure drops rapidly, because the sound level calibrator was switched OFF at that moment. Where do we find same kind of edge in the lower histogram? In Figure 19 b) a local maximum for the effective sound pressure is observed at the Fast time frame number 54. Each of these time frames lasts 250 milliseconds. By grouping the frame 54 with three earlier frames a one second long channel is created. Continuing to group four successive Fast time frames until the end of the histogram we notice that the peaks among the generated one second long periods coincide with 26th and 35th seconds in Figure 19 a). By going backwards from the Fast frame number 50 in the same manner we realize that the two very first time frames are left out. They correspond to half a second period before the fourth second in Figure 19 a). From the above mentioned comparisons we concluded that the audio channel measurement started in the middle of the third second of the noise indicator channel measurement. Therefore, we drop the time frames 1 – 2 away and map the time frames 3 – 6 to the fourth second of the noise indicator histogram (which equals time moment 12:56:06). A delay between the time axis of the audio channel compared to the noise indicator channel is 2.5 seconds (12:56:05.5 – 12:56:03). The mapping between the noise indicator channel seconds and the Fast time frames in the audio channel can also be seen in Figure 20.

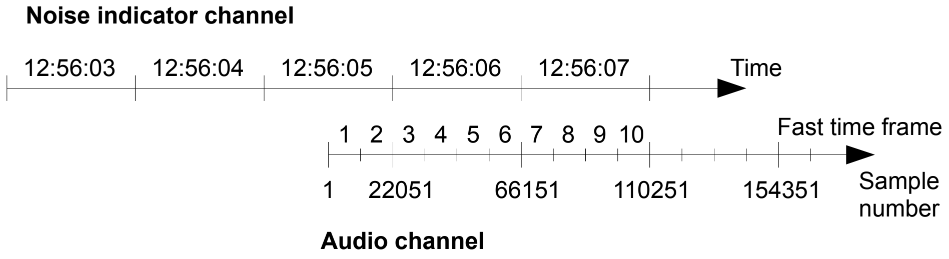


FIGURE 20 Time axes synchronization between measurement channels

This way we have managed to synchronize times between the noise indicator and audio channels. In the next section we will check the calibration of the frequency axis.

4.2 Frequency Calibration Check

Time and frequency are like two sides of a coin. For a periodic signal the frequency is the reciprocal of the period time – that is, $f = T^{-1}$. An environmental noise sample set constitute a time history for measured sound pressure values. The fluctuations in the observed time histories origin from a superposition of pressure waves from several periodic phenomena emitting with their characteristic frequencies. Individual frequency components included in these time histories can be found by a transform to the frequency domain. An FFT can hence be utilized to check whether the audio channel of the Kirri measurements resolves the frequencies correctly. We take first a look on the calibration signal in the time domain. Thereafter, its properties are checked in the frequency domain.

A detailed structure of the calibration signal is revealed by taking a short snapshot of it. A ten milliseconds long period starting at 12:56:07 is extracted from the calibration signal of KirriL measurement and drawn in Figure 21 below. From the figure it is possible to verify that the calibration signal emitted by the sound level calibrator is a 1 kHz sinusoidal as it should be.

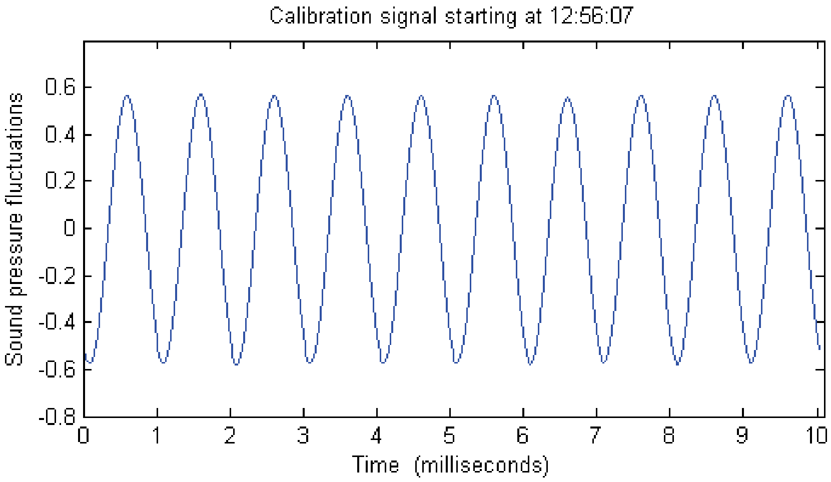


FIGURE 21 Fine structure for calibration signal of KirriL

From the noise indicator channel histogram (Figure 19 a) we observe that the calibration signal of KirriL stays constant during the first 13 seconds before the calibrator is switched off for the first time. In the audio channel the period of constant calibration level lasts only 10.5 seconds because of the 2.5 seconds delay in its start compared to the noise indicator channel. A 10.5 seconds recording with 44100 Hz sampling frequency generated 463050 digitized samples which can be used to check the calibration signal in the frequency domain.

Using a 4096 point FFT transform we are able to cover almost the whole 10.5 seconds period by taking 113 transforms from consecutive time history slices. In order to prevent aliasing we begin by lowpass filtering and downsampling the original time history. First, we read in the selected 10.5 second calibration signal, use a lowpass filter with a cutoff frequency at 4 kHz and downsample the filtered signal by a factor of 5. As a result we obtain a discrete-time signal sampled by 8820 Hz sampling rate containing frequencies up to 4 kHz. We can restrict the region of interest below this frequency, because we are concerned about the behavior of the calibration signal in the vicinity of 1 kHz. Next, all 113 slices of 4096 points each are transformed into the frequency domain using a 4096 point FFT. Finally, power spectra of all these transforms are summed up to a single calibration spectrum (Figure 22).

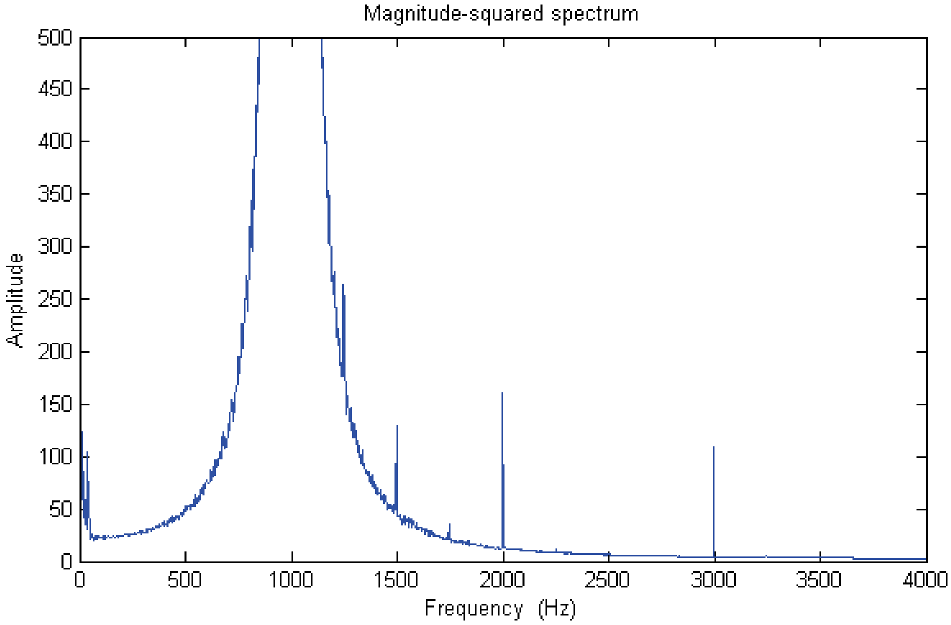


FIGURE 22 Frequency spectrum for calibration signal of KirriL

From Figure 22 it can be seen that the frequency localization of the spectrum is correct. The calibration peak is centered exactly at 1000 Hz as it should be. A couple of harmonics are detected at frequencies 1.5, 2.0 and 3.0 kHz, but their intensities are negligible (21 million counts in the calibration peak compared to 160 counts or less in the harmonics). Amplitudes in the calibration spectrum are not normalized. A normalization could be done by dividing the number of counts in each channel by the length of the calibration signal (113 * 4096 = 462848). This kind of normalization neither changes the shape of the calibration spectrum nor affects the frequency scale. Therefore, it was not applied. Instead of this only the vertical scale on the plotted figure has been modified by clipping the calibration peak. In this way it has been possible to check finer details of the correctness of frequency localization. Figure 22 shows that the audio channel of the Kirri measurement setup maps the frequency content of measured sound pressure signals correctly. The check obtained this way is of course only a one point calibration regarding the frequency scale. It is, however, an established procedure in environmental noise field measurements to use calibrators with a single frequency.

4.3 Sound Pressure Calibration

The sound pressure calibration is essentially an amplitude calibration. We need to relate the observed signal amplitude with a sound pressure level. Primary purpose of this level calibration is to provide means for correlating the measured voltage levels with the calculated sound pressure levels.

From the specification of the sound level calibrator B&K 4231 we learn that the sound pressure level of the calibration signal is 94.0 dB [BK06]. This calibration level applies to measurements conducted with a diffuse-field microphone. In our case the measurements were carried out using a free-field microphone. For this type of microphones an equivalent free-field calibration level is 93.8 dB. We must, however, make still another small correction to this calibration level before it can be used in the audio channel calibration. Reflections and attenuation in the extension cable connecting the microphone and the preamplifier to the sound level meter reduce this level. We can first check how much the noise indicator channel was affected.

A suitable period for extracting calibration data for the KirriL measurement can be found between 12:56:07 and 12:56:12. By inspecting Figure 16 and Appendix A we observe that during this time period basic environmental noise indicators recorded by the SLM (B&K 2238) stay on a constant level as given below

$$\begin{aligned} L_{Leq} &= 93.5\,dB \\ L_{LFMax} &= 93.6\,dB \\ L_{LFMin} &= 93.5\,dB \end{aligned}$$

Therefore, a 0.3 dB attenuation has occurred in the transfer of the preamplified signal from the front end electronics to the SLM. We can assume that at least the same amount of attenuation is applied on the audio channel because an additional connection from the SLM to the laptop exists. Based on this argument, the 93.5 dB sound pressure level was selected to be used for the calibration of the audio channel instead of the original calibrator level. This enables a fair comparison between the noise indicator and audio channel measurements.

While calculating sound pressure levels a base 10 logarithm function is involved. An instantaneous sound pressure p(t) fluctuates around an equilibrium (i.e., the atmospheric pressure) and gets both positive and negative values. Hence, p(t) cannot be used directly to convert sound pressure observations to sound pressure levels. Instead of instantaneous values an effective sound pressure should be used. By calculating an average power of the sound pressure signal as "a square root of the arithmetic average of a set of squared instantaneous values" [BK12] we obtain an effective sound pressure

$$p_{eff} = \sqrt{\frac{1}{T}\int_T p^2(t)\,dt} \tag{4.1}$$

Strictly speaking, we ought to let T tend to infinity inside the square root, because the signal power is only defined for continuous signals. For practical measurement purposes, however, Equation (4.1) can be used to calculate root-mean-square averages for sound pressure signals within a limited time period [Lah95]. In environmental noise assessments the Fast time constant is most often selected for the time period T. Therefore, a total time of 250 ms is allocated for each integration period. In discrete-time processing it is possible to sum up digitized samples for a period of 250 ms. While recording environmental noise during the measurements at Kirri the amplified analog signal was digitized with a 44100 Hz sampling frequency. A 250 ms period of digitized environmental noise data contains therefore 11025 samples. Effective sound pressure for the Kirri data can be calculated using a discrete-time equivalent for Equation (4.1). By denoting with N_F the number of samples within the time period T we have

$$p_{eff} = \sqrt{\frac{1}{N_F} \sum_{N_F} p^2[n]} \tag{4.2}$$

To obtain a reference value for the sound pressure level calibration of the KirriL measurement we proceeded as follows. First a six second long sequence of samples (12:56:07 – 12:56:12) was extracted from the data file where the digitized environmental noise signal was stored. Because of the time shift between the starting times of the SLM measurement and recording of audio samples the above mentioned time period begins 1.5 seconds after starting the audio channel recording (cf. Figure 20). Therefore, the corresponding data can be found between samples 66151 – 330750 in the audio data file. A total of 264600 samples in the resulting sequence were split in slices of N_F samples each. As stated above the samples count N_F equals 11025 with the audio channel measurement parameters. The effective sound pressure for all 24 slices inside the samples range of interest was calculated according to Equation (4.2). The instantaneous sound pressure p[n] in computations was represented by the sample values obtained from the audio data file. For the selected six second calibration period an average effective sound pressure $p_{eff}^{calib} = 0.4046$ with a standard deviation ± 0.0004 was obtained. From this effective sound pressure and the known sound pressure level of the calibrator signal as measured by the SLM $\left(L_p^{calib} = 93.5\,dB\right)$ it was possible to calculate a reference value p_0 for the level calibration. By substituting the above mentioned values in Equation (4.4) a calibration coefficient $p_0 = (8.552 \pm 0.009) \cdot 10^{-6}$ resulted.

$$L_p^{calib} = 20 \cdot \log_{10} \frac{p_{eff}^{calib}}{p_0} \tag{4.3}$$

$$\Rightarrow p_0 = \frac{p_{eff}^{calib}}{10^{\frac{L_p^{calib}}{20}}} \tag{4.4}$$

Knowing the reference value p_0 enables us to generate a histogram for the sound pressure levels of the audio channel. Due to the time synchronization of the two measurement channels a short subset of samples needs to be dropped out from the audio data file. The samples 1 – 22050 related to fractions of the first second are ignored. The first complete second of the audio channel histogram is mapped to time 12:56:06. For each second an equivalent continuous sound pressure (L_{Leq}) was calculated using 44100 samples. The values were calibrated by the reference value p_0 while converting to sound pressure level. Calculated L_{Leq} values expressed in a desibel scale are plotted as a histogram in Figure 23 b). Both the time axis and the sound pressure scale of the audio channel are calibrated to enable comparison with the noise indicator channel graph in Figure 23 a).

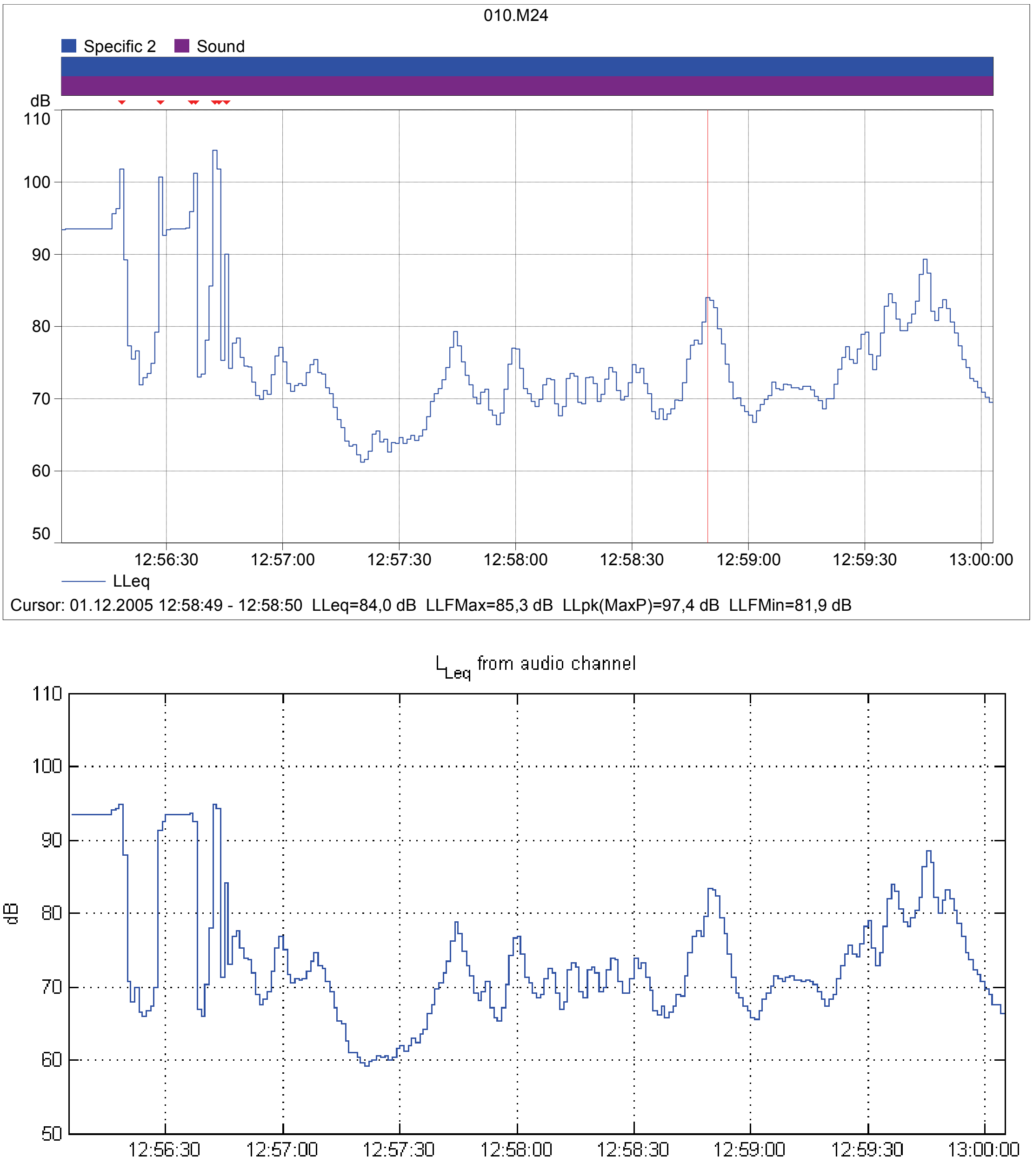


FIGURE 23 Calibrated sound pressure levels for KirriL
a) directly from B&K 2238 (upper)
b) from audio channel (lower)

The histogram in Figure 23 b) was obtained by running an M-file that was developed for MATLAB. The unweighted equivalent continuous sound pressure level L_{Leq} was computed using the equation below

$$L_{Leq}(s) = 20 \cdot \log_{10} \sqrt{\frac{1}{N} \sum_{n=1}^{N} \frac{p^2[n+(s-1) \cdot N + N_0]}{p_0^2}} \tag{4.5}$$

where the time variable indicating seconds $1 < s < 239$, the samples count within one second $N = 44100$, the number of skipped samples $N_0 = 22050$ and the reference sound pressure value p_0 was as calculated by Equation (4.4).

5 FREQUENCY WEIGHTINGS

In environmental noise assessment physical sound pressure signals are detected most often using condenser microphones. Typically the frequency response of good quality microphones used for environmental noise inventories is linear within the frequency range 20 – 10000 Hz (cf. Figure 12). As opposed to this the human auditory system is possessing a non-linear response to the above mentioned frequency range. The intention of environmental noise measurements is to assess the impact of noise on people's health, annoyance and overall wellbeing. To better estimate the symptoms, reactions and consequences of measured sound pressure signals on humans we must apply corrective actions on microphone's frequency response.

5.1 Origin of Frequency Weightings

Human auditory system's response to different frequencies is not static. The sensitivity of human ear depends on the level of sound pressure stimulus. Several correction schemes have been developed which try to adapt measurement results to sensation of hearing on different sound pressure levels. With high SPL the feedback mechanism from the neural auditory system back to the peripheral auditory system adjusts physiological parameters of hearing (see the explanation for cochlear amplification at the end of Chapter 2). Due to this feature the response of the peripheral auditory system to the sound stimulus varies. Preprocessing of sound pressure signals at higher SPL is tuned to a less sensitive mode as a result of this feedback mechanism. Protection of hearing organs is one benefit of this kind of adaptation. At environmentally noisy surroundings a less sensitive equal loudness curve describes tuning of human auditory system most adequately.

As we learned earlier in Section 2.7 equal loudness curves tell us how strong stimulus is needed for specific frequencies to produce equal sensations of loudness. What happens if we turn the question upside down and ask: "How insensitive human ear is to different frequencies?" In this case we search for an answer to a question of attenuation as a function of frequency. Roughly speaking we can rotate an equal loudness curve about the horizontal axis and we will obtain what we are looking for. The curve obtained this way informs us what weights must be applied on measured frequencies to make their amplitudes agree with the human sensation of observed sound pressure fluctuations. By averaging equal loudness curves related to sensitivity at high SPL in Figure 8 and mirroring them horizontally we get a relationship that describes the response of human ear as a function of frequency. Thus, for high SPL we have a frequency weighting curve that looks like the one shown in the figure below.

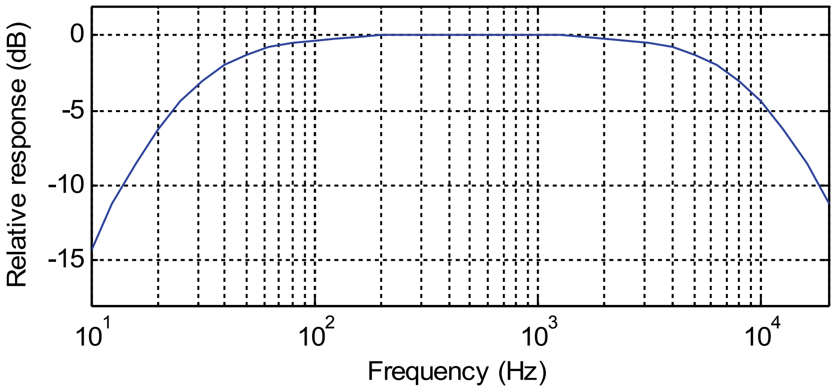


FIGURE 24 Frequency response of human auditory system to high SPL

Traditional sound level meters realize the described frequency weighting using analog components. As was shown in Chapter 1 (Figure 3) this work intends to implement frequency weightings digitally. Thus, the task comes down to designing a digital filter with a desired frequency response. Essentially this design process would present an optimization problem. The goal would be to maximize the energy transfer within the passband and minimize transmission for the stopband of the filter. Several design methods exist for this purpose (e.g. based on genetic algorithms, differential evolution, particle swarm optimization, seeker optimization algorithm [Dai10]). In this work, however, none of these methods are used. Instead, we start from an analog prototype filter and transform it to an equivalent digital design.

Analog filters for frequency weighting have existed already for several decades. Therefore, standards have evolved which specify requirements for analog frequency weighting filters. One of the most commonly used of these standards is IEC/CD 1672 for electroacoustic sound level meters [IEC96]. This standard gives a specification for a C-weighting filter which resembles human auditory system's response to high SPL. The analog C-weighting filter is characterized by zeros at an analog angular frequency of $2\pi*12200$ rad/s and double poles on real-axis at $2\pi*20.6$ and $2\pi*12200$. In the analog domain the system function of the C-weighting filter is given by

$$H_{Cw}(s)=\frac{4\pi^2\cdot 12200^2\cdot s^2}{(s+2\pi\cdot 20.6)^2\cdot(s+2\pi\cdot 12200)^2} \tag{5.1}$$

For situations where the sound pressure level is lower two other frequency weightings have been developed. By adding to $H_{Cw}(s)$ one real-axis pole with an analog angular frequency of $2\pi*158.5$ rad/s the B-weighting filter is obtained

$$H_{Bw}(s) = \frac{4\pi^2 \cdot 12200^2 \cdot s^3}{(s+2\pi\cdot 20.6)^2 \cdot (s+2\pi\cdot 12200)^2 \cdot (s+2\pi\cdot 158.5)} \tag{5.2}$$

The most commonly used weighting in environmental noise assessment is the A-weighting. This weighting is closely related to the 40 phon equal-loudness curve as shown in Figure 8. A stronger attenuation in comparison to C-weighting at low frequencies is achieved by two additional poles at $2\pi * 107.7$ and $2\pi * 737.9$

$$H_{Aw}(s) = \frac{4\pi^2 \cdot 12200^2 \cdot s^4}{(s+2\pi\cdot 20.6)^2 \cdot (s+2\pi\cdot 12200)^2 \cdot (s+2\pi\cdot 107.7) \cdot (s+2\pi\cdot 737.9)} \tag{5.3}$$

Analog prototype filters based on the aforementioned system functions are taken as a starting point for the design of digital frequency weighting filters. Using bilinear transformation these designs are converted from the analog domain to the digital domain. The steps required for this transformation are described in the following sections.

5.2 From Analog to Digital Domain

Filtering in the analog domain can be described using differential equations. In a general case the relationship between the analog filter output $y(t)$ and its input $x(t)$ can be expressed as

$$\sum_{i=0}^{N} a_i \frac{d^i y(t)}{dt^i} = \sum_{j=0}^{M} b_j \frac{d^j x(t)}{dt^j} \tag{5.4}$$

Thus the filter input and output are connected with linear combinations of their derivatives using weights a_i and b_i. Letting $N=1$ and $M=0$ gives a special case for a first order differential equation of the filter output $y(t)$

$$a_i y'(t) + a_0 y(t) = b_0 x(t) \tag{5.5}$$

In the above equation we still reside in the analog domain. Our goal is a transformation into the digital domain. Therefore, we begin by considering y'(t) at distinct points. Let us select a constant T so that we observe the signals at equally spaced points. Two successive observation point can be denoted by (n-1)T and nT. From the integral calculus we obtain an alternative expression to relate the filter output an its derivative. Namely, the difference of the output at time moments (n-1)T and nT is given by

$$y(nT) - y([n-1]T) = \int_{[n-1]T}^{nT} y'(t)\,dt \tag{5.6}$$

Solving Equation (5.5) for $y'(t)$ and substituting into (5.6) leads to

$$y(nT) - y([n-1]T) = \frac{b_0}{a_1} \int_{[n-1]T}^{nT} x(t)\,dt - \frac{a_0}{a_1} \int_{[n-1]T}^{nT} y(t)\,dt \tag{5.7}$$

By selecting T to be small enough the integrals of the analog input and output signals can be estimated by $(T/2)\cdot\{x(nT)+x([n-1]T)\}$ and $(T/2)\cdot\{y(nT)+y([n-1]T)\}$ respectively. Thus, we have derived a relationship

$$y(nT)-y([n-1]T)=\frac{b_0 T}{2\mathrm{a}_1}\{x(nT)+x([n-1]T)\}-\frac{a_0 T}{2\mathrm{a}_1}\{y(nT)+y([n-1]T)\}$$

$$\Rightarrow (2\mathrm{a}_1+a_0 T)y(nT)-(2\mathrm{a}_1-a_0 T)y([n-1]T)=b_0 T\{x(nT)+x([n-1]T)\} \qquad (5.8)$$

Taking the Z-transform on each side of Equation (5.8) and using additional notations for transformed output $Y(z)=Z\{y(nT)\}$ and input $X(z)=Z\{x(nT)\}$ gives us

$$(2\mathrm{a}_1+a_0 T)Y(z)-(2\mathrm{a}_1-a_0 T)z^{-1}Y(z)=b_0 T\{X(z)+z^{-1}X(z)\} \qquad (5.9)$$

By combining the terms containing Z-transforms of the filter output, dividing through by the resulting multiplier of $Y(z)$ and finally dividing both sides of Equation (5.9) by the Z-transform of the filter input $X(z)$ we have

$$\frac{Y(z)}{X(z)}=\frac{b_0 T(1+z^{-1})}{2\mathrm{a}_1(1-z^{-1})+a_0 T(1+z^{-1})} \qquad (5.10)$$

After the previous steps we have entered the digital domain and found a ratio between the Z-transforms of the filter output and input signals. By definition this is the transfer function of the filter which enables us to write – after reducing the right hand side of the above equation by $T(1+z^{-1})$

$$H(z)=\frac{b_0}{a_1\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}+a_0} \qquad (5.11)$$

On the other hand we can go back to the analog domain for a while to derive the filter's system function from its differential equation. By taking the Laplace-transform on both sides of Equation (5.5) and utilizing the transform properties of *L{ }* we obtain

$$a_1 s\,Y(s)+a_0 Y(s)=b_0 X(s) \qquad (5.12)$$

Solving the above equation for the ratio $Y(s)/X(s)$ and identifying it as the filter's system function $H(s)$ we have

$$H(s)=\frac{b_0}{a_1 s+a_0} \qquad (5.13)$$

Both the transfer function H(z) of Equation (5.11) and the above system function describe the behavior of the same filter. This leads us to conclude that analog domain's complex variable s and the unit delay z^{-1} in the digital domain must be related as follows

$$s = \frac{2}{T}\frac{\left(1 - z^{-1}\right)}{\left(1 + z^{-1}\right)} \tag{5.14}$$

The same dependency can be expressed in the opposite direction – that is, giving the unit delay as a function of the analog domain's complex variable

$$z^{-1} = \frac{2 - sT}{2 + sT} \tag{5.15}$$

We started from the analog domain where the continuous-time filter was specified as a first order differential equation. We then observed the output of this analog filter at discrete points of the continuous time axis. Using the Laplace- and Z-transforms we were able to derive a relationship for the s-plane variable s and the transform domain unit delay z^{-1}. The transformation between the analog and digital domains defined by Equations (5.14) and (5.15) is known as the bilinear transformation.

5.3 Digital Filter Design by Bilinear Transformation

The continuous-time system function for the C-weighting filter can be expressed in the form

$$H_{Cw}(s) = \frac{(\alpha s)^2}{(s+\beta)^2 (s+\gamma)^2} \equiv \frac{NUM}{DEN} \tag{5.16}$$

where coefficients α, β and γ are exact values for analog angular frequencies as given below instead of approximations which were shown in Equation (5.1).

$$\begin{aligned} \alpha &= 2\pi \cdot 12194.217 \\ \beta &= 2\pi \cdot 20.598997 \\ \gamma &= 2\pi \cdot 12194.217 \end{aligned}$$

Let us start by concentrating on the denominator in Equation (5.16) first. After both of the denominator terms are squared, we have

$$DEN = \left(s^2 + 2\beta s + \beta^2\right)\left(s^2 + 2\gamma s + \gamma^2\right) \tag{5.17}$$

By multiplying the two polynomials of the complex variable s term by term results in

$$\begin{aligned} DEN = \; & s^4 + 2\gamma s^3 + \gamma^2 s^2 \\ & + 2\beta s^3 + 4\beta\gamma s^2 + 2\beta\gamma^2 s \\ & + \beta^2 s^2 + 2\beta^2 \gamma s + \beta^2 \gamma^2 \end{aligned} \tag{5.18}$$

By combining coefficients for equal powers of s and renaming these coefficient sums the denominator polynomial can be represented as

$$DEN = s^4 + \delta s^3 + \varepsilon s^2 + \lambda s + \mu \tag{5.19}$$

where the coefficients for the s-domain polynomial are now denoted as

$$\begin{aligned} \delta &= 2(\beta + \gamma) \\ \varepsilon &= \beta^2 + \gamma^2 + 4\beta\gamma \\ \lambda &= 2\beta\gamma(\beta + \gamma) \\ \mu &= \beta^2\gamma^2 \end{aligned}$$

In order to apply the bilinear transformation we use Equation (5.14) to replace the continuous-time variable s with the discrete-time variable z. By denoting the coefficient $2/T$ of this equation by ω we can derive a z-domain denominator DENz by substituting s in the expression of DEN

$$DENz = DEN\Big|_{s = \omega\frac{z-1}{z+1}} \tag{5.20}$$

From the aforementioned substitution it follows that

$$DENz = \omega^4\frac{(z-1)^4}{(z+1)^4} + \delta\,\omega^3\frac{(z-1)^3}{(z+1)^3} + \varepsilon\,\omega^2\frac{(z-1)^2}{(z+1)^2} + \lambda\,\omega\frac{(z-1)}{(z+1)} + \mu \tag{5.21}$$

By expanding all rational fractions in this equation with a suitable power of (z + 1) and extracting the common denominator out of the summation the denominator DENz can be written in the form

$$\begin{aligned} DENz = \frac{1}{(z+1)^4}\{\,&\omega^4(z-1)^4 + \delta\,\omega^3(z-1)^3(z+1) \\ &+ \varepsilon\,\omega^2(z-1)^2(z+1)^2 + \lambda\omega(z-1)(z+1)^3 + \mu(z+1)^4\,\} \end{aligned} \tag{5.22}$$

Coefficients of this denominator polynomial are arranged in tabular form by evaluating the terms of the sums inside the curly braces and grouping the coefficients obtained for different powers of z on separate columns as shown below

	z^4	z^3	z^2	z^1	z^0
term 1	ω^4	$-4\,\omega^4$	$6\,\omega^4$	$-4\,\omega^4$	ω^4
term 2	$\delta\,\omega^3$	$-2\delta\,\omega^3$		$2\delta\,\omega^3$	$-\delta\,\omega^3$
term 3	$\varepsilon\,\omega^2$		$-2\varepsilon\,\omega^2$		$\varepsilon\,\omega^2$
term 4	$\lambda\omega$	$2\lambda\omega$		$-2\lambda\omega$	$-\lambda\omega$
term 5	μ	4μ	6μ	4μ	μ

Each row in this tabulated form shows the coefficients obtained from a specific term in Equation (5.22) (e.g., first term → coefficients on the first row). By summing individual contributions of different terms columnwise coefficients for different powers of the variable z can be obtained. Let us denote these coefficients as follows

$$
\begin{aligned}
v_0 &= \omega^4 + \delta\omega^3 + \varepsilon\omega^2 + \lambda\omega + \mu \\
v_1 &= -4\omega^4 - 2\delta\omega^3 + 2\lambda\omega + 4\mu \\
v_2 &= 6\omega^4 - 2\varepsilon\omega^2 + 6\mu \\
v_3 &= -4\omega^4 + 2\delta\omega^3 - 2\lambda\omega + 4\mu \\
v_4 &= \omega^4 - \delta\omega^3 + \varepsilon\omega^2 - \lambda\omega + \mu
\end{aligned}
$$

Using these notations the denominator polynomial can be written in the form

$$DENz = \frac{1}{(z+1)^4}\{v_0 z^4 + v_1 z^3 + v_2 z^2 + v_3 z + v_4\} \equiv \frac{A(z)}{(z+1)^4} \tag{5.23}$$

Similarly, Equation (5.14) relating the s-plane and z-plane variables can be applied on the numerator polynomial to transform it from the analog domain to the digital one

$$NUMz = NUM|_{s=\omega\frac{z-1}{z+1}} = \{\alpha\omega\frac{z-1}{z+1}\}^2 = \alpha^2\omega^2\frac{(z-1)^2}{(z+1)^2} \tag{5.24}$$

By combining the numerator and denominator expressions we derive a representation for the transfer function of the C-weighting filter in the digital domain

$$H_{Cw}(z) = \frac{NUMz}{DENz} = \alpha^2\omega^2\frac{(z-1)^2}{(z+1)^2}\cdot\frac{(z+1)^4}{A(z)} = \alpha^2\omega^2\frac{(z^4 - 2z^2 + 1)}{A(z)} \tag{5.25}$$

Substituting $A(z)$ with the polynomial of z as defined earlier in Equation (5.23) and reducing both the numerator and denominator polynomials by z^4 gives us

$$H_{Cw}(z) = \alpha^2\omega^2\frac{1 - 2z^{-2} + z^{-4}}{v_0 + v_1 z^{-1} + v_2 z^{-2} + v_3 z^{-3} + v_4 z^{-4}} \tag{5.26}$$

As a final step this transfer function is normalized so that the first denominator coefficient becomes unity, giving

$$H_{Cw}(z) = \frac{b_0 + b_2 z^{-2} + b_4 z^{-4}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} \equiv \frac{B_{Cw}(z)}{A_{Cw}(z)} \tag{5.27}$$

where additional notations for multipliers in the numerator and denominator polynomials have been introduced as follows

$$
\begin{aligned}
b_0 &= \alpha^2\omega^2/v_0 \\
b_2 &= -2b_0 \qquad \text{and} \qquad \{a_i = v_i/v_0,\ \mathit{where}\ \mathrm{i} = 0..4\} \\
b_4 &= b_0
\end{aligned}
$$

By substituting all numerical constants into the equations derived for the C-weighting filter's transfer function we can calculate values for the multipliers in the numerator polynomia $B_{Cw}(z)$ and the denominator polynomial $A_{Cw}(z)$. The calculated values of these multipliers are presented in the table below.

TABLE 4 Multiplier coefficients for C-weighting filter

i	0	1	2	3	4
b_i(Cw)	0.2155	0.0000	-0.4309	0.0000	0.2155
a_i(Cw)	1.0000	-2.1347	1.2793	-0.1496	0.0049

By using these feedforward and feedback coefficients the frequency response of the C-weighting filter can be drawn against the Class 1 filter specifications.

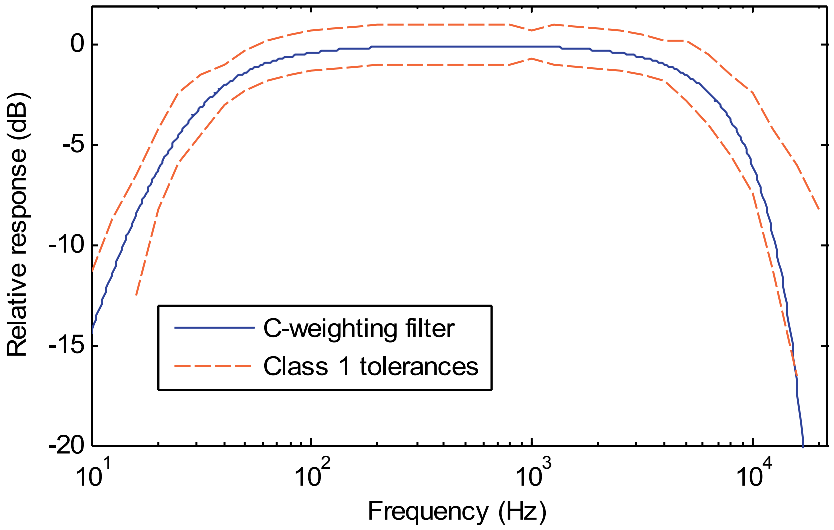


FIGURE 25 Magnitude response for C-weighting filter together with tolerance limits of Class 1 filter specification

In the figure above the magnitude response is depicted for the digital design that resulted from the bilinear transformation of an analog C-weighting filter specification. This figure also shows tolerance limits within which a C-weighting filter design's magnitude response must be confined to be regarded as a Class 1 filter. In the Class 1 C-weighting filter specification these limits are given for all 1/3-octave bands in the frequency range 10 – 20000 Hz. Class 1 tolerance limits were drawn using a function modified from a cspec.m file included in Christophe Couvreur's Octave MATLAB package[13].

