VRIJE
UNIVERSITEIT
BRUSSEL

Master thesis submitted in order to be awarded the degree of
Master of Science in Engineering Technology:
Electromechanical Engineering - aeronautics

SOCIAL ACCEPTANCE OF EVTOL UAM DRONE: CHARACTERIZATION OF BACKGROUND NOISE

Evy Heymans and Brent Van Ginderdeuren

2024-2025

Promoter: prof. dr. ir. A. Bellemans
Co-promoter: prof. dr. ir. D. Peumans & prof. dr. ir. C. Schram

ENGINEERING SCIENCES

Acknowledgements

We would like to express our gratitude to Prof. Dr. Ir. Aurélie Bellemans for providing the opportunity to carry out this master's thesis and for her supervision throughout the process.

We would also like to thank Prof. Dr. Ir. Christophe Schram and the Von Karman Institute for providing the equipment that was essential for the completion of this master's thesis.

Finally, we would like to thank Prof. Dr. Ir. Dries Peumans and Prof. Dr. Ir. Yves Rolain for their presence, guidance and additional explanations throughout the course of this master's thesis.

Evy Heymans and Brent Van Ginderdeuren

Brussels, October 4, 2025

Abstract

The emergence of electrical Vertical Take-Off and Landing (eVTOL) aircraft raises concerns about their acoustic integration in dense urban environments. A key challenge is assessing whether the characteristic tonal noise of eVTOL vehicles blends into or stands out against the urban background. This thesis addresses this challenge through the central research question:"How can a low-cost and autonomous acoustic noise measurement device be designed, constructed and calibrated to reliably capture urban background noise and evaluate the acoustic impact of eVTOL aircraft?".

To answer this question, a low-cost, autonomous noise measurement device was designed and validated for long-term urban monitoring. The prototype integrates a MEMS microphone, Raspberry Pi, a data acquisition system, 4G connectivity and solar-powered autonomy into a robust setup that can operate outdoors with minimal intervention. Calibration against a reference microphone limited the reliable frequency range to 500-3000 Hz. Within this band, an initial field test conducted on the VUB campus recorded urban sound pressure levels between 42 and 57 dB SPL. The test shows both gradual variations and transient peaks possibly caused by traffic surges and emergency sirens. Tonal components were also observed, most prominently around 1 kHz and 2 kHz. This illustrates the system's capability to detect narrowband features in the soundscape. A comparison with numerical predictions of eVTOL rotor noise suggests that tonal peaks could exceed the urban background by more than 10 dB directly beneath the vehicle, while lateral observers may experience partial masking. These first results indicate that the developed device provides a promising and affordable proof-of-concept for assessing the acoustic footprint of eVTOL aircraft in urban contexts, but further measurements are required to establish reliability and robustness.

Samenvatting

De opkomst van elektrische Vertical Take-Off and Landing (eVTOL) vliegtuigen roept vragen op over hun akoestische integratie in dichtbevolkte stedelijke omgevingen. Een belangrijke uitdaging is te onderzoeken in hoeverre het karakteristieke tonale geluid van eVTOL's wegvalt in of juist afsteekt tegen de stedelijke achtergrond. Deze masterproef behandelt deze problematiek aan de hand van de centrale onderzoeksvraag: "Hoe kan een goedkoop en autonoom akoestisch meetsysteem worden ontworpen, gebouwd en gekalibreerd om op betrouwbare wijze stedelijke achtergrondgeluiden vast te leggen en de akoestische impact van eVTOL-vliegtuigen te evalueren?

Om deze vraag te beantwoorden werd een goedkoop, autonoom meetsysteem ontwikkeld en gevalideerd voor langdurige geluidsmonitoring in stedelijke context. Het prototype combineert een MEMS-microfoon, een Raspberry Pi met data-acquisitiesysteem, 4G-connectiviteit en een op zonne-energie gebaseerde energievoorziening in een robuuste configuratie die buitenshuis kan functioneren met minimale interventie. Kalibratie tegen een referentiemicrofoon beperkte het betrouwbare frequentiebereik tot 500–3000 Hz. Binnen dit bereik registreerde een eerste meting op de VUB-campus stedelijke geluidsdrukniveaus tussen 42 en 57 dB SPL. De metingen tonen zowel geleidelijke variaties als kortstondige pieken, vermoedelijk veroorzaakt door verkeersintensivering en sirenes van hulpdiensten. Daarnaast werden tonale componenten waargenomen, met name rond 1 kHz en 2 kHz. Dit illustreert de capaciteit van het systeem om smalbandige kenmerken in het geluidslandschap te detecteren. Een vergelijking met numerieke voorspellingen van eVTOL-rotorgeluid suggereert dat tonale pieken het stedelijke achtergrondniveau met meer dan 10 dB kunnen overstijgen direct onder het toestel, terwijl waarnemers opzij deels maskering kunnen ervaren. Deze eerste resultaten tonen aan dat het ontwikkelde systeem een veelbelovend en betaalbaar proof-of-concept vormt voor de evaluatie van de akoestische voetafdruk van eVTOL's in stedelijke context. Verdere metingen zijn echter noodzakelijk om de betrouwbaarheid en robuustheid volledig te onderbouwen.

Resumé

L'émergence des aéronefs à décollage et atterrissage verticaux électriques (eVTOL) soulève des questions quant à leur intégration acoustique dans des environnements urbains denses. Un défi majeur consiste à examiner dans quelle mesure le bruit tonal caractéristique des eVTOL se fond dans, ou au contraire se détache du bruit de fond urbain. Cette thèse aborde cette problématique à travers la question de recherche centrale : "Comment concevoir, construire et étalonner un système de mesure acoustique autonome et à faible coût capable de capter de manière fiable le bruit de fond urbain et d'évaluer l'impact acoustique des aéronefs eVTOL ?"

Pour répondre à cette question, un système de mesure acoustique autonome et à faible coût a été conçu et validé pour la surveillance de longue durée en milieu urbain. Le prototype combine un microphone MEMS, un Raspberry Pi avec système d'acquisition de données, une connectivité 4G ainsi qu'une alimentation solaire, le tout intégré dans une configuration robuste pouvant fonctionner en extérieur avec une intervention minimale. L'étalonnage par rapport à un microphone de référence a limité la plage fréquentielle fiable à 500–3000 Hz. Dans cette bande, un premier essai de terrain sur le campus de la VUB a enregistré des niveaux de pression sonore urbaine compris entre 42 et 57 dB SPL. Les mesures révèlent à la fois des variations progressives et des pics transitoires, probablement dus à des intensifications du trafic et à des sirènes de véhicules d'urgence. Des composantes tonales ont également été observées, notamment autour de 1 kHz et 2 kHz, illustrant la capacité du système à détecter des caractéristiques étroites du paysage sonore. La comparaison avec des prédictions numériques du bruit des rotors d'eVTOL suggère que les pics tonals pourraient dépasser le bruit de fond urbain de plus de 10 dB directement sous l'aéronef, tandis que les observateurs latéraux pourraient percevoir un certain masquage. Ces premiers résultats montrent que le système développé constitue une preuve de concept prometteuse et abordable pour l'évaluation de l'empreinte acoustique des eVTOL en contexte urbain. Toutefois, des mesures supplémentaires sont nécessaires afin d'établir pleinement sa fiabilité et sa robustesse.

Disclosure on the use of AI

Declaration

During the preparation of this work, the author(s) used the following tool(s) and service(s):

name of tool/service		reason
ChatGPT (OpenAI)	in order to:	*improve the clarity, grammar and readability of the text. Also to obtain alternative formulations of our own sentences.*

After using this tool(s) and/or service(s), the author(s) reviewed and edited the content as needed. The author(s) take(s) full responsibility for the content of this work.

Contents

Acknowledgements i

Abstract ii

Samenvatting iii

Resumé iv

Disclosure on the use of AI v

Contents vi

List of Figures x

List of Tables xiii

Listings xiv

Nomenclature xv

1 Introduction 1

2 Literature review 3
- 2.1 Characteristics of sound . . . 3
- 2.2 Quantification of the strength of sound . . . 4
  - 2.2.1 Sound intensity . . . 4
  - 2.2.2 Sound pressure level . . . 4
    - 2.2.2.1 Signal-to-noise ratio . . . 5
- 2.3 Types of microphones . . . 6
  - 2.3.1 The Electret Condenser Microphone . . . 6
    - 2.3.1.1 The operation principle of an Electret Condenser Microphone . 6
  - 2.3.2 The MEMS microphone . . . 6
  - 2.3.3 The functioning of a MEMS microphone . . . 7
  - 2.3.4 The difference between analog and digital MEMS microphones . . . 7
  - 2.3.5 Specification of a MEMS microphone . . . 8
    - 2.3.5.1 Microphone sensitivity . . . 8
    - 2.3.5.2 Acoustic overload point . . . 9
  - 2.3.6 Overview of selected microphones for testing . . . 9
  - 2.3.7 MEMS Microphones in prior experiments . . . 10
    - 2.3.7.1 Low-Cost sensors for urban noise monitoring etworks by Picaut, J. . . . 10
    - 2.3.7.2 The design and calibration of low cost urban acoustic sensing devices . . . 10
- 2.4 Data Acquisition System . . . 11

2.4.1 Data Acquisition systems Classification . . . 11
2.4.1.1 Computer-based DAQ . . . 11
2.4.1.2 Embedded microcontroller-based DAQ . . . 11
2.4.1.3 FPGA-based DAQ . . . 11
2.4.2 Projects where DAQs are being used . . . 12
2.4.2.1 Detection of damage on inner and outer races of ball bearings using a low-cost monitoring system and deep convolutional neural networks . . . 12
2.5 Data transfer from RPI to a cloud . . . 13
2.5.1 Setting up Rclone on a Raspberry Pi . . . 13
2.6 Signal processing . . . 13
2.6.1 Spectral analysis: the Fourier transform . . . 13
2.6.1.1 Mathematical definition and interpretation . . . 14
2.6.1.2 Spectrum representation . . . 14
2.6.1.3 Application of Fourier transform in microphone arrays . . . 15
2.6.2 Sampling . . . 15
2.6.3 Possible problems . . . 16
2.6.3.1 Aliasing . . . 16
2.6.3.2 Preventing aliasing: Shannon-Nyquist theorem . . . 16
2.7 4G connectivity . . . 17
2.7.1 4G HAT LTE . . . 17
2.7.2 4G dongle . . . 18
2.8 Ensuring autonomous operation . . . 18
2.8.1 Powering the system with a solar cell . . . 18
2.8.2 The photovoltaic effect . . . 19
2.8.2.1 Discovery and meaning of the photovoltaic effect . . . 19
2.8.2.2 Structure of a silicon crystal . . . 19
2.8.2.3 Photon absorption and energy conversion . . . 19
2.8.2.4 Formation and role of the electric field . . . 19
2.8.3 Maximizing light absorption in silicon solar cells . . . 21
2.9 Batteries . . . 21
2.9.1 Operating principle of a battery . . . 21
2.9.2 The design of a battery . . . 22
2.9.2.1 The cylindrical shape . . . 22
2.9.2.2 The prismatic shape . . . 22
2.9.2.3 The pouch cell . . . 23
2.9.3 Different kinds of batteries . . . 23
2.9.3.1 Lead-acid batteries . . . 23
2.9.3.2 The nickel-cadmium battery (NiCd) . . . 23
2.9.3.3 The nickel metal-hybrid battery (NiMH) . . . 24
2.9.3.4 The lithium-ion battery (Li-ion) . . . 24
2.9.3.5 Lithium polymer battery (LiPO) . . . 24
2.9.3.6 Battery management system (BMS) . . . 25
2.9.3.7 Charge regulator . . . 25
2.10 Design of a PV system . . . 26
2.10.1 The electric load . . . 26
2.10.2 Impact of sun position and panel orientation on PV output . . . 27
2.10.3 Battery sizing . . . 27
2.10.4 Comparable projects with low-cost sound level meters . . . 28
2.10.4.1 Picault's Low-Cost Sensors for urban noise monitoring networks 28
2.10.5 Citizen Science-Based environmental noise monitoring in Flanders . . . 28

2.10.6 A Low-Cost sleep monitoring system to assess environmental noise impact on sleep . . . 29

3 Materials and Methods 31
3.1 Materials . . . 31
3.1.1 Technical Overview of the ICS-40638 Microphone . . . 31
3.1.2 INA129-EP precision, low power instrumentation amplifier . . . 32
3.1.3 The communication module . . . 33
3.1.4 The data acquisition system . . . 33
3.1.5 The ZTE MF79U modem for autonomous data transmission . . . 34
3.1.6 Sizing of the photovoltaic module . . . 35
3.1.6.1 Technical specifications of the PV module . . . 36
3.1.7 Sizing of the battery . . . 37
3.1.7.1 Selection and specification of the battery . . . 38
3.1.8 Sizing of the solar charging controller . . . 38
3.1.8.1 Technical specifications of the solar charge controller . . . 39
3.2 Methods . . . 40
3.2.1 MEMS microphone hardware design and experimental evaluation . . . 40
3.2.1.1 Microphone selection and evaluation board design . . . 40
3.2.1.2 Implementation and testing of the evaluation board . . . 40
3.2.1.3 Signal evaluation and voltage analysis using the EV_ICS-40638-FX evaluation board . . . 41
3.2.1.4 Design of the first PCB prototype . . . 43
3.2.2 Testing of the first PCB prototype . . . 44
3.2.2.1 Final PCB design . . . 45
3.2.3 Calibration of the Microphone . . . 47
3.2.3.1 Goal of the calibration . . . 47
3.2.3.2 Setup of the calibration measurements . . . 47
3.2.3.3 Time-domain representation of the unprocessed microphone signal 49
3.2.3.4 Discrete Fourier Transform . . . 50
3.2.3.5 Mitigating Spectral Leakage: Hanning Window . . . 50
3.2.3.6 Normalization of the frequency spectrum . . . 51
3.2.3.7 Improving frequency and amplitude estimation accuracy: The Grandke method . . . 51
3.2.3.8 Adjustment of the spectrum based on the transfer function . . . 53
3.2.3.9 Calibration to an absolute reference level . . . 55

4 Design 56
4.1 Hardware of the microphone setup . . . 56

5 Results 58
5.1 Results of the calibration process . . . 58
5.1.1 First calibration process . . . 58
5.1.1.1 Setup for the first calibration . . . 58
5.1.1.2 Results of the first calibration process . . . 58
5.1.1.3 Conclusion after the first calibration . . . 61
5.1.2 Second calibration measurements . . . 61
5.1.2.1 Setup of the measurements . . . 61
5.1.2.2 Results of the second calibration process . . . 61
5.1.2.3 Conclusion of the second calibration process . . . 64
5.1.3 Final calibration measurements . . . 65
5.1.3.1 Setup of the final measurements . . . 65