[13] http://www.mathworks.com/matlabcentral/fileexchange/69-octave

6 SIMULATION OF A-WEIGHTING FILTER

In the previous chapter we obtained a digital design for a frequency weighting filter. By starting from an analog prototype filter and applying bilinear transformation a fourth-order design was developed for the C-weighting filter. At the beginning of Chapter 5 specifications for three different frequency weightings were given. We will continue now by exploring the most complex one of these, namely the A-weighting. We could proceed in the same way as with the C-weighting filter and by algebraic manipulations convert the analog A-weighting filter specification to a corresponding digital design. We choose, however, to omit this exercise and rely on a function bilinear() available in MATLAB. At the end of Chapter 5 we used the Octave MATLAB package to draw Class 1 tolerance limits to verify that C-weighting filter's transfer function obtained there fulfills the specifications. One of the functions included in this package is adsgn() which implements an equivalent procedure for A-weighting filter as we applied on the analog C-weighting prototype filter. The adsgn() function starts from the analog A-weighting filter specification, multiplies the four denominator polynomials by using convolution and applies the bilinear() function to convert the filter design into the digital domain. As a result, the obtained digital design describes the filter in the z-domain as a sixth-order rational function of the unit delay. Our intention is to reduce the precision of A-weighting filter's numerator and denominator coefficients in order to reduce the computational complexity. There exists, however, a limit beyond which the filter specified using the coefficients with reduced accuracy no longer remains stable. Another potential problem due to quantization effects caused by reduction of precision is violations in regard to the A-weighting filter specification. By MATLAB simulations we should find a compromise for the multiplier coefficient accuracy which would make the computations less demanding but still provide a stable filter implementation conforming to the standards.

6.1 Stability of Original Filter Design

The MATLAB design gives the frequency response *H(z)* of an A-weighting filter in the form of sixth order numerator and denominator polynomials. This original filter design specifies the numerator and denominator coefficients in the maximum accuracy available in MATLAB (i.e., 16 decimal digits). By plotting the frequency response of this design we observe that with the 16 digits accuracy the filter fulfills both Class 1 and Class 2 requirements of the A-weighting filter specifications.

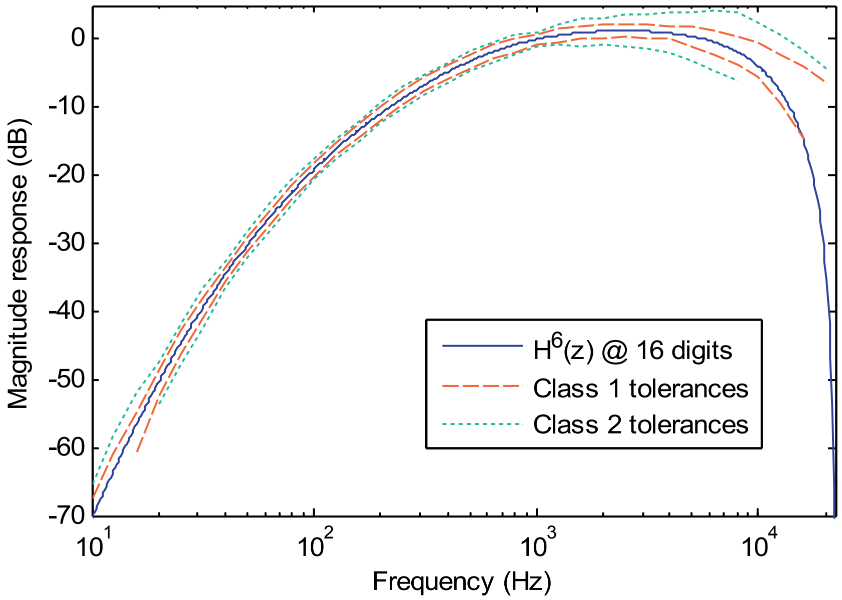


FIGURE 26 Transfer function for a monolithic A-weighting filter

The sixth-order transfer function depicted in the above figure was designed for a sampling frequency of 44100 Hz. Slightly modified versions of Octave package's aspec() function were used to draw this and other figures in Chapter 6.

Stability of this filter design can be confirmed by checking locations of its poles. In MATLAB a filter description given by the numerator and denominator polynomials (B(z) and A(z), respectively) can be converted to a representation showing the zeros and poles by a tf2zpk() function. By applying this function we detect that the poles and zeros for $H^6(z)$ are located as given in the table below.

TABLE 5 Poles and zeros of the original A-weighting filter design

poles	zeros
0.0703	-1.0000
0.0703	-1.0000
0.9848	1.0002
0.9001	0.9998
0.9971	1.0000 + 0.0002i
0.9971	1.0000 – 0.0002i

Besides of the poles and zeros the tf2zpk() function specifies also the gain coefficient which for this original filter design was 0.2557.

By examining a pole-zero plot for this filter we can verify also graphically that all poles[14] are inside the unit circle, which indicates a stable design. The most interesting region in this pole-zero plot can be found around the intersection of the unit circle and the positive real axis, which is shown in the figure below. On that area those poles are found, which are closest to the edge of the unit circle. While the accuracy of filter coefficients is reduced, these poles are the ones most likely turning the filter design into an unstable state.

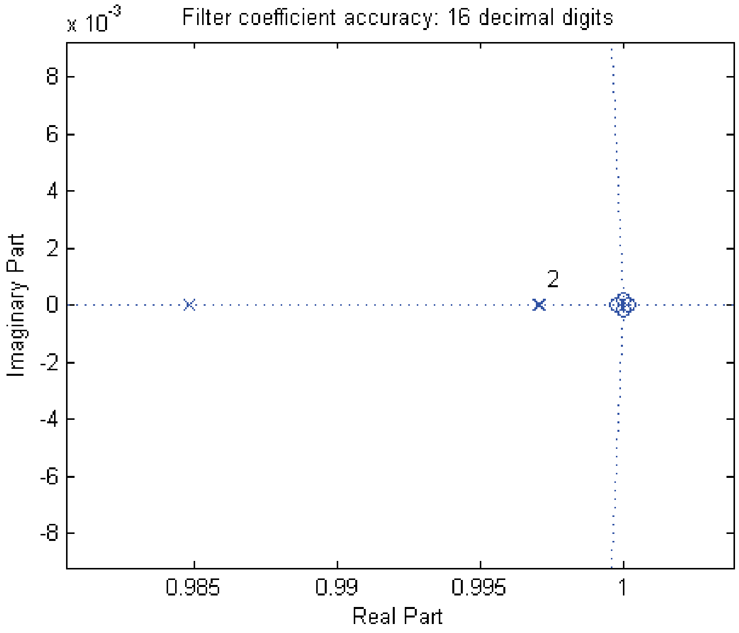


FIGURE 27 Critical poles for $H^6(z)$ with full precision

Figure 27 was created using an accuracy of 16 decimal digits for the coefficients of the original A-weighting filter design. What happens if we reduce the accuracy of the filter coefficients to 5 decimal digits? This would be desirable for an actual implementation on an embedded system platform. One possibility to accomplish the rounding in MATLAB is with variable precision arithmetic. By reducing the accuracy from 16 digits to 5 digits using this method and plotting the frequency response of the filter again with this reduced precision the following can be observed. By comparing Figures 26 and 28 we notice that within the frequency range 2 – 20 kHz reduction of the multiplier coefficients has not affected the frequency response. At lower frequencies, however, the filter design with reduced accuracy does not follow the specification at all. For frequencies below 200 Hz the reduced precision design shows a constant attenuation of about 6 dB instead of gradually enhancing the attenuation down to 70 dB when reaching the 10 Hz

[14] In the graphics the poles are marked by crosses and the zeros by open circles.

frequency as it should do. Furthermore, an additional bump appears on the magnitude response between 400 Hz and 1100 Hz, which also violates even Class 2 tolerance limits of the A-weighting specification.

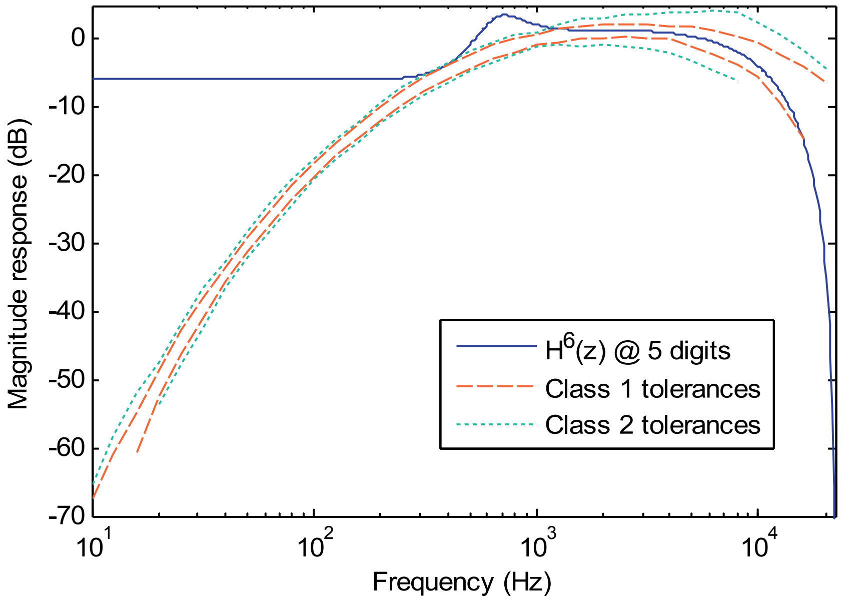


FIGURE 28 *$H^6(z)$* with reduced filter coefficient accuracy

Problems caused by directly reducing the precision of coefficients in the monolithic A-weighting filter design are visible in the pole positions as well.

TABLE 6 Poles and zeros of the reduced precision *$H^6(z)$*

poles	zeros
1.0755	-1.0031
0.9704 + 0.0890i	-0.9969
0.9704 – 0.0890i	1.0559 + 0.0591i
0.8628	1.0559 – 0.0591i
0.0712	0.9441 + 0.0528i
0.0694	0.9441 – 0.0528i

One of the poles moves outside the unit circle and therefore the design is not stable anymore. Besides of this two imaginary poles are introduced, which were not seen in the original full precision design (cf. Table 5). These changes can be observed in Figure 29, which also shows how the poles and zeros are blown up and spread on a larger area around the intersection of the real axis and the unit circle.

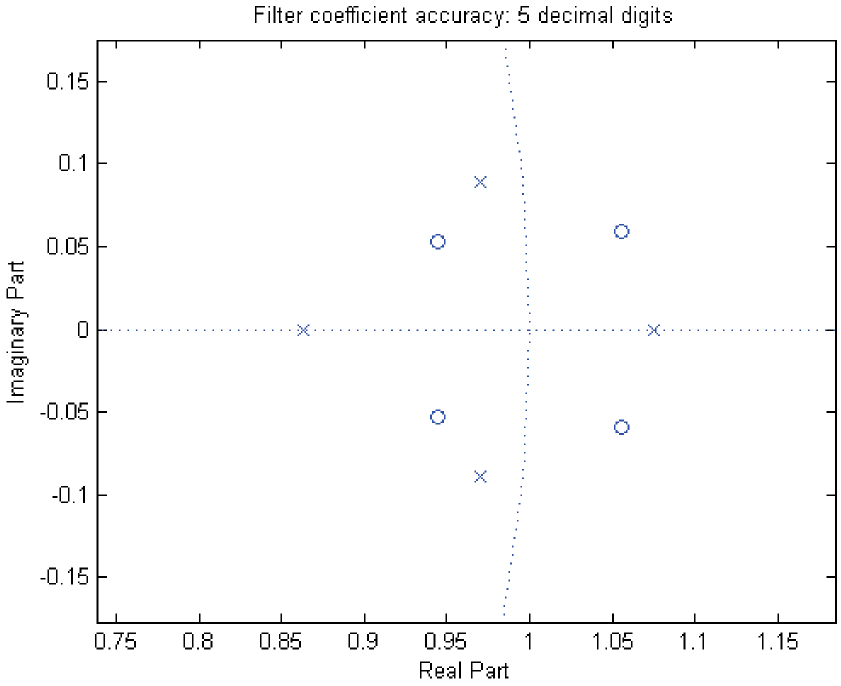


FIGURE 29 Critical poles for $H^6(z)$ with reduced precision

From the previous two figures it is evident that the approach with the accuracy of 5 decimal digits for coefficients of the sixth-order A-weighting filter is not viable. The frequency response of a filter that would be implemented using coefficients obtained this way would violate the required specification through the whole frequency range below 1 kHz.

Following the same procedure as described above we are able to iterate different coefficient precisions to search for a value within the range 5 – 16 digits, which would still give a stable solution. By carrying out this iteration procedure we observe that the filter coefficients need to be at least 8 decimal digits in order to guarantee the stability of the A-weighting filter. For a practical hardware implementation on an embedded platform we need to convert the floating point coefficients into a fixed point representation. While converting the A-weighting filter coefficients rounded to 8 decimal digits into a fixed point representation, we observe that 27 binary digits are required. Therefore, we are not able to store these coefficients in 16-bit data words. For fixed point arithmetic on an embedded platform this poses a requirement of 32-bit filter coefficients which makes the implementation ineffective. These implementation issues are explored more in Section 7.2.

As a summary of the simulations conducted thus far we can state that a direct implementation of the original sixth-order rational function with multiplier coefficients of reduced precision is not a sensible alternative.

6.2 Stability of Second-order Sections

In Section 6.1 we concluded that a direct implementation of the complete A-weighting filter requires eight decimal digits accuracy for the transfer function's coefficients. The hardware implementation, however, only permits five decimal digits accuracy. What kind of ways do we have to reduce the implementation requirements? In this section we will introduce a workaround for the encountered problem.

6.2.1 Decomposition into second order sections

Instead of directly reducing coefficient accuracy of the sixth-order rational function describing the complete A-weighting filter an alternative approach should be developed. One possible solution is to decompose the original transfer function into quadratic terms. This kind of factoring results in three components which all are of second order. Therefore, they are called second-order sections. They are still rational functions of the unit delay z^{-1} like the original function, but the order of them is reduced from six to two. Because these second-order sections contain quadratic polynomials both on the numerator and denominator they are also referred to as biquads[15]. Graphically the decomposition into lower-order components can be illustrated in the transform domain as show in the figure below.

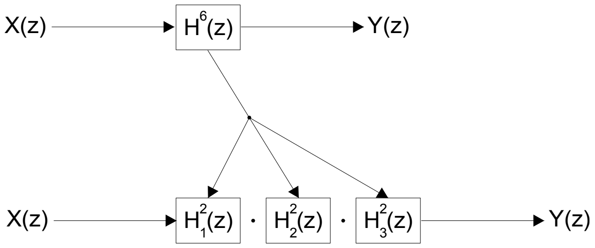


FIGURE 30 Decomposition of transfer function into second-order sections

Original filtering can be thought of as a process where a monolithic transfer function is applied on the input data. After breaking up the complete transfer function into lower-order components the input data passes through several stages before the final output is obtained. In order to check the properties of the different stages we proceeded in the following way. The original A-weighting filter design was first decomposed to second-order sections using a MATLAB function tf2sos(). This resulted in a three row matrix where each row gave the numerator and denominator coefficients for the individual biquads shown in Figure 30. The aforementioned MATLAB function is automatically taking care of ordering the second-order sections so that those ones in which poles are closest to the unit circle are placed at the end of the cascade [Bai06]. To begin with we considered the first second-order section. The multiplier coefficients for this part of the filter design were extracted from the SOS matrix to a separate vector. By a reverse conversion the first second-order section was brought back to the frequency domain. After obtaining a description corresponding to $H_1^2(z)$ we were able to draw it against the

[15] In the remainder of this text these two terms as well as an abbreviation SOS are used interchangeably.

A-weighting filter specifications.

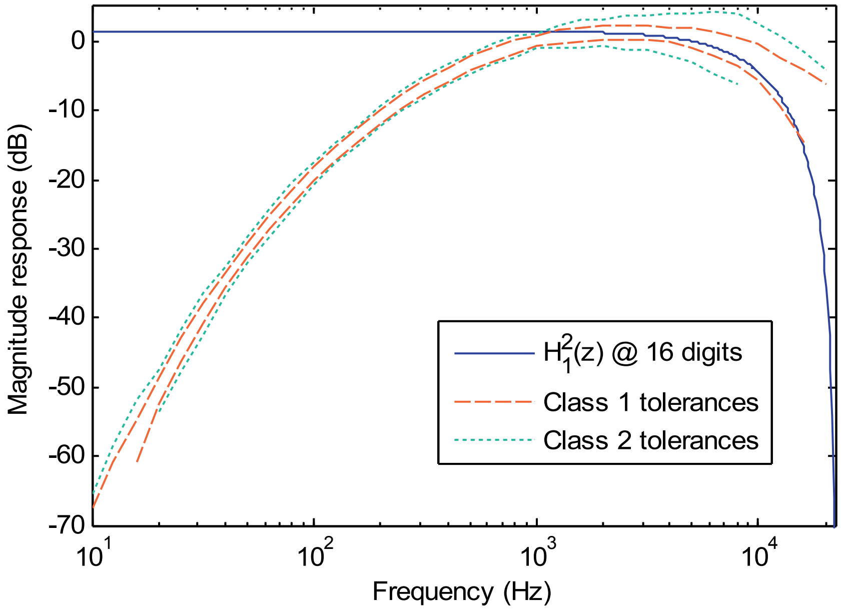


FIGURE 31 Transfer function corresponding to the 1st second-order section

From the above figure we realize that the first second-order section of the cascade structure implements a low-pass filter which mimics the attenuation of high frequency sounds in human ear. The frequency range from 2 kHz up to 20 kHz is covered with this second-order section.

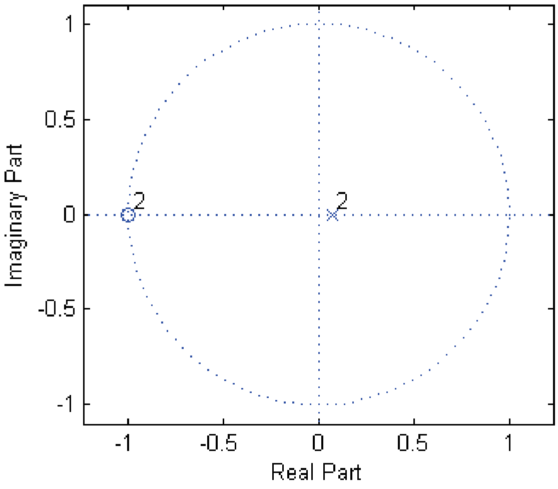


FIGURE 32 Pole-zero plot related to second-order section $H_1^2(z)$

What happens to the poles and zeros after factoring the filter design to biquads? For the first second-order section the pole-zero plot becomes as shown in Figure 32 above. By comparing the pole-zero plot of $H_1^2(z)$ to the pole-zero table of the original A-weighting filter design we notice the following. This first biquad implements the poles and zeros given on the two first rows of Table 5. There is much tolerance in regard to stability for the transfer function $H_1^2(z)$, because the double pole resides close to the origin on the real axis at 0.0703.

The above observations were valid for the first biquad while it was simulated with full precision. What happens if we reduce the accuracy of the filter coefficients? There is no noticeable difference in the transfer function corresponding to the first 2nd order section even if the filter coefficient precision is reduced from 16 decimal digits to 5 digits. Poles and zeros of the transfer function are not affected much either. The only difference compared to the full precision case is that the double pole at 0.0703 is split in to two distinct poles. These two poles are still in the proximity of 0.070 and generate therefore no problems for the stability.

For this second-order section we do not find any poles near the edge of the z-plane unit circle. We anticipate to find the poles responsible for non-stability while we proceed to other second-order sections. The stability problems are likely to arise when the accuracy of filter coefficients is reduces for the 2nd or 3rd biquad.

6.2.2 Properties of second-order section $H_2^2(z)$

The frequency characteristics for the next transfer function $H_2^2(z)$ are shown graphically in the figure below.

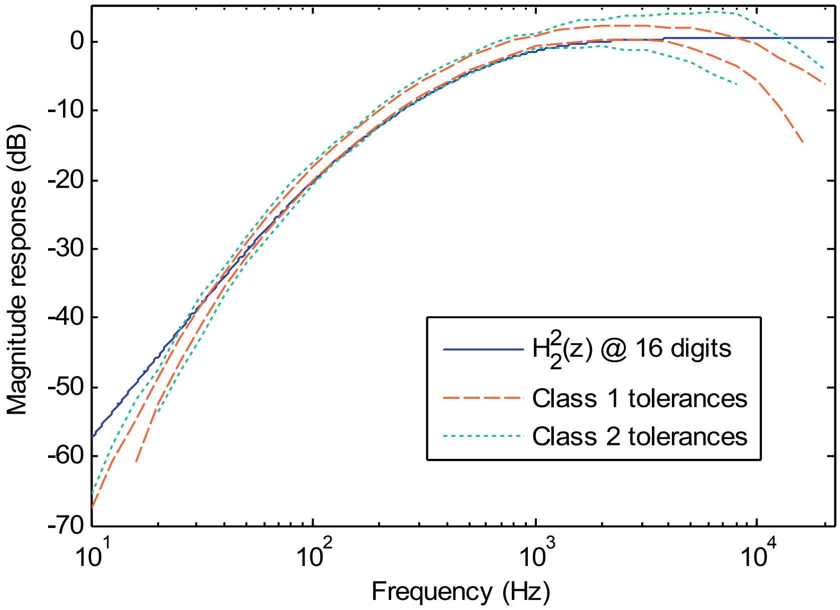


FIGURE 33 Transfer function corresponding to second-order section $H_2^2(z)$

For the second biquad we followed an equivalent procedure as with the first one. Coefficients for it were extracted from the coefficient matrix of the biquads. Conversion from the second-order section representation to a filter representation in the z-domain was performed by a MATLAB function sos2tf(). The gain of the cascade structure was embedded already in the first second-order section and therefore the gain for this 2nd biquad was unity.

As can be seen from Figure 33 this second-order section takes care of attenuation within the frequency range 60 – 2000 Hz. The magnitude response curve for frequencies within this range follows lower tolerance limits of Class 1 and Class 2 specifications. Hence, the second biquad alone would attenuate slightly too much within this range. We can, however, recall from Figure 30 that the intention while decomposing the sixth-order transfer function into separate lower-order parts was to utilize them in a sequence one after another. From Figure 31 we notice that the first second-order section applies a small amplification (~ 1.5 dB) on frequencies below 2000 Hz. When the effects of these two second-order sections are combined the frequency content of input signals in the range 60 – 2000 Hz as well as within 2000 Hz and 20000 Hz is weighted correctly.

The magnitude response was checked above. How about the stability of this second-order section?

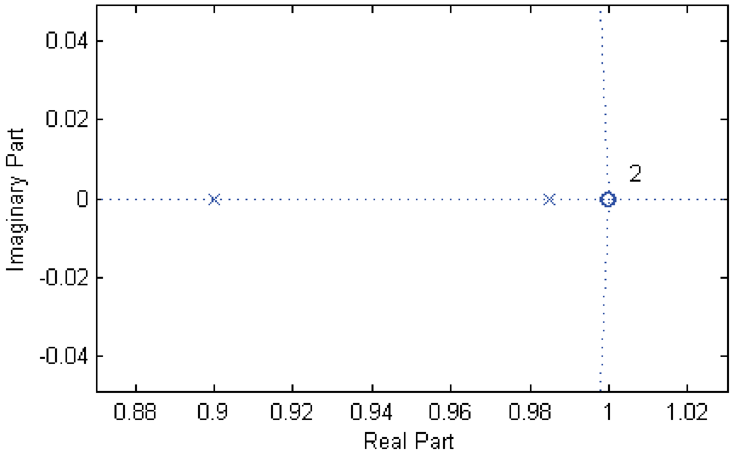


FIGURE 34 Pole-zero plot related to second-order section $H_2^2(z)$

Location of the poles and zeros is not affected when the second biquad is simulated with full coefficient precision. The first pole-zero pair (0.9848, 1.0002) of $H_2^2(z)$ is exactly the same as the pole-zero pair in the original A-weighting filter (cf. third row on Table 5). Likewise the second pole-zero pair (0.9001, 0.9998) corresponds to the original pole-zero pair of the sixth-order transfer function (cf. fourth row). The above pole-zero pairs were as obtained with full 16 digits precision. After rounding the multiplier coefficients to five digits these poles and zeros still agree with the ones given above. Only the zeros move slightly forming a double zero exactly at unity instead of being two separate ones at 1.0002 and 0.9998. Thus, the poles are strictly inside the unit circle and the design of this second-order section is stable. And the magnitude response of it – when combined with that of the first biquad – is acceptable as well. Therefore, the simulations do not reveal problems in the properties of this second-order section.

6.2.3 Problems with second-order section $H_3^2(z)$

While simulating the third second-order section we noticed differences in its characteristics depending on the coefficient precision. With full precision the two pole-zero pairs of the transfer function $H_3^2(z)$ were identical with the counterparts in the monolithic A-weighting filter (see fifth and sixth rows on Table 5). When the multiplier coefficient precision was reduced changes were detected both on the location of the poles and zeros as well as in the magnitude response of this second-order section. Instead of two imaginary zeros in the full precision filter a real double zero (at Re=1) was present in the reduced precision second-order section. An opposite change occurred for the poles of $H_3^2(z)$. A real double pole was transformed to two separate imaginary poles in the reduced precision design (found at 0.9970 ± 0.0064i). These effects are summarized in Figures 35 a) and b).

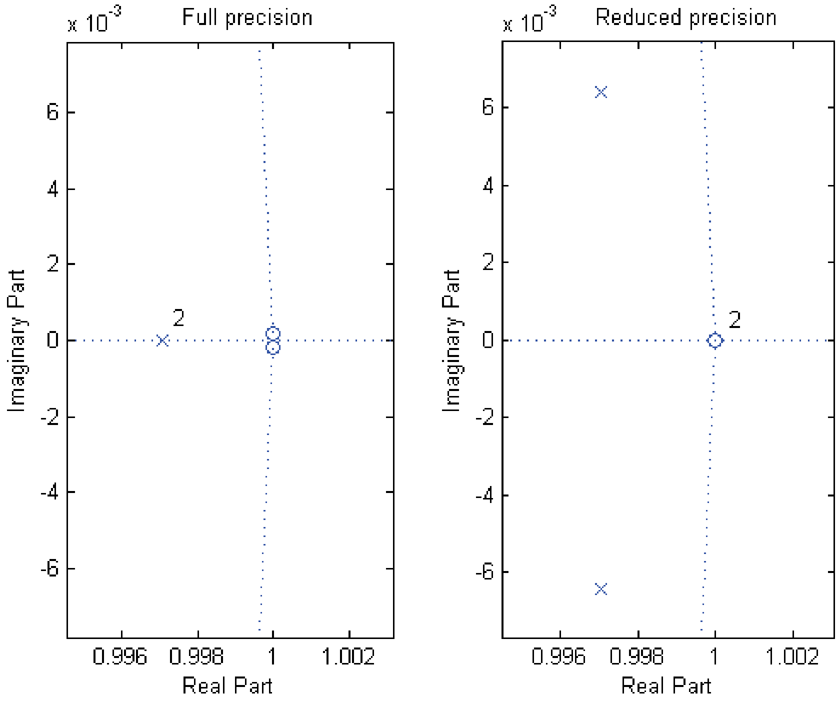


FIGURE 35 Pole-zero plots for transfer function $H_3^2(z)$

a) 16 decimal digits precision b) 5 decimal digits precision

The poles of $H_3^2(z)$ with reduced coefficient accuracy are still kept inside the unit circle. Hence, a stable design is provided, but the frequency response of this transfer function is beginning to deviate from the desired. This can be detected most clearly by taking a look at the frequency response curves of the full and reduced precision designs. These are depicted in Figure 36 which shows the transfer function for the third SOS with two different multiplier coefficient accuracies.

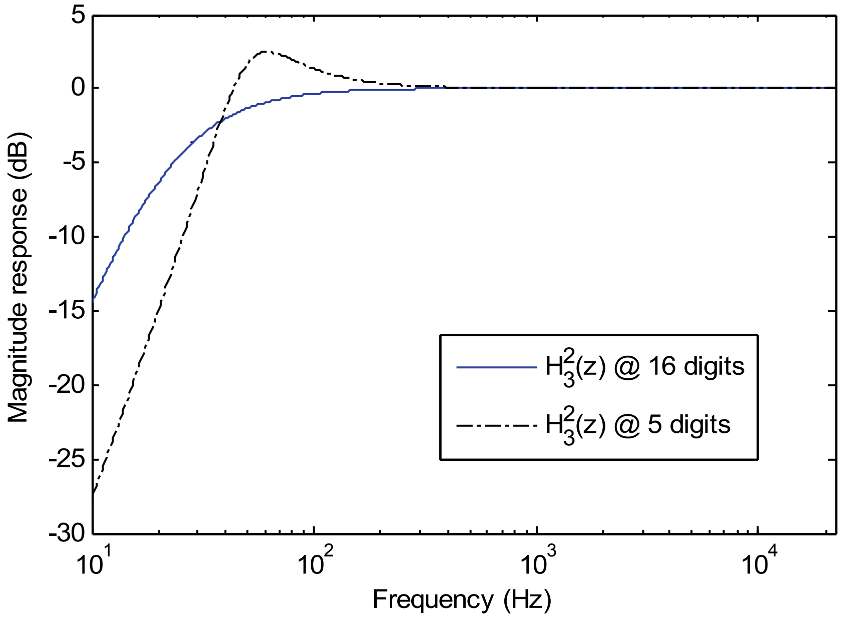


FIGURE 36 Deviation of $H_3^2(z)$ due to reduced multiplier coefficient precision

By comparing the two magnitude responses an additional bump is observed to appear at 40 – 200 Hz in the transfer function obtained with 5 decimal digits precision. While the frequency decreases below 40 Hz the response of this design drops more rapidly compared to the full precision design.

As a conclusion we can state that the transfer functions of the two first biquads do not change, but the third one is affected. It suffers most severely if the coefficient precision of the second-order sections is reduced from 16 to 5 decimal digits.

6.3 Combining second-order sections

In Section 6.2 we have seen the effects of reducing the filter coefficient precision from 16 decimal digits to 5 digits on the individual second-order sections. Earlier in Section 6.1 we also realized that this kind of reduction directly on the transfer function of the complete A-weighting filter leads to unacceptable magnitude response. What happens if, instead of reducing the multiplier coefficient precision of the sixth-order rational function directly, we generate a combined transfer function by cascading three biquads specified with this reduced accuracy? Is it going to result in as catastrophic frequency response as earlier?

The steps needed to investigate the characteristics of a cascade structure of second-order sections are as follows. The sixth-order rational function designed for the complete A-weighting filter is transformed to a second-order section representation. The transformation process provides us with a coefficient matrix for the three biquads. The coefficients obtained with accuracy of 16 decimal digits are rounded to precision of 5 decimal digits using the variable precision arithmetic function in MATLAB. A reverse conversion from the reduced precision filter coefficient matrix back to the z-domain is carried out to find out a combined transfer function. As a result of this procedure we obtain a reduced precision design describing the characteristics of the cascade structure of three second-order sections.

Rounding of the filter coefficients to 5 decimal digits deteriorated the combined transfer function behavior between 40 – 100 Hz. For these frequencies the cascade of three second-order sections violates the A-weighting specifications. Besides of this too strong attenuation is applied to the very lowest frequencies within 10 – 30 Hz.

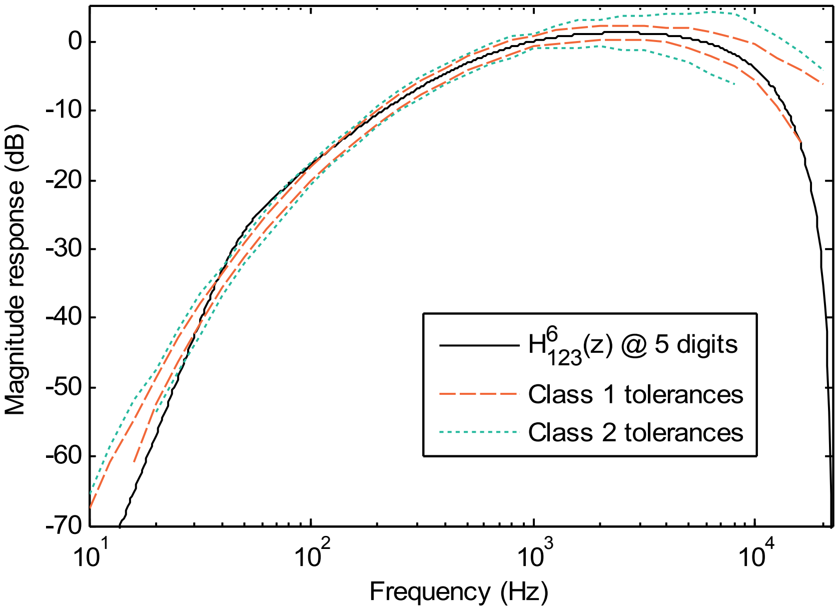


FIGURE 37 Frequency response for a cascade of three SOS with reduced accuracy

What we noticed already earlier was that the influence of the 3rd second-order section is limited to the lowest frequencies in the range of 10 – 200 Hz. Therefore, it is tempting to check how the overall transfer function performs if only two first biquads are combined. This can be done by leaving the last row of the reduced precision SOS coefficient matrix out before starting conversion back to a combined transfer function. In Figure 38 the magnitude responses are shown for the full precision A-weighting design and the combination of two reduced precision second-order sections (for clarity only upper tolerance limits of Class 1 and Class 2 specifications are marked). While comparing the transfer functions no noticeable difference can be detected from 70 Hz upwards. Below this limit the combined effect of two reduced accuracy second-order sections begins a slowly increasing overestimate of the frequency response.

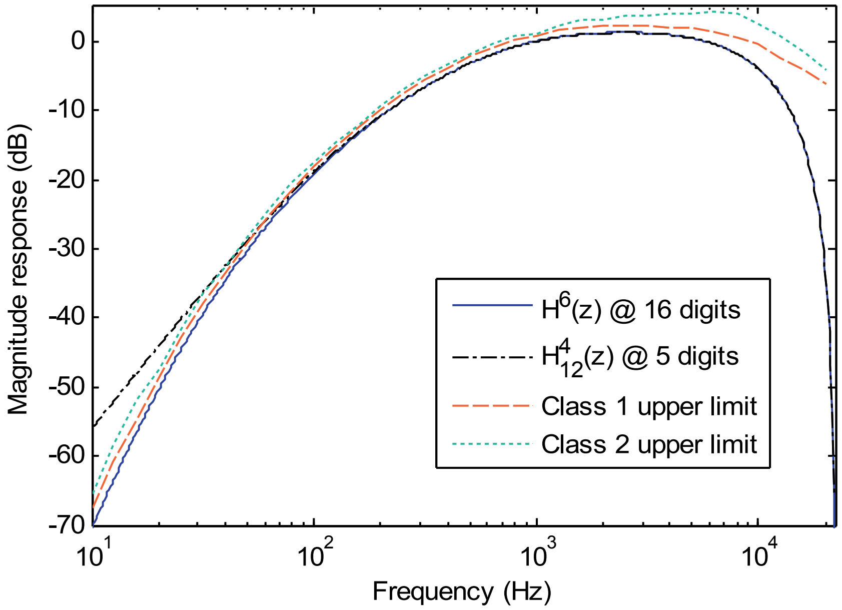


FIGURE 38 Magnitude response for $H^6(z)$ and reduced precision $H_{12}{}^4(z)$

How severe is this 'leakage' of lowest frequencies through the intended A-weighting filter? From Section 1.1 we recall that environmental noise emissions from four-stroke combustion engines fall typically on the frequency range 50 – 100 Hz. From the frequency spectrum of KirriA (Figure 18) we realize that the engine noise below 100Hz is originally extremely energetic if no frequency weighting is applied. The dynamic range of human auditory system begins at 20 Hz and therefore the human ear is not very sensitivity for these lowest frequencies. At this end of the hearing range the capability of the human peripheral auditory system to transfer energy from the outer ear via the hearing bones and the scala vestibuli to the apex of the cochlea is already heavily diminished. Furthermore, the inner hair cells, which are mainly responsible for hearing, respond weakly to these low frequencies. Therefore, the human ear applies a strong attenuation on this range, which makes the situation look like if these low frequencies would only pose a modest problem. These frequencies below 100 Hz, however, are already on the boundary between sound and vibration. Even if the human auditory system is not capable of detecting environmental noise effectively on this frequency range other pathways of influence may exist. Besides of inner hair cells, for example, we have outer hair cells inside the human cochlea. These latter hair cells possess a 10 – 20 dB higher sensitivity on the 20 – 100 Hz frequency range compared to the sensitivity of human hearing [Sal11]. Hence, even inaudible sound may stimulate outer hair cells and evoke neural impulses, which propagate further via nerve fibers to non-auditory centers of the brain causing various types of physiological responses. Another possible pathway for low-frequency sound to interfere with the human physiology is via resonances with internal organs of the human body. Recently more evidence has been presented that sound is not perceived only through the auditory pathway, but it can affect also non-auditory human cells [Les13].