5.1.3.2 Results of the final calibration process 65
5.2 Measurements of urban noise 72
5.2.1 Setup and acoustic characterization of the measurement site 75
5.2.2 Results of the measurements 76
5.2.2.1 The sound pressure levels 76
5.2.2.2 The frequency spectra 78
5.2.2.3 Solar power performance during measurements 79
5.2.2.4 eVTOL noise in context of urban background 79

6 Conclusions 81

7 Future work 83

A Codes 85
A.1 Python 85
A.2 MATLAB 95

Bibliography 101

List of Figures

2.1 A graphical representation of a sound wave, illustrating properties such as wavelength and amplitude. . . . 3
2.2 Visual representation of sound pressure level decrease with the distance (Lewis, 2013b). 4
2.3 An Electret Condenser Microphone (Adafruit, 2025). . . . 6
2.4 Microphone clipping in a time-domain plot (InvenSense, 2016). . . . 9
2.5 ICS-40300 (TDK InvenSense, 2018) & ICS-40720 (TDK InvenSense, 2019b). . . . 10
2.6 Stacked up MCC128 HATs on a Raspberry Pi. (Corporation, 2020) . . . 12
2.7 Examples of the Fourier transform of certain functions (Rolain, 2023). . . . 14
2.8 The oversampled signal and undersampled signal (Por, 2019). . . . 16
2.9 Voltage when the process is in idle state (Moyano, 2022). . . . 17
2.10 Illustrations of an 4G USB Dongle (IMILINK, 2020) and the Waveshare SIM7600G-H 4G Hat (Waveshare, 2025). . . . 18
2.11 Solar cell, solar module and solar array (Vastav et al., 2016). . . . 19
2.12 Illustration of donor and acceptor doping in a silicon semiconductor. . . . 20
2.13 Working principle of a solar cell: A photon is absorbed, creating an electron-hole pair. The built-in electric field drives the charges apart, generating an electric current (Ramrakhiani, 2017). . . . 20
2.14 Schematic Representation of a battery (Science News, 2025). . . . 21
2.15 Schematic representation of different battery configurations. (a) cylindrical cell, (b) prismatic cell and (c) pouch cell (Bilgin et al., 2015). . . . 22
2.16 The indoor noise unit and outdoor noise unit in an operational setting (Dekoninck, Wolters, et al., 2023). . . . 29
2.17 Indoor and outdoor noise levels with corresponding heart rate and motility signals (Dekoninck, Wolters, et al., 2023). . . . 30

3.1 The ICS-40638 MEMS microphone which is integrated in the noise monitoring setup (T. InvenSense, 2024). . . . 31
3.2 The INA129-EP and its simplified schematic. . . . 32
3.3 The Raspberry Pi 4 Model B.(Raspberry Pi (Trading) Ltd, 2024). . . . 33
3.4 The MCC 128 DAQ HAT, used for reading and digitizing the analog microphone signal (Measurement Computing Corporation, 2023). . . . 34
3.5 The ZTE MF79U LTE USB Modem used for mobile connectivity in the measurement setup (Amazon, 2025d). . . . 34
3.6 Monocrystalline solar panel 30W and 12V (Amazon, 2025b). . . . 37
3.7 12.8V and 12Ah $LiFePO_4$ battery selected for energy storage in the system (Amazon, 2025a). . . . 38
3.8 30A MPPT solar charge controller selected for battery regulation and system power delivery (Amazon, 2025c). . . . 39
3.9 Schematic diagram of the ICS-40638 MEMS microphone evaluation board. . . . 40
3.10 The ICS-40638 MEMS microphone evaluation board (TDK InvenSense, 2025). . . . 41
3.11 Time-domain signal of a hand clap captured by the ICS-40638 MEMS microphone evaluation board. . . . 41

3.12 Schematic diagram of the first PCB design integrating the ICS-40638 MEMS microphone with the INA129-EP instrumentation amplifier. 43
3.13 Assembled first custom PCB prototype designed for the ICS-40638 MEMS microphone. 44
3.14 First PCB prototype evaluation: clipping phenomenon observed, with 0 V reference. 44
3.15 First PCB prototype evaluation: restored signal with 2.5 V reference. 45
3.16 Schematic diagram of the final PCB design integrating the ICS-40638 MEMS microphone with the INA129-EP instrumentation amplifier. 45
3.17 Assembled final custom PCB designed for the ICS-40638 MEMS microphone. . . . 46
3.18 Measured output waveform of the final PCB design at a gain of 50. 46
3.19 Reference microphone (a) and sound calibrator (b) used for calibration. 47
3.20 The frequency response of the G.R.A.S. Type 40PL microphone (GRAS Sound & Vibration, 2025) 48
3.21 The HP 3562A Dynamic Signal Analyzer which was used to define the transfer function during the calibration measurements. 48
3.22 Example of an 1 kHz signal measured in the time-domain by the uncalibrated ICS-40638. 49
3.23 Shape of the Hann window function, which smoothly tapers the signal edges. . . . 50
3.24 Comparison between the windowed time signal and the unprocessed time signal. . 51
3.25 Spectrum from the 1 kHz sine wave signal shown in fig. 3.22. 52
3.26 Visualization of local maxima in the frequency spectrum, with red points marking dominant frequency components. 52
3.27 Example of an original spectrum (blue) and an interpolated spectrum (orange) after the method of Grandke. 54

4.1 The printed box in which the microphone is placed to conduct its measurements. . 56
4.2 The printed case in which the battery and the solar charge controller are placed. An ON/OFF-switch and 2 buttons are placed on the case to start or end the measurements. 57

5.1 A 2 kHz sine wave signal in the time domain recorded by both microphones. . . . 58
5.2 Frequency spectrum of both microphones. The excitation signal was a noise signal at attenuation level 3. 59
5.3 Frequency spectrum of both microphones. The excitation signal was a sine wave signal with a frequency of 1 kHz at attenuation 32. 59
5.4 Frequency spectrum of both microphones. The excitation signal was a chirp signal between 20 Hz and 10 kHz at attenuation level 39. 60
5.5 RMS values of the uncalibrated and the calibrated microphones at different attenuation levels. The lower the level of attenuation, the louder the sound is. 60
5.6 A chirp signal from 0-20 kHz in the time domain measured by the test microphone. 61
5.7 A chirp signal from 0-20 kHz in the frequency domain measured by the test microphone. 62
5.8 One-Sided Spectrum with Highlighted Spectral Peaks 62
5.9 Original spectrum of the measurement of the test microphone versus the interpolated spectrum after the Grandke method. 63
5.10 Original double-sided spectrum with the interpolated spectrum on the positive side. 63
5.11 The estimated model of the transfer function. 64
5.12 The inversed unprocessed transfer function. This graph has been inversed to get a more clear view of the lower frequency region. 64
5.13 An unprocessed chirp signal from 0-5 kHz in the time domain measured by the test microphone. 65
5.14 The local spectral maxima in the one-sided spectrum. These maxima are used to correct the spectrum using Grandke's method. 66

5.15 Original spectrum of the measurement of the test microphone versus the interpolated spectrum of the same microphone after the Grandke method. 66
5.16 Original double-sided spectrum with the interpolated spectrum until 5 kHz on the positive side. 66
5.17 Measured transfer function amplitude compared with the analytical model, including measurement uncertainty. 67
5.18 The coherence between the test microphone and the reference microphone. This plot shows a good coherence between 500-3000 Hz, but a lower coherence at the other frequencies. 67
5.19 Measured transfer function phase compared with the analytical model as a function of frequency. 68
5.20 Maximum likelihood residuals as a function of frequency. 69
5.21 The measured spectrum from the test microphone compared to the corrected spectrum. The corrected spectrum is the original spectrum multiplied with the inverse transfer function. 69
5.22 Average corrected spectrum of the test microphone together with the standard deviation. 70
5.23 Mean spectral amplitude of the test microphone when it has been placed in the Sound Calibrator. 70
5.24 The corrected spectrum of the ICS-40638 and the spectral value at 1 kHz of the Type 40PL. 71
5.25 The corrected spectrum of the ICS-40638 vertically shifted to equal the spectral value of the Type 40PL at 1 kHz. 71
5.26 The model of the transfer function from the 0-20 kHz range. The model shows many distortions making it not useful for further measurements. 72
5.27 Coherence between the reference and test microphones for the 0–20 kHz sweep. High coherence is observed up to about 3–4 kHz, while strong fluctuations between 4 and 16 kHz indicate reduced reliability of the measurements in this range. 72
5.28 The inverse transfer function obtained from the final calibration process. 73
5.29 The filtered frequency spectrum of a random measurement. This graph clearly shows the brick-wall band-pass filter. 73
5.30 SPL in Brussels at a certain time. Each graph represents 5 seconds of measurements. 74
5.31 Map of the VUB Etterbeek campus with indication of the measurement site (Vrije Universiteit Brussel, 2025). 75
5.32 Noise monitoring system with solar panel, solar charger and battery pack. 75
5.33 Graph of the sound pressure levels from 13:26 until 14:45. 76
5.34 Graph of the sound pressure levels from 15:45 until 16:48. 77
5.35 Graph of the sound pressure levels from 16:51 until 17:43. 77
5.36 The frequency spectrum of the measurements from 13:26 until 14:45. 78
5.37 The frequency spectrum of the measurements from 15:45 until 16:48. 78
5.38 The frequency spectrum of the measurements from 16h51 until 17h43. 79
5.39 Multirotor tonal noise at different blade-passing frequency harmonics, shown for downward-facing (top) and sideways-facing (bottom) directions (Lade et al., 2025). 80

List of Tables

2.1 Examples of sound sources with corresponding intensity levels. 5

3.1 Electrical characteristics of the ICS-40638 (T. InvenSense, 2024). 32
3.2 Electrical characteristics of the **INA129**-EP (Texas Instruments, 2015). 33
3.3 Estimated power consumption per component based on specifications. 35
3.4 Measured power consumption under startup and steady operating conditions. . . . 35
3.5 Monthly average peak sun hours at the installation site (Koninklijk Meteorologisch Instituut van België (KMI/IRM), 2020). 36
3.6 Measured output voltages from the MEMS microphone. 42

4.1 Overview of the costs of the components used in the measurement setup. 57

Listings

A.1 Signal processing in Python 85
A.2 Code that measures the dB SPL and uploads it to a Google Drive 86
A.3 Code that processes the measurement data and plots the SPL over time together with the spectrum 91
A.4 Code that calculates a rational transfer model. 95
A.5 Correction of the spectra with the inverse TF. 96
A.6 Average of the Grandke spectra of the pistonphone. 97
A.7 Absolute calibration of the microphones. 98

Acronyms

UAM	Urban Air Mobility
eVTOL	electric Vertical Take-Off and Landing
SPL	Sound Pressure Level
SNR	Signal-to-Noise Ratio
ECM	Electret Condenser Microphone
MEMS	Micro-Electro-Mechanical Systems
FET	Field-Effect Transistor
ASIC	Application-Specific Integrated Circuit
ADC	Analog-to-Digital Conversion
dBFS	decibels relative to full-scale
THD	Total Harmonic Distortion
DAQ	Data Acquisition
FPGA	Field-Programmable Gate Arrays
DNN	Deep Neutral Network
SE	Single-Ended
DIFF	Differential
GPIO	General Purpose Input/Output
API	Application Programming Interface
IFTTT	If This Then That
RPI	Raspberry PI
HDR	High Dynamic Range
USB	Universal Serial Bus
LAN	Local Area Network
LiPo	Lithium-Polymeer
NiCd	Nickel-cadmium
NiMh	Nickel-metal hydrid
Li-ion	Lthium-ion
UPS	Uninterruptible Power Supply
BMS	Battery Management System
PWM	Pulse Width Modulation
MPPT	Maximum Power Point Tracking
PV	Photovoltaic
DOD	Depth of Discharge
PIO	Program for Innovation Procuremen
IMEC	Interuniversitair Micro-Electronica Centrum
IOT	Internet of Things
WIFI	Wireless Fidelity
SIM	Subscriber Identity Module
EIN	Equivalent Input Noise

AOP	Acoustic Overload Point
THD	Total Harmonic Distortion
DC	Direct Current
AC	Alternating Current
RAM	Random Access Memory
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver
LTE	Long Term Evolution
KMI	Koninklijk Meteorologisch Instituut
IRM	Royal Meteorological Institute
PERC	Passivated Emitter Rear Cell
IP65	Ingress Protection
AGM	Absorbent Glass Mat
VDD	Voltage at the Drain
GND	Ground
PCB	Printed Circuit Board
SMD	Surface-Mounted Device
VUB	Vrije Universiteit Brussel
CHIREC	Centre Hospitalier Interrégional Edith Cavel
REF	Reference
FFT	Fast Fourier transform
CSV	Comma Separated Values
DFT	Discrete Fourier Transform
HAT	Hardware Attached on Top
ML	Maximum Likelihood
ASA	Acrylonitrile Styrene Acrylate
PLA	Polylactic Acid
UV	Ultraviolet
RMS	Root Mean Square
IEPE	Integrated Electronics Piezo-Electric
IFFT	Inverse Fast Fourier Transform
BPF	Blade-passing frequency
IIR	Infinite Impulse Response Filter
HAT	Hardware Attached on Top

Symbols

f	Frequency	Hz
T	Period	s
A	Amplitude	Pa
γ	Wavelength	m
v	Velocity	m/s
I	Intensity	W/m^2
S	Surface area	m^2
SPL	sound pressure level	Pa
T_s	Sampling period	s
F_s	Sampling frequency	Hz
f_{max}	Maximum frequency	Hz
I	Current	A
V	Voltage	V

E_r	Daily energy demand	Wh/day
P_p	Peak power requirement	W
D	Autonomy days	
G	Gain	
R_G	Resistor gain	Ω
R_b	Bit rate	Mbit/s
P	Power	W
T	Temperature	°C
G	Weight	kg
C	Capacity	F
R	Resistor	Ω
f_c	Cutoff frequency	Hz
t	Time	s
N	Samples	
n	Sample index	
u_t	Time signal	
$u_\mathrm{windowed}(t)$	Windowed time signal	
$u_\mathrm{nodc}(t)$	Time signal without DC-offset	
a	Amplitude Ratio	
l_m	Index of spectral maximum in DFT	
$X(l)$	Amplitude spectrum at bin l	dBV
f_m	Estimated frequency of spectral maximum	Hz
Δf	Frequency resolution of the DFT	Hz
A_m	Reconstructed amplitude of the spectral component	V
x_m	Interpolation factor	
$T(f)$	Transfer function	
X_test	Measured spectrum from the ICS-40638	dBV
X_ref	Measured spectrum from the Type 40-PL	dBV
n_num	Order of the numerator polynomial	
n_den	Order of the denominator polynomial	
σ_meas	Standard deviation of the measurements	

Chapter 1

Introduction

In recent years, cities have struggled with increasing traffic congestion and traditional transport systems are reaching their limits. As a potential solution, Urban Air Mobility (UAM) is being developed, offering the possibility to reduce road traffic while providing a cleaner and faster way of transporting both passengers and goods within urban areas. UAM relies on electric Vertical Take-off and Landing (eVTOL) drones. These are compact, electrically powered aircraft that can operate in dense urban areas without the need for runways. By taking off and landing vertically, these vehicles can be deployed from small landing sites and they can be operated manually by a pilot or remotely. Because they rely on electric propulsion, these aircraft are expected to have a reduced emission compared to conventional helicopters. However, despite these advantages, one of the main challenges for their acceptance is the noise they produce (European Union Aviation Safety Agency (EASA), 2025).