The response of the transfer function $H_{12}{}^{4}(z)$ within the range 20 – 100 Hz is an overestimate regarding the sensation of hearing. We argue, however, that it would be sensible to select this combined transfer function instead of the one show in Figure 37. Underestimating the influence of environmental noise containing these low frequencies would be a mistake. A slight overestimate prepares for potential harmful effects caused by absorption of vibrational energy on human body. For this reason, we prefer the combined transfer function of the two first biquads with reduced precision coefficients. Furthermore, we claim, that the five decimal digits accuracy for the filter multiplier coefficients is adequate for implementing the A-weighting filter using these two first second-order sections.

6.4 Summary of simulations

In this chapter we simulated the A-weighting filter design to find out the influence of coefficient precision on the filter properties. We first showed that the original design obtained by bilinear transformation of an analog prototype filter is stable. The transfer function of this design was also verified to fulfill the requirements of the Class 1 and Class 2 specifications posed by the IEC/CD 1672 Electroacoustics – Sound level meters standard. We then proceeded by reducing the accuracy of filter coefficients directly in the sixth-order transfer function related to the original design. We observed that in this method the quantization affects the design too much and it becomes unstable. Furthermore, the aforementioned specifications are violated by direct reduction of the coefficient accuracy.

We then proceeded on to decompose the sixth-order rational function into three lower order parts. One by one the individual biquads were checked to determine the influence of quantization. All of the resulting second-order sections were shown to be stable. Stability of the design was preserved for all three biquads even if the coefficient accuracy was reduced from 16 to 5 decimal digits. A problem, however, was noticed on the properties of the third biquad. With reduced accuracy its transfer function deviated from the desired one. It was judged that inclusion of this second-order section in filtering would be unacceptable.

Final simulations were run to find out filtering properties for a combination of two first biquads. It turned out the combined effect overestimates the frequency response slightly at frequencies below 100 Hz. Viewing this from the environmental noise perspective and human physiology we regarded it to be far better alternative than the three biquad one. The latter overestimates the response more within the range 40 – 100 Hz and underestimates it while the frequency drops below 30 Hz.

As a consequence of the simulations performed the cascade structure of two first second-order sections was found to best match the requirements of those applications we envision for the filtering algorithm.

7 IMPLEMENTATION OF A-WEIGHTING ON ARM

The A-weighting filter design was started in Chapter 5 from the analog prototype filter which was specified in continuous-time as an s-domain system function. Using bilinear transformation we converted the analog prototype into an equivalent digital filter. The properties of this filter described in the z-domain were simulated in Chapter 6. The actual implementation of the A-weighting filter requires us to transform the design back to the time domain, because on embedded systems we must process time histories of sound pressure values. We therefore begin this chapter by presenting the required steps to obtain a discrete-time algorithm corresponding to the sixth-order rational transfer function $H^6(z)$ of the A-weighting filter. Simulations conducted in Chapter 6 did show that direct form implementation of this transfer function with reduced precision coefficients and data becomes unstable and does not fulfill the A-weighting specifications. Outcome from these simulations was that a cascade structure composed of lower-order components would be preferable from the theoretical point of view. In this chapter are going to compare these two implementation alternatives still one more time from a slightly different viewpoint. A practical realization viewpoint we take should convince the reader that direct form implementation is definitely not a viable alternative on an embedded systems platform.

For the actual A-weighting implementation we selected an ARMv4 architecture. This architecture is utilized in ARM7 processor family, which is still the most popular one of the ARM processor families. In 2012 altogether 8.7 billion ARM processor-based silicon chips were sold, out of which ARM7 accounted for 36 percent [ARM12].

7.1 A-weighting Filter Description in Time Domain

Using MATLAB we were able to design a digital filter the frequency response of which fulfills the requirements of Class 1 and Class 2 specifications of the IEC/CD 1672 electroacoustics sound level meters standard. The MATLAB design gives the frequency response $H^6(z)$ in a form of a sixth-order rational function. Both the numerator and denominator polynomials $B(z)$ and $A(z)$, respectively, are ascending powers of the unit delay z^{-1} as shown below

$$H^6(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}+b_6z^{-6}}{a_0+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}+a_5z^{-5}+a_6z^{-6}} \tag{7.1}$$

Without lack of generality the numerator and denominator coefficients can be normalized so that $a_0=1$. By definition the transfer function $H^6(z)$ of the A-weighting filter is obtained as a ratio of Z-transforms of its output and input sequences. Therefore, the following relationship applies

$$H^6(z)=\frac{Y(z)}{X(z)} \tag{7.2}$$

By combining the above two equations and rearranging the terms we obtain

$$\begin{gathered}(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_5z^{-5}+a_6z^{-6})Y(z)=(b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_5z^{-5}+b_6z^{-6})X(z)\\ \Rightarrow\ Y(z)+a_1z^{-1}Y(z)+a_2z^{-2}Y(z)+\cdots+a_5z^{-5}Y(z)+a_6z^{-6}Y(z)=\\ b_0X(z)+b_1z^{-1}X(z)+b_2z^{-2}X(z)+\cdots+b_5z^{-5}X(z)+b_6z^{-6}X(z)\end{gathered} \tag{7.3}$$

It is possible to bring this equation back to the time domain by utilizing an inverse Z-transform and the translation property $(Z\{y[n-n_0]\}=z^{-n_0}Y(z))$ of the Z-transform [Nie09]

$$\begin{gathered}y[n]+a_1y[n-1]+a_2y[n-2]+a_3y[n-3]+a_4y[n-4]+a_5y[n-5]+a_6y[n-6]=\\ b_0x[n]+b_1x[n-1]+b_2x[n-2]+b_3x[n-3]+b_4x[n-4]+b_5x[n-5]+b_6x[n-6]\end{gathered} \tag{7.4}$$

From the above equation we observe that the A-weighting filter is characterized by a linear constant coefficient difference equation of the sixth order. Equation (7.4) can also be presented in the following notation by leaving only the current filter output on the left hand side of the equation

$$y[n]=\sum_{j=0}^{6}b_jx[n-j]-\sum_{i=1}^{6}a_iy[n-i] \tag{7.5}$$

The first part on the right hand side of the above equation is recognized to be dependent only on previous six input samples to the A-weighting filter. If this equation would consist only of the first sum it would mean the filter would be lacking poles altogether. Thus, the filter would be an all-zeros design which would calculate the outputs as moving averages of the environmental noise input samples. Filtering would be an MA process which could be implemented as an FIR filter. The second sum, however, brings in the poles to the design and turns it into an autoregressive moving average (ARMA) filter. An infinitely long impulse response of this recursive algorithm reminds us we are dealing with an IIR filter.

What kind of requirements exist for a direct form implementation of the above mentioned linear constant coefficient difference equation corresponding to $H^6(z)$ of the A-weighting filter? From Equations (7.4) and (7.5) we realize that the present filter input, six previous filter inputs as well as six previous filter outputs are needed while generating new filter outputs. The total number of input samples and previous outputs involved in computations is thus 13. Each of these parameters also requires a coefficient a_i or b_j with which it is weighted. Therefore, at least 26 storage elements should be available for calculating new values for $y[n]$. Preferably these should be CPU registers to enable an efficient realization of the filter design. In the following sections we will check which possibilities exist for this on an embedded systems platform.

7.2 Implementation Issues on Embedded Systems

While simulating the properties of the A-weighting filter by MATLAB the precision of the filter coefficients is of no concern. MATLAB is running simulations easily with 64-bit wordlength on nowadays' PCs and workstations. On embedded systems platforms, however, 64-bit words are hardly ever available for implementing desired applications. This calls for reduction of accuracy for constants, variables and parameters required to implement filtering on an embedded system. This reduction of precision would be desirable from embedded systems' point of view. On the other hand, however, reducing the accuracy in calculations may lead to problems related to algorithmic aspects. After a certain point reduction of coefficient precision makes quantization effects grow too much causing the digital design to become unstable or deviate from the desired filter specifications. Therefore, optimization is required in order to find a sensible compromise between the rivaling requirements.

ARM processor platform provides a 32-bit register architecture. Therefore, one might wonder why a 16-bit precision is desirable for filter coefficients and samples data. For example in the algorithm for A-weighting multiplications of filter coefficients and environmental noise sample values are required. Results of these multiplications are further summed up with previous product terms to generate final filtered output for the A-weighting algorithm. Suppose we would store both the filter coefficients and the input sample values in 32-bit words. For this purpose there should be 26 registers of 32-bit length available as mentioned at the end of the previous section. By generating product terms like $b_0 x[n]$ in the filtering algorithm of Equation (7.4) we would create 64-bit long intermediate results. If a pair of 32-bit registers would be allocated to store these temporary variables two additional internal registers would be required. This is clearly impossible on ARM v4 and v5TE architectures where the total number of 32-bit registers is 16.

One basic operation in DSP algorithms is an instruction called multiply accumulate where two register values are multiplied together and the result is added to a third register (see Figure 39). In case the filter coefficients and samples data can be squeezed into 16-bit precision we are able to use a 32-bit long register to store the multiply accumulate results. In the DSP extensions of the ARM assembly instruction set these kind of multiply accumulate operations can be executed efficiently. For this purpose the ARMv5TE architecture provides instructions which enable a 16-bit by 16-bit multiply with a 32-bit accumulate to be executed in one CPU cycle. On a 200 MHz embedded CPU execution of this kind of multiply accumulate instruction (MLA) would take only 5 ns. Therefore, it would be extremely beneficial to develop algorithms capable of running with 16-bit precision allocated for the filter multiplier coefficients and environmental noise samples.

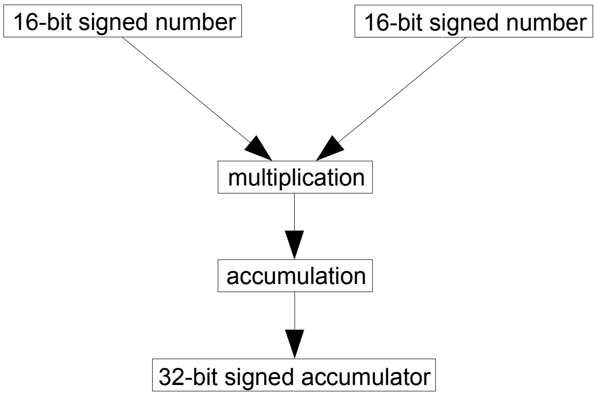


FIGURE 39 Multiply accumulate instruction of two 16-bit numbers

One design goal for embedded systems is to achieve ultra low power implementations. This desire origins from typical applications of embedded systems which must provide operation for long periods of time without human intervention (e.g., the distributed system for long-term monitoring of environmental noise mentioned in Chapter 1). Against this background every CPU cycle counts. Without doubt optimizing high level language compilers are able to produce efficient code which is more readable and much easier to maintain than assembly code. There are, however, certain drawbacks that cannot be avoided while using high level languages. One of these is the lack of control for instruction and data pipeline of the CPU. For instance on the ARM architecture the CPU may stall and introduce wait states in case values loaded from memory are used directly after a load instruction. In some architecture versions these wait states can last two CPU cycles.

While writing assembly code we have the possibility to schedule the load commands early enough so that they have time to terminate before the loaded values are actually accessed. This prevents the CPU from stalling and speeds up the execution of an algorithm. Consider for example the A-weighting which must process all of the samples obtained from frontend electronics digitizing analog environmental noise signals. Removal of wait states is important while striving towards lower power consumption. Without extra wait states the CPU manages to process a bunch of samples faster which provides a possibility for the CPU to sleep between filtering sample sets. By utilizing the CPU sleep the power consumption can be reduced.

7.3 Representation of Samples Data

Digitized samples representing sound pressure fluctuations recorded at Kirri measurements were originally stored in floating-point format. As can be seen in Figure 40 the maximum amplitude for these fluctuations is normalized to an absolute value of one.

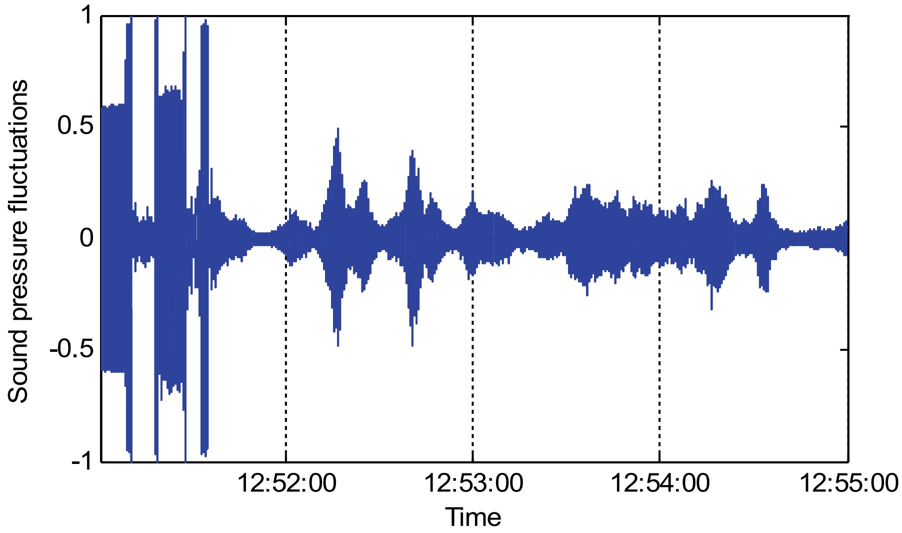


FIGURE 40 Digitized samples sequence for KirriC measurement

Finally we aim to implement digital processing of environmental noise signals on an embedded system. Typically these systems offer a more limited selection of choices for handling various data types in comparison to higher end computing systems. Quite seldom a floating-point representation exists as a hardware implementation on embedded processors. For example ARM processors provide only an integer data type for representation of signals. Therefore, an embedded system's developer must rely on the integer data type. To facilitate efficient processing of more complex data types several workarounds have been developed. Among these we have a so called Qk fixed-point representation which can be used instead of floating point representation. Fixed point arithmetic is the most efficient way of calculating with floating point numbers on ARM because the architecture does not have floating point acceleration in hardware. In the following we will select a suitable Qk number system for digital data representation of the samples data sets recorded at Kirri.

While selecting a digital representation for our measurement data (i.e., digitized samples of environmental noise) we should consider requirements for dynamic range and accuracy. In environmental noise measurements the sound pressure fluctuates between $20\mu Pa$ and $20 Pa$. This introduces a dynamic range of $1:1000000$ to be detected by a microphone. First we must estimate how big errors are allowed while using digitized sample values in the computations.

Mapping of sound pressure values into the sound pressure level scale is done according to the equation below

$$L_p = 10 \cdot \log_{10}(\frac{p^2}{p_0^2}) \tag{7.6}$$

In the above equation p is an instantaneous sound pressure and the reference sound pressure $p_0 = 20 \mu Pa$ corresponds to the threshold of hearing. The level of the sound pressure L_p is expressed in decibel.

As an example we can consider a situation where the measured sound pressure equals $20 \mu Pa.$ It is represented with sound pressure level of $0 dB$

$$L_p = 20 \cdot \log_{10}(\frac{20 \mu Pa}{20 \mu Pa}) = 20 \cdot \log_{10}(10^0) = 0 dB \tag{7.7}$$

The threshold of pain is exceeded by an instantaneous sound pressure of $20 Pa$ which maps to an $L_p = 120 dB$

$$L_p = 20 \cdot \log_{10}(\frac{20 Pa}{20 \mu Pa}) = 20 \cdot \log_{10}(10^6) = 120 dB \tag{7.8}$$

Therefore, the dynamic range required to cover sound pressure levels from the threshold of hearing up to the threshold of pain is $120 dB$ if expressed in decibel.

Errors commonly encountered in environmental noise measurements are of the order of ± (2 – 3) dB [Lah95]. In respect to total assessment errors on this level an accuracy of ± 1 dB can be regarded as acceptable for digital representation of environmental noise samples. Converting this accuracy demand backwards into sound pressures we obtain an absolute allowable error $|p_{err}|$ expressed in physical quantity

$$\begin{aligned} L_p &= 20 \cdot \log_{10}(\frac{|p_{err}|}{p_0}) = 1 dB \\ \Rightarrow \log_{10}(\frac{|p_{err}|}{p_0}) &= 1/20 = 0.05 \\ \Rightarrow |p_{err}| &= p_0 \cdot 10^{0.05} = p_0 \cdot 1.122 = 22.44 \mu Pa \end{aligned} \tag{7.9}$$

The allowable error that we tolerate was $\pm |p_{err}|.$ Therefore, it is adequate to quantize sound pressure values so that the maximum distance between two successive values in the representation of samples data is less than $2 \cdot |p_{err}|.$ Thus, the allowable error of the sound pressure becomes $e_p = 44.88 \mu Pa.$ In case we would select this maximum distance as a basic unit for discretization, the dynamic range of 20 Pa (= 120 dB) would be divided into 445632 slices. Hence, 2^{19} (i.e., 524288) numbers would be required for our digital representation of environmental noise samples. Clearly, this is impossible with 16-bit words. Fortunately, quite seldom environmental noise assessments request the complete range from 0 dB to 120 dB to be covered in a single measurement. For example along roads and streets the background noise exceeds 30 or 40 dB levels most of the time. Therefore, the measurement range can be adjusted to $30\,dB < L_p < 110\,dB$ or $40\,dB < L_p < 120\,dB.$ By limiting the measurement range in this manner the dynamic range demand reduces to 80 dB. With this practical dynamic range requirement the above accuracy estimation can be repeated. First we convert the dynamic range required for measurements from decibels to sound pressure

$$\begin{aligned}
&L_p = 20\cdot\log_{10}\left(\frac{p_{dyn}}{p_0}\right) = 80\,dB \\
&\Rightarrow \log_{10}\left(\frac{p_{dyn}}{p_0}\right) = 4 \\
&\Rightarrow p_{dyn} = p_0\cdot 10^4 = 20\cdot 10^{-6}\cdot 10^4 = 0.2\,Pa
\end{aligned} \tag{7.10}$$

While evaluating the accuracy demand for the digital representation we found out that the allowable error for the sound pressure is $e_p = 44.88\,\mu Pa.$ We can therefore select a representation were the LSB bit represents a basic unit of $40\,\mu Pa\ (< e_p).$ After this selection we should calculate an integer range that must be available for computations. In order to represent measured sound pressure fluctuations that span a dynamic range of $p_{dyn} = 0.2\,Pa$ using units of $40\,\mu Pa$ we require 5000 integer values. Hence, there is a need for a representation where at least 5000 distinct sound pressure values can be expressed.

The most convenient representation regarding a digital implementation is the one which divides the measurement range in 8192 basic units. Therefore, a sensible choice for coding of environmental noise samples is a Q13 fixed point format. Values for the fixed-point computations are obtained from the digitized environmental noise samples by multiplying them with a factor 8192. The sound pressure fluctuations around the equilibrium (i.e., atmospheric pressure) may be either positive or negative. By assuming a normalization $-1\,V < U_p < +1\,V$ for electrical signals representing these pressure fluctuations (as shown in Figure 40) a range of integers between -8192 and 8191 must be available. Therefore, the Q13 representation of environmental noise samples occupies 14 bits of storage (1 bit for 2's complement sign and 13 bits for the actual data).

By allocating 16-bit words to store environmental noise samples for computations a reserve of two bits exists at the MSB end of the storage element while storing original samples. This headroom makes simple additions and multiplications possible without a danger of an overflow. A Q13 value can be multiplied by four without corrupting the sign bit. Or adding four Q13 values together is possible still staying within the headroom. If necessary some intermediate results may be twice or four times bigger than the maximum of Q13. The final result must, of course, be brought back to the allowed scale.

In practice the mapping from the original floating-point representation into Q13 can be done as follows. Suppose the sound pressure detected at a given time t_1 by a microphone was -0.2 Pa – that is, $x(t_1) = -0.2\,Pa.$ After digitizing and converting the original measurement signal we store the sample as $x[t_1] = -8192.$ Correspondingly the digital representation for $x(t_2) = 0.1\,Pa$ would be $x[t_2] = 4096.$

7.4 Reducing Coefficient Precision

In Chapter 6 the MATLAB simulations were carried out using floating point representation. Decimal digits were therefore used as a reference to precision of the filter coefficients. In this section we consider the implementation of A-weighting filter algorithms on an embedded hardware platform. Hence, these precision references are going to be specified in bits instead of decimal digits.

The original coefficients in the sixth-order rational transfer function describing the A-weighting filter were given with a precision of 16 decimal digits. This maximum accuracy for floating point computations in MATLAB is achieved using a 53-bit mantissa with an 11-bit exponent [Qua12]. A direct implementation of the A-weighting filter would therefore require 64 bits to be allocated for each coefficient. Both the numerator and denominator of the rational transfer function contains seven coefficients. This would call for 14 storage elements of 64-bit quadwords. The filter coefficients alone would therefore require 28 registers of 32-bit length. In addition of this extra registers should be available for input samples, filtered outputs, etc. Scarce resources on embedded systems seldom provide that much register capacity for applications and hence it is a necessity to reduce the precision of the coefficients.

Our aim regarding the A-weighting filter coefficients is to pack them into 16-bit words. The filter coefficients may have either positive or negative sign which leaves 15 bits available for the actual numeric values for coefficients. The inspection of the filter coefficients after factoring the original transfer function design to second-order sections reveals that the maximum absolute value among the coefficients is slightly less than 2. In a similar way as with the environmental noise samples we want to express also these coefficients in fixed point format. While converting the floating point representation of the filter coefficients to fixed point representation we are able to multiple the maximum absolute value by 2^{14} (=16384) without exceeding the available 15 bits storage space. Selecting Q14 representation to be used for fixed point numbers the converted value of the negative coefficient in question becomes -32672. Because the smallest negative number that can be represented as a 16-bit two's complement integer is -32768 we are able to fit this coefficient in 16-bit register on the target platform. Q14 representation is fine for other coefficients as well, because their absolute values are smaller. To be more precise, the fixed point type chosen is effectively a Q15 representation. Even if the filter coefficients are converted from floating point to fixed point format by multiplying them with 2^{14} one of them occupies all 15 magnitude bits as was explained above.

In Chapter 6 we came to a conclusion that 5 decimal digits (corresponding to 16 bits) precision for the filter coefficients is adequate for an acceptable magnitude response. What happens if we must squeeze additional two bits out of the accuracy? Here we must carry out a worst-case analysis and reduce the available accuracy by the sign bit and one magnitude bit, because this reduction applies to majority of the filter coefficients. From the storage element point of view the Q14/15 fixed point format is a possible choice as we just saw. How much worse the situation regarding the magnitude response gets? Do we still have a stable filter if we approximate the original 53-bit long filter coefficients by 14-bit fixed point numbers? To check up on this we proceeded as follows.

- A-weighting filter's transfer function $H^6(z)$ was decomposed to second-order sections
- breaking up the complete transfer function resulted in three parts specifying one low-pass filter and two high-pass filters
- the low-pass filter together with that high-pass filter responsible for attenuating the mid-frequency range were selected
- conversion of the coefficients from floating point to fixed point was started by multiplying them with 2^{14}
- the filter coefficients were rounded to a precision of five decimal digits corresponding to 16 bits
- the coefficients were converted to 16 bit integer data type
- final coefficients were obtained in Q14 fixed point format to be used in ARM assembly implementation
- a reverse conversion was applied in order to check how much rounding errors affect the magnitude response of the filter
- division with 2^{14} gave floating point coefficients for the first two biquads which correspond to rounded Q14 fixed point coefficients
- the second-order section representation was transformed back to a z-domain transfer function
- the resulting transfer function corresponding to the Q14 implementation was plotted against the A-weighting filter specifications

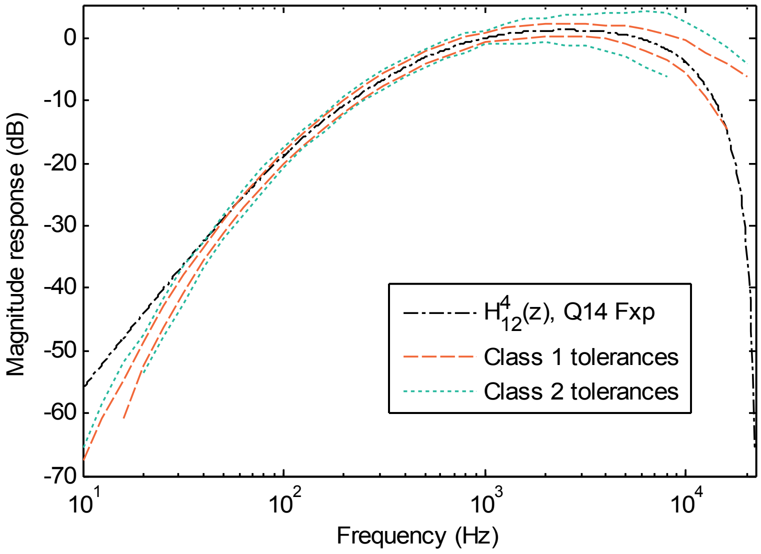


FIGURE 41 Magnitude response of A-weighting filter with Q14 fixed point coefficients

From the above figure the influence of filter coefficient quantization on the frequency response can be seen. Actually, this figure shows us the combined effect of coefficient accuracy reduction and drop-out of the third second-order section. No significant differences can be found while comparing Figures 41 and 38. Hence, we can say that quantization to 14 bits is not an issue.

We have thus validated the use of the Q14 fixed point representation for the filter coefficients in the A-weighting filter implementation. With these coefficients the filter fulfills the Class 1 and Class 2 specifications down to 60 Hz and 50 Hz, respectively.

7.5 Cascade Structure of A-weighting Filter

Theoretical simulations in Chapter 6 as well as practical implementation aspects presented in earlier sections of this chapter have pointed out disadvantages in direct form implementation of our original A-weighting filter design. Among these issues we have insufficient register capacity and restrictions on the wordlength that can be allocated for in-register variables on the ARM architecture. These problems were not present in MATLAB simulations, which were run using floating point numbers with a precision high enough to guarantee accurate results. For implementation purposes the environmental noise samples data and the filter coefficients must be quantized to fixed point representation with 13 and 15 bits precision, respectively. Reduction of precision from the original 53 bits mantissa in MATLAB introduces quantization errors on the software implementation. For higher-order IIR filter designs like our sixth-order transfer function the quantization errors start to accumulate and will have a negative influence on the filtered output. A more sensible approach for the software implementation is to factorize the higher-order transfer function to several lower-order sections. These lower-order sections can be used as building blocks of a cascade structure in a filter realization. Therefore, the original transfer function given by the sixth-order rational function $H_{Aw}^{6}(z)$ should be transformed to second-order sections. This rational transfer function consists of two sixth-order polynomials: a numerator $B_{Aw}^{6}(z)$ and a denominator $A_{Aw}^{6}(z)$. The aim of this decomposition is to split $H_{Aw}^{6}(z)$ into three 2nd-order sections $H_k^2(z)$ with quadratic polynomials (i.e., $B_k^2(z)$ and $A_k^2(z)$). After factoring the original transfer function this way it is replaced by a cascade structure of second-order parts as shown below

$$X(z) \rightarrow H_1^2(z) \cdot H_2^2(z) \cdot H_3^2(z) \rightarrow Y(z) \tag{7.11}$$

A common practice applied while factoring higher-order transfer functions to lower-order parts is to extract a gain coefficient out from the sequence of second-order sections. Including the gain in the equation and expressing it in the product form we have

$$H_{Aw}^{6}(z) = g \prod_{k=1}^{3} H_k^2(z) = g \prod_{k=1}^{3} \frac{B_k^2(z)}{A_k^2(z)} = g \prod_{k=1}^{3} \frac{\sum_{j=0}^{2} b_{jk} z^{-j}}{\sum_{i=0}^{2} a_{ik} z^{-i}} \tag{7.12}$$

By writing out summations in the numerator and denominator of the last term this expression can be converted to

$$H_{Aw}^{6}(z) = g\prod_{k=1}^{3}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{a_{0k}+a_{1k}z^{-1}+a_{2k}z^{-2}} \tag{7.13}$$

Based on the simulations performed in Chapter 6 we know that quantization effects can be reduced further by dropping the last second-order section. For an A-weighting filter approximation on ARM we therefore have two second-order sections yielding in a cascade with two quadratic terms. To facilitate the actual software implementation the transfer functions for both of the second-order sections needs to be transformed back to the time domain. Let us begin with $H_1^2(z)$ and denote the output of this first part of the cascade by $W(z)$. At this point we can also normalize the coefficients so that the first denominator coefficient $a_{01}=1$. By identifying an expression for the first second-order section from Equations (7.12) and (7.13), using the aforementioned notation and the transfer function definition we get

$$H_1^2(z) = \frac{W(z)}{X(z)} = \frac{b_{01}+b_{11}z^{-1}+b_{21}z^{-2}}{1+a_{11}z^{-1}+a_{21}z^{-2}} \tag{7.14}$$

An inverse Z-transform takes us to the time domain giving a discrete-time expression for the intermediate output $w[n]$

$$w[n] = b_{01}x[n] + b_{11}x[n-1] + b_{21}x[n-2] - a_{11}w[n-1] - a_{21}w[n-2] \tag{7.15}$$

Realization of this difference equation can be illustrated by a structure shown below

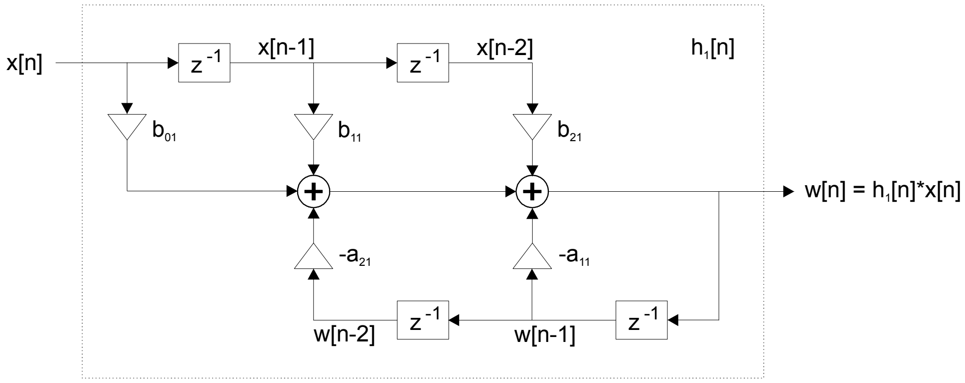


FIGURE 42 Filter structure corresponding to second-order section $H_1^2(z)$

Following an equivalent procedure as with the first biquad a filter structure corresponding to $H_2^2(z)$ can be developed. For this second stage the intermediate output of the first stage $W(z)$ is used as an input sequence.

$$H_2^2(z) = \frac{Y(z)}{W(z)} = \frac{b_{02}+b_{12}z^{-1}+b_{22}z^{-2}}{1+a_{12}z^{-1}+a_{22}z^{-2}} \tag{7.16}$$

The final time domain output $y[n]$ is obtained by multiplying $W(z)$ and the denominator of the rational function through the above equation, using the translation property $(Z\{y[n-n_0]\}=z^{-n_0}Y(z))$ of the Z-transform and applying the inverse Z-transform on this equation

$$y[n] = b_{02}w[n]+b_{12}w[n-1]+b_{22}w[n-2]-a_{12}y[n-1]-a_{22}y[n-2] \tag{7.17}$$

In the graphical representation of the second stage the input and output sequences as well as the filter coefficients differ in comparison to the first stage.

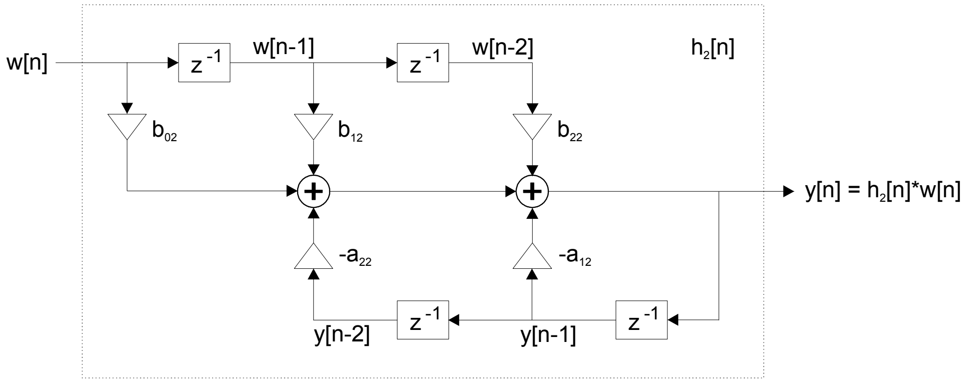


FIGURE 43 Filter structure corresponding to second-order section $H_2^2(z)$

By combining the two second-order sections shown above to a cascade structure the time domain output $y[n]$ can be expressed as a convolution involving impulse responses of these biquads and the original input sequence.

$$y[n] = h_2[n]*w[n] = h_2*(h_1[n]*x[n]) = (h_2[n]*h_1[n])*x[n] \tag{7.18}$$

How this discrete time representation of the filter output is related to a description in the z-domain? By solving Equation (7.16) for the intermediate output $W(z)$, substituting it in Equation (7.14) and expressing the filter output $Y(z)$ using the transfer functions of the second-order sections and the filter input $X(z)$ we have

$$Y(z) = H_2^2(z)\cdot H_1^2(z)\cdot X(z) \tag{7.19}$$

In Equation (7.18) we have expressed the output of the cascade filter structure in the time domain. Cascade of these second-order sections forms the base for the actual implementation on an embedded system. Equation (7.19) shows how we obtained the filter output during simulations performed in the z-domain. By comparing these two equations we realize that the latter one is obtained from the first one by applying a Z-transform on it. So they represent equivalent filtering processes in two different domains. Hence, we have proved that A-weighting filtering can be implemented in the time domain by utilizing structures shown in Figures 42 and 43.

7.6 Implementation on ARM

Hardware solutions for digital signal processing applications on embedded systems can be divided roughly in two categories. Traditionally hardware for these purposes has been designed using multicore techniques on an embedded board. A combination of a general purpose microcontroller and a separate DSP processor has been used. The microcontroller has been appropriate for implementing the control part of an application involving interaction with the user. Microcontrollers, however, lack the processing power required to run algorithms for digital signal processing tasks of embedded applications. This requirement has necessitated allocating an additional DSP processor in the hardware implementation. A dedicated DSP processor provides of course plenty of processing power, but there is a price to pay for it. Using two processor cores complicates the implementation of the system: both the hardware and software part. Synchronization between the microcontroller and the DSP processor on the hardware level as well as message passing between the processes running on separate CPUs needs extra attention in the system design. Besides of this the power consumption increases due to two CPU cores of which especially the DSP one is power hungry.

Another category of embedded systems' hardware solutions, which has emerged more recently, involves only a single microprocessor. Modern RISC processors provide already a reasonable processing power for running DSP algorithms as well. Efficient software implementations based on them, however, require coding in assembly language, which is a downside of this alternative. Advantages of the single-core solution include simplicity of implementation. Also ultra low power designs are achievable by using a single embedded processor.

The development of digital algorithms as presented in this work are targeting to distributed systems like Dimeca described in Chapter 1. For an implementation platform the ARM architecture has been chosen based on the reasons mentioned above. An additional benefit in favor of ARM is a clean and logical assembly language provided by this processor architecture. From the implementation point of view the essential features of the ARM architecture are an orthogonal register structure, a five to seven stage instruction and data pipeline, an efficient instruction set architecture and a barrel shifter.

Altogether 16 registers of 32-bit length are provided by the ARM architecture. Two of these registers are reserved for a program counter and stack pointer. Therefore, they are not available for application programs. For the rest of the 14 registers there are no distinction in use, but they can all be accessed in a similar manner. As opposed to some other CPU architectures on ARM no dedicated address or data registers exist.

This orthogonal register structure makes the algorithm implementation on ARM more flexible. The registers can be allocated more freely and the register usage can be tailored to the algorithm demands effectively.

Even if ARM is not a dedicated DSP processor there are DSP features implemented both on hardware and software on the ARM architecture. On the hardware side the most important of these is the barrel shifter. Using the barrel shifter one of the operands involved in DSP computations can easily be shifted a desired number of bit positions to the left or right. Because this feature is realized in hardware no extra CPU cycles are consumed on it. The bit shifter operand is immediately ready to be added on or multiplied with another operand not requiring extra pre-processing.

Several versions of the instruction set architecture (ISA) exist for ARM processor core families. Common ISA versions include v4 and v5TE. The latter provides extended DSP features. Digital processing of environmental noise samples described in this work is implemented on an ARM server which is based on a Chalice EBSA 21285 development system. It is a v4 ISA architecture running on a 200 MHz Intel StrongARM SA-110 processor.

The filter realization involves a cascade structure of two second-order sections. Coefficients for these biquads are kept in the processor's registers and a bunch of environmental noise samples are loaded, filtered and stored back to the external memory. We recall that the algorithm to be implemented is

$$y[n]=b_0x[n]+b_1x[n-1]+b_2x[n-2]-a_1y[n-1]-a_2y[n-2]$$

The above mentioned algorithm is implemented by a cascade of direct form structures as shown in Figures 42 and 43.

Digitized environmental noise samples are loaded into the CPU registers so that the most recent value x[n] and two delayed samples are available all the time. Product terms $b_jx[n-j]$ are calculated and added to a new filter output y[n] using the multiply accumulate instruction. On each iteration of the algorithm the input samples are aging by an additional delay and the oldest obsolete one is dropped out from the CPU registers. A load for a new sample from the external memory is initiated while the algorithm execution proceeds by processing the filter output histories $y[n-i]$. This way it is possible to take full advantage of the data pipeline on the ARM architecture. The load instruction for the new sample has enough time to terminate while the product terms involving the filter output histories are accumulated on the new filter output value. To be precise, the output calculated this way is an intermediate output denoted by $w[n]$ in Figure 42. The implemented assembly code for the first second-order section stores the intermediate output $w[n]$ in Q13 format to the external memory on the ARM server. Another biquad in the cascade structure is implemented with a similar assembly code fragment. For this second-order section the intermediate output sequence $w[n]$ from the first stage of the cascade structure acts as an input. Unrolling of control loops is utilized in both of these assembly code fragments to reduce the register demand. With this kind of hand-written low-level optimization we have managed to implement in-register filtering for individual second-order sections.

While implementing second-order sections for A-weighting with version 4 of the ARM instruction set architecture we faced a restriction related to the register access. The ARM v4 ISA only provides a 32-bit access on internal CPU registers. The A-weighting filter coefficients as well as the samples for environmental noise data fit in 16-bit words. On the v4 ARM architecture, however, 32-bit longwords must be allocated for all data items needed in the A-weighting filter implementation. Up to half of the internal register space is thus wasted. This limits the number of internal storage elements available for algorithms. Therefore, only a limited number of operations could be performed on values read from external memory to CPU registers. This limitation forced us to split the ARM v4 implementation of the cascade structure into two separate parts. Each of these code fragments had to read the external memory to obtain the input data values required for processing. Besides of this the first SOS had to write its output to a temporary storage in the external memory from which the second SOS retrieved it. The A-weighting implementation on the ARM v4 architecture was not optimal, because the load and store instructions from the external memory are considerable slower compared to accessing the internal CPU registers directly.

On version 5 of the ARM instruction set architecture (v5TE) there exist a new DSP feature to access the upper and lower 16-bit words of a 32-bit register separately. This opens up a possibility to pack two filter coefficients represented in Q15 format into one 32-bit register. Equally well a CPU register can hold two environmental noise samples expressed in Q13 fixed point representation. This feature enables to double the number of internal CPU registers used for storage of the filter multiplier coefficients and input data samples. Therefore, a more efficient ARM assembly algorithm can be developed, which is able to process a batch of environmental noise samples during one pass of the filtering code. This in-register implementation of the A-weighting cascade structure avoids intermediate store and subsequent load instructions to and from the external memory at interconnection between SOS1 and SOS2.

8 VERIFICATION OF A-WEIGHTING

In Chapters 6 and 7 we were dealing with simulation of the A-weighting filter design and its implementation on an ARM architecture. The simulation was carried out with a 64-bit wordlength, which caused only minor deviations from an ideal infinite wordlength case. In the implementation, however, 16-bit memory elements were allocated for the filter input data and for the filter coefficients. Likewise, the multiplication results were quantized back to 16-bit accuracy for further filtering steps even if the intermediate product terms were held in 32-bit registers. Therefore, several error sources originating from the finite wordlength exist for this practical implementation of A-weighting. Already the measured environmental noise signal contains an error component due to the quantization noise in the A/D conversion phase. The second source of errors origins from the necessity to quantize the 64-bit filter coefficients used in simulations down to 16 bits for the implementation. Additional quantization errors are generated while the product terms during the filtering must be quantized to fit them back in 16-bit storage elements.

In this chapter the ARM implementation of A-weighting will be verified. We shall begin by briefly presenting statistical properties for a time history of digitized sound pressure samples acquired during one of our experimental measurements. Following this, a comparison for direct form vs. cascade form implementation approaches is presented. A well known fact is that the sensitivity to finite wordlength effects depends on the chosen filter structure. Hence, sensitivities of two implementation alternatives on filter coefficient quantization are compared. We then continue by considering filter coefficient quantization and arithmetic roundoff errors specifically for the cascade structure that was chosen for A-weighting filter realization in this thesis. For example issues associated with representation of input data and filter multipliers, prevention of possible arithmetic overflows as well as estimation of errors due to quantization are examined. Thereafter, we continue with a verification example using as an input the KirriA data set described in Chapter 3. The ARM implementation will be compared against a MATLAB reference to check how well it has survived the finite wordlength effects. The verification example of the A-weighting filter realization consists of two different parts. This example starts by analyzing filtering results for the first second-order section separately. We then finalize this verification by evaluating the combined performance of two second-order sections in a cascade structure.

8.1 Input Samples Quantization and Statistics

In our experimental measurement setup (see Figure 14) analog to digital conversion was conducted using an external A to D converter. A certain amount of quantization noise is introduced on discretized signals due to the A/D conversion of physical sound pressure signals. We do not consider these input quantization errors here, because the external ADC responsible for the A/D conversion phase is not part of the digital A-weighting filter realization which is the main interest in this work. This external ADC is regarded as an independent data source supplying a time history of digitized sound pressure samples for the ARM implementation of A-weighting. Instead of quantization noise due to A/D conversion we concentrate on statistical properties of input samples in this section.

As an example we select a four minute long KirriA measurement. The first minute containing the calibration period is discarded from the KirriA audio data and we concentrate on the actual measurement part. This three minute long time history of digitized sound pressure samples is drawn in Figure 44. In this samples sequence we are interested in knowing how often a certain amplitude exists in the measured sound pressure signal.

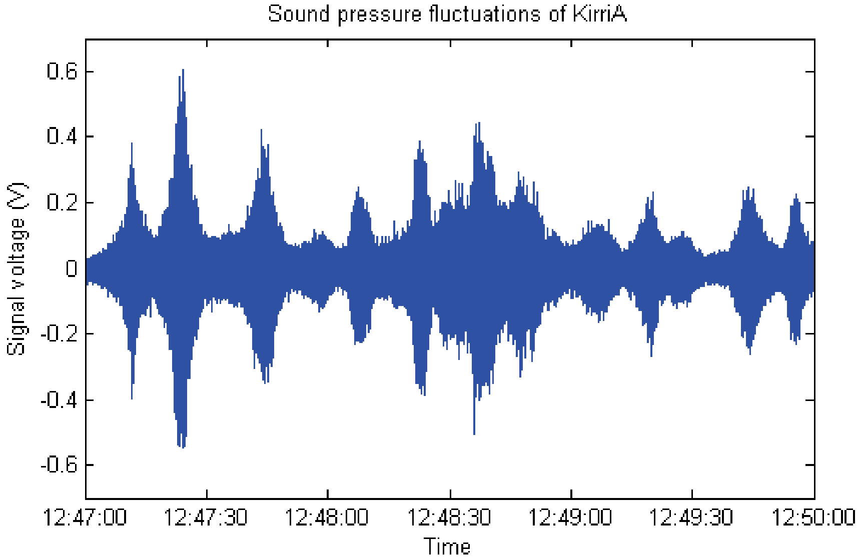


FIGURE 44 Time history for KirriA sound pressure

The audio channel in our measurement setup provided us with floating point data where the signal amplitude was limited in the range [-1, 1 [. In order to investigate the statistical distribution of amplitudes more in detail we proceeded as follows. By calculating a standard deviation for the amplitudes of sound pressure variations recorded from the KirriA audio channel we found out it to possess a value $\sigma \approx 0.05042$. Signal levels of the digitized sound pressure samples sequence were normalized by dividing them with this standard deviation. A normalized sound pressure samples sequence was obtained where normalized amplitudes were expressed in multiples of the standard deviation of the KirriA measurement signal. The interval between the maximum and minimum values of the normalized amplitude was discretized using a step size of 2σ. As a result of this discretization the normalized amplitude range was divided into 228 slices. Statistics for the amplitude slices were obtained by counting how many times different normalized signal levels fell within each slice. Finally probabilities for normalized amplitude levels were calculated by dividing the counts in each slice by the total number of samples (i.e., 7938001) in the three minute long samples sequence.

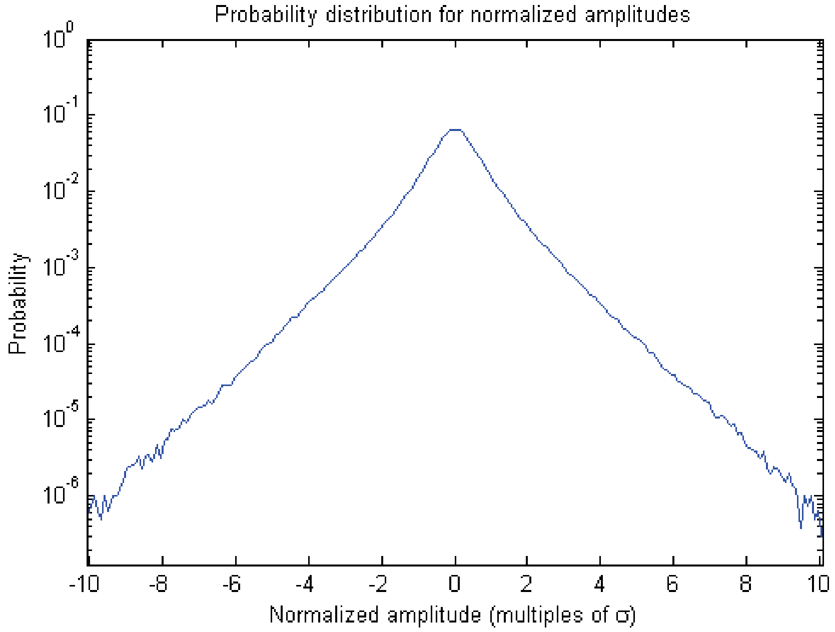


FIGURE 45 Normalized amplitude histogram for KirriA measurement

A normalized amplitude histogram generated using the above mentioned procedure gives us probabilities with which a specific normalized amplitude is expected to be present in the KirriA audio data file. The normalized amplitude histogram shown in the figure above presents probabilities as a function of the normalized signal level. From the normalized amplitude histogram we are able to estimate probabilities for certain normalized signal levels. For example the probability for a normalized signal level +- 3σ is about 10^{-3}. Correspondingly +- 9 σ signal levels are rare only occurring at a probability less than $2*10^{-6}$. This histogram also shows that for the majority of input samples in the KirriA audio data file the amplitude is close to zero.

The physical meaning for a zero amplitude signal is a silent moment while the measured sound pressure is equivalent to the ambient atmospheric pressure. What interests us in regard to input signal statistics is how far the measured sound pressure signal deviates from this equilibrium as well as how often it reaches a specific distance from the equilibrium position. For this purpose we can modify slightly the previous procedure with which the normalized amplitude histogram was generated. Instead of counting separately both the negative and positive amplitudes we consider only absolute values of the normalized amplitudes. By successively setting normalized absolute amplitude limits, which fall further and further away from the equilibrium, we count the number of occurrences of absolute amplitudes exceeding these limits. This kind of cumulative amplitude histogram can be created by scanning the complete three minute long KirriA sound pressure signal with the described procedure.

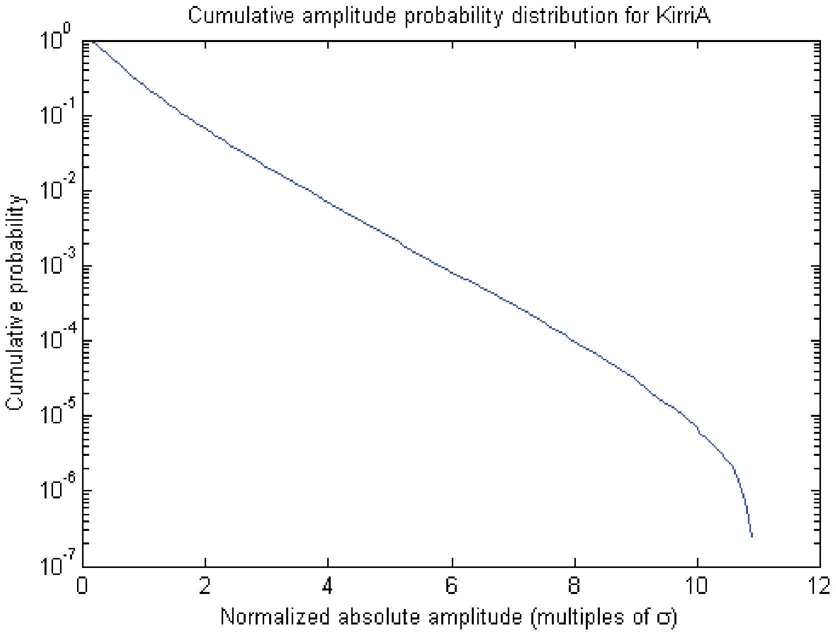


FIGURE 46 Cumulative amplitude histogram for KirriA

From the above figure we observe that for about 25 % of the KirriA samples the normalized absolute amplitude exceeds one standard deviation. Likewise the normalized absolute amplitude of 2σ is exceeded by 7 % of the samples, 3σ by 2 % and so on. Another way to interpret this figure is to give percentages of the opposite – that is, 75 %, 93% and 98% of the measured samples stay within 1σ, 2σ and 3σ, respectively.

In Chapter 9 possibilities to utilize knowledge gained in this section related to the statistical properties of the measured sound pressure signal will be discussed.

8.2 Filter Sensitivity on Coefficient Quantization

As was already noticed in Chapter 6 the quantization of filter coefficients affected the characteristics of the A-weighting design. During simulations conducted with the sixth-order filter design we saw it to be very sensitive to the quantization. The original filter design with 64-bit multiplier coefficients fulfilled the required specifications nicely. If, however, these multipliers were quantized to 16-bit the sixth-order filter design turned into an unstable one (cf. Table 6). This result demonstrates a well-known fact that a straightforward utilization of direct form structures should be avoided in realization of higher-order filter designs [Taw93]. In Chapter 6 the sixth-order transfer function $H_{Aw}^{6}(z)$ was split into a cascade of two second-order sections to approximate the original design. The sixth-order design and the cascade of second-order sections were compared in Chapter 6 by observing the pole-zero behavior while reducing the coefficient accuracy. Decomposition into lower-order components and exclusion of the last biquad helped to alleviate the pole-crowding problem and as a result a stable design was restored.

The difference between the original design and the cascade structure can also be characterized by calculating coefficient sensitivities for these filter structures. The sensitivity of a digital filter is related to the displacement of poles and zeros of the filter design while the filter coefficients are quantized. Most essential parameters affecting the sensitivity are the pole locations, which are specified by the roots of the denominator polynomial. While a feedback coefficient (see Figure 49) in the denominator polynomial of the transfer function is quantized to a reduced accuracy during the filter implementation the corresponding pole will be displaced from its original location. In an extreme case when a pole moves outside the unit circle the filter design becomes unstable as we noticed in Chapter 6.

In the design of digital filters a preferable goal is to find low sensitivity[16] structures which are more robust to coefficient quantization effects [Vai93]. In low sensitivity designs only small pole displacements are encountered. Several different measures exist to estimate the sensitivity of filter structures on reduction of the filter coefficient accuracy. For example filter coefficient sensitivity, pole sensitivity for a single pole, sensitivity sum of all poles, sensitivity function or transfer function sensitivity can be evaluated.

In order to obtain the above mentioned measures both analytical methods to derive closed-form expressions for the sensitivity as well as symbolic sensitivity analysis methods have been developed. Results of the sensitivity analysis can be presented either as a single sensitivity value, a two dimensional plot or a three dimensional contour. Most of these analysis methods include partial derivatives in respect to the multiplier coefficients. Some examples of sensitivity measures are given below.

To measure quantitatively the success of obtaining a robust filter design the coefficient sensitivity is defined according to [Neu93] as

$$S^2(e^{j\omega}) = \sum_{i=1}^{N-1} [\frac{\partial H(z)}{\partial k_i}]^2 , \tag{8.1}$$

where N equals the filter order and the summation is over all multiplier coefficients k_i. The coefficient sensitivity defined in this manner is a frequency dependent measure which tells us what kind of influence multiplier quantization induces on the overall transfer function of a filter.

Magnitude sensitivity of a filter structure is evaluated in [Sig96] by calculating a combined effect of multiplier quantization on filter's magnitude response. Instead of checking each multiplier's influence separately the robustness of the filter structure on

[16] Quite often in other contexts the judgment is just the opposite: high sensitivity is regarded as a desirable feature. For example high sensitivity of microphones and other transducers is advantageous, because they are able to detect even very weak signals.

reduced precision is analyzed using a sensitivity sum S(ν) as defined below.

$$S(\nu) = \sum_{i=1}^{N-1} \frac{\partial |H(\nu)|}{\partial k_i}, \tag{8.2}$$

where the frequencies are normalized by the sampling frequency (i.e., $\nu = f / f_s$).

In [Dol06] a sensitivity measure for the filter coefficient quantization is based on the upper limit of variations in the pole radius r and pole angle θ. A pole displacement related to quantization of a specific filter coefficient is characterized by changes in the radius and angle of the pole, Δr and Δθ, respectively. The pole sensitivity of the filter structure can be estimated by plotting Δr and Δθ with a fixed multiplier quantization error while the pole radius and angle are allowed to vary within a certain interval. Two- or three-dimensional plots obtained with this method can be used to compare different filter structures in respect of the pole sensitivity.

We have chosen to use a procedure described in [Hil09a, Hil09b] to compare sensitivities of the original design and our A-weighting implementation. This alternative may not be the most advanced one, but is sufficient for our purposes at the present stage of the research. Following a method explained in an example at the end of [Hil09a] and using a Finite Wordlength Realizations toolbox (FWRtoolbox[17]) developed for MATLAB we have calculated sensitivity estimates for the sixth-order structure and the cascade of two second-order sections. The original 6th order design received a sensitivity value of $3.5890*10^{13}$ whereas the sensitivity for the cascade of two second-order sections was much lower i.e., 16725. This result implies that the cascade structure selected for our ARM implementation of A-weighting can resist coefficient quantization considerably better. The obtained result is consistent with our simulation results presented in Chapter 6.

8.3 Headroom and Prevention of Overflows

During the simulation of the A-weighting filter all multiplier coefficients were expressed in 64-bit floating point format. In the ARM implementation 16-bit storage elements were allocated for these filter coefficients instead of 64-bit floating point numbers. The conversion from floating point representation to fixed-point numbers was performed differently for feedback and feedforward multipliers which are shown in Figure 49. The feedback multipliers a_{ik} were converted prior to the actual filtering by multiplying them with 2^{14} and rounding to a 16-bit integer. The precalculated values were used as multiplier constants in the ARM filtering code. The feedforward multipliers b_{jk}, however, were embedded directly into the ARM assembly as bitshifts by 14 bits. Originally two of the feedforward multipliers (i.e., b_{11} and b_{12}) have an absolute value of two. The bitshift by 14 bits makes them to occupy all 15 magnitude bits available in the 16-bit storage elements. Hence, the magnitude of the multipliers is coded in 15 bits and the fixed-point representation we observe for these b_{jk} is effectively Q15, which leaves no headroom for us. For all other multiplier coefficients an additional bit of headroom is available in the filtering algorithm. We have taken advantage of the

[17] FWRtoolbox, https://gforge.inria.fr/projects/fwrtoolbox/

knowledge of this difference in multiplier properties while implementing the filtering algorithm as is explained later in this section.

As we recall from Chapter 7 the A-weighting implementation requires product terms of types a_{ik}* y[n – i] and b_{jk}* x[n – j] to be calculated. As explained above the A-weighting filter coefficients a_{ik} and b_{jk} must be interpreted as Q15 fixed-point numbers, because of the multipliers b_{11} and b_{12}. Earlier in the implementation phase a Q13 fixed-point representation was chosen for the digitized sound pressure samples sequence x[n]. This leaves a two bit headroom available while performing calculations required in A-weighting. The filter inputs x[n] and outputs y[n] are stored in Q13 representation. After multiplying a Q13 sound pressure sample by a Q15 filter coefficient we gain an additional bit for the headroom. The product term is stored in a 32-bit register as a Q28 fixed-point number. Contrary to two 16-bit words which both required a two's complement sign bit only a single bit is lost for expressing the sign in the 32-bit sum of products term. Therefore a three bit headroom is available in the destination register of multiply-accumulate instructions as depicted in the figure below.

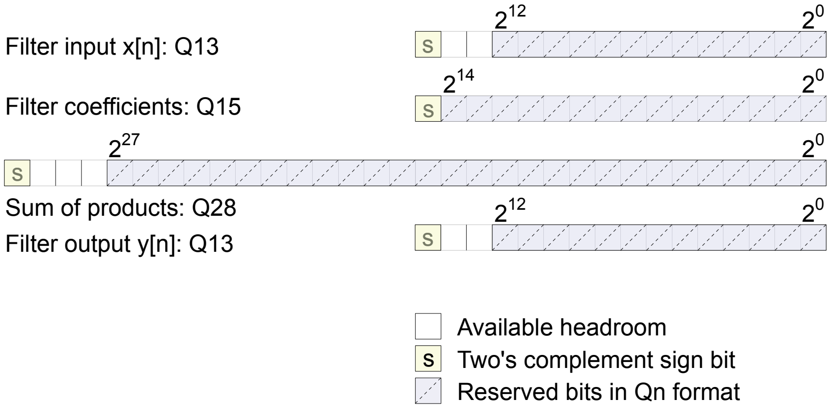


FIGURE 47 Available headroom in A-weighting algorithm on ARM

Both second-order sections in the cascade implementation generate five product terms (cf. Equations 7.15 and 7.17). These product terms are accumulated to the destination register one after another. The order of summations in the algorithm has been arranged so that those product terms involving the multipliers b_{11} and b_{12} are placed at the end of SOS1 and SOS2 filtering, respectively. By this arrangement we have managed to guarantee that the first four multiply-accumulate instructions neither in SOS1 nor SOS2 will consume all of the free headroom space we have at our disposal. The accumulation of the fifth product term into the sum of products may generate a Q31 result as shown in Figure 48. The results are not overflowing physically out of an intermediate 32-bit storage register in which the numbers are held in two's-complement format. A logical overflow, however, may occur. While we add two Q28 numbers together we may require an additional bit for the sum. In case the addition instruction overflows logically the result becomes a Q29 number, but fortunately the extra bit can be allocated from the headroom space in the destination register.

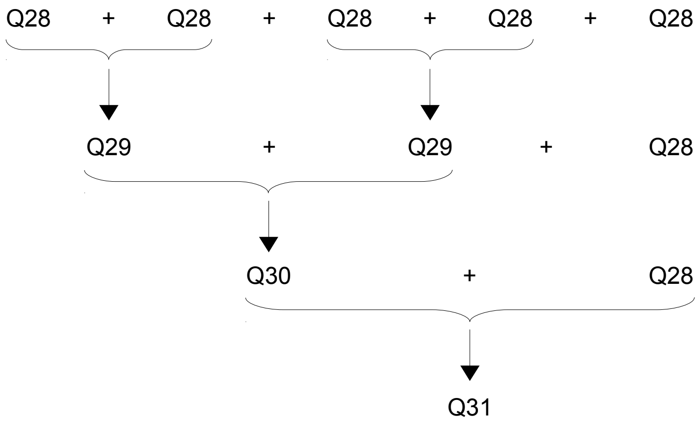


FIGURE 48 Possible logical overflows in summation of product terms

So, even if all additions inside one SOS would overflow logically the three bit headroom would be sufficient to guarantee that the A-weighting algorithm on ARM will not encounter a physical overflow while accumulating the product terms. This finding verifies that we have developed an algorithm for a second-order section on ARM which will not encounter overflows. Our A-weighting implementation, however, requires a cascade of two second-order sections. The Q13 output of the first SOS needs to be fed into the second SOS for an additional filtering step. Issues arising while cascading these two structures will be examined in the next section.

8.4 Arithmetic Quantization Errors

Individual second-order sections in the cascade were implemented as direct form I structures (cf. Figures 42 and 43). Let us consider the first second-order section of our A-weighting filter the block diagram of which is redrawn below for convenience.

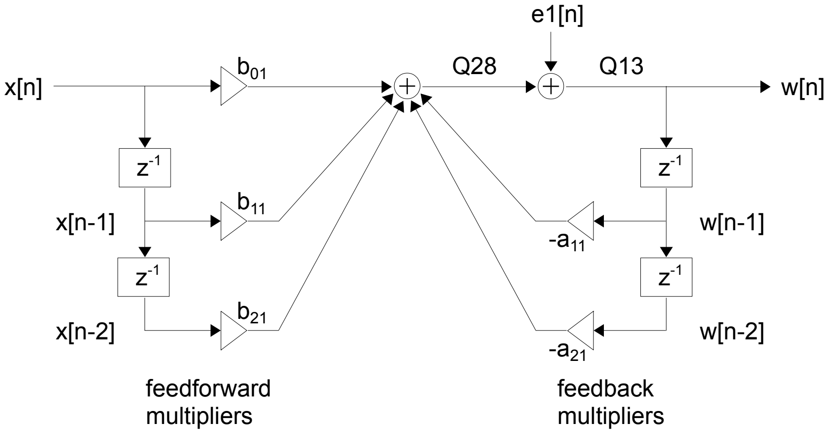


FIGURE 49 Statistical model for quantization errors on SOS1

The last category of quantization errors faced while realizing a digital filter design on an embedded platform is due to arithmetic roundoff. After calculating the sum for the product terms in the first SOS the result must be quantized back to Q13 representation again. The quantization becomes actual when all the product terms b_{01}*x[n], b_{11}*x[n-1], b_{21}* x[n-2], -a_{11}*w[n-1] and -a_{21}*w[n-2] have been accumulated to an intermediate storage in the multiply-accumulate register. Therefore, we utilize a quantization after addition scheme [Kru02]. The quantization from the Q28 representation into Q13 fixed-point numbers introduces an error e_1[n] on the first stage of the A-weighting filter as shown in the figure above.

Feedforward multipliers which implement the zeros of the transfer function do not contribute to quantization noise. This benefit is due to the fact that the feedforward multipliers of our A-weighting implementation are powers of two. Therefore, they can be implemented exactly by utilizing bitshifts instead of ordinary multiplications. As a result only a zero-valued tail is appended at the end of those product terms involving the feedforward multipliers. Later when the sum of all five product terms is quantized back to the Q13 format the feedforward product terms do not contribute to quantization errors. Their zero-valued tail embedded into the sum of product terms is just discarded. The feedback product terms, on the contrary, generate a non-zero tail in the process of multiplication when the wordlength doubles. Quantization of these product terms related to the poles is the cause for the observed errors.

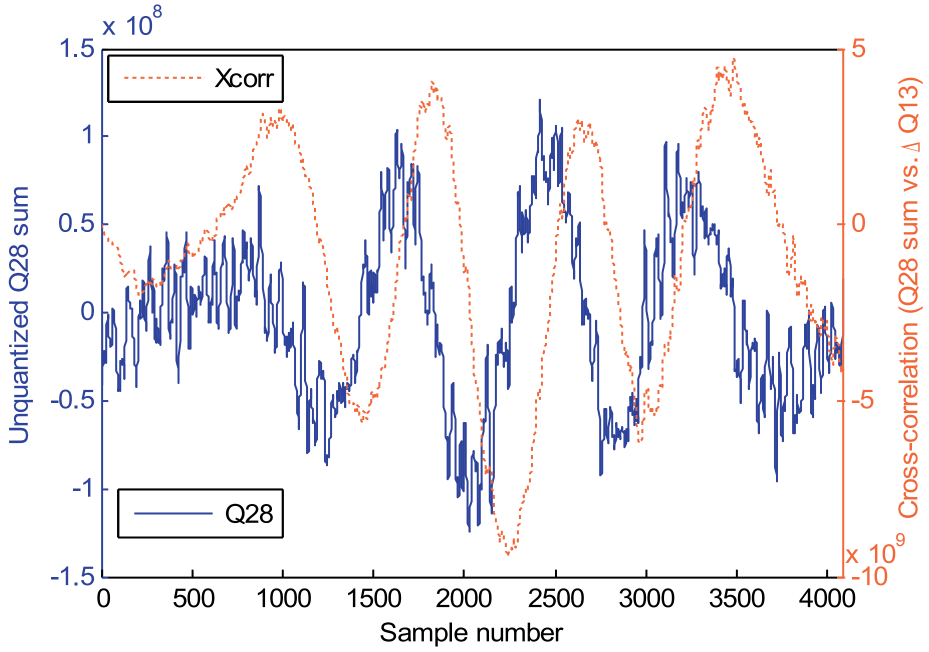


FIGURE 50 Correlation of quantization error with unquantized Q28 sum of products

The current A-weighting implementation on ARM applies truncation to quantize sums of product terms in the first biquad from Q28 fixed point integers to Q13 representation. This quantization is a random process which generates an error sequence obeying a certain probability law. Quantization noise models commonly assume the sequence $e_1[n]$ of these errors to be uncorrelated with the unquantized samples sequence. This assumption is a prerequisite for example in an additive noise model [Opp99], a statistical roundoff noise model [Pad09] and a quantization noise propagation model [Roc12]. To check the applicability of these noise models we generated a plot for the correlation between the unquantized samples and quantization error sequences of the first second-order section in the ARM A-weighting filter. If the aforementioned assumption would be fulfilled quantization errors could be modeled using a statistical noise model based on a white noise error process and uniform distribution of errors over the quantization step. In our current A-weighting implementation, however, the quantization error seems to be correlated with the unquantized sum of product terms. As can be observed from Figure 50 the quantization error due to truncation of the sum of products from Q28 to Q13 is highly correlated with the original Q28 samples sequence. Therefore, we are not able to model arithmetic quantization noise of our ARM implementation with uniformly distributed white noise.

Another way to gain insight into the characteristics of the truncation process is to plot an autocorrelation sequence for the truncation error. This sequence can be created by a MATLAB xcorr() function.

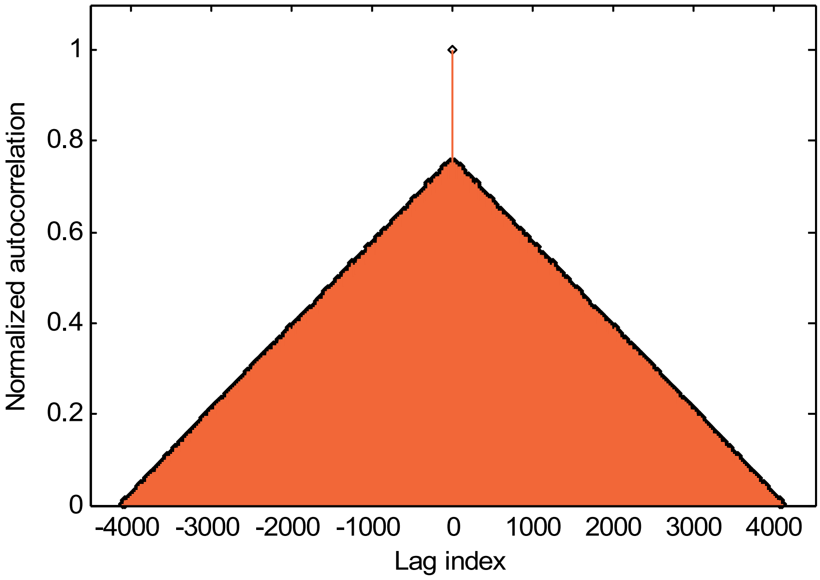


FIGURE 51 Autocorrelation sequence for quantization noise due to truncation

From the above figure we observe that the truncation error sequence can not result from a random white noise process. If it would, the autocorrelation sequence should just possess a delta peak at zeroth lag. The truncation error sequence is not generated by a wide-sense stationary random process either, because in that case the normalized autocorrelation sequence would be a triangle with the top edge at unity on the abscissa. Therefore, the truncation error sequence must have been generated by another type of random process which cannot be modeled by additive white noise. Neither can the linear noise model be applied. This is why the above mentioned quantization noise models are not applicable for the SOS error analysis in our case.

Because of the aforementioned limitations we shall estimate neither the magnitude of arithmetic quantization errors generated inside individual 2nd order sections nor the propagation of these errors through further stages of the filter. Instead of a separate analysis for arithmetic quantization errors we estimate their influence together with the coefficient quantization noise. A convenient measure for this purpose is a signal to quantization noise ratio (SQNR) which can be defined as follows [Aam08]

$$SQNR = \frac{P_s}{P_{qe}} = 10 \cdot \log_{10} \left\{ \frac{\sum_{n=1}^{N} (y_{ref}[n])^2}{\sum_{n=1}^{N} (y_{ref}[n] - y_A[n])^2} \right\}, \tag{8.3}$$

where

- P_s is the average signal power for the ideal reference output,
- P_{qe} is the average power of the quantization noise sequence,
- y_{ref} is the reference output sequence generated by MATLAB using double precision arithmetic and
- y_A is the ARM SOS2 output obtained using filter coefficients and input data quantized to Q15 and Q13 fixed-point representation

The above mentioned SQNR value given in decibel includes effects of roundoff both from the filter multiplier quantization as well as from the quantization of intermediate sums of products inside the second-order sections of the cascade structure. The SQNR value calculated using Equation (8.3) shows a lousy performance of 17 dB. In Chapter 9 we will discuss possible workarounds for this problem, which is caused by an overly conservative approach applied in scaling of the cascade structure.

8.5 Verification Example using KirriA Audio Data

8.5.1 Input Samples Sequence for Verification

In order to verify the proper operation of the ARM implementation developed for A-weighting we proceed as follows. In Chapter 3 we described the experimental measurements conducted along the Kirri motorway. For a verification example we select sound pressure samples from one of the audio channel data sets recorded during these measurements, namely KirriA. Already at the beginning of Chapter 8 we used this particular samples set to calculate statistics for the signal amplitude. For the purpose of verification we first extract a short period of digitized audio signal from the KirriA data set where both low and high frequency components are present. This kind of samples slice is found at the top of the first peak in the KirriA time history. The selected slice of 4096 samples starts at time 12:47:11.621 (sample 3158500) in Figure 44. The extracted samples sequence accounts for a 93 ms time period from the amplified sound pressure signal recorded and digitized via the audio channel during the KirriA measurement.

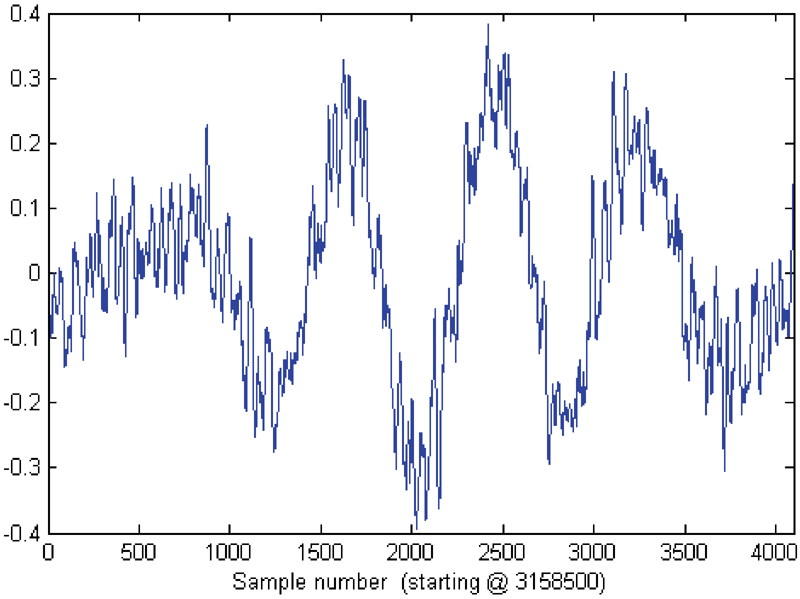


FIGURE 52 KirriA samples sequence for verification

Within this time window we are able to detect signal frequencies which are above 22 Hz (≈ 2 * (1 / 93ms)). This will suffice for us because the frequency range of the human auditory system extends from 20 Hz upwards.

8.5.2 Verification of the First Second-Order Section

As a first step of the actual verification we intend to check how the first second-order section of the ARM implementation performs. The ARM filtered output is compared with a reference output generated by MATLAB. As explained in Chapter 7 the A-weighting filter on ARM has been implemented by a cascade of two 2nd sections. The first one of them (i.e., SOS1) is acting as an LPF and is intended to filter high frequencies upon the upper edge of the human auditory range. A region of interest in the verification of SOS1 is found between frequencies 2 kHz and 20 kHz. We can concentrate only on this frequency range, because below it the effect of SOS1 is just a slight constant amplification (see Figure 31). The purpose of the extra amplification is to lift the combined transfer function within the range 50 Hz – 3 kHz in order to fulfill the filter specification (see Figure 33). A direct comparison of time domain signals is less informative in the verification of SOS1, because SOS1 affects the higher end of the frequency range as mentioned above. Therefore, the time domain signals from verification stages 'MATLAB SOS1 reference' and 'ARM SOS1 output' as depicted in Figure 53 are not plotted for visual inspection. The ARM SOS1 output is compared with the MATLAB reference in the time domain by checking the correlation between these outputs.