To address this challenge, the eVTOLUTION project was initiated. The project focuses on the development of advanced tools and methods that equip eVTOL aircrafts with improved aerodynamic performance and reduced noise emissions (European Commission, 2025). Conventional design approaches often struggle to deal with the complex aerodynamic and acoustic interactions between rotors, propulsion systems and the airframe. Low-fidelity simulations provide results quickly, but are not sufficiently accurate, while high-fidelity simulations offer more detail but are costly. These limitations hinder the ability of current design processes to deliver fast and accurate results.

eVTOLUTION proposes the implementation of a multi-fidelity approach to aircraft design and optimization. This approach combines different levels of simulation, ranging from simple models to highly detailed ones, in order to balance speed and accuracy in the design process. By combining data-driven surrogate models with physical simulations, the project aims to accelerate while maintaining robustness and reliability.

In addition to its aerodynamic focus, eVTOLUTION also addresses the acoustic challenges associated with eVTOL aircraft. A central objective of the project is to analyze and rank the main sources of noise, such as rotor self-noise, rotor-rotor interaction noise and rotor-rotor-airframe interaction noise. The results of this analysis will be used to reduce the noise emissions of optimized rotors trough the application of flow-control and noise-control technologies. In addition, the project studies how optimized departure and approach trajectories can help to further reduce the noise footprint in urban areas. These activities are important to prepare the certification process and to increase the acceptability of eVTOL technology in cities (eVTOLUTION, 2025).

Achieving these goals requires not only advanced simulation methods but also reliable noise measurements of urban sound, since urban environments are characterized by high levels of background noise. Therefore it is important to assess whether the noise generated by eVTOL aircraft blends into this background noise or stands out as an additional source of disturbance. For this purpose, this master's thesis contributes to the eVTOLUTION project by focusing on the design and validation of a low-cost acoustic measurement device. This device must be capable of measuring urban noise with high accuracy.

The developed system integrates a MEMS microphone with a Raspberry Pi 4, which records and processes the acoustic signals before transmitting them via a 4G connection to a cloud server. To enable autonomous operation in outdoor conditions, the system is powered by a solar panel with integrated battery storage. In this way, the design is not only affordable but also robust and adaptable to different research and monitoring contexts.

This thesis proceeds in several stages. It first focuses on the development of the prototype, in which both hardware and software are integrated into a functional system. The performance of the setup is then tested by generating sound with loudspeakers and analyzing the microphone response with an oscilloscope. Once the system's functionality had been confirmed, the microphone was calibrated against a reference microphone in order to evaluate its accuracy. Finally, field test have been performed in Brussels to assess the performance of the prototype in a real urban setting.

By providing an affordable, autonomous and reliable noise monitoring device, the thesis equips the eVTOLUTION project with a practical tool to monitor and analyze the complex noise environment of cities. The developed microphone makes it possible to assess how eVTOL aircraft influence city noise and contribute to the development of quieter and more sustainable air mobility aircraft that meet society's demand for sustainable and livable cities.

The central research question of this thesis is: "How can a low-cost and autonomous acoustic noise measurement device be designed, constructed and calibrated to reliably capture urban background noise and evaluate the acoustic impact of eVTOL aircraft?".

2.1 Characteristics of sound

The sound we perceive is created by a time-dependent change in pressure as a vibration in a medium, such as water, air or solid material. These changes are caused by a vibrating surface or turbulent motion of a fluid (Hansen, 1951). Sound moves in the form of longitudinal waves, as the air particles move in the same direction as the wave travels (Rumsey, 2009).

We can characterize sinusoidal sound waves by their frequency (f), period (T), amplitude (A) and wavelength (λ). The frequency of a sinusoidal sound wave is the number of periods per second and is expressed in Hertz (Hz) (Hansen, 1951). The period of such a wave is defined as the time interval between two successive peaks and can be calculated using eq. (2.1).

$$T = \frac{1}{f} \tag{2.1}$$

The amplitude of the sound wave, is defined as the maximum pressure difference between any instantaneous pressure and its equilibrium position (Giancoli, 2016). The wavelength (λ)corresponds to the distance the wave propagates during one period and is determined by its speed (Rumsey, 2009). The speed of the sound wave depends on the medium through which it propagates. In air, sound travels at an approximate speed (v) of 340 m/s under normal atmospheric conditions (Malcoci, 2014). However the speed can vary depending on the temperature, pressure and humidity of the air (Rumsey, 2009).

The properties discussed so far apply only to pure sinusoidal waves, which correspond to sound that is present at a single frequency. In reality, sound consists of a mix of pure sinusoidal pressure changes, resulting in a continuous frequency spectrum. For real sounds, there is no simple relationships between the frequency (f), period (T), amplitude (A) and wavelength (λ). Nevertheless, many sounds can be broken down into a combination of sinusoidal waves (Hansen, 1951). When the waveform becomes aperiodic, the sound is classified as noise (Rumsey, 2009).

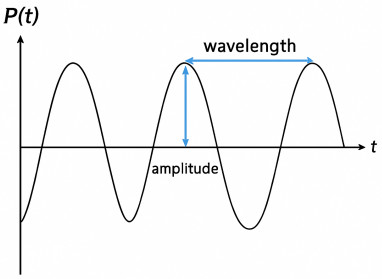


Figure 2.1: A graphical representation of a sound wave, illustrating properties such as wavelength and amplitude.

2.2 Quantification of the strength of sound

2.2.1 Sound intensity

The strength of sound can be described by its intensity. Sound intensity is defined as the amount of energy transmitted per unit time, through a surface that is oriented perpendicular to the direction of propagation (Giancoli, 2016). Intensity (I) is expressed in Watts per square meter (W/m^2) and can be calculated using eq. (2.2), where W is the power in watts and S is the surface area in square meters (Hansen, 1951).

$$I = \frac{W}{S} \tag{2.2}$$

The human ear is capable of detecting sound intensities ranging from 10^{-12} to $1\,W/m^2$ (Giancoli, 2016).

2.2.2 Sound pressure level

Sound pressure reflects how sound power affects its surroundings (Rumsey, 2009). A louder sound, such as a conversation, generates a larger sound pressure compared to a softer sound, such as the drop of a needle (Vinck and Degeest, 2024).

Sound Pressure Level (SPL) is measured in Pascal. This is not very practical due to the large dynamic range of pressures the human ear can observe. The ear can detect a minimum sound pressure of 20 µPa and a maximum of 60 Pa. This means that a linear scale based on the square of the sound pressure would be extremely large, requiring 10^{13} unit divisions to cover the range of human hearing. To reduce this range, a logarithmic scale is used instead, specifically the decibel scale (Hansen, 1951).

For microphone measurements, the standard reference sound pressure level (SPL) is defined as 94 dB SPL, which corresponds to a sound pressure of 1 Pa. This reference level is used to measure microphone sensitivity and compare different models (InvenSense, 2016). Because the reference level of 0 dB SPL is set at 20×10^{-6} Pa, a sound pressure of 1 Pa corresponds to:

$$\text{SPL} = 20\log_{10}\left(\frac{1}{20 \times 10^{-6}}\right) = 94\,\text{dB SPL} \tag{2.3}$$

As shown in fig. 2.2, the sound pressure level decreases by 6 dB for every doubling of the distance from the sound source. This phenomenon becomes especially relevant in the far-field region, which occurs when the distance from the source exceeds approximately 10 times the acoustic wavelength (λ). A higher-sensitivity microphone may be necessary here to compensate for sound attenuation.

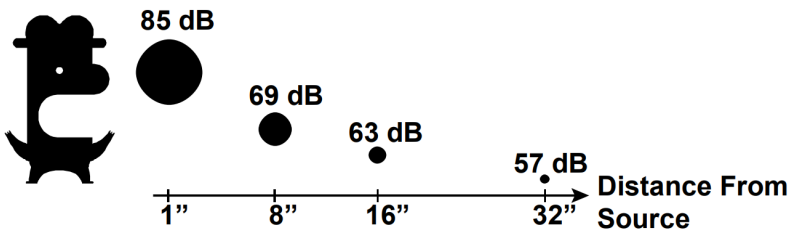


Figure 2.2: Visual representation of sound pressure level decrease with the distance (Lewis, 2013b).

Based on Nurjannah et al. (2017) and Lewis (2013b), table 2.1 can be set up. It provides an overview of various sounds sources along with their corresponding intensity levels and sound intensity. This information is crucial in understanding the range of sound pressure levels encountered in different environments and their potential impact on human perception and health. This classification of sound intensity levels is essential for designing sound measurement instruments and noise regulation policies.

Table 2.1: Examples of sound sources with corresponding intensity levels.

The Source of Sound	**Intensity Levels (dB)**	**Intensity of Sound (W/m²)**
Jet plane at a distance of 30 m	140	10^2
Threshold of pain	120	1
Loud rock concert indoors	120	1
Heavy Machine Shops	100	1×10^{-2}
Average Factory	75	3×10^{-5}
Busy road traffic	70	1×10^{-5}
Ordinary conversation (50 cm)	65	3×10^{-6}
Slow radio	40	1×10^{-8}
Whisper	20	1×10^{-10}
Leaf rustling	10	1×10^{-11}
Threshold of hearing	0	1×10^{-12}

2.2.2.1 Signal-to-noise ratio

In general, the signal-to-noise ratio (SNR) of a microphone and the measurements represents the ratio of the measured signal power and the measured noise power of the device. It is measured as the difference in decibel between the noise power level and a standard sine wave signal emitted at 1 kHz, with a 94 dB Sound Pressure Level. This specification accounts for both the microphone element, the analog signal conditioning, the analog-to-digital converter and the numeric noise of the internal ASIC. This ensures an accurate representation of the microphone's noise performance. The signal-to-noise ratio is typically weighted by an A-weighted value over a 20 kHz bandwidth to align the measurement with the auditory sensitivity. It is important to ensure consistency both in weighting and in bandwidth when comparing the signal-to-noise ratio across different microphone systems. A higher signal-to-noise ratio indicates superior performance, as it allows for a better dynamic range in the signal capture because of the minimal background noise interference. The equivalent input noise can be derived from the signal-to-noise ratio by subtracting it from 94 dB SPL, providing an additional perspective on the microphone's noise characteristics (InvenSense, 2016).

Zhu et al. (2023) gives a practical example about how the signal-to-noise ratio plays a crucial role in underwater noise monitoring, particularly in detecting low-noise ships that utilize acoustic stealth technologies. Traditional hydrophones struggle with low-SNR conditions due to their reliance on acoustic pressure measurements, which are highly susceptible to isotropic noise. To address this limitation, the MEMS noise listener presented in this study integrates a vector hydrophone, which simultaneously measures acoustic pressure and particle velocity. This approach enhances the signal-to-noise ratio by suppressing isotropic noise and improving the detectability of weak underwater acoustic signals. Experimental results indicate that the proposed system achieves a 10 dB gain in SNR compared to conventional sound pressure detection methods. Additionally, the ship noise monitoring experiment demonstrated a measured SNR of 10.38 dB, ensuring reliable detection. In contrast, when the SNR falls below 6 dB, the system's ability to distinguish ship noise from background noise diminishes, rendering the data invalid. These findings highlight the significance of high-signal-to-noise ratio measurement techniques in marine noise monitoring, enabling the detection of quieter targets in challenging acoustic environments.



Figure 2.3: An Electret Condenser Microphone (Adafruit, 2025).

2.3 Types of microphones

Microphones play a crucial role in capturing sound and converting it into an electrical signal. In this section, we will focus on two specific type of microphones: Electret Condenser Microphones (ECMs) and MEMS (Micro-Electro-Mechanical Systems) microphones.

2.3.1 The Electret Condenser Microphone

Electret Condenser Microphones (ECM) as shown in fig. 2.3 are widely used sound sensors. They are characterized by their affordability and good performance for general audio applications. ECMs consist of a permanently charged electret diaphragm and a backplate forming a capacitor. Sound waves cause the diaphragm to vibrate, leading to changes in the capacitance and this change is converted to an electrical signal (Bakhoum and Cheng, 2011)(Picaut, Can, Fortin, Ardouin, and Lagrange, 2020).

2.3.1.1 The operation principle of an Electret Condenser Microphone

The core of an electret microphone consists of a thin diaphragm that is placed in front of a fixed backplate. This diaphragm moves when sound waves reach it, causing fluctuations in the distance between the diaphragm and the backplate. As a result, the capacitance between these two elements changes and this change is converted into a small electric signal. However, this signal is typically too weak to be used directly, so an internal field-effect transistor (FET) amplifier is integrated into the microphone to boost the signal to a more higher level and reduce the impedance of the sensor (Bakhoum and Cheng, 2011) (Cho, 2018).

2.3.2 The MEMS microphone

Micro-Electromechanical Systems (MEMS) microphones are a type of transducer that uses micro electromechanical systems to convert acoustic pressure waves into electrical signals. They integrate a MEMS transducer, typically a variable capacitor, with an Application-Specific Integrated Circuit (ASIC) for signal amplification or conversion. This technology leverages manufacturing processes similar to those used in the silicon chip industry (Lewis, 2013a).

2.3.3 The functioning of a MEMS microphone

At the core of a MEMS microphone is a capacitive sensing mechanism, which consists of three major components:

1. Diaphragm: A flexible, movable plate that deflects in response to incoming sound waves.
2. Backplate: A fixed plate that forms a capacitor with the diaphragm.
3. Bias Voltage: Maintains an electric field between the plates.

When sound waves hit the diaphragm, the diaphragm moves. It alters the capacitance between the plates (C. B. King and Monti, 2022). This change is converted into an electrical signal by a preamplifier and processed further by signal conditioning for an analog microphone or signal conditioning and conversion to a data stream for a digital microphone (Lewis, 2013a).

2.3.4 The difference between analog and digital MEMS microphones

To make a comparison between analog and digital MEMS microphones, Picaut, Can, Fortin, and Ardouin (2020), C. B. King and Monti (2022) and Lewis (2013a) share the following:

1. **Signal Output:**
   - Analog MEMS Microphones:
     - Outputs an analog signal directly from the preamplifier.
     - The signal is prone to interference.
     - The signal requires external analog-to-digital conversion (ADC) for digital systems.
     - The output impedance is typically a few hundred Ohms and care must be taken to match the impedance of the subsequent stages to avoid signal attenuation.
   - Digital MEMS Microphones:
     - Integrate an ADC within the microphone package to convert the signal to a digital data stream.
     - These microphones are less susceptible to noise and interference, making them suitable for environments with high electromagnetic activity.
2. **Design and integration:**
   - Analog MEMS Microphones:
     - Simpler design with fewer internal components, resulting in lower power consumption.
     - More flexible for systems that require a custom ADC.
   - Digital MEMS Microphones:
     - Reduce circuit design complexity.
     - Require careful clock and signal management.
     - No need for external ADCs.
3. **Susceptibility to Noise and Interference:**
   - Analog MEMS Microphones:
     - More prone to noise and interference, especially over longer transmission paths.
     - Proper shielding is advised to minimize electrical interference.