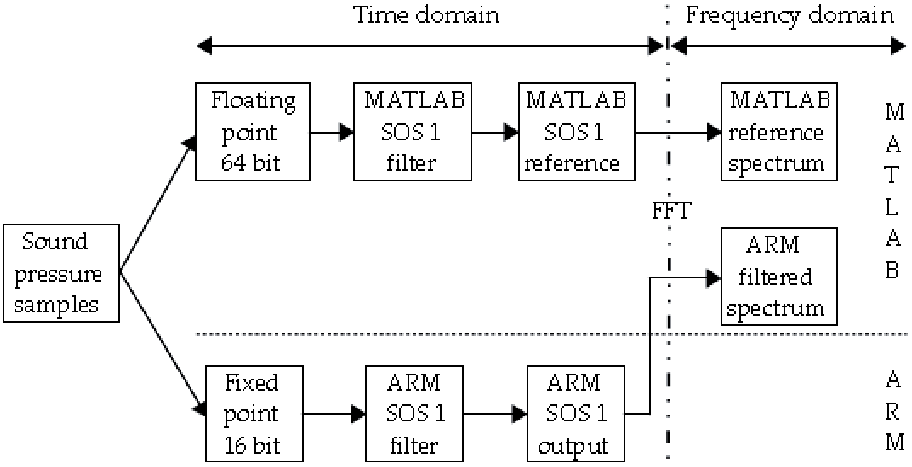


FIGURE 53 Verification of the first second-order section

As a filtering result we obtain a sequence of values indexed with time indices in the same way as the samples data in the filter input. Hence, a pairwise comparison of output vector elements in the ARM and MATLAB channels is possible. A two dimensional correlation plot is generated in which the amplitudes of corresponding channels are shown as a scatter plot. If the ARM realization of SOS1 is consistent with the MATLAB reference it dictates as a straight line in the correlation plot. The ARM filtered SOS1 output is plotted as a function of the MATLAB reference output for SOS1 in Figure 54.

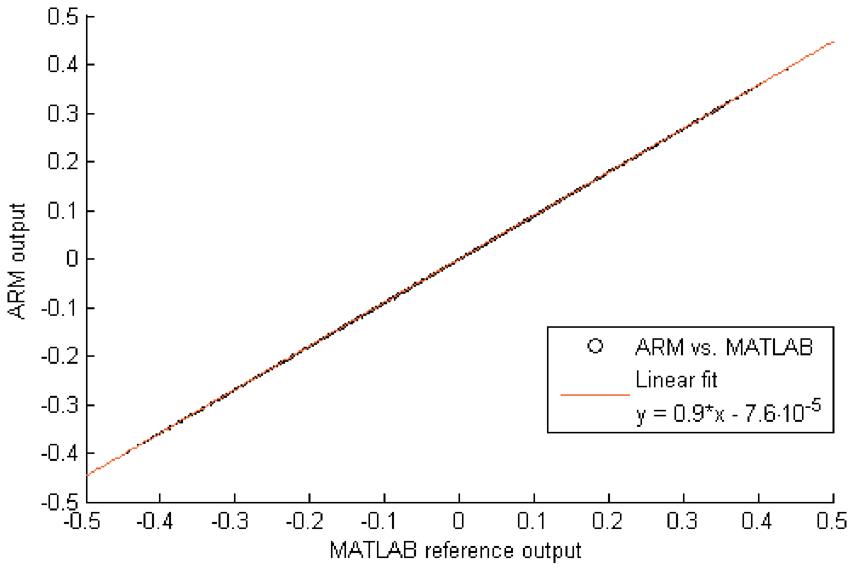


FIGURE 54 Correlation between ARM and MATLAB time domain SOS1 outputs

From the above figure it can be seen how well the ARM output amplitude of a certain time domain sample correlates with the MATLAB reference amplitude of the same sample. A good correlation observed visually is confirmed additionally by calculating a norm of residuals for the linear fit which is 0.067732.

8.5.3 Verification of the Cascade Implementation

Using a similar procedure as with the first second-order section we have verified the cascade structure of the ARM A-weighting filter. Instead of only one SOS we used a cascade of two second-order sections to implement the desired filtering as shown in Figure 55. While we constructed a cascade by attaching another second-order section behind SOS1 the low frequency characteristics on the filter output were modified.

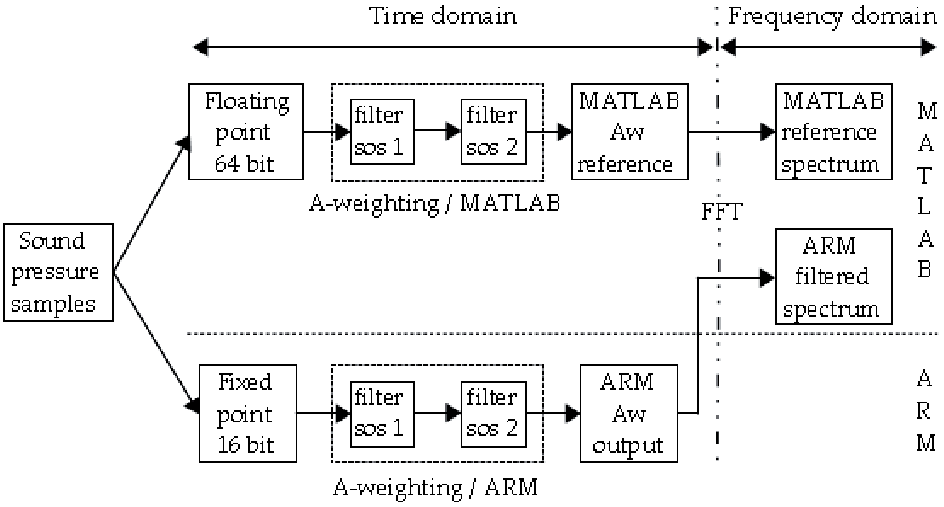


FIGURE 55 Verification of A-weighting implementation on ARM

The additional SOS in the cascade acts as a high pass filter and affects low frequency characteristics of the filter output. This SOS has an equivalent function as the apical end of the cochlea in the inner ear. While reaching toward the apex the tapered spiral geometry of the cochlea makes human ear's frequency response to gradually add more attenuation on lower frequencies. Correspondingly, the second SOS in the A-weighting cascade on ARM shows a smooth transition from the passband to the stopband (see Figure 33) while going toward lower frequencies. With frequencies around 50 Hz the attenuation applied by human ear and SOS2 is 20 – 30 dB. In the measured environmental noise signal used for the verification the above mentioned frequencies are present. They origin from a diesel engine of a heavy truck and hence contain more energy than other traffic noise components superimposed on the measured signal. The influence of SOS2 on low frequencies is noticed already in the time domain by visual inspection of the 'ARM output' and the original unfiltered 'Input samples' sequence which are shown in Figure 56. It can be observed that SOS2 has attenuated effectively those slow fluctuations with high amplitude related to the low frequency engine noise.

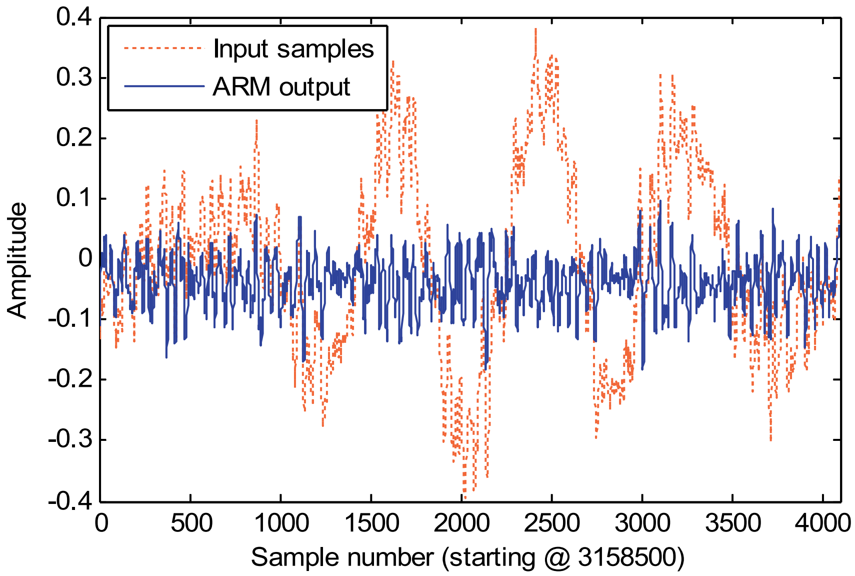


FIGURE 56 Input and output sequences of ARM A-weighting filter

What remains in the ARM A-weighted output is a signal where the mid frequencies are dominating, because SOS1 already filtered high frequencies. A more detailed content of the A-weighted signal is shown at the end of this chapter while comparing A-weighted frequency domain outputs of ARM and MATLAB. We also observe that the maximum amplitude of the ARM Aw output is less than half of the maximum input amplitude. By calculating output variances for the filtered and unfiltered signals we can conclude that the ARM A-weighting has removed about 60 – 65 % of the incoming signal energy.

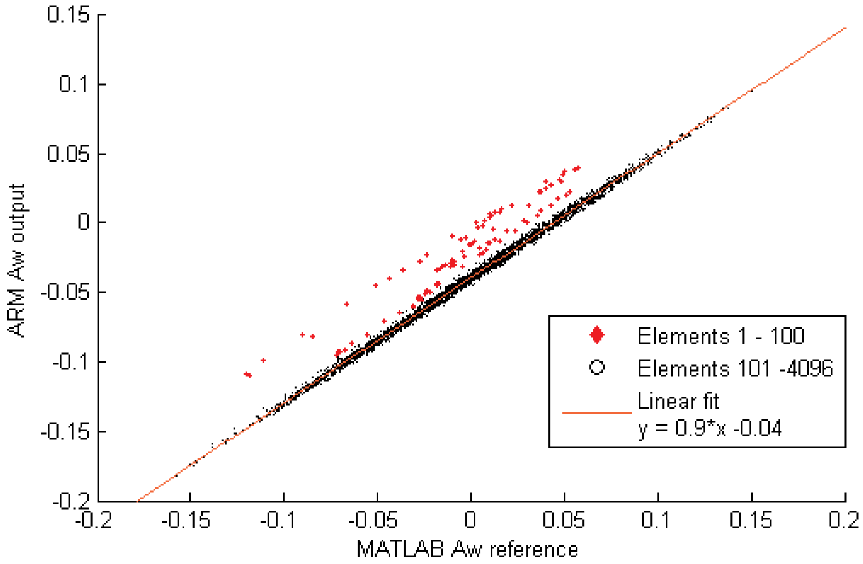


FIGURE 57 Correlation of A-weighting outputs between ARM and MATLAB

Let us stay still a while in the time domain and compare the ARM Aw output with the MATLAB Aw reference by checking their correlation. In the same manner as in the previous section we have generated a correlation plot for the time domain outputs of ARM and MATLAB. More spread is observed in the two dimensional correlation plot of outputs for the cascade structure compared to correlation of the SOS1 outputs earlier. A more detailed investigation of the output data reveals that this spread is most clearly present at the beginning of the output sequence. In Figure 57 the correlation between the first one hundred elements of the ARM and MATLAB filter outputs is marked with red diamonds. This figure shows that the time domain relation between the ARM output and the MATLAB reference can be described quite well using a linear fit (i.e., y = 0.89897*x – 0.039836) if the leading 100 elements of the verification sequence are excluded. As can be seen these trailing element pairs of the outputs are most noticeably off-axis from a regression line which is fitted in the data set. In case of the cascade structure the goodness of fit calculated using a norm of residuals (0.279) is worse than the value obtained for the linear fit of SOS1 output data (0.068). If, however, the first one hundred output vector elements would be discarded while calculating the norm of residuals for the cascade structure would decrease to a value 0.139.

As a final verification step of our A-weighting implementation we examine ARM and MATLAB frequency domain outputs. According to the upper sequence of actions depicted in Figure 55 a MATLAB reference spectrum was created for the A-weighting filter output. By comparing this reference spectrum with the filter output from the ARM implementation following observations can be made.

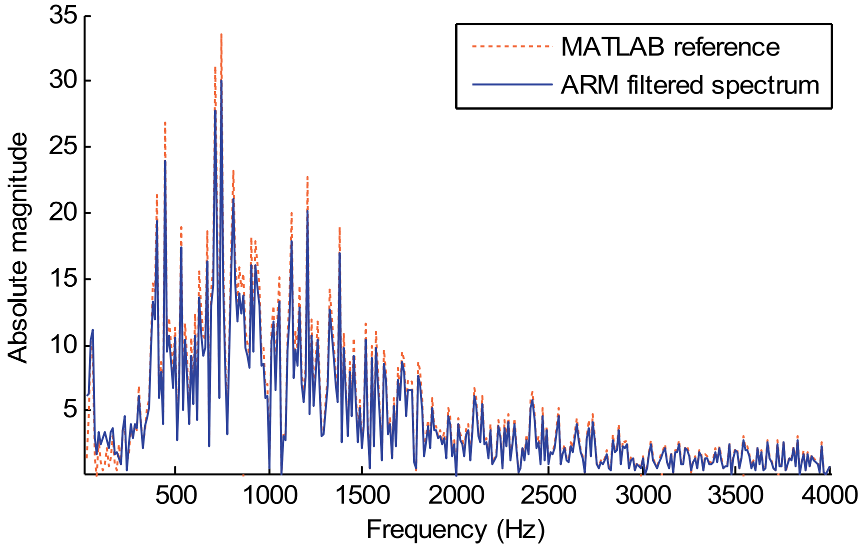


FIGURE 58 Comparison of frequency responses between ARM and MATLAB

The local maxima and minima of the ARM filtered spectrum are located on the same frequencies as in the MATLAB reference. Thus, the quantization errors of the ARM implementation have not affected the frequency scale of the filter output. The amplitude of the frequency peaks in the ARM filtered spectrum shows a systematic error which reduces the peak height within a frequency range 250 – 4096 Hz compared to the MATLAB reference. The difference of peak heights, however, is a constant ratio of 0.9 as was observed already while examining the time domain outputs in Figure 57. By applying a correction based on the linear fit shown there the peak height of the ARM filtered spectrum could be increased by 10 percent. This kind of correction would counteract the low input gain of the ARM A-weighting cascade which was caused by taking a too cautious approach in the filter scaling. Additionally, we notice from the above figure that for low frequencies below 200 Hz the ARM implementation is not filtering as effectively as the MATLAB reference. This feature was already pointed out at the end of the simulations chapter (cf. Figure 38). There we also presented reasons for selecting this implementation alternative for A-weighting on ARM. According to our judgment it is better to let a bit too much energy through on these low frequencies instead of neglecting their importance. Therefore, we do not find the observed lack of attenuation on the lower edge of the frequency range a serious problem.

9 DISCUSSION AND FUTURE WORK

In Chapter 8 we verified that the A-weighting filter on ARM fulfills the requirements posed by the intended target application. We compared the ARM A-weighting filter with a reference generated by MATLAB. A representative sequence of digitized sound pressure samples from the KirriA measurement served as an input for this comparison. The same test input sequence was first fed through a 64-bit MATLAB chain to produce an 'infinite' precision reference for the filter output. A 16-bit fixed-point ARM implementation of the A-weighting filter was then compared against the MATLAB generated reference. As a conclusion based on the performed comparison we could verify that our ARM implementation is adequate.

The realized A-weighting filter is targeted for an embedded systems platform which introduced specific boundary conditions on the realization. One major constraint implying severe consequences was the necessity to quantize filter coefficients and product terms to reduce the required wordlength and computational complexity. We had to make compromises to balance between the target application requirements and the available computational resources. We started by analyzing the impact of quantization on the A-weighting filter implementation. We managed to show that a cascade of two second-order sections chosen for the ARM implementation resulted in a lower sensitivity filter in comparison to the original sixth order design.

One aim of this work was to develop a first prototype for the A-weighting filter on the ARM platform. The described implementation is not optimal and it has some weaknesses related to scaling and signal-to-noise ratio of the filter structure as will be explained later in this chapter. The goal of this thesis, however, is already achieved with the modest performance obtained by using a conservative approach in the realization. These research results can be regarded as a proof-of-concept for an assembly implementation of A-weighting on an embedded ARM platform.

In this final chapter we comment also briefly those environmental noise monitoring systems presented in Section 1.4. Application possibilities for the ARM assembly implementation of A-weighting exist in several of the WSN and PSN systems reviewed in that section. Because none of the methods currently available in environmental noise mapping are flawless we propose a hybrid method for compilation of environmental noise maps. We finish by envisioning future directions for development of the target application that was described in Section 1.5.

9.1 A-weighting Realization Aspects

One example of those compromises mentioned on the previous page was scaling of the A-weighting filter on the ARM platform. We have applied sum scaling on the input of the first SOS by using a scaling constant L derived from a sum of the impulse response sequence. The sum calculated from values of the A-weighting filter's impulse response sequence turned out to be 2.2211. Hence the first power of two – that is, 2^1 – would not have been enough for scaling purposes. The next power of two (i.e., 2^2) is a good alternative, because it can be implemented as a bitshift. Using the value L=4 we have coded 1/L scaling into the A-weighting filtering by reserving a two bit headroom in the fixed-point representation of the input samples sequence. As a matter of fact, second-order sections comprising the filter are unscaled. The scaling constant is applied on the input of the cascade structure in the form of this two bit headroom. As shown in Section 8.3 this kind of scaling has guaranteed that the digitized sound pressure samples sequence can be filtered without a physical overflow. Thus, the currently applied sum scaling prevents overflows completely during the filtering. An advantage of this choice is the avoidance of severe side effects caused by overflows in the algorithm (e.g., a limit cycle type known as overflow oscillations). It can rightly be argued that the applied scaling is too conservative and decreases the signal-to-noise ratio (SNR) of the realized filter. While we selected Q13 fixed-point integers to represent input data in our ARM implementation we actually applied an adjustment on the input gain of the A-weighting filter. The signal level at the filter output as well as the SNR were inevitably affected by this input gain adjustment.

Another scaling alternative would have been to realize dynamic range scaling for both of the second-order sections of the ARM filter. During the realization phase of the A-weighting filter choices regarding the implementation strategy had to be made. One of these was the decision not to apply dynamic range scaling on second-order sections of the cascade structure. By scaling internal branch nodes to an L2-norm of unity a complete avoidance of overflows would have resulted like in the sum scaling. Dynamic range scaling, however, would have introduced new noise sources on the realization.

TABLE 7 Feedforward multiplier coefficients for two scaling alternatives

	Current SOS1 + SOS2 cascade		Dynamic range scaled cascade	
	k = 1	k = 2	k = 1	k = 2
b0k	1.0000	1.0000	0.0040	95.8708
b1k	2.0000	-2.0000	0.0081	-191.7415
b2k	1.0000	1.0000	0.0040	95.8708

In Table 7 feedforward multipliers for our current second-order sections are given together with the corresponding values[18] in case dynamic range scaling would have been

[18] Calculated according to a procedure outlined in [Mit01]

applied to the cascade structure. By comparing sum scaling and dynamic range scaling approaches the following differences can be observed. While applying sum scaling we operate on the input signal before the cascade structure. We do not need to touch the values of the feedforward multipliers, because this scaling is achieved by reserving a two bit headroom in the fixed-point representation of the input data (see Section 7.3). Therefore, no run-time delays are introduced in the current filtering algorithm which applies sum scaling.

All current feedforward multipliers of the cascade structure designed for A-weighting are powers of two. An additional simplification for the filtering algorithm is provided by the fact that four of these multipliers take a value on unity (see the second and third columns in Table 7). Even the last two feedforward multipliers have an absolute value of two. Hence, all of these filter coefficients can be inserted to the ARM assembly code as bitshifts without a need to allocate any register space for them. The utilization of bitshifts allows us to fit the complete filtering of one second-order section in the available register space on the ARM ISA v4 architecture. The in-register execution of the algorithm is naturally much more efficient compared to continuous reload and rewrite cycles between ARM CPU registers and external RAM memory. Furthermore, no additional noise sources are generated on the feedforward stages of filtering while using bitshifts instead of multiply instructions. And finally, a multiplierless implementation for the feedforward part of SOS1 and SOS2 is achieved by replacing the actual multiply instructions with bitshifts.

Dynamic range scaling, on the other hand, would have affected the multiplier coefficients inside the second-order sections comprising the cascade structure. As a consequence of this the algorithm implementation for A-weighting would have been more complicated. From Table 7 we observe that in order to realize dynamic range scaling for the A-weighting filter non-trivial multiplier coefficients should have been used inside second-order sections. Because of these non-trivial multiplier coefficients additional product terms would have had to be added into the filtering algorithm to replace bitshifts. The multiplier coefficients in these new product terms would have required registers to be allocated for storage. One consequence of the dynamic range scaling would therefore have been the lack of internal register storage space which would have prevented the in-register realization of second-order sections from us. The need to use multiply instructions instead of bitshifs together with extra load and store instructions would have had an impact on the run-time performance of the ARM implementation as well. Moreover, some new noise sources would have been introduced due to additional multiply instructions.

Both of the scaling methods mentioned thus far are derived from characteristic features of the filter structure. In sum scaling and dynamic range scaling the coefficients were obtained by examining the properties of the impulse response and cascade structure of the A-weighting filter, respectively. In the future statistical properties of the input signal to be filtered could be taken into account to implement scaling in a less conservative manner.

Complete overflow avoidance was applied in the A-weighting implementation described in this work. Instead of complete avoidance an approach to minimize the probability for arithmetic overflows could be chosen in the future. The future scaling approach ought to take advantage of the knowledge acquired for the input samples statistics (cf. Section 8.1). Instead of the complete overflow avoidance a certain probability should be allowed for the overflows based on the cumulative amplitude histogram generated from the input samples sequence. This relaxation in avoidance of overflows could provide a less conservative realization where the signal to be filtered would not be attenuated excessively. By modifying the A-weighting scaling in this way the SNR of our ARM implementation could be increased.

A-weighting realization on an embedded systems platform requires intermediate product terms to be quantized. Depending on the type of quantization the probability distribution of errors varies. Arithmetic quantization errors in the current version of our ARM assembly implementation are generated by truncation of the intermediate product terms. A proper error analysis could not be performed, because the error sequence correlates with unquantized intermediate product terms as was shown in Figure 50. Thus, quantization noise models mentioned in Section 8.4 were not applicable.

In case of rounding the quantization errors take a probability distribution similar to that of a random white noise process. This fact can be seen from the figure below where the autocorrelation function for the SOS1 quantization error is plotted assuming the rounding scheme would have been applied.

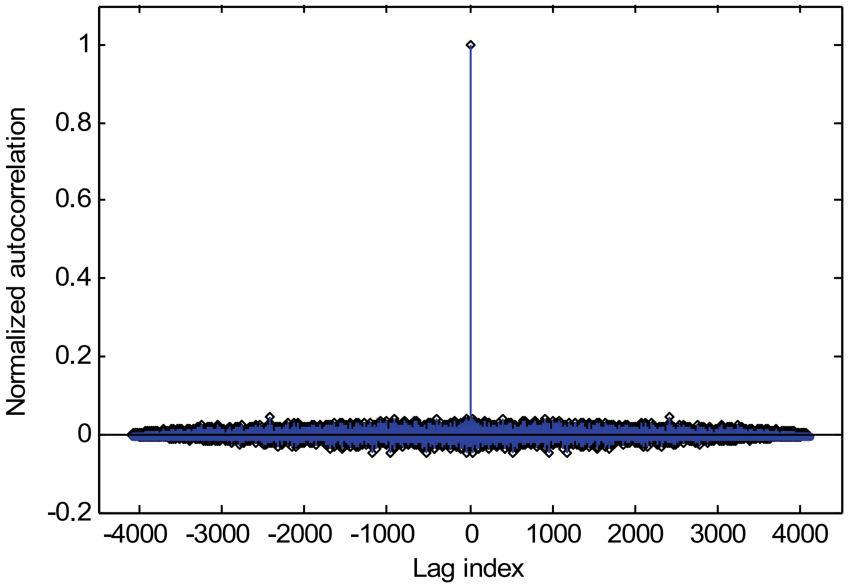


FIGURE 59 Autocorrelation sequence for quantization noise due to rounding

The autocorrelation function depicted in Figure 59 resembles that of a white noise random process. Therefore, we can state that the quantization process of SOS1 would have fulfilled the requirements of the linear noise model if the intermediate Q28 sums would have been rounded. In future development versions of the A-weighting filter on ARM the quantization inside second-order sections will be implemented by rounding instead of truncation. This modification enables a proper quantization error analysis to be executed using the standard linear noise model.

A well known fact is that the filter structure influences quantization processes in filtering. Because of this connection the original sixth order filter was not implemented directly. Even if the direct form structure is not suitable for implementing higher-order transfer functions directly it is a sensible choice for first- and second-order section realizations. Especially in our case this implementation scheme is favorable, because of a good match between the direct form structure and the ARM Instruction Set Architecture. A DSP like multiply accumulate (MLA) instruction provided by the instruction set together with a barrel shifter on the hardware level is an efficient combination for realizing direct form second-order structures. Our cascade realization for the A-weighting filter comprised of two second-order sections. The first of these implemented sections performed extremely well as we noticed in subsection 8.5.2. By using a simple sum scaling we were able to develop an in-register realization for the first SOS. Both of the second-order sections, however, could not be fitted into the ARM CPU registers simultaneously. This constraint is caused by a limitation in the version 4 of the ARM ISA. This version is a 32-bit architecture, but a restriction exists in it which forces a 16-bit long operand to allocate the complete 32-bit register. Therefore, we had to divide filtering in two steps. First a set of digitized environmental noise samples were filtered by SOS1 the output of which were stored back to external memory. Following this first step the ARM CPU registers were initialized with the multiplier coefficient of SOS2 and the SOS1 filtered output was processed further. As can be observed in Figure 57 we have a problem while loading SOS1 filtered output from external memory to CPU registers for the second filtering step. The correlation between the MATLAB reference and ARM SOS2 output for the first one hundred elements deviates from the regression line. Rest of the output samples generated by the SOS1 + SOS2 cascade correlate again nicely with the MATLAB reference.

In the future the A-weighting filter implementation should be ported to a more recent ARM ISA architecture version where the upper and lower 16-bit words inside a 32-bit register can be accessed separately. As a result, the number of available 16-bit registers could be doubled and this change would open up a possibility for in-register realization of both second-order sections simultaneously. An advantage of keeping the multiplier coefficients for both of the second-order sections in CPU registers at the same time is reduced number of load and store operations in filtering. Those input samples that are loaded from external memory can pass through the complete cascade structure and be filtered by both of the second-order sections before storing them back. We also anticipate that problems encountered between SOS1 and SOS2 can be removed when filtering is executed in one pass without extra load and store operations between the CPU registers and external memory.

9.2 Implementation of WSN Systems

Work related to wireless sensor network systems started for more than a decade ago in the form of academic research projects. Scientific knowledge and practical experience gained during the early years of this research slowly disseminated from universities and research institutes to a wider field of utilization. As a result of this pioneering academic development some WSN components have been provided more recently as commercial products. A good example of this slow dissemination process is the Mica and Telos mote hardware, which were mentioned in subsection 1.3.2. These mote class WSN nodes were originally introduced in 2002 and 2005, respectively. There have been a couple of spin-off companies (e.g., Crossbow Technology Inc., Moteiv Corporation, and Sentilla Corporation) which have marketed initial versions of Mica and Telos motes. Currently latest versions of these wireless motes (i.e., MicaZ and TelosB) are available from MEMSIC Inc., which is not specialized only in wireless sensor networks, but has a broader product portfolio.

Another trend in the evolution of WSN hardware has been a move from simple microcontroller based motes toward a more powerful platforms. Two examples of the latter category are the ARM processor based boards Gumstix and SunSPOT. Likewise from subsection 1.3.1 it can be observed that a similar shift of interest has occurred in WSN research projects. This developmental process is demonstrated by Glagsweb in which the initial hardware was solely based on 8-bit microcontrollers, whereas version 2012 of that system is equipped with 32-bit ARM processors.

While choosing radio frequency for communication and data transfer in a WSN system several criteria needs to be considered. The most popular RF frequency band utilized has been 2.4 GHz. As we pointed out the popularity of this frequency band is not altogether without problems. We addressed implementation issues faced by WSN developers in relation to the 2.4 GHz band as well as benefits of using lower frequencies in subsection 1.3.2. In this subsection we explored issues related to interference and transmission range of RF links. The choice, however, can not be based solely on performance of these RF characteristics, but one must evaluate other parameters as well (e.g., antenna size, data transfer speed, availability of development tools). A sensible balance should be found between a diversity of parameters affecting development, operation and maintenance of an intended environmental noise measurement system.

One goal in the design of the CiNet sensor device was to minimize the number of peripheral components connected to the microcontroller [Hak10b]. An additional step toward this goal would be a software implementation of A-weighting. By replacing the current analog CiNet implementation of A-weighting with a digital one the required hardware components (i.e., six capacitors and five resistors) for this analog filter could be dropped off from the sensor node design. This modification was rejected by CiNet developers because the microcontroller utilized on CiNet sensor nodes was lacking the required processing power (see the end of Section 1.5). In this modification we have one application possibility for the digital ARM A-weighting implementation described in this thesis. As can be observed from the hardware alternatives reviewed in subsection 1.3.2 suitable HW platforms currently exist with which digital A-weighting could be realized on WSN nodes.

9.3 PSN Systems in Environmental Noise Mapping

Other application possibilities for our ARM A-weighting implementation can be found in participatory sensor network applications which were explored in subsection 1.4.2. The current Ear-Phone implementation relies on platform independent high-level programming language (Java) which is not optimal from the CPU usage point of view. In Ear-phone development, minimization of CPU utilization and power consumption were left for future work. On the target platforms used in Ear-Phone (i.e., Nokia N95 and HP iPAQ) only fixed point arithmetic is available. The ARM assembly realization of A-weighting filter presented in this thesis is based exclusively on fixed point type arithmetic. Therefore, it could be applied on the HP iPAQ version of the Ear-Phone application, because the iPAQ is equipped with an Xscale PXA270 running v.5TE of the ARM instruction set architecture. Equally well the ARM A-weighting implementation is suited for the N95 in which an embedded ARM1136 processor is running ARM ISA v.6 instruction set. An additional benefit of using this ARM A-weighting in conjunction with Ear-Phone would be the replacement of the Class 2 filter (0 – 8 kHz) with a Class 1 filter (50 – 20000 Hz). Still another and in some respect easier and more straightforward application platform for the digital ARM A-weighting implementation was mentioned in subsection 1.4.3. The first external acoustic sensor board described there was lacking environmental noise measurement features, but the ARM-Cortex M3 processor on this external microphone board could provide an ideal platform for executing the assembly algorithm for A-weighting.

In participatory sensor networks mobile phones should be aware of their position while sensing environmental noise. Otherwise, necessary corrections on the measured environmental noise levels cannot be applied. An example of corrections applied in environmental noise measurements conducted by standard sound level meters is a -3 dB reduction of the measured SPL if the SLM microphone is within a 3 meters' distance from a reflective surface (e.g., a wall). One possible position for a PSN phone is on the palm of a participating noise mapper. In this case, reflections of traffic noise from the mobile phone user's body affect the amount of sound pressure sensed by the phone's microphone. The Ear-Phone group has made some context-awareness tests and noticed that the measurement errors may vary between 2.7 dB and 4.1 dB depending on the mobile phone's position as given in subsection 1.4.2. The extra noise generated by the participating noise mapper should also be estimated and corrected in the results.

Let us continue by concentrating on NoiseTube, because it is the most recent and most comprehensive evaluation in the field of participatory noise sensor networks. The NoiseTube group has evaluated thoroughly the calibration issues of mobile phones used in participatory noise mapping. As explained in subsection 1.4.2 they measured first a mobile phone frequency response in an anechoic room using professional grade measurement equipment. These calibration experiments resulted in seven frequency response curves associated with sound pressure levels in the range 60–90 dB. The NoiseTube group compared the mobile phone frequency response against a reference response measured by a laboratory reference DAQ system. Their conclusion from these frequency-dependent calibration experiments was that this type of calibration is not required, because the frequency response of the tested mobile phone was quite similar to that of the laboratory reference system on those sound pressure levels which are important in urban environments.

According to the NoiseTube group low SPL are not important in urban areas and therefore they diminish the value of three frequency characteristic curves with lowest SPL (i.e., 60 dB, 65 dB and 70 dB). Later in their article, however, they present measurement data obtained from a participatory noise sensing experiment which indicate that an average, minimum and maximum sound pressure level for a daytime noise map have been 61 dB, 44 dB and 71 dB, respectively. These measurement results clearly show that sound pressure levels below 70 dB cannot be excluded in the urban context.

Despite a clear contradiction between their judgment and these measurement results they use this incorrect judgment to reason frequency-dependent calibration out. Moreover, significant deviations from the reference exist in frequency response curves measured with three lowest sound pressure levels. For example the mobile phone response to 4 kHz test signal applied on 60 dB, 65 dB and 70 dB SPL deviates 8–12 dB from the response of the laboratory reference system. On the other hand at 2.5 kHz these three SPLs generate more of less equal responses for the NoiseTube phone and the reference system. These calibration results reveal that large differences compared to the reference are present in the response of the tested mobile phone even between nearby 1/3 octave bands.

Nevertheless, a frequency independent white noise calibration method was selected by the NoiseTube group. If the frequency independent white noise calibration method is accepted, we know that calibrated mobile phones can deliver rather accurate results in case the measured environmental noise contains all frequencies in equal amounts. This is not, however, a relevant assumption while road traffic noise is measured. In road traffic noise emissions two main components can be identified, namely a low-frequency component of engine noise and a higher frequency contribution from the wheels. A typical road traffic noise spectrum is composed of a frequency peak at around 50–150 Hz and a broader frequency bump within a range of 800–2000 Hz. Therefore, a mobile phone used to sense road traffic noise may assess noise levels incorrectly if it relies on calibration coefficients obtained from a procedure based on the white noise frequency content.

The NoiseTube group also claims that low frequency noise is not important in urban areas. In line with this claim they state: "Pink noise is the spectrum of choice when there is a focus on low frequency sounds, which is not particularly the case in an urban setting." This argument was another reason to select a frequency independent white noise calibration method. It should, however, be pointed out that because octave bands are wider at higher frequencies a white noise input evokes an auditory perception as if more high frequency content were present. Pink noise, on the contrary, is perceived as having equal amounts of energy in all octave bands. Therefore, from the human perspective, it is more common to use pink noise excitation to assess human perception on auditory stimuli. Another remark is that we regard low frequencies important also in urban settings. The use of A-weighting just creates an illusion that low frequencies are not present in urban environmental noise. What makes the low frequency component of urban noise a problem is the fact that the abatement of lower frequencies requires massive constructions for which space does not exists in urban surroundings.

Because of those contradictions mentioned above we disagree with the reasoning to abandon the frequency-dependent calibration method. Even sound pressure levels within the range 60–70 dB are important in urban noise assessment and therefore the frequency characteristics of noise sensors on these lower SPLs cannot be ignored. The NoiseTube group writes in their article: "we have put much effort in calibration experiments, determining frequency characteristics as well as white noise characteristics for 11 mobile phones". It is unfortunate if all efforts expended on determining frequency characteristics of mobile phones in the context of these noise measurements are wasted. Further development of the frequency-dependent calibration method would be highly desirable.

The claim “Participatory noise mapping works!” presented by the NoiseTube group is correct to a certain extent. As can be seen from the results of the participatory noise sensing campaign arranged by them a small group of citizens was able to provide noise measurement data for urban noise mapping. Participatory noise mapping is able to catch up rush-hour characteristics of urban noise. A correlation is seen between the mobility of urban noise sources and participatory noise mappers. Most of the measurement data for mapping was obtained during the daytime on specific peak hours which coincide with rush-hours in traffic.

There was not much mobile sensing activity at nighttime as was states in their article: “very little measurements were made between midnight and 8 am”. This specific result from NoiseTube measurement campaigns reveals a challenge which is faced in participatory noise mapping. Namely, the amount of noise measurement data obtained is reversely proportional to the priority of assessment during the day, evening and night periods. As the CiNet group pointed out authorities in cities need to have a method to assess also the nighttime environmental noise. Assessment of nighttime noise levels is essential, because during the night people need silence most in order to get rest. The Environmental Noise Directive prioritizes the nighttime noise in calculations of Lden values by requiring 10 dB, 5 dB and 0 dB penalties on Lnight, Levening and Lday values, respectively. In the NoiseTube measurement campaign, however, it was observed that mobile phone users' participation during the night was minimal. Only 5% of the measured LAeq,1s values were acquired during nighttime, whereas 21% origin from evenings and 74% from days. This behavioral feature in activity of participatory noise mappers influences the distribution of sensed noise data and complicates generation of comprehensive urban noise maps. From a sparse (both in time and space) nighttime data set it is difficult to obtain enough statistics to run analysis tools (e.g., compressive sensing) for participatory noise mapping. Lack of nighttime observations also proves that the common claim about better temporal granularity presented in favor of participatory noise mapping over WSN monitoring does not hold generally.

NoiseTube measurement campaigns were executed with mobile phones which used no windscreens on their microphones. In these field measurements wind was shown to cause almost 10 dB errors in the worst case on an LAeq,4min noise level. An observation that the wind induced error is not seen any more if an LAeq noise indicator is calculated for a 25 or 81 minute long walk was regarded by the NoiseTube group as a 'surprisingly good result'. This is good news, if a NoiseTube phone is used as a personal noise dose meter. During a longer period (1–8 h) wind induced errors are averaged away and the daily noise dose may be rather close to a correct value. However, if the aim is to allow a NoiseTube user to participate in collective noise mapping, half an hour or one hour averages are not any more applicable. Suppose a pedestrian needs to walk 25 minutes before the LAeq value settles close to the correct level. This would mean over a two kilometer long route assuming a walking speed of 5 km/h. An LAeq,25min value would not have much use in noise mapping, because it could only be used if the grid size would be 2 km * 2 km or larger (cf. the 40 m * 40 m grid size applied in the analysis of the Ademloos noise mapping experiment (Phase 2) which was explained at the end of subsection 1.4.2). Depending on the intended application type of NoiseTube wind induced errors may need to be taken into account. As a personal noise dose meter a NoiseTube phone might cope without a windscreen. In contrast, a collaborative noise mapping application would require the NoiseTube phone to be equipped with a windscreen.

One possible application type for participatory noise mapping is in prioritizing environmental noise abatement actions. Participatory noise maps could provide indication of those urban routes which are taken more frequently by pedestrians or cyclists. This information would help to estimate the number of citizens exposed on environmental noise in different urban areas. Consequently, scarce resources in environmental noise abatement could be allocated more efficiently.

The Ennowa group has also conducted measurements to evaluate deviations in mobile phone sensing characteristics among a set of identical phones. Their observations were consistent with NoiseTube calibration experiments stressing the importance to apply individual calibration settings for each phone – even if they would be of the same model from a specific manufacturer (cf. the beginning of subsection 1.4.2). In general, it can be stated that calibration still requires extra development efforts (both on WSN and PSN systems) even if several groups have already worked extensively on it.

Based on this subsection, it can be concluded that participatory noise mapping still has some unresolved issues. Hence, other methods like WSN systems needs to be developed further. One component applicable in this development is the A-weighting filter implemented on ARM. This filter can also be embedded into PSN smartphones. The use of the ARM assembly routine in A-weighting answers some open challenges of PSN implementations (e.g., need for a Class 1 compliant A-weighting response, urge to optimize CPU usage, requirement for lower power consumption).

9.4 Environmental Noise Assessment Methods

Criticism presented by PSN developers on the coverage of the END initiated strategic noise maps is at least partly justified. By taking for example a look at the strategic noise map of Finland on the NOISE web service (accessible via the address given at the end of Section 1.2) we notice almost a black hole on this part of the European map. Except for the Helsinki region hardly any colored areas can be found for different types of noise sources in the map. Only by observing Noise Contour Maps for Major Roads Lden indicator we are able to find noise mapping info also outside the Helsinki region. Lden noise levels have been reported to the European noise database also from roads around Turku, Tampere, Jyväskylä, Oulu, Rovaniemi, Kuopio, Joensuu, Savonlinna, Mikkeli and Kotka. Still, the strategic noise map for Finland is quite sparse – both in spatial as well as in temporal sense. The main reason for the lack of spatial coverage in the Finnish strategic noise map is that the only agglomeration in Finland exceeding the END phase 1 population limit of 250000 can be found around Helsinki. Therefore, the Helsinki region was the only area in Finland which had to compile an agglomeration wide strategic noise map fulfilling the 2007 requirements of the END. A few active spots in the Finnish noise map outside the Helsinki region are related to major roads passing through the aforementioned ten cities. Compilation of noise contour maps for these locations is not due to obligations set on agglomeration, but on major roads which deliver six million vehicle passages per year. For this reason, these noise maps are provided by the national Finnish Transport Agency instead of regional authorities.

Temporal sparseness of these strategic noise maps is another issue criticized by PSN developers. Support for this criticism can be given by taking an example of update intervals related to strategic noise maps we. At the time of this writing (May 15, 2013) the current END initiated strategic noise map compiled for the complete European Union area presents noise exposure data and noise contour maps for which data is reported by national environmental authorities on October 31, 2012. Furthermore, noise data reported by these national authorities does not describe the environmental noise situation that was prevailing at the end of October. Before these authorities were able to report noise exposure data for the NOISE map there must have been a period of computational strategic noise map simulations. These simulations, in turn, are based on environmental noise measurements or traffic statistics evaluated on an earlier date. As a consequence, the currently accessible NOISE map shows a snapshot of an environmental noise situation that dates back at least half a year, in worst cases more than five years.

Currently, four different methods (i.e., traditional SLMs, WSNs, PSNs and computational noise modeling SW) are available for environmental noise monitoring and noise mapping. All of them have their own strengths and weaknesses. Hence, none of them alone can provide the definitive answer to the problem of environmental noise assessment. This is why we suggest a combination of all these approaches to be used in order to stepwise strive toward more accurate environmental noise mapping. As a first phase of the suggested process current mobile phone sensing systems (e.g., NoiseTube, NoiseTubePrime, Ear-Phone) would identify urban areas where environmental noise levels potentially exceed the recommended limits. In the second step those users who have participated actively in collaborative noise mapping inside the potential problem areas are provided with external acoustic sensor boards (cf. Section 1.4.3) for their disposal. Sensor nodes in this case would comprise of the users' smartphone accompanied with an external smart microphone board. With this arrangement accuracy of environmental noise estimates aggregated by these participatory sensor networks could be increased. The third phase of the suggested gradually increasing accuracy process would concentrate on those urban regions which were verified in the second phase to violate – with high probability – against the END. Static WSN measurement systems could be deployed on these problem areas to continuously monitor environmental noise levels. WSN nodes installed on these strategic points would resolve the temporal sparseness issue by acquiring noise indicator data all the time. Finally, in the fourth step the most relevant WSN nodes could be selected to be used as real-time data providers for noise map generation. Measurement data acquired by these WSN nodes could be input to computational noise modeling software which would simulate the propagation of environmental noise from measurement locations to surrounding urban neighborhood and generate close to real-time strategic noise maps. In all phases of the described process high-quality Class 1 SLMs could be utilized as a reference against which the accuracy and reliability of other methods could be estimated and compared. In Finland the tasks in these four environmental noise assessment methods could, for example, be integrated into Tarkkailija alert, Harava enquiry and Liiteri information services[19].

[19] http://www.ym.fi/en-US/Land_use_and_building/Programmes_and_strategies/eServices_for_Housing_and_Building_SADe/eServices_for_Housing_and_Building

9.5 Future work

The principal aim of this thesis was to implement a digital A-weighting filter to be used on an embedded ARM platform as a first step toward digital processing of environmental noise samples. A-weighting is one of the frequency weightings defined in standards applied on environmental noise measurements. As explained in Section 5.1 two similar frequency weightings, namely B-weighting and C-weighting, also exist even if A-weighting is the most commonly used one. However, these two additional frequency weightings have simpler definitions (cf. Equations 5.1 and 5.2) and can therefore be implemented in a more straightforward way. Implementation of the B- and C-weighting filters is first of the future development tasks for digital processing of environmental noise samples.

Even if utilization of A-weighting in environment noise measurements has evolved to a de facto stardard a few words of criticism are still justified. One should remember that A-weighted noise indicators measured in environmental noise inventories are averages over the complete frequency range of human hearing. For example standard measurements of equivalent A-weighted sound pressure level – that is, LAeq – hide frequency characteristic details of the measured signal. Even with equal LAeq indicator values the frequency contents of the actual physical sound pressure signal in different measurement locations may vary considerably. Consider a situation where one of the measured signals is flat in regard to intensities on different frequencies whereas another signal possesses an intense peak around a certain frequency. Human response to these two environmental noise signals may differ because some frequencies arise more annoyance than others. Therefore, a single A-weighted noise indicator may be misleading if noise annoyance is to be assessed.

In the future we plan to address the aforementioned problem by realizing more accurate assessment methods. Instead of just generating a single A-weighted broadband noise indicator to assess environmental noise we want to provide more detailed measurement results for the frequency content (e.g., (fractional) octave band histograms and narrowband frequency spectra). Final goal for the work described in this thesis is to utilize it in the development of a wireless measurement system for long-term monitoring of environmental noise. The development steps envisioned for the earlier mentioned Dimeca project are as follows:

* Phase 1: a system to produce basic environmental noise indicators

* Phase 2: a system capable of generating octave and fractional octave band histograms

* Phase 3: a system which enables creation of frequency spectra utilizing either FFT or wavelet transforms

Hansa Ecuras Oy is continuing development of the aforementioned phases of Dimeca. One research partner already found for this development is the Industrial Ontologies Group at the Agora Center in the University of Jyväskylä. The UBIWARE concept developed by the IOG group provides smart semantic middleware for ubiquitous computing applications. In Dimeca the UBIWARE tools could be applied to implement a hardware independent middleware between measurement stations and the measurement center. Different phases of Dimeca are going to realize measurement stations with several types of hardware components and protocol versions. The UBIWARE middleware enables to isolate the hardware and firmware details from the upper layers of the Dimeca architecture. Advantage of this is that the evolution of Dimeca can be achieved without much additional effort on the measurement center side when hardware and firmware improvements become available on the measurement station or sensor level. The UBIWARE tools might even provide automatic reconfigurability of the Dimeca system at some extent.

Application areas for the Dimeca system would be long-term monitoring of road traffic and railway noise as well as noise around airports. With additional sensors Dimeca could also be used to study environmental noise induced effects on human health. By equipping Dimeca with wireless sensors to detect changed of test persons physiological response (blood pressure, level of stress hormones etc.) it would be possible to map the influence of noise on sleep disturbances. This way testing could be performed in a person's own home reducing the error sources due to unfamiliar laboratory environment.

Neither processing of inner ear evoked neural impulses nor psychological effects caused by environmental noise (annoyance, changes in social behavior, etc.) were considered in this thesis. These aspects are at least equally important as the physiological ones. Within the scope of this research, however, it was not possible to consider the psychological effects of environmental noise on human well-being. Even if these effects are extremely important we had to set a boundary on a point where the neural auditory system receives the environmental noise generate electric impulses. Therefore, this thesis concentrates only on auditory signal processing up to the inner hair cells where the action potentials are evoked.

One aim of this thesis is to act like a glue and arise interest of experts in multiple disciplines to come together and strive to tackle this wide-spread and wicked environmental problem. Global monitoring of the environmental noise situation could increase knowledge of the severity of this problem and stress the importance of environmental noise abatement actions. The writer sincerely hopes that this thesis can initiate a multidisciplinary cooperation to develop a large scale monitoring system for environmental noise issues.

YHTEENVETO (FINNISH SUMMARY)

Ympäristömelu on yhä kasvava ongelma, joka on otettu huomioon viime vuosina myös eurooppalaisessa ympäristölainsäädännössä. Ympäristömelutilanteen seurannan velvoitteet lisääntyivät vuonna 2012, kun Euroopan Unionin ympäristömeludirektiivin toinen vaihe astui voimaan. Kyseinen direktiivi velvoittaa yli 100000 asukkaan väestökeskittymät raportoimaan ympäristömelun vuotuiset keskiarvot Euroopan komissiolle ja päivittämään tehdyt arviot viiden vuoden välein. Elinympäristömme tilan seurantaan on tähän mennessä kehitetty menetelmiä, joiden avulla pystytään monitoroimaan hitaasti vaihtelevien ympäristöparametrien muutoksia. Tällaisten parametrien (ilman lämpötila, suhteellinen ilmankosteus, ilmanlaatuun vaikuttavat hiukkaspitoisuudet jne.) seurantaa varten on jo olemassa laajoilla alueilla toimivia mittaus- ja seurantajärjestelmiä. Ympäristömelu on kuitenkin huomattavasti vaativampi fysikaalinen suure monitoroitavaksi sisältämänsä suuren informaatiomäärän ja nopeiden vaihteluidensa vuoksi. Tämän lisensiaattityön (YMPÄRISTÖMELUSIGNAALIEN DIGITAALINEN KÄSITTELY) tarkoituksena on esitellä ratkaisu, jolla ympäristömelun hajautettuun pitkäaikaiseen seurantaan soveltuvaan järjestelmään vaadittava yksi peruskomponentti pystytään toteuttamaan virrankulutukseltaan pienellä sulautetulla prosessorialustalla.

Lisensiaattityöni jakautuu kahteen suurempaan kokonaisuuteen. Lukujen 1 – 4 muodostamassa ensimmäisessä osassa käsitellään ympäristömeluun, sen mittaamiseen ja ihmisen kuuloon liittyviä asioita. Tässä osassa on tarkoitus tuoda erityisesti esiin ympäristömeluongelman laajuus, nykyisillä mittausmenetelmillä tehtyjen meluselvitysten riittämättömyys sekä se millä tavalla ihmiskorva rekisteröi ja painottaa ympäristömelussa esiintyviä erilaisia taajuuskomponentteja. Ympäristömelun haittoja arvioitaessa on tärkeää ottaa huomioon ihmisen fysiologian vaikutus ympäristömelusignaalien havaitsemiseen. Yksi tunnusomainen piirre ihmiskorvalle on sen herkkyyden muuttuminen ympäristömelun taajuuden funktiona. Ihmiskorva painottaa havaitsemaansa ympäristömelua siten, että kuuloalueen alapäässä (~ 20–100 Hz) ja yläpäässä (~ 10–20 kHz) olevia taajuuksia vaimennetaan voimakkaasti. Tässä suhteessa ihmiskorva eroaa selvästi sähköisistä mittalaitteista, joilla on lineaarinen taajuusvaste. Ihmiskorvan epälineaarista taajuusvastetta kuvaamaan on kehitetty erilaisia painotuskäyriä, joista ns. A–painotus on vakiintunut yleisimmin käytettäväksi. Ympäristömelua mittaavissa laitteissa ja järjestelmissä on siis oltava mahdollisuus korjata mitatun signaalin taajuusvaste vastaamaan ihmiskorvan havaintoa.

Lisensiaattityöni ensimmäinen osa sisältää myös lyhyen kuvauksen laajemmasta ympäristömelun hajautetusta mittausjärjestelmästä, jonka kehittämisessä lisensiaattityöni tuloksia voitaisiin hyödyntää. Esimerkiksi lämpötilan monitorointiin verrattuna ympäristömelun seuranta vaatii luonteensa vuoksi paljon enemmän tiedonkäsittelykapasiteettia. Ympäristömelun pitkäaikainen monitorointi hajautetusti laajalla alueella asettaa kehitettävälle järjestelmälle reunaehtoja, jotka ovat ristiriidassa suuren prosessointikapasiteetin tarpeen kanssa. Seurantajärjestelmän tulisi olla toteutettavissa vähän tehoa vaativilla komponenteilla, jotta monitorointia voitaisiin jatkaa yhtäjaksoisesti ilman huolto- ja ylläpitotoimia pitkiäkin aikoja.

Lisensiaattityöni toisessa osakokonaisuudessa (luvut 5 – 8) keskitytään siihen miten ympäristömelumittauksissa oleellinen A–taajuuspainotus saadaan toteutettua digitaalisesti sulautettua prosessorialustaa käyttäen. Lähtemällä liikkeelle analogiatasossa annetusta A–painotussuotimen kuvauksesta johdetaan aluksi bilineaarisen muunnoksen avulla esitys vastaavalle digitaaliselle suotimelle. Saadun kuudennen kertaluvun rationaalifunktion todetaan täyttävän A–painotussuotimelle asetetut vaatimukset mikäli sen toimintaa simuloidaan MATLAB-ohjelmointiympäristössä käytössä olevalla 64–bitin sananpituudella. Suotimen toteutusvaiheessa hyödynnettävä sulautettu järjestelmä ei kuitenkaan tarjoa mahdollisuutta 64–bitin prosessoriarkkitehtuurin käyttöön. Siksi MATLAB-simulointeja jatkettiin digitaalisen A–painotussuotimen käyttäytymisen selvittämiseksi, kun suotimen siirtofunktiota kuvaavan rationaalifunktion kertoimien tarkkuutta pienennetään. Simulointien tuloksena todettiin, että kuudennen kertaluvun siirtofunktio muuttuu epästabiiliksi, jos suotimen kertoimien tallentamiseen on 64–bitin sijasta käytössä vain 16–bitin pituisia talletuselementtejä. Tällainen reunaehto tulee eteen toteutettaessa suodatusalgoritmia sulautetulla 32–bittisellä prosessorialustalla. A–painotussuotimen suunnittelua jatkettiin hajottamalla kuudennen kertaluvun rationaalifunktio kolmeen toisen kertaluvun komponenttiin. Simuloinnit osoittivat, että liittämällä peräkkäin kaksi näistä toisen kertaluvun komponenteista saadaan aikaan kaskadirakenne, joka pyöristetyilläkin kertoimilla toteuttaa A–painotussuotimelle määritellyn spesifikaation riittävällä tarkkuudella.

Taajuustasossa suoritettujen simulointien jälkeen A–painotussuotimen suunnitelma palautettiin takaisin aikatasoon varsinaista toteutusta varten. Luvussa 7. on käyty läpi vaatimuksia, joita sulautettu prosessorialusta asettaa A–painotuksen käytännön toteutukselle. Simuloinneissa käytössä olleen liukulukuesityksen sijaan digitoitujen äänenpainesignaalin arvojen ja suotimen kertoimien talletusmuoto on muutettu kiinteän pisteen esitykseksi. Perusteena tälle on ollut se, että toteutukseen valitulla ARM prosessorialustalla on pyritty pieneen tehonkulutukseen eikä se sen vuoksi sisällä liukulukuyksikköä. Hyödyntämällä kokonaislukutyyppistä kiinteän pisteen esitystä painotusalgoritmin suoritusaikaa pystytään lyhentämään. Tämä on oleellista mikäli puheena olevaa A–painotustoteutusta halutaan hyödyntää hajautetun langattoman anturiverkon kehittämisessä. Anturiverkon mittausasemilla olevat sulautetut prosessorit voidaan asettaa osaksi aikaa virransäästötilaan, jos pakollinen A–painotusalgoritmi saadaan suoritettua riittävän lyhyessä ajassa.

Digitaalinen toteutus A–painotussuotimelle on tehty ohjelmoimalla MATLAB-simulointien perusteella valittu, kahdesta toisen asteen komponentista muodostettu, kaskadirakenne ARM prosessoriarkkitehtuurin assembly–kielellä. Assembly–kielen käyttö on mahdollistanut ARM v4 arkkitehtuurin liukuhihnan hyödyntämisen siten, ettei suodatusalgoritmin suoritus ole viivästynyt prosessorin odotustilojen takia. Tähän on päästy käynnistämällä ympäristömelusignaalista digitoitujen näytteiden haku ulkoisesta muistista jo etukäteen. Suodatusalgoritmin toteutusta on voitu jatkaa keskeytyksettä prosessorin rekistereissä tallessa olevilla aikaisemmilla syötteillä samalla, kun liukuhihna on kuljettanut uudet syötteet algoritmin edellisistä vaiheista vapautuneisiin rekistereihin. Tällainen käsin suoritettu assembly-kielisten komentojen suoritusjärjestyksen optimointi on ollut välttämätöntä aiemmin mainittujen reunaehtojen täyttämiseksi.

Edellä mainittujen MATLAB-simulointien ja ARMin assembly-kielellä koodatun suodatusalgoritmin avulla tässä työssä on osoitettu, että sulautetulla ARM-alustalla pystytään hyvin pienillä resurssitarpeilla toteuttamaan A-painotus, jota edellytetään käytettävän standardien mukaisissa ympäristömelumittauksissa.

Kokeellisten ympäristömelumittausten, MATLAB-simulointien ja digitaalisen A-painotussuotimen toteutuksen lisäksi tähän työhön sisältyy kirjallisuuskatsaus (ks. aliluku 1.4) uusista langattomista menetelmistä, joita on kokeiltu EU:n ympäristömeludirektiivin edellyttämien strategisten melualuekarttojen tuottamiseen. Osa katsauksessa mainituista menetelmistä perustuu langattomiin anturiverkkoihin, joissa ympäristömelua rekisteröivät langattomat anturisolmut mittaavat ja digitoivat ympäristömelusignaalia jatkuvasti tietyssä kiinteässä asennuspaikassa. Langattomien anturiverkkojen lisäksi digitaalista langatonta tiedonsiirtoa on hyödynnetty myös mittaussovelluksissa, joissa ympäristömelusignaalia monitoroidaan älypuhelinten avulla. Näissä yleisöosallisuuteen (eng. crowdsourcing, participatory sensor networks) perustuvissa verkoissa kännyköiden käyttäjät (osatuottajat) voivat osallistua yhteisöllisesti tuotettujen melualuekarttojen luomiseen. Asentamalla älypuhelimelleen DSP sovelluksen käyttäjä voi muuttaa puhelimensa langattomaksi äänitasomittariksi, josta mitattujen ja digitoitujen äänenpainesignaalien perusteella lasketut äänenpainetasot kerätään Internetissä toimivalle palvelimelle. Tällaiselta julkiselta www-palvelimelta yhteisöllisesti tuotetut melualuekartat ovat kaikkien käytettävissä samalla tavalla kuin EU:n koostamat strategiset melualuekartat.

Kirjallisuuskatsauksessa mainituista uusista menetelmistä löytyy parannettavia osa-alueita, joissa tässä tutkimuksessa ARM-arkkitehtuurille kehitettyä A-painotussuodinta pystytään hyödyntämään. Sekä perinteisillä ympäristömelun mittaustavoilla (Luku 3) että näillä uusilla menetelmillä on omat heikkoutensa ja vahvuutensa. Siksi mikään niistä yksinään ei tarjoa lopullista ratkaisua ympäristömeluselvityksille asetettuihin haasteisiin. Tämän tutkimuksen lopussa on ehdotettu kyseiseen ongelmaan ratkaisumallia, jossa yhdistämällä perinteisten ja uusien menetelmien parhaita ominaisuuksia saataisiin aikaan entistä parempi työkalu ympäristömeluongelman laajuuden ja siitä aiheutuvien haittojen arviointiin sekä toteutettujen meluntorjuntatoimien vaikuttavuuden seurantaan.

REFERENCES

[Aam08] T.M. Aamodt and P. Chow. Compile-Time and Instruction-Set Methods for Improving Floating- to Fixed-Point Conversion Accuracy. *ACM Transactions on Embedded Computing Systems,* 7(3, Article 26):1-27. ISSN: 1539-9087. ACM, April 2008. DOI: dx.doi.org/10.1145/1347375.1347379

[Abb12] S. Abbate, M. Avvenuti, D. Cesarini and A. Vecchio. Estimation of Energy Consumption for TinyOS 2.x-Based Applications. *Procedia Computer Science, ANT 2012 and MobiWIS 2012,* 10(0):1166-1171. ISSN: 1877-0509. Elsevier, August 2012. DOI: dx.doi.org/10.1016/j.procs.2012.06.167

[Aib01] R. Aibara, J.T. Welsh, S. Puria and R.L. Goode. Human Middle-Ear Sound Transfer Function and Cochlear Input Impedance. *Hearing Research,* 152(1-2):100-109. ISSN: 0378-5955. 2001. DOI: dx.doi.org/10.1016/S0378-5955(00) 00240-9

[Alb10] A. Albarbar, F. Gu and A.D. Ball. Diesel Engine Fuel Injection Monitoring using Acoustic Measurements and Independent Component Analysis. *Measurement,* 43(10):1376-1386. ISSN: 0263-2241. Elsevier, December 2010. DOI: dx.doi.org/10.1016/j.measurement.2010.08.003

[Amj00] H. Amjadi. Electret Membranes and Backelectrodes for Application in Micromechanical Transducers. *Journal of Electrostatics,* 48(3-4):179-191. ISSN: 0304-3886. Elsevier, March 2000. DOI: dx.doi.org/10.1016/S0304-3886(99) 00064-9

[Amu05] A. H. Amundsen and R. Klaeboe. *A Nordic Perspective on Noise Reduction at the Source,* pp. 1-48. ISBN: 82-480-0576-3. Institute of Transport Economics, November 2005. https://www.toi.no/publikasjoner/nordisk-perspektiv-paa-stoeyreduksjon-ved-kilden-article18917-8.html (Accessed: December 20, 2013)

[Ana09] G. Anastasi, M. Conti and M. Di Francesco. Extending the Lifetime of Wireless Sensor Networks through Adaptive Sleep. *IEEE Transactions on Industrial Informatics,* 5(3):351-365. ISSN: 1551-3203. IEEE, August 2009. DOI: dx.doi.org/10.1109/TII.2009.2025863

[Ana12] G. Anastasi, E. Bini and G. Lipari. Extracting Data from WSNs: A Service-Oriented Approach. In G. Anastasi, E. Bellini, E. Di Nitto, C. Ghezzi, L. Tanca and E. Zimeo (Eds.): *Methodologies and Technologies for Networked Enterprises, Lecture Notes in Computer Science Volume 7200,* pp. 329-356. ISBN: 978-3-642-31738-5. Springer Berlin Heidelberg, 2012. DOI: dx.doi.org/10.1007/978-3-642-31739-2_17

[And13] H. Andersson, L. Jonsson and M. Ögren. Benefit Measures for Noise Abatement: Calculations for Road and Rail Traffic Noise. *European Transport Research Review,* 5(3):135-148. ISSN: 1867-0717. Springer Berlin Heidelberg, 2013. DOI: dx.doi.org/10.1007/s12544-013-0091-3

[Ans12] J. Ansari. *Design considerations for medium access control in resource constrained embedded wireless networks,* pp. 1-246. RWTH Aachen, Fakultät für Elektrotechnik und Informationstechnik, Lehrstuhl und Institut für Vernetzte Systeme; Aachen, Germany. December 2012. Ph.D. Thesis.

http://darwin.bth.rwth-aachen.de/opus3/volltexte/2012/4283

[ARM12] We Are Connecting Intelligence. *Annual Report & Accounts 2012,* pp. 1-156. ARM Holdings, February 2013. http://financialreports.arm.com/pdf/ARM_Reporting_2012.pdf (Accessed: December 09, 2013)

[Aya11] A. Ayadi. Energy-Efficient and Reliable Transport Protocols for Wireless Sensor Networks: State-of-Art. *Wireless Sensor Network,* 3(3):106-113. ISSN: 1945-3086. Scientific Research Publishing, March 2011.
DOI: dx.doi.org/10.4236/wsn.2011.33011

[Bac12] E. Baccelli, O. Hahm, M. Wählisch, M. Günes and T. Schmidt. RIOT: One OS to Rule them all in the IoT. *Research Report,* (RR-8176):1-16. ISSN: 0249-6399. INRIA, December 2012. http://hal.inria.fr/hal-00768685/

[Bai06] G.S. Baicher. Optimization of Finite Word Length Coefficient IIR Digital Filters Through Genetic Algorithms – A Comparative Study. In L. Jiao, L. Wang, X. Gao, J. Liu and F. Wu (Eds.): *Advances in Natural Computation, Lecture Notes in Computer Science Volume 4222,* pp. 641-650. ISBN: 978-3-540-45909-5. Springer Berlin Heidelberg, September 2006.
DOI: dx.doi.org/10.1007/11881223_79

[Baj11] N.B. Bajema, J. Trevathan, N.W. Bergmann, I. Atkinson, W. Read, A. Scarr, Yong Jin Lee and R. Johnstone. Benefits of building wireless sensor networks on commodity hardware and software stacks. *In Proceedings of the Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2011)*, pp. 282-287. IEEE, December 2011.
DOI: dx.doi.org/10.1109/ISSNIP.2011.6146554

[Bek60] G. von Békésy and E.G. Vreber. *Experiment in Hearing,* pp. 1-745. McGraw-Hill, 1960.

[Ber12] L.L. Beranek and T. Mellow. *Acoustics: Sound Fields and Transducers,* pp. 1-720. ISBN: 978-0-12-391421-7. Academic Press, September 2012.

[Bha05] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson and R. Han. MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mobile Networks and Applications,* 10(4):563-579. ISSN: 1383-469X. Kluwer Academic Publishers, August 2005. DOI: dx.doi.org/10.1007/s11036-005-1567-8

[BK94] Brüel & Kjær. *Measurement Microphones (2nd ed),* pp. 1-37. BR 0567-12. Brüel&Kjær Sound&Vibration Measurement A/S, Nærum, Denmark, August 1994. http://www.bksv.com/doc/br0567.pdf (Accessed: December 11, 2013)

[BK98a] Brüel & Kjær. *Brüel & Kjær 2238 Mediator, Integrating Sound Level Meter, Basic SLM Software BZ 7126, User Manual BB1119-11.* Brüel&Kjær Sound&Vibration Measurement A/S, Nærum, Denmark, August 1998.

[BK98b] Brüel & Kjær. *Brüel & Kjær 2238 Mediator, Integrating Sound Level Meter, Logging SLM Software BZ 7124, User Manual BB1125-11.* Brüel&Kjær Sound& Vibration Measurement A/S, Nærum, Denmark, November 1998.

[BK06] Brüel & Kjær. *Sound Calibrator Type 4231, Product Data,* pp. 1-4. BP 1311-16. Brüel&Kjær Sound&Vibration Measurement A/S, Nærum, Denmark, November 2006. http://www.bksv.com/doc/bp1311.pdf (Accessed: December 6, 2013)

[BK12] Brüel & Kjær. *Transducers and Conditioning, Issue 17,* pp. 1-176. BF 0236-11. Brüel&Kjær Sound&Vibration Measurement A/S, Nærum, Denmark, 2012. http://www.bksv.com/doc/bf0236.pdf (Accessed: December 11, 2013)

[Bla11] N. Blanes and C. Nugent. Noise Observation and Information Service for Europe – NOISE. *Implementation of Environmental Noise Directive, its Next Phase and Exploitation of Results in Europe,* pp. 1-34. Keynote speech at the 12th National Noise Abatement Days in Jyväskylä. Center for Economic Development, Transport and the Environment, Finland, March 2011. http://ely-centralen.fi/fi/uutiset/uutiset_alueilta/Sivut/Meluntorjuntapaivien_esitykset_ja_valokuvia_tapahtumasta.aspx (Accessed: December 3, 2013)

[Bon07] X. Bonnefoy, M. Braubach, M. Davidson and N. Robbel. A Pan-European Housing and Health Survey: Description and Evaluation of Methods and Approaches. *International Journal of Environment and Pollution,* 30(3):363-383. ISSN: 1741-5101. Inderscience Publishers, August 2007. http://inderscience.metapress.com/content/0779H76P35337V4N

[Bou11] T. Bourdenas, D. Wood, P. Zerfos, F. Bergamaschi and M. Sloman. Self-adaptive routing in multi-hop sensor networks. *In Proceedings of the 7th International Conference on Network and Service Management (CNSM 2011)*, pp. 1-9. ISBN: 978-1-4577-1588-4. IEEE, October 2011. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6103949

[Bra04] L.S. Brakmo, D.A. Wallach and M.A. Viredaz. µSleep: a Technique for Reducing Energy Consumption in Handheld Devices. *In Proceedings of the 2nd inter-national conference on Mobile systems, applications, and services (MobiSys '04)*, pp. 12-22. ISBN: 1-58113-793-1. ACM, June 2004.
DOI: dx.doi.org/10.1145/990064.990069

[Bra07] W. Bracke, P. Merken, R. Puers and C.V. Hoof. Generic Architectures and Design Methods for Autonomous Sensors. *Sensors and Actuators A: Physical,* 135(2):881-888. ISSN: 0924-4247. Elsevier, April 2007.
DOI: dx.doi.org/10.1016/j.sna.2006.07.028

[Brö11] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang and R. Lemmens. New Generation Sensor Web Enablement. *Sensors,* 11(3): 2652-2699. ISSN: 1424-8220. MDPI AG, March 2011.
DOI: dx.doi.org/10.3390/s110302652

[Bud12] M. Budde, M. Berning, M. Busse, T. Miyaki and M. Beigl. The TECO Envboard: A mobile sensor platform for accurate urban sensing – and more. *In Proceedings of the Ninth Interna-tional Conference on Networked Sensing Systems (INSS 2012)*, pp. 1-2. ISBN: 978-1-4673-1784-9. IEEE, June 2012.
DOI: dx.doi.org/10.1109/INSS.2012.6240573

[Bue06] M. Buettner, G.V. Yee, E. Anderson and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. *In Proceedings of the 4th international conference on Em-bedded networked sensor systems (SenSys '06)*, pp. 307-320. ISBN: 1-59593-343-3. ACM, 2006.
DOI: dx.doi.org/10.1145/1182807.1182838

[Bur14] C. Buratti, E. Belloni and E. Moretti. Façade Noise Abatement Prediction: New Spectrum Adaptation Terms Measured in Field in Different Road and Railway Traffic Conditions. *Applied Acoustics,* 76(0):238-248. ISSN:0003-682X

Elsevier, February 2014. DOI: dx.doi.org/10.1016/j.apacoust. 2013.08.016

[Bur07] N. Burri, P. von Rickenbach and R. Wattenhofer. Dozer: Ultra-Low Power Data Gathering in Sensor Networks. *In Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN 2007)*, pp. 450-459. ISBN: 978-1-59593-638-7. IEEE, April 2007. DOI: dx.doi.org/10.1109/IPSN.2007.4379705

[Cam12] A. Camillo and C. Petrioli. Hands on IRIS: Lessons learned from implementing a cross layer protocol stack for WSNs. *In Proceedings of the 55th IEEE Global Telecommunications Conference (GLOBECOM 2012)*, pp. 175-181. ISBN: 978-1-4673-0919-6. IEEE, December 2012. http://www-mobile.ecs.-soton.ac.uk/home/conference/Globecom_12/papers/p175-camillo.pdf

[Can12] A. Cano, J.L. Añón, C. Reig, C. Millán-Scheiding and E. López-Baeza. Automated Soil Moisture Monitoring Wireless Sensor Network for Long-Term Cal/Val Applications. *Wireless Sensor Network,* 4(8):202-209. ISSN: 1945-3086. Scientific Research Publishing, August 2012. DOI: dx.doi.org/10.4236/wsn.2012.48030

[Cer01] F. Cervera, L. Sanchis, J.V. S\'anchez-P\'erez, R. Mart\'\inez-Sala, C. Rubio, F. Meseguer, C. L\'opez, D. Caballero and J. S\'anchez-Dehesa. Refractive Acoustic Devices for Airborne Sound. *Phys.Rev.Lett.,* 88(2):023902. American Physical Society, December 2001. DOI: dx.doi.org/10.1103/PhysRevLett.88.023902

[Che09] C.-Y. Chen, Y.-T. Chen, Y.-H. Tu, S.-Y. Yang and P.H. Chou. EcoSpire: An Application Development Kit for an Ultra-Compact Wireless Sensing System. *IEEE Embedded Systems Letters,* 1(3):65-68. ISSN: 1943-0663. IEEE, October 2009. DOI: dx.doi.org/10.1109/LES.2009.2037984

[Che10] Y. Chen, T. Chien and P.H. Chou. Enix: a lightweight dynamic operating system for tightly constrained wireless sensor platforms. *In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 183-196. ISBN: 978-1-4503-0344-6. ACM, 2010. DOI: dx.doi.org/10.1145/1869983.1870002

[Chh14] D. Chhan, C. Thompson and K. Aho. Two-Dimensional Analysis of Fluid Motion in the Cochlea Resulting from Compressional Bone Conduction. *Journal of Sound and Vibration,* 333(3):1067-1078. ISSN: 0022-460X. Elsevier, February 2014. DOI: dx.doi.org/10.1016/j.jsv.2013.09.031

[CHI10] CHILImodule Datasheet. *Doc. No 0S4-0000-00100H.* CHILIdevices International, 2010. http://www.chilidevices.com/site/products/documents/datasheets/0S4-0000-00100H%20-%20CHILImodule%20Datasheet.pdf (Accessed: February 12, 2013)

[Chr11] D. Christin and M. Hollick. We must move – We will move: On mobile phones as sensing platforms. *In Proceedings of the 10th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, pp. 25-28. TU Darmstadt, September 2011. http://www.seemoo.tu-darmstadt.de/dl/seemoo/FGSN11.pdf

[Col11] U.M. Colesanti, S. Santini and A. Vitaletti. DISSense: An adaptive ultralow-power communication protocol for wireless sensor networks. *In Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS 2011)*, pp. 1-10. IEEE, June 2011.