  - Digital MEMS Microphones:
    - Immune to electromagnetic interference after signal digitization.

4. **Applications:**
   - Analog MEMS Microphones:
     - Preferred in space-constrained designs due to smaller package size.
     - Ideal for systems with external ADCs or where flexibility in analog signal processing is required.
   - Digital MEMS Microphones:
     - Useful for applications where minimizing interference and maximizing design simplicity are priorities.

2.3.5 Specification of a MEMS microphone

2.3.5.1 Microphone sensitivity

Microphone sensitivity is a fundamental parameter that determines the ratio between the microphone input sound level and the electrical output signal. It is defined as the ratio between the magnitude of the electrical output, which can be a voltage or a digital signal and the acoustic input (sound pressure). The sensitivity of a microphone determines the SNR of the channel. This has direct implications for signal processing and this is a crucial factor in selecting a microphone for specific applications. Sensitivity is typically measured using a 1 kHz sine wave at a sound pressure level of 94 dB, which corresponds to a pressure of 1 Pa. This serves as a standard reference for comparing different microphones (Lewis, 2013b) (InvenSense, 2016).

The sensitivity of microphones can be expressed in different ways, depending on the type of microphone. For analog microphones, sensitivity is often expressed in decibels relative to 1 Volt (dBV) or in millivolts per Pascal (mV/Pa). The conversion between these units is given by eq. (2.4), where the reference value of the output is equal to 1000 mV/Pa. A higher sensitivity means that a microphone generates a larger output voltage for the same sound pressure (InvenSense, 2016).

$$Sensitivity_{\mathrm{dBV}} = 20\log\left(\frac{Sensitivity_{\mathrm{mV/Pa}}}{Output_{\mathrm{REF}}}\right) \tag{2.4}$$

For digital microphones, sensitivity is often expressed in dBFS (decibels relative to full-scale digital output). This refers to the ratio between the output value and the maximum digital value. Sensitivity is determined by the difference between the maximum acoustic input signal and the 94 dB SPL reference. This relationship is defined by eq. (2.5), where the reference output is the full-scale digital output level (Lewis, 2013b)(InvenSense, 2016).

$$Sensitivity_{\mathrm{dBFS}} = 20\log\left(\frac{Sensitivity_{\mathrm{\%FS}}}{Output_{\mathrm{REF}}}\right) \tag{2.5}$$

While higher sensitivity can result in a stronger output, it does not always mean the microphone is better. A higher sensitivity can reduce the dynamic range and cause distortion at high SPL levels. The dynamic range of a microphone is determined by the difference between the lowest detectable signal, better known as noise floor and the highest input signal that can be processed without distortion (Lewis, 2013b).

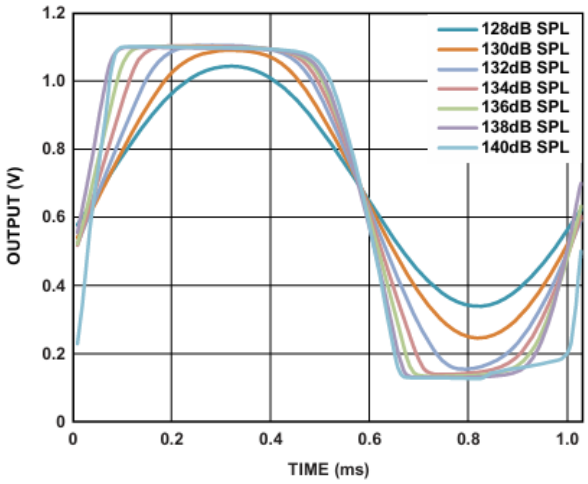


Figure 2.4: Microphone clipping in a time-domain plot (InvenSense, 2016).

2.3.5.2 Acoustic overload point

The acoustic overload point is the sound pressure level at which the total harmonic distortion (THD) of the microphone's output reaches 10%. This point is also referred to as the clipping point of the microphone. When the sound pressure level exceeds this level, it causes severe nonlinear distortion in the microphone's output signal, meaning the microphone can no longer accurately capture incoming sound without distortion. This reduces the measured pressure as the gain of the microphone is reduced for large signals. The acoustic overload point can be determined by examining the total harmonic distortion plot and identifying the SPL level at which the THD reaches 10%. Some datasheets also present the microphone's clipping characteristics as a time-domain plot, as shown fig. 2.4, to illustrate how the output signal behaves at increasing sound pressure levels (InvenSense, 2016).

2.3.6 Overview of selected microphones for testing

To ensure an informed selection of a suitable MEMS microphone, three different models have been considered for evaluation. The different models are:

1. ICS-40300: The ICS-40300 is shown in fig. 2.5, is designed for applications requiring extended low-frequency response and high SPL handling. It features a single-ended analog output and a low noise floor, making it well suited for acousting sensing applications. The high SPL tolerance makes it ideal for live sound recording where accurate audio capture of loud sources is necessary (TDK InvenSense, 2018) .

2. ICS-40638: The ICS-40638 is a high dynamic range microphone capable of handling extreme sound pressure levels while maintaining low distortion. It has a high power supply rejection ratio of $-81\,\mathrm{dB}$, ensuring immunity to power supply noise. The microphone also has a tight sensitivity tolerance, making it well-suited for applications requiring high consistency across multiple units (TDK InvenSense, 2019a).

3. ICS-40720: The ICS-40720, illustrated in fig. 2.5, is an ultra-low noise MEMS microphone that provides a differential analog output and is optimized for high SNR. Its $70\,\mathrm{dB}$ SNR and linear response make it ideal for microphone arrays. The ICS-40720 offers a high dynamic range and low equivalent input noise. Its differential output also helps to reject common-mode noise, making it particularly useful in noisy environments (TDK InvenSense, 2019b).



Figure 2.5: ICS-40300 (TDK InvenSense, 2018) & ICS-40720 (TDK InvenSense, 2019b).

2.3.7 MEMS Microphones in prior experiments

2.3.7.1 Low-Cost sensors for urban noise monitoring etworks by Picaut, J.

In this study, MEMS microphones were employed as a crucial component of a low-cost system designed for urban noise visualization. The MEMS microphones were connected to a digital recorder, which provided a compact and portable solution for capturing sound data. Their role was to measure the sound pressure levels of various noise sources in urban environments, enabling the visualization of sound fields.

The measurements involved placing two microphones, one reference and one scanning, near the noise sources. Calibration was performed using external sound calibrators to ensure accurate data capture. The collected data were recorded as digital files, processed further on a desktop computer and analyzed using advanced acoustic visualization techniques such as planar near-field acoustical holography.

The results demonstrated that MEMS microphones, despite their low cost, can provide reliable measurements over a wide frequency range. They accurately captured sound pressure and particle velocity, allowing for the detailed visualization of noise sources like an enclosed loudspeaker and a small air compressor. These findings validated the system's capability to identify noise characteristics effectively, showcasing MEMS microphones as a viable tool for urban noise monitoring and sound visualization applications (Picaut, Can, Fortin, Ardouin, and Lagrange, 2020).

2.3.7.2 The design and calibration of low cost urban acoustic sensing devices

In this study, MEMS microphones were integrated into a low-cost sensor network designed for long-term urban noise monitoring. These microphones were central to the system, offering a compact, cost-effective and reliable solution for capturing sound data in complex urban environments. MEMS microphones were chosen for their consistent performance, resistance to environmental factors such as temperature and humidity and suitability for large-scale deployment.

The sensors equipped with MEMS microphones were deployed across various urban locations in New York City to capture sound pressure levels over extended periods. Calibration procedures were employed to ensure the accuracy of the microphones. The collected sound data were processed locally by the sensors using embedded computing systems.

The results highlighted the capability of MEMS microphones to deliver reliable sound pressure level measurements that adhered to type 2 specifications for sound level meters. The system successfully identified urban noise trends and patterns, demonstrating its utility for creating dynamic sound maps and informing urban noise management policies. Despite a limitation in sensitivity at low noise levels, the MEMS microphones performed effectively in the typically

high-noise urban environment. This study established MEMS microphones as a promising component for scalable, affordable and efficient urban noise monitoring systems (Mydlarz, 2015).

2.4 Data Acquisition System

Data acquisition systems (DAQ) are devices designed to collect information from the surrounding environment in digital form. The process of acquiring data provides insights into the specific conditions being studied. Typically, DAQ devices interface with a variety of sensors that measure different physical phenomena. Most systems capture signals from transducers that generate analog outputs, which require further processing. To enable digital processing, these analog signals are converted into digital form by means of an analog-to-digital converter (ADC) (Abdallah, 2009).

2.4.1 Data Acquisition systems Classification

Data Acquisiton systems can be classified into three main categories based on their architecture and flexibility: computer-based DAQ systems, embedded microcontroller-based systems and FPGA-based systems. Each of these approaches has distinct advantages and limitations, making them suitable for different applications (Abdallah, 2009).

2.4.1.1 Computer-based DAQ

Computer-based DAQ systems rely on the processing power of a computer to handle data manipulation, visualization, storage and decision-making. These systems can either be internal, where DAQ modules are installed as expansion cards in a computer, or external, where standalone DAQ devices connect via ports like USB, serial or parallel interfaces. While computer-based DAQs offer strong computation capabilities, they are often associated with high costs, large physical size and significant power consumption (Abdallah, 2009).

2.4.1.2 Embedded microcontroller-based DAQ

Embedded microcontroller-based DAQ systems offer a more compact and energy-efficient alternative. These systems are widely used in fields such as healthcare, the automotive industry and environmental monitoring. Their key advantages lie in their low cost, small form factor and reduced power consumption. However, they operate with a fixed hardware architecture, which limits their adaptability. Once designed and implemented, modifications or upgrades are difficult without replacing the entire system (Abdallah, 2009).

2.4.1.3 FPGA-based DAQ

A more flexible and reconfigurable alternative is found in FPGA-based DAQ systems. Unlike fixed-architecture microcontroller systems, Field-Programmable Gate Arrays (FPGAs) allow reprogramming in the field, enabling system upgrades, fault tolerance and optimized performance without the need for hardware replacement. This makes FPGA-based DAQs particularly advantageous for high-speed, multi-channel data acquisition while maintaining a compact design and efficient power usage.

2.4.2 Projects where DAQs are being used

2.4.2.1 Detection of damage on inner and outer races of ball bearings using a low-cost monitoring system and deep convolutional neural networks

The goal of this project was to develop a low-cost monitoring system for detecting damage to ball bearings. Vibration data is collected using a set of affordable sensors, such as a solid-state accelerometer, microphone and piezo sensor, together with an MCC128 DAQ HAT on a Raspberry Pi 4B. A Deep Neutral Network (DNN) is then trained to detect damage to the inner and outer race of bearings based on time-frequency representations such as spectrograms and scalograms (You et al., 2024).

The MCC128 DAQ HAT is used as the main data acquisition system within the low-cost solution. The results show that the MCC128 DAQ HAT with a Raspberry Pi 4B can collect signals up to 20 kHz and that its performance is comparable to high-end commercial DAQ systems, but at only 3% of the cost (You et al., 2024).

The MCC128 is a 16-bit high-speed voltage measurement DAQ HAT (Hardware Attached on Top) designed for use with the Raspberry Pi. The MCC128 provides 8 single-ended (SE) or 4 differential (DIFF) analog input channels, making it a versatile solution for voltage measurements in embedded and industrial applications (You et al., 2024) (Corporation, 2020).

The MCC128 offers a high sampling rate up to 100 kS/s per board, with the ability to stack up to 8 MCC128 HATs, as shown in fig. 2.6, achieving an aggregate throughput of 320 kS/s. This stacking capability allows for scalable data acquisition, making it suitable for multi-channel sensing applications. The device supports multiple input voltage ranges (±10 V, ±5 V, ±2 V, ±1 V), providing flexibility for various measurement needs. For enhanced synchronization and control, the MCC128 includes an external scan clock and a digital trigger input, allowing precise timing of data acquisition. The external clock feature enables users to synchronize multiple board, making it suitable for complex monitoring systems (Corporation, 2020).



Figure 2.6: Stacked up MCC128 HATs on a Raspberry Pi. (Corporation, 2020)

The MCC128 interfaces directly with the 40-pin GPIO of the Raspberry Pi and is compatible with multiple Raspberry Pi models. It is powered via the 5V supply from the Raspberry Pi, eliminating the need for an additional power source. Configuration settings are stored in an onboard EEPROM, which allows for automatic GPIO setup when the HAT is connected. Software support for the MCC128 is provided through the MCC DAQ HAT Library, an open-source library available on GitHub, with APIs in C, C++ and Python. This library facilitates easy integration into various applications, including real-time monitoring and data logging. Example programs include console-based applications, web server integration and IFTTT (If This Then That) automation for cloud-based monitoring (Corporation, 2020).

2.5 Data transfer from RPI to a cloud

To retrieve and analyze the given data from the microphone, this data has to be uploaded from the Raspberry Pi (RPI) onto a cloudserver so it can be reachable from anywhere. Grindstaff, 2019 and Ginzburg, 2023 provide a program that is easy to configure on a RPI called Rclone.

Rclone is a third-party, open-source application that can be configured to connect to several popular cloud storage services such as Google Drive, Box, Onedrive and Dropbox (Grindstaff, 2019).

2.5.1 Setting up Rclone on a Raspberry Pi

To configure Rclone on a Raspberry PI, both Ginzburg, 2023 and Grindstaff, 2019 provide a step-by-step plan as following:

1. Open a terminal window on the Raspberry Pi either by clicking on the terminal icon at the bottom of the screen or by typing Ctrl+Alt+T.
2. Run the following command in the opened terminal window to download and install the latest, stable release of rclone: *wget https://downloads.rclone.org/rclone-current-linux-arm.zip*
3. After successful installation, run the following configuration command: *rclone config.*
4. A list of options will be given, Select 'new remote' by entering "n".
5. Provide a name for the newly created remote sync location.
6. A list of file sharing services will then be given (e.g, Google Drive, Box, Dropbox, OneDrive).
7. Select the desired level of access given to the selected cloud account. Full access is recommended.
8. A list host of options will be given. Use autoconfig to accept the default options without any other inputs.
9. The last input will make the program open a browser and requires the user to login to the chosen cloud service.
10. Finally, the program will give an overview of the selected choices. Be sure to look for any errors.
11. Exit the Rclone configuration.

2.6 Signal processing

2.6.1 Spectral analysis: the Fourier transform

The Fourier transform is a mathematical tool that decomposes a continuous-time signal into its frequency components. This transformation is fundamental in signal processing, communication and physics, as it allows the analysis of signals in the frequency domain rather than in the time domain (Rolain, 2023) (Bracewell, 1989).