DOI: dx.doi.org/10.1109/DCOSS.2011.5982162

[CRO07] Crossbow. Crossbow Announces Crossbow Imote2, the Most Powerful Mote Platform for Wireless Sensor Network Research. *Crossbow Imote2, Press release.* Crossbow Technology Inc, January 23, 2007. http://www.businesswire.com/news/home/20070123005191/en/Crossbow-Announces-Crossbow-Imote2-Powerful-Mote-Platform (Accessed: February 13, 2013)

[Cul04] D. Culler, D. Estrin and M. Srivastava. Guest Editors' Introduction: Overview of Sensor Networks. *Computer,* 37(8):41-49. ISSN: 0018-9162. IEEE Computer Society, August 2004. DOI: dx.doi.org/10.1109/MC.2004.93

[Dai10] C. Dai, W. Chen and Y. Zhu. Seeker Optimization Algorithm for Digital IIR Filter Design. *IEEE Transactions on Industrial Electronics,* 57(5):1710-1718. ISSN: 0278-0046. IEEE, May 2010.
DOI: dx.doi.org/10.1109/TIE.2009.2031194

[Dau12] S. Dauwe, T. Van Renterghem, D. Botteldooren and B. Dhoedt. Multiagent-Based Data Fusion in Environmental Monitoring Networks. *International Journal of Distributed Sensor Networks,* 2012(Article ID 324935):1-15. ISSN: 1550-1477. Hindawi Publishing, 2012.
DOI: dx.doi.org/10.1155/2012/324935

[Dho13] E. D'Hondt, M. Stevens and A. Jacobs. Participatory Noise Mapping Works! An Evaluation of Participatory Sensing as an Alternative to Standard Techniques for Environmental Monitoring. *Pervasive and Mobile Computing (Special issue on Pervasive Urban Applications), 9(5):681-694*. ISSN: 1574-1192.
Elsevier, October 2013. DOI: dx.doi.org/10.1016/j.pmcj.2012.09.002

[DiM12] P. Di Marco, P. Park, C. Fischione and K.H. Johansson. Analytical Modeling of Multi-Hop IEEE 802.15.4 Networks. *IEEE Transactions on Vehicular Technology,* 61(7):3191-3208. ISSN: 0018-9545. IEEE, September 2012.
DOI: dx.doi.org/10.1109/TVT.2012.2201221

[Dol06] G.J. Dolecek and S.K. Mitra. Symbolic Sensitivity Analysis of IIR Digital Filters using MATLAB. *International Journal of Control,* 79(11):1331-1339. ISSN: 1366–5820. Taylor & Francis, March 2006.
DOI: dx.doi.org/10.1080/00207170600725594

[Don10] W. Dong, C. Chen, X. Liu and J. Bu. Providing OS Support for Wireless Sensor Networks: Challenges and Approaches. *IEEE Communications Surveys & Tutorials,* 12(4):519-530. ISSN: 1553-877X. IEEE, November 2010.
DOI: dx.doi.org/10.1109/SURV.2010.032610.00045

[Dou13] M. Doudou, D. Djenouri and N. Badache. Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials, 15*(2):528-550. ISSN: 1553-877X. IEEE, May 2013. DOI: dx.doi.org/10.1109/SURV.2012.040412.00075

[Dro12] G. Drosatos, P.S. Efraimidis, I.N. Athanasiadis, E. D'Hondt and M. Stevens. A Privacy-Preserving Cloud Computing System for Creating Participatory Noise Maps. *In Proceedings of the 36th Annual Computer Software and Applications Conference (COMPSAC 2012)*, pp. 581-586. ISSN: 0730-3157. IEEE, July 2012. DOI: dx.doi.org/10.1109/COMPSAC.2012.78

[Dun04] A. Dunkels, B. Gronvall and T. Voigt. Contiki – a lightweight and flexible operating system for tiny networked sensors. *In Proceedings of the 29th An-*

nual IEEE International Conference on Local Computer Networks, 2004, pp. 455-462. ISSN: 0742-1303. IEEE, November 2004.
DOI: dx.doi.org/10.1109/LCN.2004.38

[Dut10] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.M. Liang and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. *In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 1-14. ISBN: 978-1-4503-0344-6. ACM, November 2010. DOI: dx.doi.org/10.1145/1869983.1869985

[Dwi09] A.K. Dwivedi, M.K. Tiwari and O.P. Vyas. Operating Systems for Tiny Networked Sensors: A Survey. *International Journal of Recent Trends in Engineering (IJRTE),* 1(2):152-157. ISSN: 1797-9617. Academy Publisher, May 2009. http://ijrte.academypublisher.com/vol01/no02/ijrte0102152157.pdf

[Dyo12] V. Dyo, S.A. Ellwood, D.W. Macdonald, A. Markham, N. Trigoni, R. Wohlers, C. Mascolo, B. Pásztor, S. Scellato and K. Yousef. WILDSENSING: Design and Deployment of a Sustainable Sensor Network for Wildlife Monitoring. *ACM Transactions on Sensor Networks (TOSN),* 8(4, Article No. 29):1-33. ISSN: 1550-4859. ACM, September 2012.
DOI: dx.doi.org/10.1145/2240116.2240118

[ECE09] Making THE Link: Transport Choices for our Health, Environment and Pros-perity. TRANSPORT HEALTH AND ENVIRONMENT PAN-EURO-PEAN PROGRAMME (THE PEP). *Third High-level Meeting on Transport, Health and Environment.* United Nation's Economic Commission for Europe / WHO Europe, January 2009. http://www.unece.org/fileadmin/DAM/thepep/en/hlm/documents/2009/Amsterdam_Declaration_ENG.pdf (Accessed: November 30, 2013)

[EEC86] Council Directive 86/188/EEC of 12th May 1986 on the protection of workers from the risks related to noise at work. *Official Journal of the European Communities, L 137, 24/5/1986,* pp. 28-34. The Council of the European Communities, 1986. http://new.eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.1986.137.01.0028.01.ENG

[ENR13] Eurostat. Second estimate for the second quarter of 2013. *Eurostat News Release 130/2013 issued on 4 September 2013, Euroindicators,* pp. 1-6. Eurostat, September 2013. http://epp.eurostat.ec.europa.eu/cache/ITY_PUBLIC/2-04092013-BP/EN/2-04092013-BP-EN.PDF (Accessed: November 29, 2013)

[EPC02] Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise. *Official Journal of the European Communities, L 189, 18/07/2002,* pp. 12–25. Publications Office of the European Union, July 2002. http://new.eur-lex.europa.eu/legal-content/EN/TXT/?qid=1385628985019&uri=CELEX:32002L0049

[EPC03] Directive 2003/10/EC of the European Parliament and of the Council of 6 February 2003 on the minimum health and safety requirements regarding the exposure of workers to the risks arising from physical agents (noise). *Official Journal of the European Union, L 42, 15/2/2003,* pp. 38-44. Publications Office of the European Union, February 2003. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2003:042:0038:0044:EN:PDF

[Far11] M.O. Farooq and T. Kunz. Operating Systems for Wireless Sensor Networks: A Survey. *Sensors,* 11(6):5900-5930. ISSN: 1424-8220. MDPI AG, May 2011. DOI: dx.doi.org/10.3390/s110605900

[Fer12] F. Ferrari, M. Zimmerling, L. Mottola and L. Thiele. Low-power wireless bus. *In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pp. 1-14. ISBN: 978-1-4503-1169-4. ACM, November 2012. DOI: dx.doi.org/10.1145/2426656.2426658

[Fil08] L. Filipponi, S. Santini and A. Vitaletti. Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring. In S. Nikoletseas, B. Chlebus, D. Johnson and B. Krishnamachari (Eds.): *Distributed Computing in Sensor Systems, Lecture Notes in Computer Science Volume 5067*, pp. 492-497. ISBN: 978-3-540-69169-3. Springer Berlin Heidelberg, 2008.
DOI: dx.doi.org/10.1007/978-3-540-69170-9_35

[Fis12] D.K. Fisher and P.J. Gould. Open-Source Hardware is a Low-Cost Alternative for Scientific Instrumentation and Research. *Modern Instrumentation,* 1(2):8-20. ISSN: 0090-6778. Scientific Research Publishing, April 2012.
DOI: dx.doi.org/10.4236/mi.2012.12002

[Gao11] D. Gao, L. Zhang and H. Wang. Energy Saving with Node Sleep and Power Control Mechanisms for Wireless Sensor Networks. *The Journal of China Universities of Posts and Telecommunications,* 18(1):49-59. ISSN: 1005-8885. Elsevier, February 2011. DOI: dx.doi.org/10.1016/S1005-8885(10)60027-1

[Gao12] C. Gao, I. Kivelä, X. Tan and I. Hakala. A Transmission Scheduling for Data-Gathering Wireless Sensor Networks. *In Proceedings of the 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC 2012)*, pp. 292-297. ISBN: 978-1-4673-3084-8. IEEE, September 2012. DOI: dx.doi.org/10.1109/UIC-ATC.2012.48

[Gar12] N. Garg and O. Sharma. Measurement Accuracy of Secondary Standards of Sound Pressure in Comparison to Primary Standards. *MAPAN – Journal of Metrology Society of India,* 27(4): 219-229. ISSN: 0970-3950. Springer-Verlag, December 2012. DOI: dx.doi.org/10.1007/s12647-012-0033-x

[Gna09] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis. Collection tree protocol. *In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1-14. ISBN: 978-1-60558-519-2. ACM, November 2009. DOI: dx.doi.org/10.1145/1644038.1644040

[Goo94] R.L. Goode, M. Killion, K. Nakamura and S. Nishihara. New Knowledge about the Function of the Human Middle Ear: Development of an Improved Analog Model. *American Journal of Otology,* 15(2):145-154. Lippincott Williams & Wilkins, March 1994. http://www.etymotic.com/publications/erl-0028-1994.pdf (Accessed: November 26, 2013)

[Gra01] J. Gramtorp and E. Frederiksen. Frequency Response for Measurement Microphones – a Question of Confidence. In H.K. Zaveri (Ed.): *Brüel&Kjær Technical Review 2001–1 Properties and Calibration of Laboratory Standard Microphones; Uncertainties in Microphone Frequency Responses,* pp. 24-35. ISSN: 0007–2621. Brüel&Kjær Sound&Vibration Measurement A/S, Nærum, Denmark, 2001. http://bksv.com/doc/bv0054.pdf (Accessed: December 06, 2013)

[Gree96] D.D. Greenwood. Comparing Octaves, Frequency Ranges, and Cochlear-Map Curvature Across Species. *Hearing Research,* 94(1-2):157-162. ISSN: 0378-5955. Elsevier, May 1996. DOI: dx.doi.org/http://dx.doi.org/10.1016/0378-5955(95)00229-4

[Gun10] V.C. Gungor, Bin Lu and G.P. Hancke. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *IEEE Transactions on Industrial Electronics,* 57(10):3557-3564. ISSN: 0278-0046. IEEE, October 2010. DOI: dx.doi.org/10.1109/TIE.2009.2039455

[Hak06] I. Hakala and M. Tikkakoski. From vertical to horizontal architecture: a cross-layer implementation in a sensor network node. *In Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense '06),* pp. 1-6. ISBN:1-59593-427-8. ACM, May 2006. DOI: dx.doi.org/10.1145/1142680.1142688

[Hak10a] I. Hakala, T. Hongell and J. Luomala. CiNetView – Graphic Interface for Wireless Sensor Network Deployment and Monitoring. *In Proceedings of the Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010),* pp. 395-401. IEEE, July 2010. DOI: dx.doi.org/10.1109/SENSORCOMM.2010.65

[Hak10b] I. Hakala, I. Kivelä, J. Ihalainen, J. Luomala and C. Gao. Design of Low-Cost Noise Measurement Sensor Network: Sensor Function Design. *In Proceedings of the First International Conference on Sensor Device Technologies and Applications (SENSORDEVICES 2010)*, pp. 172-179. IEEE, July 2010. DOI: dx.doi.org/10.1109/SENSORDEVICES.2010.39

[Hak12] I. Hakala and T. Hongell. Neighbourtables – A Cross-Layer Solution for Wireless CiNet Network Analysis and Diagnostics. *Sensors & Transducers Journal,* 14-2(3):228-241. ISSN: 1726-5479. International Frequency Sensor Association, March 2012. http://www.sensorsportal.com/HTML/DIGEST/P_SI_211.htm (Accessed: September 09, 2012)

[Hal12] J.I. Halonen, J. Vahtera, S. Stansfeld, T. Yli-Tuomi, P. Salo, J. Pentti, M. Kivimäki and T. Lanki. Associations between Nighttime Traffic Noise and Sleep: The Finnish Public Sector Study. *Environmental Health Perspectives,* 120(10):1391-1396. US National Library of Medicine, October 2012. DOI: dx.doi.org/10.1289/ehp.1205026

[Hän11a] M. Hänninen, J. Suhonen, T.D. Hämäläinen and M. Hännikäinen. Link Quality-Based Channel Selection for Resource Constrained WSNs. In J. Riekki, M. Ylianttila and M. Guo (Eds.): *Advances in Grid and Pervasive Computing, Lecture Notes in Computer Science Volume 6646,* pp. 254-263. ISBN: 978-3-642-20753-2. Springer Berlin Heidelberg, 2011. DOI: dx.doi.org/10.1007/978-3-642-20754-9_26

[Hän11b] M. Hänninen, J. Suhonen, T.D. Hämäläinen and M. Hännikäinen. Practical monitoring and analysis tool for WSN testing. *In Proceedings of the 2011 Conference on Design and Architectures for Signal and Image Processing (DASIP 2011)*, pp. 1-8. ISBN: 978-1-4577-0620-2. IEEE, November 2011. DOI: dx.doi.org/10.1109/DASIP.2011.6136894

[Han13] A. Hansell, H.E. Laszlo and S. Floud. Review of Evidence of Noise Related Health Effects. In J. Lekaviciute, S. Kephalopoulos, S. Stansfeld and C.

Clark (Eds.): *European Network of Noise and Health, Final Report of EU Project no.226442 (FP-7-ENV-2008-1), Joint Research Centre's Scientific and Policy Reports,* pp. 20-36. ISBN: 978-92-79-28593-6. ISSN: 1831-9424. Publications Office of the European Union, 2013. DOI: dx.doi.org/10.2788/83694

[Hei09] M. Heinonen-Guzejev. *Noise sensitivity – medical, psychological and genetic aspects,* pp. 1-88. ISBN: 978-952-10-5177-7. ISSN: 0355-7979. University of Helsinki, Faculty of Medicine, Department of Public Health, Finland. January 2009. Ph.D. Thesis. http://hdl.handle.net/10138/20384

[Her12] F. Hermans, L. Larzon, O. Rensfelt and P. Gunningberg. A Lightweight Approach to Online Detection and Classification of Interference in 802.15.4-Based Sensor Networks. *ACM SIGBED Review – Special Issue on the 3rd International Workshop on Networks of Cooperating Objects (CONET 2012),* 9(3):11-20. ISSN: 1551-3688. ACM, July 2012.
DOI: dx.doi.org/10.1145/2367580.2367582

[Hil02] J.L. Hill and D.E. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro,* 22(6):12-24. ISSN: 0272-1732. IEEE, November/ December 2002. DOI: dx.doi.org/10.1109/MM.2002.1134340

[Hil09a] T. Hilaire. New L2-Dynamic-Range-Scaling Constraints for Low Parametric Sensitivity Realizations. *In Proceedings of the 17th European Signal Processing Conference (EUSIPCO 2009)*, pp. 988-992. European Association for Signal, Speech, and Image Processing, August 2009. http://www.eurasip.org/Proceedings/Eusipco/Eusipco2009/contents/papers/1569192832.pdf (Accessed: December 10, 2013)

[Hil09b] T. Hilaire. On the Transfer Function Error of State-Space Filters in Fixed-Point Context. *IEEE Transactions on Circuits and Systems II: Express Briefs,* 56(12):936-940. ISSN: 1549-7747. IEEE, December 2009.
DOI: dx.doi.org/10.1109/TCSII.2009.2034193

[Hua10] K.L. Huang, S.S. Kanhere and W. Hu. Are you contributing trustworthy data?: the case for a reputation system in participatory sensing. *In Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems (MSWIM '10)*, pp. 14-22. ISBN: 978-1-4503-0274-6. ACM, October 2010. DOI: dx.doi.org/10.1145/1868521.1868526

[Hud97] A.J. Hudspeth. How Hearing Happens. *Neuron,* 19(5):947-950. Cell Press, November 1997. http://www.dafml.unito.it/anatomy/ponzetto/UDITO/how%20hearing%20happens.pdf (Accessed: December 10, 2013)

[Hui11] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams Over IEEE 802.15.4-Based Networks. *Internet Engineering Task Force RFC Series,* (RFC 6282):1-24. ISSN: 2070-1721. IETF, September 2011. http://www.rfc-editor.org/info/rfc6282 (Accessed: April 04, 2013)

[Ian12] F. Iannello, O. Simeone and U. Spagnolini. Medium Access Control Protocols for Wireless Sensor Networks with Energy Harvesting. *IEEE Transactions on Communications,* 60(5):1381-1389. ISSN: 0090-6778. IEEE, May 2012.
DOI: dx.doi.org/10.1109/TCOMM.2012.030712.110089

[IEC96] IEC/CD 1672: Electroacoustics-Sound Level Meters. IEC Geneva, November 1996.

[IEE05] IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002). *IEEE Standard* *for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements. – Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs),* pp. 1-580. IEEE, June 2005. DOI: dx.doi.org/10.1109/IEEESTD.2005.96290

[IEE11] IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006). *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),* pp. 1-314. IEEE, June 2011. DOI: dx.doi.org/10.1109/IEEESTD.2011.6012487

[IEE12a] IEEE Std 802.15.6-2012. *IEEE Standard for Local and metropolitan area networks Part 15.6: Wireless Body Area Networks,* pp. 1-271. IEEE, February 2012. DOI: dx.doi.org/10.1109/IEEESTD.2012.6161600

[IEE12b] IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* pp. 1-2793. IEEE, March 2012. DOI: dx.doi.org/10.1109/IEEESTD.2012.6178212

[IEE12c] IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011). *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Speci-fications for Low-Data-Rate, Wireless, Smart Metering Utility Networks,* pp. 1-252. IEEE, April 2012. DOI: dx.doi.org/10.1109/IEEESTD.2012.6190698

[Jac11] F. Jacobsen. *An Elementary Introduction to Acoustics,* pp. 1-64. Technical University of Denmark, Department of Electrical Engineering, Acoustic Technology, November 2011. http://web-files.ait.dtu.dk/fjac/p_home_page/notes/Introduction_to_acoustics.pdf (Accessed: September 11, 2013)

[Jan08] I. Jantunen, H. Laine, P. Huuskonen, D. Trossen and V. Ermolov. Smart Sensor Architecture for Mobile-Terminal-Centric Ambient Intelligence. *Sensors and Actuators A: Physical, Special Issue: The 20th European conference on Solid-State Transducers – Eurosensors 2006,* 142(1):352-360. ISSN: 0924-4247. Elsevier, March 2008. DOI: dx.doi.org/10.1016/j.sna.2007.04.014

[Jau07] T. Jauhiainen, H.S. Vuorinen and M. Heinonen-Guzejev. Ympäristömelun vaikutukset (Effects of Noise). *The Finnish Environment 3/2007,* pp. 1-79. ISBN: 978-952-11-2564-5. ISSN: 1796-1637. Ministry of the Environment, Environmental Protection Department, Finland, March 2007. http://hdl.handle.net/10138/38400

[Jon12] P. Jones and K. Lucas. The Social Consequences of Transport Decision-Making: Clarifying Concepts, Synthesising Knowledge and Assessing Implications. *Journal of Transport Geo-graphy, Special Issue: Social Impacts and Equity Issues in Transport,* 21(0):4-16. ISSN: 0966-6923. Elsevier, March 2012. DOI: dx.doi.org/10.1016/j.jtrangeo.2012.01.012

[Jur10] R. Jurdak, A.G. Ruzzelli and G.M.P. O'Hare. Radio Sleep Mode Optimization in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing,* 9(7):955-968. ISSN: 1536-1233. IEEE, July 2010. DOI: dx.doi.org/10.1109/TMC.2010.35

[Kan13] S.S. Kanhere. Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. In C. Hota and P. Srimani (Eds.): *Distributed Computing and Internet Technology, Lecture Notes in Computer Science Volume 7753,* pp. 19-26. ISBN: 978-3-642-36070-1. Springer Berlin Heidelberg, 2013. DOI: dx.doi.org/10.1007/978-3-642-36071-8_2

[Kan10] E. Kanjo. NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping. *Mobile Networks and Applications,* 15(4):562-574. ISSN: 1383-469X. Kluwer Academic Publishers, August 2010. DOI: dx.doi.org/10.1007/s11036-009-0217-y

[Kel12] M. Keller, J. Beutel and L. Thiele. How was your journey? Uncovering routing dynamics in deployed sensor networks with multi-hop network tomography. *In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pp. 15-28. ISBN: 978-1-4503-1169-4. ACM, November 2012. DOI: dx.doi.org/10.1145/2426656.2426659

[Kha13] W.Z. Khan, Y. Xiang, M.Y. Aalsalem and Q. Arshad. Mobile Phone Sensing Systems: A Survey. *IEEE Communications Surveys and Tutorials,* 15(1):402-427. ISSN: 1553-877X.IEEE, February 2013. DOI: dx.doi.org/10.1109/SURV.2012.031412.00077

[Kim11] N. Kim, K. Homma and S. Puria. Inertial Bone Conduction: Symmetric and Anti-Symmetric Components. *Journal of the Association for Research in Otolaryngology,* 12(3):261-279. ISSN: 1525-3961. Springer-Verlag, June 2011. DOI: dx.doi.org/10.1007/s10162-011-0258-3

[Kiv11a] I. Kivelä, C. Gao, J. Luomala and I. Hakala. Design of Noise Measurement Sensor Network: Networking and Communication Part. *In Proceedings of the Fifth International Conference on Sensor Technologies and Applications (SENSORCOMM 2011)*, pp. 280-287. IARIA, August 2011. http://www.thinkmind.org/index.php?view=article&articleid=sensorcomm_2011_11_30_10122 (Accessed: March 28, 2013)

[Kiv11b] I. Kivelä, C. Gao, J. Luomala, J. Ihalainen and I. Hakala. Design of Networked Low-Cost Wireless Noise Measurement Sensor. *Sensors & Transducers Journal,* 10(Special Issue): 171-190. ISSN:1726-5479. International Frequency Sensor Association, February 2011. http://www.sensorsportal.com/HTML/DIGEST/Special_Issue_Feb_2011.htm (Accessed: April 18, 2013)

[Klu09] K. Klues, C.M. Liang, J. Paek, R. Musăloiu-E, P. Levis, A. Terzis and R. Govindan. TOSThreads: thread-safe and non-invasive preemption in Tiny OS. *In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 127-140. ISBN: 978-1-60558-519-2. ACM, November 2009. DOI: dx.doi.org/10.1145/1644038.1644052

[Ko11] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, J. Vasseur, M. Durvy, A. Terzis, A. Dunkels and D. Culler. Industry: beyond interoperability: pushing the performance of sensor network IP stacks. *In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*, pp. 1-11. ISBN: 978-1-4503-0718-5. ACM, November 2011. DOI: dx.doi.org/10.1145/2070942.2070944

[Koh09] M. Kohvakka. *Medium Access Control and Hardware Prototype Designs for Low-Energy Wireless Sensor Networks,* pp. 1-206. ISBN: 978-952-15-2189-8.

ISSN: 1459-2045. Tampere University of Technology, Faculty of Computing and Electrical Engineering, Department of Computer Systems, Finland, 2009. Ph.D. Thesis. http://URN.fi/URN:NBN:fi:tty-200906231082

[Kru02] A. Krukowski, R.C.S. Morling and I. Kale. Quantization Effects in the Polyphase N-Path IIR Structure. *IEEE Transactions on Instrumentation and Measurement,* 51(6):1271-1278. ISSN: 0018-9456. IEEE, December 2002. DOI: dx.doi.org/10.1109/TIM.2002.808032

[Lah95] T.Lahti. *Akustinen mittaustekniikka,* pp. 1-151. Teknillinen korkeakoulu, Sähkötekniikan osasto, Akustiikan ja äänenkäsittelyn laboratorio, 1995. https://noppa.aalto.fi/noppa/kurssi/s-89.3430/materiaali/S-89_3430_akustinen_mittaustekniikka.pdf (Accessed: December 20, 2013)

[Lee11] Y.J. Lee, J. Trevathan, I. Atkinson, W. Read, T. Myers and R. Johnstone. The evolution of the SEMAT sensor network management system. *In Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2011)*, pp. 229-234. ISBN: 978-1-4577-0675-2. IEEE, December 2011. DOI: dx.doi.org/10.1109/ISSNIP.2011.6146633

[Lei11] T. Lei, X. Zhao and F. Hui. A TinyOS scheduling strategy and its implementation. *In Proceedings of the 3rd International Conference on Communication Software and Networks (ICCSN 2011)*, pp. 216-219. ISBN: 978-1-61284-485-5. IEEE, May 2011. DOI: dx.doi.org/10.1109/ICCSN.2011.6014255

[Lem10] S. Lembo, J. Kuusisto and J. Jukka Manner. In-Depth Breakdown of a 6LoWPAN Stack for Sensor Networks. *International Journal of Computer Networks & Communications,* 2(6):204-223. ISSN: 0974-9322. Academy & Industry Research Collaboration Center, November 2010. DOI: dx.doi.org/10.5121/ijcnc.2010.2614

[Les13] Nd. Lestard, R.C. Valente, A.G. Lopes and M.A.M. Capella. Direct Effects of Music in Non-Auditory Cells in Culture. *Noise and Health,* 15(66):307-314. ISSN: 1998-4030. Medknow Publications, August 2013. DOI: dx.doi.org/10.4103/1463-1741.116568

[Lev05] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler. TinyOS: An Operating System for Sensor Networks. In W. Weber, J. Rabaey and E. Aarts (Eds.): *Ambient Intelligence,* pp. 115-148. ISBN: 978-3-540-23867-6. Springer Berlin Heidelberg, 2005. DOI: dx.doi.org/10.1007/3-540-27139-2_7

[Lin12] P. Lindgren, H. Mäkitaavola, J. Eriksson and J. Eliasson. Leveraging TinyOS for integration in process automation and control systems. *In Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON 2012)*, pp. 5779-5785. ISSN: 1553-572X. IEEE, October 2012. DOI: dx.doi.org/10.1109/IECON.2012.6389040

[Liu03] T. Liu and M. Martonosi. Impala: a middleware system for managing autonomic, parallel sensor systems. *In Proceedings of the Ninth ACM SIGPLAN symposium on Principles and practice of parallel programming (PPoPP '03)*, pp. 107-118. ISBN: 1-58113-588-2. ACM, October 2003. DOI: dx.doi.org/10.1145/781498.781516

[Liu04] T. Liu, C.M. Sadler, P. Zhang and M. Martonosi. Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet. *In Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys '04)*, pp. 256-269. ISBN: 1-58113-793-1. ACM, June 2004. DOI: dx.doi.org/10.1145/990064.990095

[Liu09] S. Liu, K. Fan and P. Sinha. CMAC: An Energy-Efficient MAC Layer Protocol using Con-vergent Packet Forwarding for Wireless Sensor Networks. *ACM Transactions on Sensor Networks,* 5(4, Article No. 29):1-34. ISSN: 1550-4859. ACM, November 2009. DOI: dx.doi.org/10.1145/1614379.1614381

[Liu10] Y. Liu, K. Liu and M. Li. Passive Diagnosis for Wireless Sensor Networks. *IEEE-ACM Transactions on Networking,* 18(4):1132-1144. ISSN: 1063-6692. IEEE Press, August 2010. DOI: dx.doi.org/10.1109/TNET.2009.2037497

[Mac12] E. Mackensen, M. Lai and T.M. Wendt. Performance analysis of an Bluetooth Low Energy sensor system. *In Proceedings of the IEEE 1st International Symposium on Wireless Systems (IDAACS-SWS 2012)*, pp. 62-66. ISBN: 978-1-4673-4678-8. IEEE, September 2012. DOI: dx.doi.org/10.1109/IDAACS-SWS.2012.6377634

[Mai10] N. Maisonneuve, M. Stevens and B. Ochab. Participatory Noise Pollution Monitoring using Mobile Phones. *Information Polity,* 15(1):51-71. ISSN: 1875-8754. IOS Press, January 2010. DOI: dx.doi.org/10.3233/IP-2010-0200

[Mam07] F. Mammano, M. Bortolozzi, S. Ortolano and F. Anselmi. Ca^{2+} Signaling in the Inner Ear. *Physiology,* 22(2):131-144. ISSN: 0886-1714. American Physiological Society, April 2007. DOI: dx.doi.org/10.1152/physiol.00040.2006

[Man12] A. Manninen. Uusi SI-järjestelmä toteuttaa Maxwellin unelman. *Arkhimedes 2/2012,* (2):10-20. ISSN: 0004-1920. Finnish Physical Society / Finnish Mathematical Society, August 2012. http://www.doria.fi/handle/10024/78586

[Man06] D. Manoussaki, E.K. Dimitriadis and R.S. Chadwick. Cochlea's Graded Curvature Effect on Low Frequency Waves. *Physical Review Letters,* 96(8):088701. American Physical Society, March 2006.
DOI: dx.doi.org/10.1103/PhysRevLett.96.088701

[Man08] D. Manoussaki, R.S. Chadwick, D.R. Ketten, J. Arruda, E.K. Dimitriadis and J.T. O'Malley. The Influence of Cochlear Shape on Low-Frequency Hearing. *Proceedings of the National Academy of Sciences,* 105(16):6162-6166. National Academy of Sciences of the USA, April 2008.
DOI: dx.doi.org/10.1073/pnas.0710037105

[Mar04] K. Martinez, J.K. Hart and R. Ong. Environmental Sensor Networks. *Computer,* 37(8):50-56. ISSN: 0018-9162. IEEE Computer Society, August 2004.
DOI: dx.doi.org/10.1109/MC.2004.91

[Mar08] I. Marín, J. Arias, E. Arceredillo, A. Zuloaga, I. Losada and J. Mabe. LL-MAC: A Low Latency MAC Protocol for Wireless Self-Organised Networks. *Microprocessors and Microsystems,* 32(4): 197-209. ISSN: 0141-9331. Elsevier, 2008. DOI: dx.doi.org/10.1016/j.micpro.2007.10.003

[Mar09] K. Martinez, J. Hart and R. Ong. Deploying a Wireless Sensor Network in Iceland. In N. Trigoni, A. Markham and S. Nawaz (Eds.): *GeoSensor Networks, Lecture Notes in Computer Science Volume 5659,* pp. 131-137. ISBN: 978-3-642-02902-8. Springer Berlin Heidelberg, 2009.
DOI: dx.doi.org/10.1007/978-3-642-02903-5_13

[Mar11] K. Martinez and P. Basford. Robust wireless sensor network performance analysis. *In Proceedings of the 2011 IEEE Sensors*, pp. 203-206. ISSN: 1930-0395. IEEE, October 2011. DOI: dx.doi.org/10.1109/ICSENS.2011.6127105

[Mar12] K. Martinez, P.J. Basford, D. De Jager and J.K. Hart. A wireless sensor network system deployment for detecting stick slip motion in glaciers. *In Proceedings of the IET Conference on Wireless Sensor Systems (WSS 2012)*, pp. 12-14. ISBN: 978-1-84919-625-3. IET, June 2012. DOI: dx.doi.org/10.1049/cp.2012.0580

[Mas10] D.L. Mascarenas, E.B. Flynn, M.D. Todd, T.G. Overly, K.M. Farinholt, G. Park and C.R. Farrar. Experimental Studies of using Wireless Energy Transmission for Powering Embedded Sensor Nodes. *Journal of Sound and Vibration, Special Issue: Structural Health Monitoring Theory Meets Practice,* 329(12):2421-2433. ISSN: 0022-460X. Elsevier, June 2010. DOI: dx.doi.org/10.1016/j.jsv.2009.10.023

[MEM11a] MEMSIC. MicaZ, Wireless Measurement System. 6020-0065-05 Rev A. *micaz_datasheet-t.pdf.* MEMSIC Inc, 2011. http://www.memsic.com/wireless-sensor-networks/MPR2400CB (Accessed: April 13, 2013)

[MEM11b] MEMSIC. TelosB Mote Platform. 6020-0094-03 Rev A. *telosb_datasheet.pdf.* MEMSIC Inc, 2011. http://www.memsic.com/wireless-sensor-networks/MPR2400CB (Accessed: April 13, 2013)

[MEM11c] MEMSIC. IRIS, Wireless Measurement System. 6020-0124-02 Rev A. *IRIS_Datasheet.pdf.* MEMSIC Inc, 2011. http://www.memsic.com/wireless-sensor-networks/XM2110CA (Accessed: April 13, 2013)

[MEM11d] MEMSIC. LOTUS, High-Performance Wireless Sensor Network Platform. *6020-0705-01_A_LOTUS.pdf.* MEMSIC Inc, 2011. http://www.memsic.com/wireless-sensor-networks/LPR2400 (Accessed: April 13, 2013)

[Mik13] K. Mikhaylov and J. Tervonen. Data Collection From Isolated Clusters in Wireless Sensor Networks Using Mobile Ferries. *In Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2013),* pp. 903-909. ISBN: 978-0-7695-4952-1. IEEE, March 2013. DOI: dx.doi.org/ 10.1109/WAINA.2013.87

[Mit01] S.K. Mitra. *Digital signal processing: a computer-based approach (International ed),* pp. 1-866. ISBN: 0-07-118175-X. McGraw-Hill, 2001.

[Moe10] S. Moeller, A. Sridharan, B. Krishnamachari and O. Gnawali. Routing without routes: the backpressure collection protocol. *In Proceedings of the 9th ACM/ IEEE International Conference on Information Processing in Sensor Networks (IPSN '10),* pp. 279-290. ISBN: 978-1-60558-988-6. ACM, April 2010. DOI: dx.doi.org/10.1145/1791212.1791246

[Møl77] A.R. Møller. Occupational Noise as a Health Hazard: Physiological Viewpoints. *Scandinavian Journal of Work, Environment & Health,* 3(2):73-79. ISSN: 1795-990X. SJWEH, June 1977. http://www.jstor.org/stable/40964614

[Mon07] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler. Transmission of IPv6 Packets Over IEEE 802.15.4 Networks. *Internet Engineering Task Force RFC Series,* (RFC 4944):1-30. ISSN:2070-1721. IETF, September 2007. http://www.rfc-editor.org/info/rfc4944 (Accessed: April 04, 2013)

[Moo97] B.C.J. Moore. *An Introduction to the Psychology of Hearing (4th ed., 3rd pr., 2001),* pp. 1-373. ISBN: 0-12-505627-3. Academic Press, 1997.

[MOT06] Moteiv. Tmote Sky: Datasheet. *tmote-sky-datasheet.pdf.* Moteiv Corporation, February 2006. http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf (Accessed: February 12, 2013)

[Mot11] L. Mottola and G.P. Picco. MUSTER: Adaptive Energy-Aware Multisink Routing in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing,* 10(12):1694-1709. ISSN: 1536-1233. IEEE, October 2011.
DOI: dx.doi.org/10.1109/TMC.2010.250

[Mur89] P. Murphy, K. Hubschi, N. de Rooij and C. Racine. Subminiature Silicon Integrated Electret Capacitor Microphone. *IEEE Transactions on Electrical Insulation,* 24(3):495-498. ISSN: 0018-9367. IEEE, June 1989.
DOI: dx.doi.org/10.1109/14.30895

[Nac05] L. Nachman, R. Kling, R. Adler, J. Huang and V. Hummel. The Intel® mote platform: a Bluetooth-based sensor network for industrial monitoring. *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 437-442. ISBN: 0-7803-9201-9. IEEE, April 2005. DOI: dx.doi.org/10.1109/IPSN.2005.1440968

[Nac08] L. Nachman, J. Huang, J. Shahabdeen, R. Adler and R. Kling. IMOTE2: Serious Computation at the Edge. *In Proceedings of the International Wireless Commu-nications and Mobile Computing Conference (IWCMC '08)*, pp. 1118-1123. ISBN: 978-1-4244-2201-2. IEEE, August 2008.
DOI: dx.doi.org/10.1109/IWCMC.2008.194

[Neu93] Y. Neuvo. Digital Filter Implementation Considerations. In S.K. Mitra and J.F. Kaiser (Eds.): *Handbook for Digital Signal Processing,* pp. 337-417. ISBN: 0-471-61995-7. John Wiley & Sons, 1993.

[Nie09] A. Niemi. *Fourier-analyysi ja Laplace-muunnos, (7. uudistettu painos),* pp. 1-260. Jyväskylä: Antti Niemi, 2009.

[Nie13] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby and C. Gomez. Transmission of IPv6 Packets Over Bluetooth Low Energy. *IPv6 over Low power WPAN Working Group Draft,* pp. 1-16. IETF, February 2013. http://tools.ietf.org/pdf/draft-ietf-6lowpan-btle-12.pdf (Accessed: April 03, 2013)

[Nob03] R. Nobili, A. Vetešnik, L. Turicchia and F. Mammano. Otoacoustic Emissions from Residual Oscillations of the Cochlear Basilar Membrane in a Human Ear Model. *Journal of the Association for Research in Otolaryngology,* 4(4):478-494. ISSN: 1525-3961. Springer-Verlag, December 2003.
DOI: dx.doi.org/10.1007/s10162-002-3055-1

[Ola12] M. Olama, S. Smith, T. Kuruganti and Xiao Ma. Performance study of hybrid DS/FFH spread-spectrum systems in the presence of frequency-selective fading and multiple-access interference. *In Proceedings of the IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2012)*, pp. 1-5. ISBN: 978-1-4673-1861-7. IEEE, May 2012.
DOI: dx.doi.org/10.1109/CQR.2012.6267102

[Opp99] A.V. Oppenheim, R.W. Schafer and J.R. Buck. *Discrete-Time Signal Processing (2nd ed),* pp. 1-870. ISBN: 0-13-083443-2. Prentice Hall, 1999.

[Pad09] W.T. Padgett and D.V. Anderson. Fixed-Point Signal Processing. In J. Moura (Eds.): *Synthesis Lectures on Signal Processing,* pp. 1-121. ISBN: 978-15-982-925-89. ISSN: 1932-1236. Morgan & Claypool, September 2009.