2.6.1.1 Mathematical definition and interpretation

For a given continuous-time function u(t), the Fourier transform U(f) is defined by eq. (2.6). Here $U(f)$ represents the spectrum of the signal in the frequency domain.

$$U(f) = \int_{-\infty}^{+\infty} u(t) e^{-j2\pi ft} dt \tag{2.6}$$

The inverse Fourier transform, as given in eq. (2.7), allows reconstruction of the original signal.

$$u(t) = \int_{-\infty}^{+\infty} U(f) e^{j2\pi ft} df \tag{2.7}$$

This signal $u(t)$ is decomposed into a sum of weighted sinusoidal functions at different frequencies. Figure 2.7 provides examples of the resulting Fourier transform (Rolain, 2023).

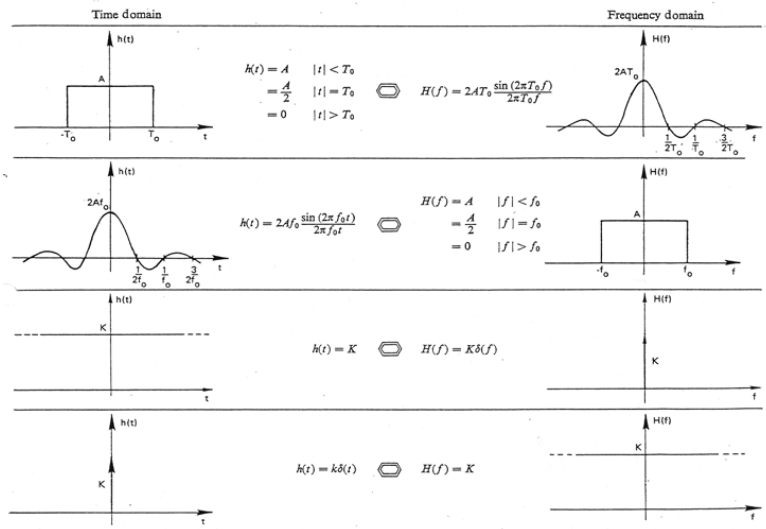


Figure 2.7: Examples of the Fourier transform of certain functions (Rolain, 2023).

2.6.1.2 Spectrum representation

The spectrum of a signal consists of two key components: the amplitude spectrum $A(f)$, which defines the magnitude of the frequency components and the phase spectrum $\phi(f)$, which describes the phase of shift of each frequency component. For example, for a damped exponential function e^{-at}, the amplitude spectrum is shown in eq. (2.8) and the phase spectrum is illustrated in eq. (2.9), which can be graphically represented on a logarithmic frequency axis (Rolain, 2023) (Bracewell, 1989).

$$A(f) = \frac{1}{\sqrt{a^2 + (2\pi f)^2}} \tag{2.8}$$

$$\phi(f) = -\tan^{-1}\left(\frac{2\pi f}{a}\right) \tag{2.9}$$

To construct the spectrum of a given function, one can follow these steps (Rolain, 2023):

1. Compute the Fourier Integral: Apply the definition of the Fourier transform to derive $U(f)$.
2. Analyze properties: Determine amplitude and phase spectra.
3. Apply Transform theorems like linearity, time-shifting and scaling to simplify calculations.

For periodic signals, the Fourier transform does not strictly exist in the traditional sense. Instead, they can be analyzed using Fourier series, which result in a discrete line spectrum consisting of Dirac delta functions at integer multiples of the fundamental frequency (Rolain, 2023)(Bracewell, 1989).

2.6.1.3 Application of Fourier transform in microphone arrays

The article *"Some Analysis of Microphone Arrays for Speech Data Acquisition"* by Silverman (1987) applies Fourier analysis to the design and optimization of microphone arrays. The author uses the Fourier transform to decompose speech signals into frequency components and analyze how different directions affect signal reception.

Specifically, the exponential form of the Fourier series is used to model the transfer function of the microphone array. This allows for the characterization of the array's frequency response and the optimization of how signals from different directions are either enhanced or suppressed. Fourier analysis is also employed in the implementation of digital signal processing techniques, such as interpolation filters and frequency-domain compensation.

This work demonstrates that the Fourier transform is not just a theoretical tool but also a practical method for developing acoustic sensor arrays and noise reduction systems. In real-world applications, such as speech recognition and teleconferencing, Fourier-based frequency analysis enables the design of microphone arrays that can focus on desired speech sources while minimizing interference from unwanted noise.

2.6.2 Sampling

Sampling is the process of converting a continuous-time signal into a discrete-time signal by measuring its amplitude at specific time intervals. This transformation is performed by a sample-and-hold circuit, which captures and maintains the instantaneous value of the signal at each sampling instant. In most practical applications, sampling occurs at regular intervals dictated by a clock signal, ensuring a uniform time separation between successive samples. The sampling process plays a crucial role in digital signal processing, as it forms the basis for representing and analyzing real-world continuous signals in a digital format (Rolain, 2023).

In signal processing, the sampling period (T_s) and the sampling frequency (F_s) are fundamental parameters that determine how a continuous-time signal is converted into a discrete-time representation. The sampling period represents the time interval between consecutive samples, while the sampling frequency is its reciprocal:

$$F_s = \frac{1}{T_s} \tag{2.10}$$

A higher sampling frequency results in a smaller sampling period, allowing for the representation of high-frequency components up to $F_s/2$.

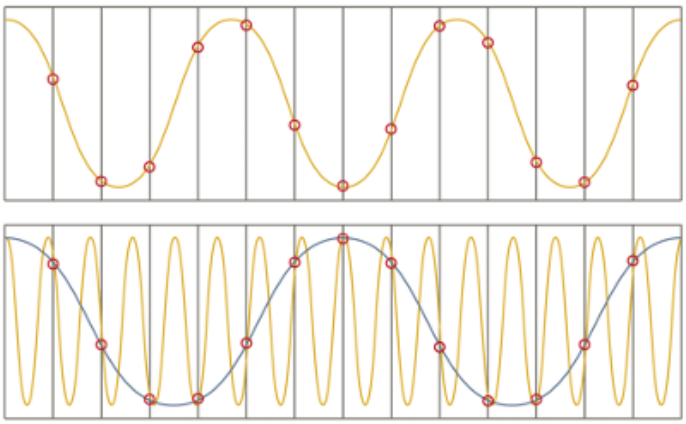


Figure 2.8: The oversampled signal and undersampled signal (Por, 2019).

2.6.3 Possible problems

2.6.3.1 Aliasing

Aliasing is a distortion effect that occurs when a continuous-time signal is sampled at a rate that is insufficient to capture its frequency content accurately. Specifically, it arises due to the overlap of periodic repetitions of the signal spectrum in the frequency domain. When the sampling frequency is too low, different frequency components become indistinguishable, leading to an incorrect representation of the original signal. This effect makes it impossible to accurately reconstruct the original signal from its sampled version, potentially introducing misleading frequency components into the digital representation (Rolain, 2023) (Dmochowski, 2009).

Por (2019) provides a good visual example of aliasing as shown in fig. 2.8. It shows that aliasing occurs when a signal is undersampled. The sampling of two different signals is shown. Each of the signals are sampled at the frequency. It is clearly visible that the top signal is oversampled because its frequency is lower than half the sampling frequency. Therefore there are more than two samples per period of this sine wave signal. The original signal can be totally reconstructed. On the other hand, the bottom sine wave signal is undersampled. There are less than two samples per period of this sine wave and when reconstruction is attempted, illustrated with a blue line, the original signal cannot be recovered. It reconstructs another signal at a much lower frequency. This is the effect called aliasing.

2.6.3.2 Preventing aliasing: Shannon-Nyquist theorem

The Shannon-Nyquist sampling theorem establishes a fundamental criterion for ensuring that a sampled signal can be perfectly reconstructed without losing any information. According to the theorem, a bandlimited signal X(t) with a maximum frequency f_{max} can be perfectly reconstructed if it is sampled at a rate that is at least twice the highest frequency present in the signal (Por, 2019) (Rolain, 2023).

Mathematically, this is presented in eq. (2.11).

$$f_{\text{sample}} \geq 2f_{\text{max}} \tag{2.11}$$

where f_{sample} is the sampling frequency and f_{max} is the maximum frequency component of the original signal. If this is the case, there has been no loss of any information during the sampling

process and in theory, the original signal can be reconstructed from the sampled signal. The Nyquist frequency is defined in eq. (2.12).

$$f_{\text{Nyquist}} = \frac{f_{\text{sample}}}{2} \tag{2.12}$$

A signal that contains spectral components at frequencies exceeding the Nyquist frequency cannot be accurately reconstructed from its sampled version. Such signals are referred to as undersampled. Conversely, if a signal contains only frequency components below the Nyquist frequency, the original signal can be perfectly reconstructed from the sampled data. These signals are referred to as oversampled. When a signal is bandlimited to a frequency equal to the Nyquist frequency, it is said to be critically sampled (Por, 2019).

2.7 4G CONNECTIVITY

2.7.1 4G HAT LTE

Moyano (2022) explores the implementation of an autonomous solar energy system utilizing a Raspberry Pi for 4G communication. The study focuses on integrating the SIM7600E-H 4G (fig. 2.10), a module enabling wireless connectivity for data transmission over mobile networks. This allows the system to process HDR images and upload them remotely to a database.

Moyano (2022) provides detailed power consumption measurements for the 4G module when active. Using a USB tester (UM25C), voltage and current variations were recorded during both idle and transmission states. Under full load conditions, the Raspberry Pi and 4G module consume an average of 1.3 A at 5.2 V, resulting in a power usage of approximately 6.7 W. Voltage fluctuations were observed, especially during 4G data transmission, where voltage peaks occur due to the modem's processing requirements. When the process is idle, the voltage measurements are more or less constant, as shown in fig. 2.9. The only appreciable voltage peak here, corresponds to the 4G connection through the modem.

To ensure a stable power supply, the system integrates a bq24074 charge controller and a PowerBoost 1000 Charger, which transforms the 3.7 V LiPo battery output to 5 V, the required voltage for both the Raspberry Pi and the 4G module. The voltage stability is crucial for continuous operation, as drops below 4.5 V can cause connectivity loss with the SIM7600E-H module

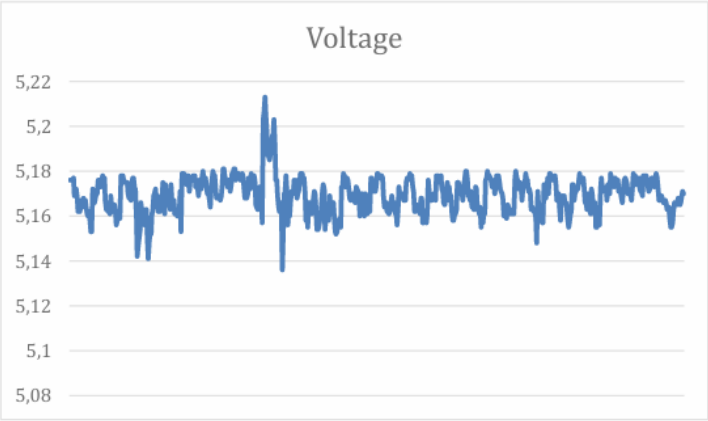


Figure 2.9: Voltage when the process is in idle state (Moyano, 2022).

2.7.2 4G dongle

A dongle is a tiny piece of computer hardware that plugs into a port on another device to provide functionality or enable a pass-through to such a device. This dongle connects the system to the internet at high rates of speed and assigns it a unique IP address (Reddy et al., 2022).

The use of a 4G dongle, illustrated in fig. 2.10, as an alternative internet connection offers several key advantages, particularly in environments where network access is restricted. Mamat and Basiran (2022) highlight that a 4G dongle is a cost-effective, easy-to-install and easy-to-configure solution that provides a reliable internet experience comparable to traditional wired broadband. Unlike fixed broadband connections, a 4G dongle requires no complex infrastructure changes, making it a highly portable and flexible option for users who need independent internet access. Another significant advantage is the ability to establish a redundant connection, ensuring continuous internet access even when a primary Local Area Network (LAN) is slow or restricted.



Figure 2.10: Illustrations of an 4G USB Dongle (IMILINK, 2020) and the Waveshare SIM7600G-H 4G Hat (Waveshare, 2025).

2.8 Ensuring autonomous operation

For the measurement setup to be effectively deployed in diverse environments, it must operate independently of the local power grid. Relying on access to conventional electricity is not feasible in many field conditions, as the system may need to function in remote areas where no power source is available. To ensure continuous and autonomous operation, an alternative and sustainable energy supply is therefore required. In the following section, solar panels are introduced as a suitable solution to provide the necessary power for the system.

2.8.1 Powering the system with a solar cell

A photovoltaic cell, also referred to as a solar cell, is a semiconductor device, typically made of silicon, that transforms light energy from both natural sunlight and artificial sources into electrical power. When multiple solar cells are connected, they form a solar module, also known as a solar panel (Ramrakhiani, 2017). Since a single solar panel generates only a limited amount of energy, multiple panels are connected to form a solar array, allowing for greater energy production (Vastav et al., 2016). Through the photovoltaic effect, each solar cell generates electricity (DC), which contributes to the overall power output of the panel (Ramrakhiani et al., 2017).

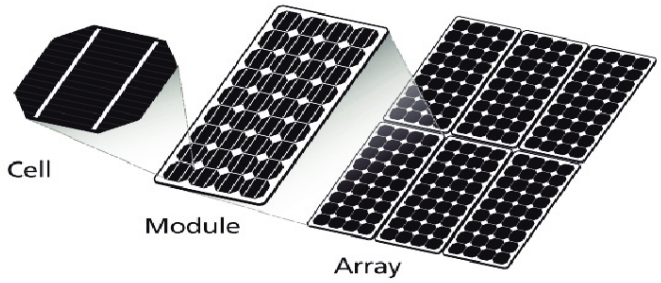


Figure 2.11: Solar cell, solar module and solar array (Vastav et al., 2016).

2.8.2 The photovoltaic effect

2.8.2.1 Discovery and meaning of the photovoltaic effect

The principle of the photovoltaic effect was first revealed in 1839 by Edmond Becquerel, a physicist from France. He found that illuminating specific substances can lead to the generation of electrical current (Boxwell, 2017). The term 'photovoltaic' is derived from a combination of two terms. The first part 'photo' comes from the Greek word for light, while the second part 'voltaic', is derived from the name of Alessandro Volta, an Italian scientist known for his groundbreaking contributions to electricity (Dixit et al., 2023).

2.8.2.2 Structure of a silicon crystal

A solar cell is made of crystalline silicon, which consists of atoms. Atoms are formed of positively charged protons, negatively charged electrons and neutral neutrons. The nucleus of an atom contains both protons and neutrons, while the electrons move in orbits around the nucleus. An atom is electrically neutral because it contains an equal number of protons and electrons.