DOI: dx.doi.org/10.2200/S00220ED1V01Y200909SPR009

[Per12] C. Perera, A. Zaslavsky, P. Christen, A. Salehi and D. Georgakopoulos. Connecting mobile things to global sensor network middleware using system-generated wrappers. *In Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '12)*, pp. 23-30. ISBN: 978-1-4503-1442-8. ACM, May 2012. DOI: dx.doi.org/10.1145/2258056.2258062

[Pir10] S. Pirrera, E. De Valck and R. Cluydts. Nocturnal Road Traffic Noise: A Review on its Assessment and Consequences on Sleep and Health. *Environment International,* 36(5):492-498. ISSN: 0160-4120. Elsevier, July 2010. DOI: dx.doi.org/10.1016/j.envint.2010.03.007

[Pol04] J. Polastre, J. Hill and D. Culler. Versatile low power media access for wireless sensor networks. *In Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pp. 95-107. ACM, November 2004. DOI: dx.doi.org/10.1145/1031495.1031508

[Pol05] J. Polastre, R. Szewczyk and D. Culler. Telos: enabling ultra-low power wireless research. *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 364-369. IEEE, April 2005. DOI: dx.doi.org/10.1109/IPSN.2005.1440950

[Puc11] D. Puccinelli, O. Gnawali, S. Yoon, S. Santini, U. Colesanti, S. Giordano and L. Guibas. The Impact of Network Topology on Collection Performance. In P.J. Marrón and K. Whitehouse (Eds.): *Wireless Sensor Networks, Lecture Notes in Computer Science Volume 6567,* pp. 17-32. ISBN: 978-3-642-19185-5. Springer Berlin Heidelberg, 2011. DOI: dx.doi.org/10.1007/978-3-642-19186-2_2

[Puc12] D. Puccinelli, S. Giordano, M. Zuniga and P.J. Marrón. Broadcast-free collection protocol. *In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pp. 29-42. ISBN: 978-1-4503-1169-4. ACM, November 2012. DOI: dx.doi.org/10.1145/2426656.2426660

[Qua12] A. Quarteroni, F. Saleri and P. Gervasio. *Calcolo Scientifico. Esercizi e problemi risolti con MATLAB e Octave, UNITEXT,* pp. 1-456. ISBN: 978-88-470-2745-9. ISSN: 2038-5757. Springer Milan, July 2012. DOI: dx.doi.org/10.1007/978-88-470-2745-9

[Qui94] É. Quinet. The Social Costs of Transport: Evaluation and Links with Internalisation Policies. *Internalising The Social Costs of Transport,* pp. 1-44. OECD/ECMT, 1994. http://internationaltransportforum.org/IntOrg/ecmt/taxes/pdf/94socCosts3e.pdf (Accessed: November 29, 2013)

[Rad12] D. Radu, C. Avram, A. Astilean, B. Parrein and Jiazi Yi. Acoustic noise pollution monitoring in an urban environment using a VANET network. *In Proceedings of the IEEE International Conference on Automation Quality and Testing Robotics (AQTR 2012)*, pp. 244-248. IEEE, May 2012. DOI: dx.doi.org/10.1109/AQTR.2012.6237711

[Rak07] P. Rako. Hop, Jump, and Spread: Wireless Machine-to-Machine Interfaces. *EDN,* 52(12):52-66. UBM Tech, June 2007. http://www.edn.com/design/communications-networking/4315324/Hop-jump-and-spread-wireless-machine-to-machine-interfaces (Accessed: March 08, 2013)

[Ran10] R.K. Rana, C.T. Chou, S.S. Kanhere, N. Bulusu and W. Hu. Ear-phone: an end-to-end participatory urban noise mapping system. *In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 105-116. ISBN: 978-1-60558-988-6. ACM, April 2010. DOI: dx.doi.org/10.1145/1791212.1791226

[Ran11] R.K. Rana. *Addressing three wireless sensor network challenges using sparse approximation methods*, pp. 1-202. University of New South Wales, Faculty of Engineering, Computer Science & Engineering, Sydney, Australia. June 2011. Ph.D. Thesis. http://handle.unsw.edu.au/1959.4/50856

[Rap03] Y. Raphael and R.A. Altschuler. Structure and Innervation of the Cochlea. *Brain Research Bulletin, Functional Anatomy of Ear Connections,* 60(5-6):397-422. ISSN: 0361-9230. Elsevier, June 2003. DOI: dx.doi.org/10.1016/S0361-9230(03)00047-9

[Ree12] J.H. Reed, J.T. Bernhard and Jung-Min Park. Spectrum Access Technologies: The Past, the Present, and the Future. *Proceedings of the IEEE,* 100(Special Centennial Issue):1676-1684. ISSN: 0018-9219. IEEE, May2012. DOI: dx.doi.org/10.1109/JPROC.2012.2187140

[RIO02] *RION NL-32 Sound Level Meter User Manual,* RION Ltd., Japan, 2002.

[Roc12] R. Rocher, D. Menard, P. Scalart and O. Sentieys. Analytical Approach for Numerical Accuracy Estimation of Fixed-Point Systems Based on Smooth Operations. *IEEE Trans- actions on Circuits and Systems I: Regular Papers,* 59(10):2326-2339. ISSN: 1549-8328. IEEE, October 2012. DOI: dx.doi.org/10.1109/TCSI.2012.2188938

[Rua11] Z. Ruan, E.C.-H. Ngai and J. Liu. Wireless Sensor Deployment for Collaborative Sensing with Mobile Phones. *Computer Networks,* 55(15):3224-3245. ISSN: 1389-1286. Elsevier, October 2011. DOI: dx.doi.org/10.1016/j.comnet.2011.06.017

[Saa13] A. Saarinen. Valtioneuvoston periaatepäätös ja valtakunnallinen toimintaohjelma meluntorjunnasta. *Ympäristö ja Terveys; Special Issue of the Twelfth National Noise Abatement Days in Finland,* 44(2):22-26. ISSN: 0358-3333. Suomen Ympäristö- ja Terveysalan Kustannus, March 2013. http://www.ymparistojaterveys.fi/lehdet/ytlehti/sisallot/vuosi-2013/2-meluntorjunta/

[Sal11] A.N. Salt and J.A. Kaltenbach. Infrasound from Wind Turbines could Affect Humans. *Bulletin of Science, Technology & Society,* 31(4):296-302. ISSN: 1552-4183. SAGE Publications, August 2011. DOI: dx.doi.org/10.1177/0270467611412555

[San08] S. Santini, B. Ostermaier and A. Vitaletti. First experiences using wireless sensor networks for noise pollution monitoring. *In Proceedings of the workshop on Real-world wireless sensor networks (REALWSN '08)*, pp. 61-65. ISBN: 978-1-60558-123-1. ACM, April 2008. DOI: dx.doi.org/10.1145/1435473.1435490

[San09] S. Santini, B. Ostermaier and R. Adelmann. On the use of sensor nodes and mobile phones for the assessment of noise pollution levels in urban environments. *In Proceedings of the Sixth International Conference on Networked Sensing Systems (INSS 2009)*, pp. 1-8. ISBN: 978-1-4244-6313-8. IEEE, June 2009. DOI: dx.doi.org/10.1109/INSS.2009.5409957

[Sav02] A. Savvides and M.B. Srivastava. A distributed computation platform for wireless embedded sensing. *In Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02)*, pp. 220-225. ISSN: 1063-6404. IEEE, September 2002.
DOI: dx.doi.org/10.1109/ICCD.2002.1106774

[Sch03] P.R. Scheeper, B. Nordstrand, J.O. Gulløv, B. Liu, T. Clausen, L. Midjord and T. Storgaard-Larsen. A New Measurement Microphone Based on MEMS Technology. *Journal of Microelectromechanical Systems,* 12(6):880-891. ISSN: 1057-7157. IEEE, December 2003. DOI: dx.doi.org/10.1109/JMEMS.2003.820260

[She12] Z. Shelby, S. Chakrabarti, E. Nordmark and C. Bormann. Neighbor Discovery Optimization for IPv6 Over Low-Power Wireless Personal Area Networks (6LoWPANs). *Internet Engineering Task Force RFC Series,* (RFC 6775): 1-24. ISSN: 2070-1721. IETF, November 2012. http://www.rfc-editor.org/info/rfc6775 (Accessed: April 04, 2013)

[SI06] *The International System of Units (SI, 8th edition).* Bureau International des Poinds et Mesures, Paris, France. March 2006. http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf (Accessed: December 06, 2013)

[Sig96] S.R. Signell, T.G. Kouyoumdjiev, K.H. Mossberg and C.G.L. Harnefors. Design and Analysis of Bilinear Digital Ladder Filters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 43(2):69-81. ISSN: 1057-7122. IEEE, February 1996. DOI: dx.doi.org/10.1109/81.486429

[Sil12] T.H. Silva, P.O.S. Vaz de Melo, J.M.d. Almeida and A.A.F. Loureiro. Uncovering properties in participatory sensor networks. *In Proceedings of the 4th ACM international workshop on Hot topics in planet-scale measurement (HotPlanet '12)*, pp. 33-38. ISBN: 978-1-4503-1318-6. ACM, June 2012.
DOI: dx.doi.org/10.1145/2307836.2307847

[Sim07] G. Simon, M. Molnar, L. Gonczy and B. Cousin. Dependable k-coverage algorithms for sensor networks. *In Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC 2007)*, pp. 1-6. ISSN: 1091-5281. IEEE, May 2007. DOI: dx.doi.org/10.1109/IMTC.2007.379153

[Sip07] P. Sipari, K. Mäkela, T. Järvi, A. Saarinen and J. Hirvonen. Noise in built-up areas and noise abatement. In S. Malin (Ed.): *MELUTTA Project: Final Report, Reports of the Ministry of the Environment 20/2007,* pp. 117-171. ISBN: 978-952-11-2831-8. Ministry of the Environment, Environmental Protection Department, Finland, September 2007. http://hdl.handle.net/10138/41381

[Sør12] M. Sørensen, Z.J. Andersen, R.B. Nordsborg, S.S. Jensen, K.G. Lillelund, R. Beelen, E.B. Schmidt, A. Tjønneland, K. Overvad and O. Raaschou-Nielsen. Road Traffic Noise and Incident Myocardial Infarction: A Prospective Cohort Study. *PLoS ONE,* 7(6):e39283. ISSN: 1932-6203. PLOS, June 2012. DOI: dx.doi.org/10.1371/journal.pone.0039283

[Sot05] R. Sottek and K. Genuit. Models of Signal Processing in Human Hearing. *AEÜ – International Journal of Electronics and Communications,* 59(3):157-165. ISSN: 1434-8411. Elsevier, June 2005. DOI: dx.doi.org/10.1016/j.aeue.2005.03.016

[Sri11] S. Sridevi and M. Usha. Taxonomy of transport protocols for Wireless Sensor Networks. *In Proceedings of the International Conference on Recent Trends in Information Technology (ICRTIT 2011)*, pp.467-472. ISBN:978-1-4577-0588-5. IEEE, June 2011. DOI: dx.doi.org/10.1109/ICRTIT.2011.5972350

[Sug08] R. Sugihara and R.K. Gupta. Programming Models for Sensor Networks: A Survey. *ACM Transactions on Sensor Networks,* 4(2, Article 8):1-29. ISSN: 1550-4859. ACM, March 2008. DOI: dx.doi.org/10.1145/1340771.1340774

[Suh06] J. Suhonen, M. Kohvakka, M. Hännikäinen and T. Hämäläinen. Design, Implementation, and Experiments on Outdoor Deployment of Wireless Sensor Network for Environmental Moni-toring. In S. Vassiliadis, S. Wong and T. Hämäläinen (Eds.): *Embedded Computer Systems: Architectures, Modeling, and Simulation, Lecture Notes in Computer Science Volume 4017,* pp. 109-121. ISBN: 978-3-540-36411-5. ISSN: 0302-9743. Springer Berlin Heidelberg, 2006. DOI: dx.doi.org/10.1007/11796435_13

[Suh10] J.G. Suh, H.Y. Kim and Y. Suzuki. Measurement of Resonance Frequency and Loss Factor of a Microphone Diaphragm using a Laser Vibrometer. *Applied Acoustics,* 71(3):258-261. ISSN: 0003-682X. Elsevier, March 2010. DOI: dx.doi.org/http://dx.doi.org/10.1016/j.apacoust.2009.09.007

[Suh12] J. Suhonen. *Designs for the Quality of Service Support in Low-Energy Wireless Sensor Network Protocols,* pp. 1-125. ISBN: 978-952-15-2907-8. ISSN: 1459-2045. Tampere University of Technology, Faculty of Computing and Electrical Engineering, Department of Computer Systems, Finland, August 2012. Ph.D. Thesis. http://URN.fi/URN:ISBN:978-952-15-2907-8

[SUN10] Sun Microsystems. SunSPOT, Main Board Technical Datasheet, Rev 8.0. Oracle America Inc, October 2010. http://www.sunspotworld.com/docs/Yellow/eSPOT8ds.pdf (Accessed: February 18, 2013)

[Taw93] A. Tawfik, P. Agathoklis and F. El-Guibaly. A tool for analyzing finite word-length effects in fixed-point digital filter implementations. *In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'93)*, pp. 116-119. IEEE, May 1993. DOI: dx.doi.org/10.1109/PACRIM.1993.407208

[TIN11] Tinynode. *Tinynode 584 User's Manual (v.1.3)*, pp. 1-14. Tinynode SA, January 2011. http://www.tinynode.com/?q=system/files/TN584_Users_Manual_v_1_3.pdf (Accessed: April 12, 2013)

[TRA12] Trafi. Ajoneuvokanta 2012. *11064-Rekisterissa_olevat_maakunnittain_31_12_2012.pdf.* Finnish Transport Safety Agency, December 2012. http://www.trafi.fi/palvelut/tilastot/tieliikenne/ajoneuvokanta/ajoneuvokanta_2012 (Accessed: November 18, 2013)

[UKsi05] The Control of Noise at Work Regulations 2005. *UK Statutory Instruments 2005 No.1643, Health and Safety,* pp. 1-13. ISBN: 0-11-072984-6. The Stationery Office Limited, June 2005. http://www.legislation.gov.uk/uksi/2005/1643/pdfs/uksi_20051643_en.pdf

[UN12] UNEP. Global Environment Outlook 5; Environment for the future we want. *GEO5_report_full_en.pdf.* United Nations Environment Programme, May 2012. http://www.unep.org/geo/pdfs/geo5/GEO5_report_full_en.pdf (Accessed: November 19, 2013)

[Vai93] P.P. Vaidyanathan. Robust Digital Filter Structures. In S.K. Mitra and J.F.

Kaiser (Eds.): *Handbook for Digital Signal Processing,* pp. 419-490. ISBN: 0-471-61995-7. John Wiley & Sons, 1993.

[Vas12] J. Vasseur, M. Kim, K. Pister, N. Dejean and D. Barthel. Routing Metrics used for Path Cal-culation in Low-Power and Lossy Networks. *Internet Engineering Task Force RFC Series,* (RFC 6551):1-30. ISSN: 2070-1721. IETF, March 2012. http://www.rfc-editor.org/info/rfc6551 (Accessed: April 02, 2013)

[VNa06] Valtioneuvoston asetus työtekijöiden suojelemisesta melusta aiheutuvilta vaaroilta. 85/26.1.2006. http://www.finlex.fi/fi/laki/alkup/2006/20060085

[VNp07] Valtioneuvoston periaatepäätös meluntorjunnasta. *The Finnish Environment 7/2007,* pp. 1-26. ISBN: 952-11-2618-5. ISSN: 1796-170X. Ministry of the Environment, Finland, March 2007. http://hdl.handle.net/10138/41509

[Völ11] P. Völgyesi, S. Szilvási, J. Sallai and A. Lédeczi. External smart microphone for mobile phones. *In Proceedings of the Fifth International Conference on Sensing Technology (ICST 2011)*, pp. 171-176. ISSN: 2156-8065. IEEE, November 2011. DOI: dx.doi.org/10.1109/ICSensT.2011.6136957

[Wan08] M. Wang, J. Cao, J. Li and S.K. Dasi. Middleware for Wireless Sensor Networks: A Survey. *Journal of Computer Science and Technology,* 23(3):305-326. ISSN: 1000-9000. Springer US, 2008. DOI: dx.doi.org/10.1007/s11390-008-9135-x

[Wer06] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees and M. Welsh. Deploying a Wireless Sensor Network on an Active Volcano. *IEEE Internet Computing,* 10(2):18-25. ISSN: 1089-7801. IEEE, March-April 2006. DOI: dx.doi.org/10.1109/MIC.2006.26

[Whe07] A. Wheeler. Commercial Applications of Wireless Sensor Networks using ZigBee. *IEEE Communications Magazine,* 45(4):70-77. ISSN: 0163-6804. IEEE, April 2007. DOI: dx.doi.org/10.1109/MCOM.2007.343615

[WHO04a] WHO. *Environmental Health Indicators for Europe. A Pilot Indicator-based Report,* pp. 1-58. WHO Regional Office for Europe, June 2004. http://www.euro.who.int/__data/assets/pdf_file/0003/140925/E82938.pdf (Accessed: November 28, 2013)

[WHO04b] WHO. *Development of Environment and Health Indicators for European Union Countries: Results of a Pilot Study,* pp. 1-117. WHO European Centre for Environment and Health, July 2004. http://www.euro.who.int/document/E85061.pdf (Accessed: June 19, 2008)

[WHO11] WHO. Tools for the Monitoring of Parma Conference Commitments. *Report of the Meeting 25-26 November 2011 in Bonn, Germany,* pp. 1-37. WHO European Centre for Environment and Health, November 2011. http://www.euro.who.int/__data/assets/pdf_file/0019/134380/e94788.pdf (Accessed: December 3, 2013)

[Wil12] S. Wille, N. Wehn, I. Martinovic, S. Kunz and P. Göhner. AmICA – A Flexible, Compact, Easy-to-Program and Low-Power WSN Platform. In P. Sénac, M. Ott and A. Seneviratne (Eds.): *Mobile and Ubiquitous Systems: Computing, Networking, and Services, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,* pp. 381-382. ISBN: 978-3-642-29153-1. Springer Berlin Heidelberg, 2012. DOI: dx.doi.org/10.1007/

978-3-642-29154-8_45

[Win12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *Internet Engineering Task Force RFC Series,* (RFC 6550):1-158. ISSN: 2070-1721. IETF, March 2012. http://www.rfc-editor.org/info/rfc6550 (Accessed: January 24, 2013)

[Yan11] D. Yang, Z. Wang, B. Li, Y. Luo and X. Lian. Quantitative Measurement of Pass-by Noise Radiated by Vehicles Running at High Speeds. *Journal of Sound and Vibration,* 330(7):1352-1364. ISSN: 0022-460X. Elsevier, March 2011 DOI: dx.doi.org/10.1016/j.jsv.2010.10.001

[Yib11] C. Yibo, K.-M. Hou, H. Zhou, H.-L. Shi, X. Liu, X. Diao, H. Ding, J.-J. Li and C. de Vaulx. 6LoWPAN Stacks: A Survey. *In Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2011),* pp. 1-4. ISSN: 2161-9646. IEEE, September 2011. DOI: dx.doi.org/10.1109/wicom.2011.6040344

[YM04] Ympäristöministeriö. *Meluntorjunnan valtakunnalliset linjaukset ja toimintaohjelma (National guidelines and action plan for noise abatement), The Finnish Environment 696,* pp. 1-64. ISBN: 952-11-1688-9. ISSN: 1238-7312. Ministry of the Environment, Finland, April 2004. http://hdl.handle.net/10138/40452

[Zhe06] J. Zheng and M.J.Lee. A comprehensive performance study of IEEE 802.15.4 In S. Phoha, T.F. La Porta and C. Griffin (Eds.): *Sensor network operations,* pp. 218-237. ISBN: 978-0-471-78417-3. Wiley-IEEE Press, 2006. DOI: dx.doi.org/10.1002/9780471784173.ch4

APPENDIX Noise Indicator Data for KirriL Measurement

Instrument:		2238
Application:		BZ7124 version 1.2
Start Time:		01.12.2005 12:56:03
End Time:		01.12.2005 13:00:03
Elapsed Time:		0:04:00
Bandwidth:		Broad band
Detector 1/2	RMS	Peak
Range:		30,0-110,0 dB
	Time	Frequency
Detector 1:	S F I	L
Detector 2:	Peak	L
Statistic	F	L
Criterion Level:		100,0 dB
Threshold:		0,0 dB
Exchange Rate		3 and 4
Exposure Time:		7:30:00
Peaks Over:		65,0 dB
Instrument S/N:		2356921
Microphone S/N:		2304137
Input:		Microphone
Windscreen Corr.:		On
S. I. Correction:		Frontal
Calibration Time:		01.12.2005 08:38:51
Calibration Level:		93,8 dB
Sensitivity:		-31,8 dB
Microphone:		2304137

Start date	Start time	LLeq	LLFMax	LLpk(MaxP)	LLFMin	Overload
01.12.2005	12:56:03	93,4	93,5	97,2	93,4	0
01.12.2005	12:56:04	93,5	93,5	96,8	93,5	0
01.12.2005	12:56:05	93,5	93,6	97,2	93,5	0
01.12.2005	12:56:06	93,5	93,5	97,0	93,5	0
01.12.2005	12:56:07	93,5	93,6	96,9	93,5	0
01.12.2005	12:56:08	93,5	93,6	96,9	93,5	0
01.12.2005	12:56:09	93,5	93,6	96,9	93,5	0
01.12.2005	12:56:10	93,5	93,6	96,9	93,5	0
01.12.2005	12:56:11	93,5	93,6	96,8	93,5	0
01.12.2005	12:56:12	93,5	93,6	97,0	93,5	0
01.12.2005	12:56:13	93,5	93,5	97,0	93,5	0
01.12.2005	12:56:14	93,5	93,6	97,0	93,5	0
01.12.2005	12:56:15	93,5	93,5	97,0	93,5	0
01.12.2005	12:56:16	95,6	98,5	109,2	93,1	0
01.12.2005	12:56:17	96,3	98,6	106,8	94,1	0
01.12.2005	12:56:18	101,8	108,4	117,6	95,2	1
01.12.2005	12:56:19	89,2	105,8	96,9	79,1	0
01.12.2005	12:56:20	77,3	81,8	88,0	72,7	0
01.12.2005	12:56:21	75,5	78,4	85,4	71,4	0
01.12.2005	12:56:22	76,6	80,0	86,5	73,4	0
01.12.2005	12:56:23	71,9	76,4	81,6	69,9	0
01.12.2005	12:56:24	72,9	75,5	82,2	68,4	0
01.12.2005	12:56:25	73,5	76,3	81,4	69,6	0
01.12.2005	12:56:26	74,9	76,6	83,0	71,2	0
01.12.2005	12:56:27	79,2	84,8	91,7	72,2	0
01.12.2005	12:56:28	100,7	107,6	117,3	82,7	1
01.12.2005	12:56:29	92,6	97,3	97,6	92,7	0
01.12.2005	12:56:30	93,4	93,6	98,1	93,2	0
01.12.2005	12:56:31	93,5	93,6	98,0	93,5	0
01.12.2005	12:56:32	93,5	93,6	98,0	93,5	0
01.12.2005	12:56:33	93,5	93,7	98,3	93,5	0
01.12.2005	12:56:34	93,5	93,6	98,1	93,6	0
01.12.2005	12:56:35	93,6	93,8	98,9	93,5	0
01.12.2005	12:56:36	95,9	100,0	115,7	93,6	1
01.12.2005	12:56:37	101,2	108,8	117,1	82,7	1
01.12.2005	12:56:38	73,0	82,5	83,4	70,0	0
01.12.2005	12:56:39	73,4	75,9	85,7	68,4	0
01.12.2005	12:56:40	78,1	84,9	91,1	70,7	0
01.12.2005	12:56:41	85,6	88,0	93,4	82,7	0
01.12.2005	12:56:42	104,4	110,8	118,0	86,5	1
01.12.2005	12:56:43	101,8	107,4	118,0	94,6	1
01.12.2005	12:56:44	75,3	94,4	92,2	74,1	0
01.12.2005	12:56:45	90,0	97,2	116,6	74,8	1
01.12.2005	12:56:46	74,2	81,6	87,2	73,5	0
01.12.2005	12:56:47	77,7	78,9	89,9	74,5	0
01.12.2005	12:56:48	78,4	79,4	90,0	77,5	0
01.12.2005	12:56:49	75,7	77,9	88,8	74,8	0
01.12.2005	12:56:50	74,5	75,3	87,8	74,1	0
01.12.2005	12:56:51	74,4	75,2	85,9	73,9	0
01.12.2005	12:56:52	72,3	74,4	85,0	70,5	0

Start date	Start time	LLeq	LLFMax	LLpk(MaxP)	LLFMin	Overload
01.12.2005	12:56:53	70,4	72,3	84,1	69,1	0
01.12.2005	12:56:54	69,9	72,0	81,4	68,1	0
01.12.2005	12:56:55	71,1	74,1	83,6	68,7	0
01.12.2005	12:56:56	70,6	71,6	82,9	70,0	0
01.12.2005	12:56:57	73,3	75,7	87,5	70,9	0
01.12.2005	12:56:58	75,9	77,0	87,5	75,0	0
01.12.2005	12:56:59	77,1	77,9	89,1	76,7	0
01.12.2005	12:57:00	75,1	76,9	87,5	73,5	0
01.12.2005	12:57:01	72,1	74,0	84,0	70,6	0
01.12.2005	12:57:02	71,0	71,6	82,6	70,6	0
01.12.2005	12:57:03	71,8	73,1	84,4	71,0	0
01.12.2005	12:57:04	72,1	73,3	84,5	71,0	0
01.12.2005	12:57:05	71,8	73,4	85,3	70,9	0
01.12.2005	12:57:06	73,6	75,4	85,1	72,6	0
01.12.2005	12:57:07	74,7	76,6	87,9	72,5	0
01.12.2005	12:57:08	75,4	76,6	87,6	74,3	0
01.12.2005	12:57:09	73,6	74,7	86,4	72,8	0
01.12.2005	12:57:10	73,4	74,8	86,0	72,4	0
01.12.2005	12:57:11	71,5	72,8	83,6	70,9	0
01.12.2005	12:57:12	70,7	72,2	83,3	68,4	0
01.12.2005	12:57:13	68,8	69,9	81,2	67,8	0
01.12.2005	12:57:14	67,1	69,4	80,0	65,0	0
01.12.2005	12:57:15	66,0	69,0	77,3	64,0	0
01.12.2005	12:57:16	64,1	65,8	75,5	61,9	0
01.12.2005	12:57:17	63,4	65,1	75,6	62,1	0
01.12.2005	12:57:18	63,6	65,9	74,7	61,7	0
01.12.2005	12:57:19	62,2	64,9	73,6	60,2	0
01.12.2005	12:57:20	61,2	62,6	72,8	60,2	0
01.12.2005	12:57:21	61,6	63,6	74,3	59,8	0
01.12.2005	12:57:22	62,7	64,5	73,1	61,1	0
01.12.2005	12:57:23	65,1	67,6	74,6	61,1	0
01.12.2005	12:57:24	65,5	67,7	74,9	62,0	0
01.12.2005	12:57:25	64,0	66,7	74,5	62,1	0
01.12.2005	12:57:26	64,4	66,0	75,1	61,5	0
01.12.2005	12:57:27	62,6	64,5	73,5	61,3	0
01.12.2005	12:57:28	63,9	67,2	76,0	60,1	0
01.12.2005	12:57:29	63,8	66,9	74,1	62,9	0
01.12.2005	12:57:30	64,6	67,4	75,8	62,2	0
01.12.2005	12:57:31	63,8	66,0	75,6	62,1	0
01.12.2005	12:57:32	64,4	66,2	77,1	62,9	0
01.12.2005	12:57:33	64,9	66,7	77,0	63,5	0
01.12.2005	12:57:34	64,2	65,6	75,7	62,4	0
01.12.2005	12:57:35	64,8	66,4	77,7	63,5	0
01.12.2005	12:57:36	65,7	66,9	77,8	64,8	0
01.12.2005	12:57:37	67,5	68,9	80,2	66,2	0
01.12.2005	12:57:38	69,6	73,4	86,3	67,6	0
01.12.2005	12:57:39	70,7	72,1	82,3	69,7	0
01.12.2005	12:57:40	71,4	72,2	84,9	70,9	0
01.12.2005	12:57:41	72,6	73,3	84,6	71,4	0
01.12.2005	12:57:42	74,3	75,3	87,7	72,8	0

Start date	Start time	LLeq	LLFMax	LLpk(MaxP)	LLFMin	Overload
01.12.2005	12:57:43	77,1	78,8	91,5	74,9	0
01.12.2005	12:57:44	79,3	80,1	90,8	78,5	0
01.12.2005	12:57:45	77,3	79,2	90,4	76,2	0
01.12.2005	12:57:46	75,1	76,4	87,4	74,5	0
01.12.2005	12:57:47	73,2	74,5	86,7	72,3	0
01.12.2005	12:57:48	71,9	73,2	84,3	70,5	0
01.12.2005	12:57:49	70,2	71,3	83,1	69,4	0
01.12.2005	12:57:50	69,3	71,3	80,4	68,6	0
01.12.2005	12:57:51	70,9	73,3	83,1	67,4	0
01.12.2005	12:57:52	71,3	73,0	82,2	68,6	0
01.12.2005	12:57:53	68,4	70,1	79,6	67,1	0
01.12.2005	12:57:54	67,7	69,1	79,9	66,4	0
01.12.2005	12:57:55	66,4	67,2	78,8	65,7	0
01.12.2005	12:57:56	68,0	69,0	80,5	65,8	0
01.12.2005	12:57:57	71,3	73,5	85,6	68,7	0
01.12.2005	12:57:58	74,8	75,8	87,3	73,3	0
01.12.2005	12:57:59	77,0	77,6	89,4	75,6	0
01.12.2005	12:58:00	76,9	77,6	88,1	76,3	0
01.12.2005	12:58:01	74,2	76,6	87,0	72,7	0
01.12.2005	12:58:02	71,4	72,8	84,6	70,6	0
01.12.2005	12:58:03	70,7	71,8	82,7	70,0	0
01.12.2005	12:58:04	69,6	70,1	82,3	69,1	0
01.12.2005	12:58:05	68,9	69,6	81,1	68,4	0
01.12.2005	12:58:06	69,9	70,9	82,6	68,2	0
01.12.2005	12:58:07	71,9	73,2	85,2	70,2	0
01.12.2005	12:58:08	72,8	73,5	85,1	72,4	0
01.12.2005	12:58:09	72,6	73,2	85,9	72,1	0
01.12.2005	12:58:10	69,3	72,3	81,9	67,5	0
01.12.2005	12:58:11	67,6	68,9	80,7	66,3	0
01.12.2005	12:58:12	68,9	70,8	81,6	67,1	0
01.12.2005	12:58:13	72,8	74,3	87,9	70,6	0
01.12.2005	12:58:14	73,5	74,3	86,4	72,3	0
01.12.2005	12:58:15	73,1	74,6	85,1	71,1	0
01.12.2005	12:58:16	69,5	71,5	82,3	68,7	0
01.12.2005	12:58:17	69,3	70,5	81,6	68,5	0
01.12.2005	12:58:18	72,9	74,2	86,7	70,3	0
01.12.2005	12:58:19	73,0	73,8	85,7	72,3	0
01.12.2005	12:58:20	72,1	73,6	84,4	70,4	0
01.12.2005	12:58:21	69,6	70,6	81,9	69,1	0
01.12.2005	12:58:22	70,6	71,6	82,7	69,3	0
01.12.2005	12:58:23	72,7	74,0	84,8	71,1	0
01.12.2005	12:58:24	74,3	75,3	86,5	73,3	0
01.12.2005	12:58:25	73,7	75,5	86,3	72,6	0
01.12.2005	12:58:26	71,1	72,6	82,6	70,1	0
01.12.2005	12:58:27	69,8	70,8	82,6	68,8	0
01.12.2005	12:58:28	70,3	71,8	81,7	68,6	0
01.12.2005	12:58:29	72,2	73,0	82,3	71,2	0
01.12.2005	12:58:30	74,7	75,7	82,0	72,4	0
01.12.2005	12:58:31	73,6	75,1	81,2	71,5	0
01.12.2005	12:58:32	74,2	76,1	82,4	72,3	0

Start date	Start time	LLeq	LLFMax	LLpk(MaxP)	LLFMin	Overload
01.12.2005	12:58:33	72,1	73,6	80,7	71,4	0
01.12.2005	12:58:34	70,7	72,4	80,4	69,2	0
01.12.2005	12:58:35	68,2	70,4	79,7	65,7	0
01.12.2005	12:58:36	67,2	69,7	78,3	65,4	0
01.12.2005	12:58:37	68,6	70,1	78,3	66,3	0
01.12.2005	12:58:38	67,1	69,1	79,6	65,0	0
01.12.2005	12:58:39	67,9	69,5	79,0	66,2	0
01.12.2005	12:58:40	68,6	69,8	77,9	66,5	0
01.12.2005	12:58:41	69,8	70,9	81,5	67,9	0
01.12.2005	12:58:42	69,7	70,8	80,7	68,0	0
01.12.2005	12:58:43	72,2	73,1	85,3	69,5	0
01.12.2005	12:58:44	75,5	76,6	89,4	73,0	0
01.12.2005	12:58:45	77,4	79,1	91,1	76,1	0
01.12.2005	12:58:46	78,1	79,4	90,8	77,5	0
01.12.2005	12:58:47	77,6	78,3	90,3	77,0	0
01.12.2005	12:58:48	80,6	82,0	93,3	77,8	0
01.12.2005	12:58:49	84,0	85,3	97,4	81,9	0
01.12.2005	12:58:50	83,6	85,2	95,9	83,0	0
01.12.2005	12:58:51	82,6	83,5	94,9	81,2	0
01.12.2005	12:58:52	79,7	81,4	92,1	78,3	0
01.12.2005	12:58:53	77,6	78,7	91,4	76,7	0
01.12.2005	12:58:54	74,8	76,8	87,8	73,1	0
01.12.2005	12:58:55	72,3	73,9	84,8	71,0	0
01.12.2005	12:58:56	70,0	72,2	81,2	69,1	0
01.12.2005	12:58:57	70,1	71,4	80,8	68,3	0
01.12.2005	12:58:58	69,0	70,7	81,2	67,1	0
01.12.2005	12:58:59	68,2	69,8	79,2	67,0	0
01.12.2005	12:59:00	67,7	69,2	79,0	66,5	0
01.12.2005	12:59:01	66,7	68,1	78,7	64,8	0
01.12.2005	12:59:02	68,3	69,0	80,4	66,8	0
01.12.2005	12:59:03	69,2	70,3	82,8	67,9	0
01.12.2005	12:59:04	69,9	71,1	82,1	68,9	0
01.12.2005	12:59:05	70,4	71,1	83,9	70,0	0
01.12.2005	12:59:06	72,3	73,7	84,2	70,4	0
01.12.2005	12:59:07	71,4	73,6	85,7	70,7	0
01.12.2005	12:59:08	71,2	71,8	82,7	70,7	0
01.12.2005	12:59:09	72,0	73,6	84,1	70,9	0
01.12.2005	12:59:10	71,9	72,8	84,3	71,4	0
01.12.2005	12:59:11	71,5	72,2	84,1	71,2	0
01.12.2005	12:59:12	71,5	72,6	84,8	70,9	0
01.12.2005	12:59:13	71,3	72,2	84,4	70,8	0
01.12.2005	12:59:14	71,7	73,2	86,0	70,9	0
01.12.2005	12:59:15	71,7	73,0	83,9	71,1	0
01.12.2005	12:59:16	71,2	72,6	83,3	70,0	0
01.12.2005	12:59:17	70,3	71,4	82,1	69,6	0
01.12.2005	12:59:18	69,7	71,1	82,4	68,6	0
01.12.2005	12:59:19	68,6	71,0	80,9	67,2	0
01.12.2005	12:59:20	70,0	70,9	81,4	69,0	0
01.12.2005	12:59:21	70,0	71,8	83,5	68,7	0
01.12.2005	12:59:22	72,0	73,0	84,3	71,1	0

Start date	Start time	LLeq	LLFMax	LLpk(MaxP)	LLFMin	Overload
01.12.2005	12:59:23	74,1	75,8	86,9	72,1	0
01.12.2005	12:59:24	75,7	76,9	88,6	74,4	0
01.12.2005	12:59:25	77,2	79,0	89,1	75,8	0
01.12.2005	12:59:26	75,4	77,0	88,6	74,6	0
01.12.2005	12:59:27	74,9	76,2	88,4	74,1	0
01.12.2005	12:59:28	76,9	78,3	88,9	75,3	0
01.12.2005	12:59:29	78,9	79,5	89,7	77,9	0
01.12.2005	12:59:30	79,2	80,3	90,7	77,1	0
01.12.2005	12:59:31	76,1	77,5	89,1	74,9	0
01.12.2005	12:59:32	74,0	75,5	86,6	72,9	0
01.12.2005	12:59:33	75,9	76,9	89,0	74,4	0
01.12.2005	12:59:34	79,1	80,6	91,4	76,8	0
01.12.2005	12:59:35	82,8	84,5	96,2	80,4	0
01.12.2005	12:59:36	84,5	85,0	96,2	83,8	0
01.12.2005	12:59:37	83,3	84,8	94,7	82,4	0
01.12.2005	12:59:38	81,0	83,3	94,8	79,5	0
01.12.2005	12:59:39	79,4	80,4	91,1	78,1	0
01.12.2005	12:59:40	79,4	80,1	92,9	78,3	0
01.12.2005	12:59:41	80,5	81,3	92,3	79,7	0
01.12.2005	12:59:42	81,7	82,6	94,8	80,5	0
01.12.2005	12:59:43	83,5	85,1	96,6	82,3	0
01.12.2005	12:59:44	87,2	88,4	99,1	85,1	0
01.12.2005	12:59:45	89,3	90,6	101,5	87,7	0
01.12.2005	12:59:46	87,4	89,4	101,7	85,3	0
01.12.2005	12:59:47	82,1	85,4	95,7	80,0	0
01.12.2005	12:59:48	80,8	82,0	92,7	80,0	0
01.12.2005	12:59:49	82,6	83,6	94,1	81,3	0
01.12.2005	12:59:50	83,7	84,5	95,0	83,0	0
01.12.2005	12:59:51	82,5	84,1	92,8	81,5	0
01.12.2005	12:59:52	80,6	82,5	93,1	79,6	0
01.12.2005	12:59:53	79,1	79,7	90,5	78,6	0
01.12.2005	12:59:54	77,3	79,3	89,0	76,0	0
01.12.2005	12:59:55	75,4	77,1	89,2	74,3	0
01.12.2005	12:59:56	74,3	75,3	86,3	73,6	0
01.12.2005	12:59:57	72,8	73,9	86,0	72,2	0
01.12.2005	12:59:58	72,4	73,2	84,5	71,9	0
01.12.2005	12:59:59	71,5	72,3	83,3	70,6	0
01.12.2005	13:00:00	70,9	72,2	83,0	69,5	0
01.12.2005	13:00:01	70,2	72,0	83,8	69,0	0
01.12.2005	13:00:02	69,5	70,8	81,4	68,2	0