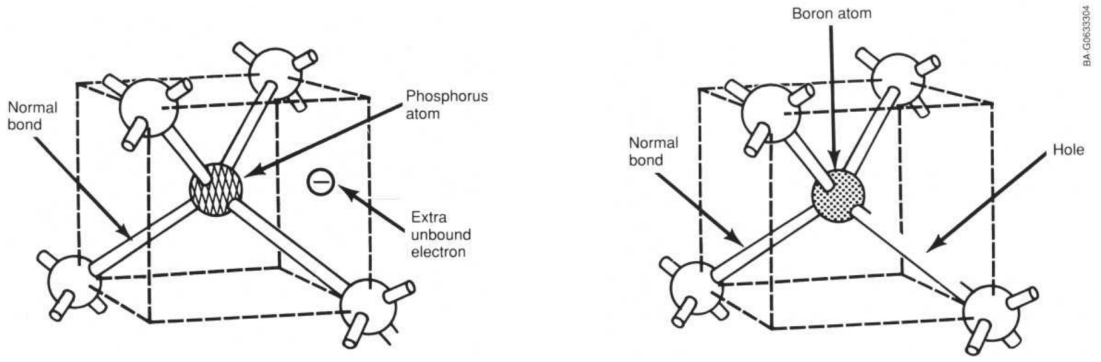
A silicon atom has 14 electrons that move around its nucleus. Each silicon atom has four valence electrons, which can be donated, accepted or shared with other atoms. This allows silicon atoms to bond with one another, forming a crystal structure. A crystal structure consists of multiple silicon atoms, where each silicon atom is bonded to four neighboring silicon atoms through a covalent bond (Cook et al., 1991).

2.8.2.3 Photon absorption and energy conversion

A solar cell generates electricity by utilizing light energy, which consists of photons, which are small packets of energy. The amount of energy a photon carries depends on the wavelength of the light. When sunlight strikes a solar cell, it can be reflected, absorbed or passed through (Solanki, 2015). If a photon is absorbed by the silicon, its energy is transferred to valence electrons, allowing them to move to a higher energy level, increasing their ability to move freely through the crystal. When an electron is released from its connection to the atom, it leaves behind an empty space called a hole, which behaves as a positive charge. Electrons from neighboring atoms can fill these holes, creating new holes in the process. As this continues, the holes appear to move through the material (Ramrakhiani, 2017) (Cook et al., 1991).

2.8.2.4 Formation and role of the electric field

Electricity is produced through the electric field that forms when two different types of semiconductors come into contact. This field directs positive and negative charges to move in opposite directions, creating the foundation for generating electrical power.



(a) Phosphorus atom in silicon introduces an extra free electron (Cook et al., 1991).

(b) Boron atom in silicon creates a hole due to the absence of an electron (Cook et al., 1991).

Figure 2.12: Illustration of donor and acceptor doping in a silicon semiconductor.

To improve the conductivity of silicon, additional atoms are introduced into its crystal structure through a process called doping. There are two types of doping: n-type and p-type (Cook et al., 1991). N-type doping involves adding phosphorus atoms to the silicon crystal, as illustrated in fig. 2.12a . Since silicon has four valence electrons and phosphorus has five, one electron remains unbound after bonding, allowing it to move freely (Zeman, 2003). In addition, there is p-type doping, which involves introducing boron atoms into the silicon crystal. Unlike silicon, which has four valence electrons, boron has only three. As a result, boron cannot fully bond with the surrounding silicon atoms, creating a hole, as demonstrated in fig. 2.12b (Cook et al., 1991).

When n-type and p-type silicon come into contact, the electrons from the n-type region near the junction move toward the p-type side, leaving behind a positively charged layer at the n-side of the junction. Simultaneously, the holes shift in the opposite direction, creating a negatively charged layer on the other side. This movement of charges creates an electric field at the junction (Cook et al., 1991), forming what is known as the depletion region (Tiwari and Dubey, 2009). The strength of the resulting electric field depends on the concentration of dopant atoms introduced into the silicon.

As discussed in section 2.8.2.3, when a solar cell is exposed to a photon, it excites an electron, creating an electron-hole pair. Under normal conditions, these electrons and holes would quickly recombine, preventing any electricity from being generated. However, due to the electric field at the p-n junction, electrons are driven toward the n-side, while holes move toward the p-side.

If an external electrical circuit is connected to the p-n junction, the separated electrons from the n-side will flow through a load and return to the p-side via the external circuit. As the electrons

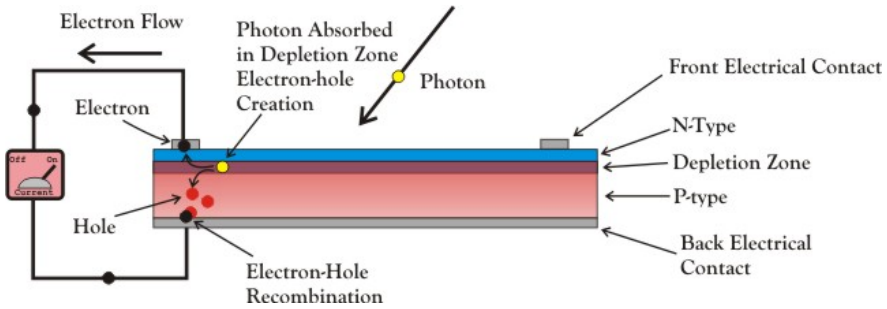


Figure 2.13: Working principle of a solar cell: A photon is absorbed, creating an electron-hole pair. The built-in electric field drives the charges apart, generating an electric current (Ramrakhiani, 2017).

reach the p-side, they will recombine with the holes, completing the cycle and generating an electric current, as illustrated in fig. 2.13. This process is the fundamental mechanism by which a solar cell converts light into usable electrical energy (Cook et al., 1991).

2.8.3 Maximizing light absorption in silicon solar cells

When silicon is untreated it can behave like a mirror, reflecting up to 30% of incoming light. This reflection reduces the efficiency of a solar cell, as less light is absorbed and converted into electricity. To minimize this effect, the top layer of the solar cell can be coated with a thin layer of silicon monoxide. This coating can reduce light reflection by approximately 10%, allowing more light to enter the cell.

Another effective method to enhance light absorption is texturing the surface of the solar cell. This process creates a rough surface by chemically etching the silicon, forming tiny cone- or pyramid shaped structures. When light strikes these textured surfaces, it does not reflect directly away from the cell. Instead, the cone or pyramid structures guide the light toward another surface within the solar cell, giving it a second chance to be absorbed. This significantly increases light absorption, leading to a more efficient solar cell (Cook et al., 1991).

2.9 Batteries

Photovoltaic solar panels generate electricity during daylight hours. However, energy production significantly decreases as natural light decreases. To optimize the use of solar energy and ensure a continuous power supply, storing excess electricity is essential. Battery storage systems offer an effective solution by storing the excess energy produced during the day for later use, such as during nighttime or periods of low sunlight. Various battery technologies are available, each with unique characteristics in terms of efficiency, lifespan and energy storage capacity.

2.9.1 Operating principle of a battery

A battery consists of one or more electrochemical cells. Each cell is composed of a positive electrode called the cathode and a negative electrode known as the anode. Between the cathode and the anode is the electrolyte, a medium that allows ion transport between the electrodes while maintaining electrical isolation (Horlings, 2024) (Wessling et al., 2025). The components of a battery are shown in fig. 2.14.

When the battery needs to supply energy to a device, the active material in the cathode undergoes oxidation, meaning that electrons are removed from the cathode and transferred to the

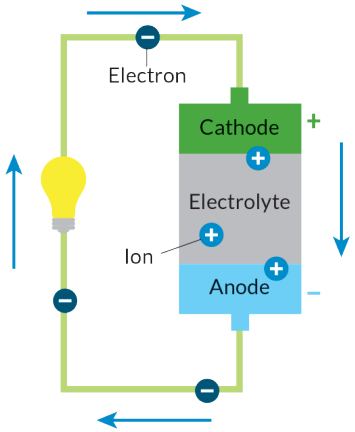


Figure 2.14: Schematic Representation of a battery (Science News, 2025).

anode. These electrons do not travel directly through the electrolyte but instead flow through an external circuit, generating an electric current that powers the connected device. During discharge, the electrons move back to the positive electrode, where they trigger a reduction reaction with the active material (Wessling et al., 2025).

2.9.2 The design of a battery

Batteries, as discussed in section 2.9.1, consist of one or more cells. These cells vary in shape and size based on their specific purpose. The most common types are cylindrical, prismatic and pouch cells (Horlings, 2024). An overview of these cell types is provided in fig. 2.15.

2.9.2.1 The cylindrical shape

The cylindrical cell is composed of four layers of material, which are rolled into a cylindrical shape. These layers include the cathode, anode and two separator layers (Rahn and Wang, 2013). The separators act as barriers, preventing the anode and cathode from touching each other, which could otherwise cause a short circuit. The cathode is connected to the battery's top terminal, the positive pole, while the anode is connected to the bottom terminal, the negative pole (Horlings, 2024).

We connect the cells in a series configuration to generate a higher voltage. Additionally, the current capacity can be increased by expanding the electrode surface area, which can be achieved by increasing the cell's diameter or length.

One of the main advantage of the cylindrical shape is that it allows for efficient cooling. However, a drawback is that this shape reduces space efficiency in battery modules, as empty spaces are formed between the cells when they are packed together (Horlings, 2024).

2.9.2.2 The prismatic shape

A prismatic cell has an aluminum alloy casing and an internal structure composed of rolled or stacked layers, including the cathode, anode and separators (Chen, 2022). Unlike cylindrical cells, which leave empty spaces when packed together, prismatic cells maximize space efficiency due to their rectangular design, making them easier to arrange in battery modules. One advantage of this improved space efficiency is that prismatic cells can achieve a higher energy density. However, a drawback is that they are more challenging to cool and are more expensive to manufacture (Horlings, 2024).

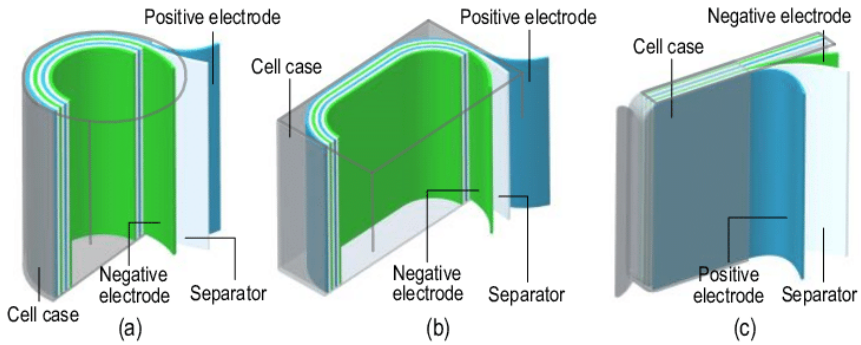


Figure 2.15: Schematic representation of different battery configurations. (a) cylindrical cell, (b) prismatic cell and (c) pouch cell (Bilgin et al., 2015).

2.9.2.3 The pouch cell

The pouch cell features an internal layered structure similar to other battery types, but its outer casing is made of aluminum-plastic film, as illustrated in fig. 2.15 (Chen, 2022). This design reduces weight and allows for greater flexibility in shaping and customizing the cell (Horlings, 2024). Pouch cells are approximately 20% lighter than prismatic cells of the same capacity and can store 50% more energy than prismatic cells of the same volume.

One of the main benefits of pouch cells is that their outer aluminum-plastic film expands or ruptures when a safety hazard occurs, reducing the risk of explosions and improving the safety of the battery. However, this flexible casing offers lower mechanical strength, making pouch cells structurally weaker than cylindrical and prismatic cells (Chen, 2022). Despite this drawback, they remain the most cost-effective option (Horlings, 2024).

2.9.3 Different kinds of batteries

Having established an understanding of battery cell design, it is essential to explore the various types of batteries that utilize these structural configurations. As battery technology continues to evolve, different battery types are designed to meet specific performance criteria, such as charge-discharge rates, cycle life and power output. The choice of battery chemistry significantly impacts its energy density, efficiency and environmental sustainability, making certain types more suitable for specific applications than others.

The most commonly used battery types include lead-acid, nickel-cadmium (NiCd), nickel-metal hydride (NiMh), lithium-ion (Li-ion), lithium polymer (LIPO) batteries. Each of these technologies has distinct properties that define their efficiency and practical use (Horlings, 2024) (Soloveichik, 2011).

2.9.3.1 Lead-acid batteries

Lead-acid batteries were invented in 1859 by Gaston Planté. They operate through an electrochemical reaction between lead (Pb) and lead dioxide (PbO_2) in a sulfuric acid (H_2SO_4) electrolyte, generating a voltage of approximately 2.04 V per cell (Soloveichik, 2011). These batteries are primarily used in automotive applications, emergency power supplies (UPS), solar energy storage and industrial applications.

One of the key advantages of lead-acid batteries is their reliability, as they are resistant to overcharging and have a low self-discharge rate. However, their biggest benefit is their cost-effectiveness, making them a practical choice for a wide range of applications (Horlings, 2024).

Despite these advantages, lead-acid batteries have significant limitations. A major drawback is that large lead electrodes are not mechanically strong enough and are unable to sustain high current loads efficiently, making them unsuitable for large-scale energy storage applications (Soloveichik, 2011). Additionally, their excessive weight and relatively low energy density, typically ranging from 30 to 50 Wh/kg (Marin-Garcia et al., 2020), limit their efficiency compared to modern battery technologies. Furthermore, the inclusion of lead, a toxic material, poses significant risks to both human health and the environment (Horlings, 2024).

2.9.3.2 The nickel-cadmium battery (NiCd)

The nickel-cadmium battery, invented in 1899 by Waldemar Jungner, was developed as a replacement for the lead-acid battery. It relies on an electrochemical reaction where nickel hydroxide ($Ni(OH)_2$) serves as the positive electrode and cadmium (Cd) as the negative electrode. Compared to lead-acid batteries, nickel-cadmium batteries offer a longer lifespan, better resistance to low temperatures and improved charge-discharge efficiency with minimal capacity loss

(Horlings, 2024). NiCd-batteries are available in both cylindrical forms, such as AA-batteries and prismatic forms (Keizer et al., 2008).

However, a major drawback of NiCd-batteries is the memory effect, where repeated charging before a full discharge, gradually reduces the battery's capacity, leading to a shorter operational life (Horlings, 2024). Furthermore, they produce a lower voltage of approximately 1.29 V per cell (Soloveichik, 2011) (Keizer et al., 2008), although their energy density is approximately 50 to 75 Wh/kg, which exceeds that of lead-acid batteries (Marin-Garcia et al., 2020). Additionally, similar to lead-acid batteries, NiCd-batteries contain a toxic heavy metal, cadmium, which poses significant environmental hazards (Horlings, 2024).

2.9.3.3 The nickel metal-hybrid battery (NiMH)

The nickel metal-hydride battery was introduced to the market in 1989 as an improved version of the nickel-cadmium battery. By replacing the cadmium-based negative electrode with a hydrogen-absorbing alloy, the NiMH-battery became a more environmentally friendly alternative while also offering better resistance to high temperatures and a higher energy density of approximately 80 Wh/kg (Horlings, 2024) (Keizer et al., 2008). Additionally, they produce a higher voltage than NiCd-batteries, approximately 1.35 V per cell (Soloveichik, 2011). However, their performance under high current loads is lower compared to NiCd-batteries (Keizer et al., 2008).

The most significant advancement, was the elimination of the memory effect, allowing the battery to be recharged at any time without capacity loss, making it a more efficient and user-friendly energy storage solution. NiMH-batteries continue to be extensively utilized today, especially in the form of AA- and AAA-batteries (Horlings, 2024)(Soloveichik, 2011).

2.9.3.4 The lithium-ion battery (Li-ion)

A lithium-ion battery consists of a positive electrode made of lithium metal oxides and a negative electrode composed of graphite carbon. These batteries can generate a high voltage per cell, typically ranging from 3.3 to 4.2 V and have a high energy density (Soloveichik, 2011), ranging between 75 and 125 Wh/kg (Marin-Garcia et al., 2020). This means they can store a large amount of energy in a small and lightweight package (Soloveichik, 2011).

One key advantage of lithium-ion batteries is their low self-discharge rate, which ensures they retain most of their charge even when not in use (Horlings, 2024). Additionally, they offer fast charging times and excellent performance at high discharge rates, making them ideal for high-power applications. However, lithium-ion batteries also come with some drawbacks. They are more expensive compared to other battery types and their performance declines in low temperatures. Moreover, they require additional safety measures to prevent overheating and potential explosions (Soloveichik, 2011).

To maximize the lifespan and maintain optimal performance, it is essential to avoid full charging and complete discharging of a lithium-ion battery. Maintaining the charge level between 20% and 80% reduces stress on the cell, extending its overall lifespan (Horlings, 2024).

2.9.3.5 Lithium polymer battery (LiPO)

Another common type of rechargeable lithium-based battery is the lithium-polymer (LiPO) battery. Similar to lithium-ion batteries, it operates using a positive and a negative electrode. However, instead of a liquid electrolyte, the battery contains a solid, porous, polymer-based electrolyte (Karlina et al., 2021). This solid electrolyte allows for more efficient use of internal space within the battery cell, making it possible to design thinner and more compact batteries, as solid electrolytes generally occupy less volume than liquid ones (Marin-Garcia et al., 2020).

The nominal voltage of a typical LiPO cell is approximately 3.7 V, although this value can vary slightly depending on the specific chemical composition used within the cell (Dileepan et al., 2017).

In addition to its structural advantages, the LiPO battery offers a high power density, typically ranging between 130 and 200 Wh/kg, exceeding that of conventional lithium-ion batteries. This performance benefit, combined with their reduced weight and smaller size, makes LiPO batteries especially suitable for applications where space and weight are limited. Moreover, they do not suffer from the memory effect, meaning they do not require complete discharge before recharging and have a low self-discharge rate, allowing them to retain charge over extended periods of inactivity.

Despite these advantages, LiPo batteries also have certain limitations. The use of a solid polymer electrolyte results in lower conductivity compared to liquid electrolytes. This reduced conductivity leads to a higher internal resistance, which can affect the battery's overall performance. As a result, LiPo batteries are generally more expensive to produce and require precise charge management to ensure safe and reliable operation.

In cases where a LiPo battery is damaged, physical swelling may occur. This is an indication that the cell is no longer safe for use and should be replaced immediately. Furthermore, improper handling or incorrect charging procedures can significantly increase the risk of failure, including the potential for explosion (Marin-Garcia et al., 2020).

2.9.3.6 Battery management system (BMS)

To ensure the battery operates safely and efficiently while maximizing its lifespan, a Battery Management System (BMS) is used for continuous monitoring and control. The BMS is essential for temperature regulation: battery cells are sensitive to extreme heat, which can lead to degradation and safety risks, while extremely low temperatures can reduce capacity and overall performance. Additionally, the BMS prevents deep discharge, which can cause irreversible damage to the battery and regulates charging to avoid overcharging, ensuring optimal energy storage and extended battery lifespan.

Furthermore, The BMS ensures that all cells in a battery pack are evenly charged and discharged. This is crucial because imbalances in charge levels between cells can lead to inefficiencies, reducing the overall capacity and durability of the battery pack. While a BMS is primarily used for multi-cell battery systems, it can also be applied to single-cell batteries, ensuring safe and reliable operation in various applications.(Horlings, 2024) (Chen, 2022).

2.9.3.7 Charge regulator

While a Battery Management System (BMS) is crucial for multi-cell battery systems, a charge regulator plays an essential role in managing the charging process of single-cell batteries. A charge regulator, also known as a charge controller, ensures that the battery receives a stable and controlled power supply, preventing fluctuations that could negatively impact performance and longevity.

This is particularly important in applications where power generation is inconsistent, such as solar energy systems. As solar panels converts sunlight into electricity, variations in solar intensity throughout the day can cause fluctuations in the generated power. These fluctuations can result in sudden increases or decreases in the generated energy, leading to irregularities in the power supply. Such variations can be harmful to both the battery and connected devices.

The charge regulator is connected to both the battery and the load, enabling safe charging of the battery while ensuring a stable and efficient power supply to the connected system (Moyano,

2022). There are two main types of charge controllers: Pulse Width Modulation (PWM) and Maximum Power Point Tracking (MPPT) (Pop and Avram, 2019).

PWM charge controllers regulate the charging current based on the battery's state of charge and the real-time energy generated by the solar panel (Moyano, 2022). They operate at the same nominal voltage as the battery and use voltage pulses to incrementally charge it. A critical limitation of PWM controllers is that the voltage of the solar panel must closely match the voltage of the battery. For instance, a 12 V solar panel must be paired with a 12 V battery to ensure proper operation and efficient energy transfer.

In contrast, MPPT charge controllers offer greater flexibility. One of their key advantage is that the input voltage from the solar panel does not need to match the battery voltage. This flexibility is achieved through an integrated step-down converter, which allows, for instance, a 24V solar panel to be connected to a 12V battery (Pop and Avram, 2019). MPPT controllers are designed to optimize energy harvesting by continuously operating at the panel's most efficient combination of voltage and current, known as the Maximum Power Point (MPP) (Pop and Avram, 2019) (Moyano, 2022).

2.10 Design of a PV system

2.10.1 The electric load

In order to design a reliable photovoltaic system, it is essential to determine three fundamental parameters: the total daily energy consumption, the average number of peak sunlight hours per day at the installation site and the operating voltage of the PV system. These factors ensure that the energy generated by the system meets the daily load demand.

To estimate the total daily energy consumption, expressed in watt-hours per day (Wh/day), a detailed analysis of all electrical loads must be carried out. This involves identifying each component that will be powered by the PV system, determining its power rating in watts (W) and estimating the average number of hours it operates per day. The energy required for each device is then calculated by multiplying its power rating by the duration of daily use. Summing the energy requirements of all devices provides the total daily energy demand (Al-Shamani et al., 2015). In order to account for system losses and avoid designing a system that underperforms, the total calculated load should be multiplied by a correction factor. This factor, typically valued at 1.3, provides a safety margin to ensure the system consistently meets the energy demand under real-world conditions (Muzaffar, 2021). The adjusted energy requirement (E_r) is therefore calculated as shown in eq. (2.13).

$$E_r = 1.3 * \text{daily average energy consumption} \tag{2.13}$$

Next, the peak power requirement (P_p) of the photovoltaic system must be determined. The peak power is calculated by dividing the total adjusted daily energy demand (E_r) by the average number of effective sunlight hours per day at the specific installation site, as shown in eq. (2.14) (Al-Shamani et al., 2015).

$$P_p = \frac{E_r}{\text{average sun hours per day}} \tag{2.14}$$

Once the peak power requirement of the photovoltaic system has been established, the next step is to determine the total current that the system must be capable of delivering. The total required current (I_{tot}) can be calculated by dividing the peak power output (P_p) of the system

by its operating DC voltage (V_{DC}). This relationship is expressed mathematically in eq. (2.15) (Al-Shamani et al., 2015).

$$I_{tot} = \frac{P_p}{V_{DC}} \tag{2.15}$$

2.10.2 Impact of sun position and panel orientation on PV output

The energy output of a photovoltaic panel is directly influenced by the amount of solar irradiance it receives. As the sun changes its position throughout the day, the intensity and angle at which sunlight strikes the panel also vary, leading to fluctuations in energy production. To optimize performance, it is crucial to position the PV module at a perpendicular angle to the sun when it reaches its highest point in the sky. As a result, the optimal tilt angle of a solar panel is approximately equal to the latitude of the installation site.

When the tilt angle is fixed, the annual energy output of the PV panel will still vary due to seasonal shifts in the sun's position in the sky. During summer months, the sun appears higher in the sky compared to winter months, leading to differences in solar exposure throughout the year. A PV panel positioned at a horizontal tilt angle, 0 degrees, is vulnerable to the build-up of dust, dirt and snow. This build-up can obstruct sunlight and reduce efficiency, which requires more frequent cleaning or maintenance to maintain optimal energy production. Conversely, a panel mounted vertically at a 90 degree tilt receives significantly less direct sunlight, especially during summer, which leads to reduced overall performance. In the Northern Hemisphere, a fixed tilt angle of approximately 30 degrees is commonly recommended to achieve optimal energy output over the entire year.

In addition to tilt, orientation is a key factor in maximizing energy output. PV panels should be oriented towards the south, since the sun is mostly positioned in the southern part of the sky throughout the day. This orientation ensures that the PV system captures the maximum possible solar energy (Franklin, 2017).

2.10.3 Battery sizing

To ensure that a photovoltaic system can continue to operate during periods without solar energy, it is essential to install an appropriately sized battery energy storage system. The capacity of the battery storage system must be capable of supplying the required energy load for a given number of days without sunlight, commonly referred to as the days of autonomy.

The required battery capacity, expressed in ampere-hours (Ah), can be determined by multiplying the total daily energy consumption (E_r) by the number of autonomy days (D) and dividing the result by the product of the maximum allowable level of discharge (MDOD) and the voltage of the battery (V_b), as shown in eq. (2.16) (Al-Shamani et al., 2015).

$$\text{Battery capacity} = \frac{E_r * D}{MDOD * V_b} \tag{2.16}$$

2.10.4 Comparable projects with low-cost sound level meters

2.10.4.1 Picault's Low-Cost Sensors for urban noise monitoring networks

Picaut, Can, Fortin, and Ardouin (2020) explores the rapid development and deployment of low-cost acoustic sensors in urban environments. Traditional noise monitoring methods, such as simulation-based noise maps and high-grade Class 1 measurement stations, are often limited in scalability, realism and affordability. As a response to these limitations, the authors highlight a growing trend in recent years toward distributed sensor networks built with low-cost hardware components, benefiting from advances in microelectronics, miniaturized microphones and wireless data transmission.

The review analyzes over twenty technical implementations, comparing platforms such as Raspberry Pi, STM32 microcontrollers and custom-built boards. In terms of sensing technology, MEMS microphones are increasingly preferred due to their low cost, compact size, good signal-to-noise ratio and robustness to environmental factors such as temperature and humidity. While digital MEMS microphones are often praised for their internal analog-to-digital conversion, Picaut, Can, Fortin, and Ardouin (2020) also reference several effective systems based on analog MEMS microphones, where the signal is conditioned and digitized externally before processing.

In this project, a Raspberry Pi is used in combination with an analog MEMS microphone to capture environmental sound levels. The analog signal is amplified and digitized using an external ADC before being processed on the Raspberry Pi. The system calculates A-weighted sound pressure levels (dBA) in real time and transmits these measurements via a 4G USB dongle to a cloud server for remote monitoring and analysis. This configuration enables flexible and autonomous operation without reliance on local WiFi infrastructure.

The design choices in this setup align well with the recommendations of Picaut, Can, Fortin, and Ardouin (2020), who outline essential performance criteria for low-cost urban noise sensors, including a measurement range of 30–105 dBA, reliable frequency weighting, adequate dynamic range and robust data communication. By combining an analog MEMS microphone with an embedded computing platform and mobile network transmission, this project demonstrates a practical and scalable approach for continuous, real-world noise monitoring.

2.10.5 Citizen Science-Based environmental noise monitoring in Flanders

The paper titled "Region-wide Environmental Noise Monitoring in Flanders through Citizen Science: Protocol to Integrate Surveys and Measurements" presents a project that explores how environmental noise in Flanders can be monitored in an affordable way. The project, known as "Ambient Noise in Flanders", is funded by the Flemish government under the Program for Innovation Procurement (PIO) which was carried out between April 2023 and March 2025.

Within this program, two sub-projects were initiated: one focused on sleep studies involving both indoor and outdoor measurements and another that involves citizens in measuring environmental noise. This second sub-project is referred to as the Ambient Noise in Flanders project. The approach involves the development of a device capable of collecting outdoor sound pressure levels.

To keep the project manageable and affordable, the project involves citizens from across Flanders. Participants receive a measurement kit that allows them to perform noise monitoring autonomously. The kit includes a sound level meter based on the Asus Tinkerboard S R2.0, combined with a MEMS microphone. The design is inspired by the 'Klankentappers' project by Makerspace-Antwerp and Imec, which used a Raspberry Pi 3 combined with a MEMS microphone. The system is designed for outdoor noise monitoring, with measurement locations extending beyond traditional facades and bedroom windows to more flexible settings such as gardens, parks and public spaces.

The main goal is to collect data that represents all social groups and different types of sound environments, such as areas near busy roads, airports, railways and industrial zones. This broad approach helps to accurately understand how much noise people are exposed to in different places and how they experience that noise in daily life (Dekoninck, Thomas, et al., 2023).

2.10.6 A Low-Cost sleep monitoring system to assess environmental noise impact on sleep

The study "Development of a sleep disturbance protocol collecting simultaneous indoor and outdoor noise, motility and heart rate", addresses the second component of the Flemish Program for Innovation Procurement (PIO), namely a sleep monitoring system designed to capture sound pressure levels inside and outside the home. This study aims to investigate how environmental noise, produced both inside and outside the home, can affect sleep quality.

The hardware setup used in this study is based on the same configuration described in section 2.10.5, consisting of the Asus Tinkerboard S R2.0 paired with a MEMS microphone. The Tinkerboard offers improved performance and includes an aluminum casing, which allows for efficient heat dissipation without the need for a cooling fan, thereby lowering the overall cost.

The microphone unit is powered via a flexible cable and housed in a 3D-printed mount, as illustrated in fig. 2.16, which facilitates easy installation and use without technical knowledge or complex tools. The design also provides flexibility in sensor configuration, allowing for the replacement of the MEMS microphone with a more robust analog microphone if long-term durability becomes a concern.

To enable wireless transmission of data from the outdoor microphone to the indoor processing unit, an ESP32-WROOM-23E module is used. This microcontroller communicates with the indoor unit via WIFI. During the pilot phase of the project, a SIM-based network module was integrated into the noise monitoring system to provide mobile network connectivity. In a later stage, this temporary solution will be replaced by a standard modem.

An Internet of Things (IoT) platform, namely Thingsboard, is utilized to collect data from the measurement devices. The data is then transmitted to a centralized data server for further processing and reporting.



Figure 2.16: The indoor noise unit and outdoor noise unit in an operational setting (Dekoninck, Wolters, et al., 2023).

In this experiment, the accuracy of the measurement device was evaluated in an anechoic chamber. The results revealed that the MEMS microphone frequency response was less reliable in both the high-frequency range above 10 kHz and the low-frequency range below 100 Hz. To address this problem, a correction filter was implemented in the signal processing software. The filter adjusts for variations across frequencies, resulting in more reliable data output. For calibration of the microphone, a ring test was conducted, enabling comparison across multiple devices to ensure consistent and accurate performance. A ring test is a calibration method where several measurement units are placed around a sound source in an anechoic room. This allows their readings to be directly compared to a reference microphone.

From the initial analysis of the collected data, it can be observed that indoor noise levels fluctuated between 35 and 40 dB, while outdoor noise levels ranged from 35 to 70 dB. These findings provide early evidence supporting the system's capability to detect meaningful variations in noise exposure. In addition, the results confirm that heart rate is responsive to changes in environmental noise, as illustrated in fig. 2.17. In this specific case, the noise sources were local road traffic and a railway located approximately 30 meters from the monitoring site (Dekoninck, Wolters, et al., 2023).

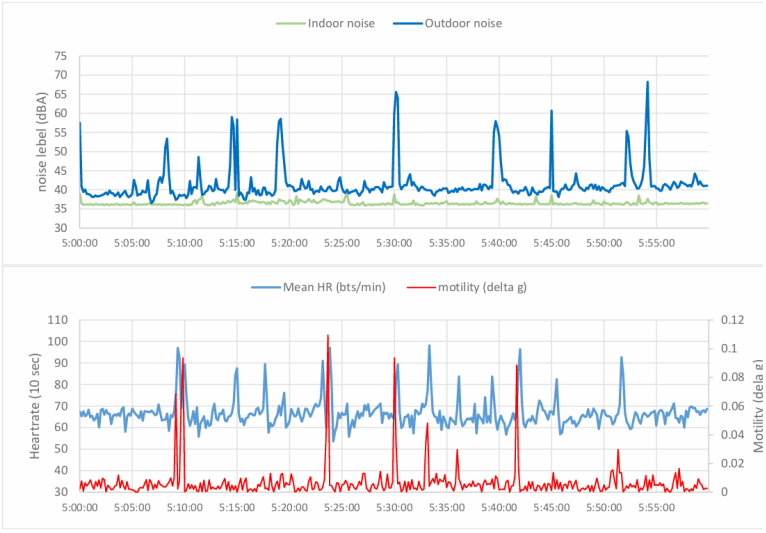


Figure 2.17: Indoor and outdoor noise levels with corresponding heart rate and motility signals (Dekoninck, Wolters, et al., 2023).

The goal of this project is to develop a low-cost and reliable device for environmental noise measurements in outdoor conditions. To meet this objective, several key requirements were considered during the design process. The system must be capable of accurately capturing acoustic data, functioning autonomously without access to a fixed power supply and operating continuously for extended periods of time. The following sections provide an overview of the selected materials, explain how they were integrated into a functional system and describe the methods used to enable reliable data.

3.1 Materials

3.1.1 Technical Overview of the ICS-40638 Microphone

The microphone used in this setup for acoustic measurements is the analog MEMS microphone ICS-40638, as illustrated in fig. 3.1. This microphone includes a MEMS microphone element, an impedance converter and a differential output amplifier. The output polarity is non-inverted, meaning that the output signal is not inverted compared to the input sound. As discussed in section 2.3.6, this sensor is capable of handling high sound pressure levels up to 138 dB SPL while maintaining low distortion, making it well suited for environmental noise monitoring.

With a signal-to-noise ratio (SNR) of 63 dBA, the microphone provides a clear separation between the measured acoustic signal and the inherent electrical noise generated by the microphone itself. Additionally, its frequency response, ranging from 35 Hz to 20 kHz, ensures that the microphone can capture a broad spectrum of audible sounds, from low-frequency ambient noise to high-frequency peaks.

Energy efficiency is another important aspect of this component. When powered at 3.3 V, the microphone operates at a low current consumption of 190 µA, which makes it an excellent choice for systems relying on limited energy sources such as solar-powered configurations.

Based on its equivalent input noise (EIN) of 31 dBA SPL and an acoustic overload point (AOP) of 138 dBA SPL, the effective dynamic range of the microphone is approximately 107 dB. This defines the range in which the sensor can provide accurate and linear sound pressure level measurements (TDK InvenSense, 2019a). An overview of its specifications is summarized in table 3.1.



Figure 3.1: The ICS-40638 MEMS microphone which is integrated in the noise monitoring setup (T. InvenSense, 2024).

Table 3.1: Electrical characteristics of the ICS-40638 (T. InvenSense, 2024).

Parameter	Conditions	Min	Typ	Max	Units
Supply Voltage	Standard mode	1.52	–	3.63	V
Supply Current		–	170	190	µA
Sensitivity	1 kHz, 94 dB SPL, differential	-44	-43	-42	dBV
Signal-to-Noise Ratio	20 kHz bandwidth, A-weighted	–	63	–	dBA
Total Harmonic Distortion	105 dB SPL	–	0.2	–	%
Equivalent Input Noise	20 kHz bandwidth, A-weighted	–	31	–	dBA SPL
Acoustic overload point	10% THD	–	138	–	dB SPL

3.1.2 INA129-EP precision, low power instrumentation amplifier

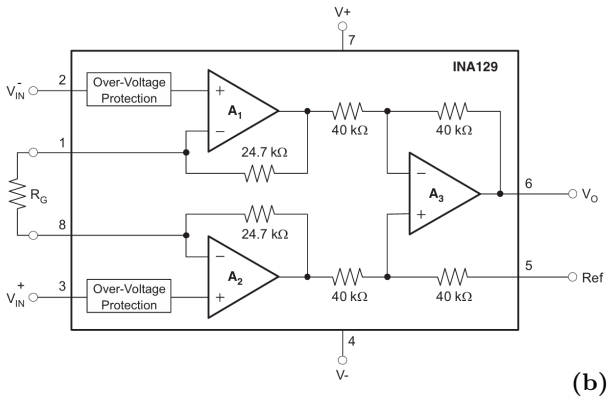
In order to amplify the analog output signal of the MEMS microphone, the INA129-EP is utilized as represented in fig. 3.2b. This is a differential instrumentation amplifier specifically designed for precise measurement and test applications. The amplifier features an integrated input protection circuit and input buffer amplifiers with its key electrical characteristics summarized in table 3.2.

The simplified schematic in fig. 3.2a illustrates the internal configuration of the INA129-EP, which consist of three operational amplifiers and an external resistor used to set the amplifier's gain. This design allows the input signal to be amplified by a factor ranging from 1 to 10000, offering flexibility depending on the specific needs of the measurement setup. The gain is determined according to eq. (3.1).

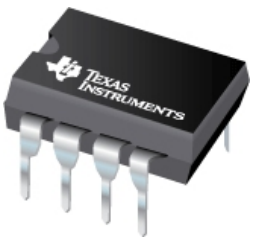
$$G = 1 + \frac{49.4\,\text{k}\Omega}{R_G} \tag{3.1}$$

Moreover, the INA129-EP offers the advantage of a low DC offset voltage, which enhances measurement accuracy, especially when dealing with very small input signals. This makes it a reliable choice for applications where precision and signal integrity are critical.

The amplifier operates over a wide power supply range of ± 2.25 V and ± 18 V and consumes a current of only 750 µA. This low power consumption makes it suitable for energy-efficient systems, such as battery-powered or solar-powered devices (Texas Instruments, 2015).



(a) Simplified schematic of the INA129-EP (Texas Instruments, 2015).



(b) The INA129-EP, used to amplify the analog signal of the MEMS-microphone (Texas Instruments, 2025).

Figure 3.2: The INA129-EP and its simplified schematic.

Table 3.2: Electrical characteristics of the **INA129**-EP (Texas Instruments, 2015).

Parameter	Min	Max	Units
Supply Voltage	±2.25	±18	V
Operating Temperature	−55	125	°C
Range of Gain	1	10 000	–
Quiescent Current	–	750	µA

3.1.3 The communication module

For the processing and transmission of data, a Raspberry Pi 4 Model B was chosen as the central computing unit. This single-board computer is equipped with a quad-core 64-bit processor operating at 1.5 GHz and is available with up to 8 GB of RAM. The Raspberry Pi used in this project has 2 GB of RAM available. Its computational capabilities and memory resources enable real-time audio signal processing, local data storage and concurrent network communication.

A key advantage of the Raspberry Pi 4 is its ability to run a full-featured Linux-based operating system which facilitates the use of advanced software libraries and tools. In this project, Python was utilized for signal processing tasks, while Rclone was used to automatically synchronize the recorded data with an external server via a connected USB 4G dongle. In addition, the Raspberry Pi 4 includes a 40-pin GPIO header that supports a variety of communication protocols such as SPI, I2C and, UART. This feature enables seamless integration with external hardware modules, such as the MCC128 DAQ HAT, which was used for analog signal acquisition as discussed in section 3.1.4.

Compared to a traditional microcontroller platform such as the Arduino Uno, the Raspberry Pi 4 offers significant advantages in terms of processing power, multitasking capabilities and software support. Microcontrollers are typically limited to running a single program without an operating system and lack the computational capacity, memory and input/output capabilities required for tasks involving real-time audio acquisition, file system management and network communication (Raspberry Pi (Trading) Ltd, 2024).

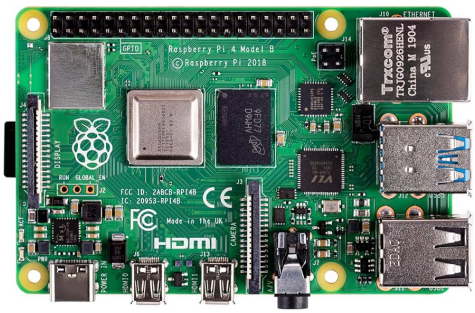


Figure 3.3: The Raspberry Pi 4 Model B.(Raspberry Pi (Trading) Ltd, 2024).

3.1.4 The data acquisition system

As part of this project, the MCC 128 DAQ HAT is used as the data acquisition system to convert analog voltage signals into digital measurements. This HAT was directly connected to the 40-pin GPIO header of a Raspberry Pi 4. The analog signals were provided by the MEMS ICS-40638 microphone.

The choice for the MCC 128 was motivated by its combination of high resolution, measurement accuracy and compatibility with the Raspberry Pi platform. Since the Raspberry Pi 4 does not include built-in analog-to-digital conversion, an external HAT module was required to process the analog signals. The MCC 128 features a 16-bit analog-to-digital converter which is essential

for accurately capturing the small voltage variations generated by a MEMS microphone at low sound levels. It also supports a maximum sampling rate of 100 kS/s, enabling reliable digitization of audio signals.

The MCC 128 allows measurements in both single-ended and differential configurations, with selectable voltage input ranges from $\pm$ 1 V to $\pm$ 10 V. This flexibility enables optimal alignment of the input signal with the dynamic range of the microphone (Corporation, 2020).

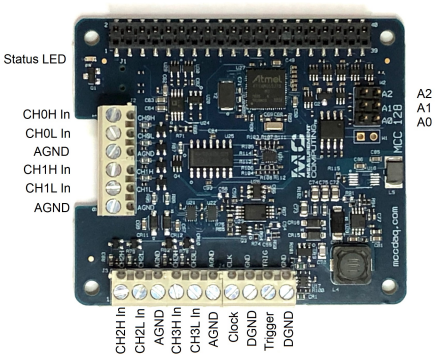


Figure 3.4: The MCC 128 DAQ HAT, used for reading and digitizing the analog microphone signal (Measurement Computing Corporation, 2023).

3.1.5 The ZTE MF79U modem for autonomous data transmission

This project aims at the realization of an automated measurement system designed to continuously and remotely monitor urban noise, that also transmits the collected data to a cloud storage. Achieving this requires a stable and autonomous internet connection on the measurement site, that is independent of existing local WIFI networks. To fulfill this requirement, the ZTE MF79U Wingle CAT4-4G wireless USB modem, illustrated in fig. 3.5, is connected to the Raspberry Pi via a USB port. This provides the necessary mobile connectivity for real-time or periodic data transmission.

The ZTE MF79U is a compact LTE category 4 modem that support download speeds of up to 150 Mbps and upload speeds up to 50 Mbps. Although the device features an integrated WiFi hotspot capable of supporting up to ten devices, within the scope of this project, it is solely used to provide a connection to the Raspberry Pi. This hotspot operates through the use of a SIM card that is inserted into its 3FF SIM card slot. Furthermore, the dongle is equipped with two external antenna connectors (TS-9), which can be utilized to improve the signal reception. With physical dimensions of 101.6 x 31 x 14 mm, the device is both portable and easy to integrate into the remote measurement setup (Amazon, 2025d).



Figure 3.5: The ZTE MF79U LTE USB Modem used for mobile connectivity in the measurement setup (Amazon, 2025d).

3.1.6 Sizing of the photovoltaic module

To enable the autonomous operation of the noise monitoring system, an appropriate sizing of the photovoltaic panel was essential. Following the methodology outlined in section 2.10.1, the first step in designing a photovoltaic system, is estimating the total daily energy consumption. This is achieved by identifying each electrical component and determining its nominal power consumption. Since all components are intended to operate continuously for a 24-hour period, their individual daily energy consumption was calculated by multiplying the power consumption by 24. Table 3.3 provides a detailed overview of these values.

Table 3.3: Estimated power consumption per component based on specifications.

Component	Voltage (V)	Current (mA)	Power (W)	h/day	Wh/day
Raspberry Pi	–	–	2.1	24	50.4
ICS-40638	3.3	0.19	0.00063	24	0.015
INA129-EP	5.0	2	0.01	24	0.24
MCC-128	5.0	135	0.675	24	16.2
Dongle	–	–	0.4	24	9.6
Total					**76.46**

The power consumption of the Raspberry Pi and the dongle were obtained through direct measurements using a wattmeter. These measurements were taken under realistic load conditions to ensure an accurate representation of their typical power usage. For the remaining components, the power values were estimated based on manufacturer datasheets. The operating voltage and maximum supply current of each component were extracted, after which the power consumption was calculated using eq. (3.2).

$$P = V \times I \tag{3.2}$$

In addition to the theoretical power estimation based on individual components, supplementary measurements were performed using a wattmeter to validate the system's actual energy usage. These measurements included both the initial power required during startup and the steady operating conditions. The startup power consumption, as shown in table 3.4, was observed to peak at approximately 4.6 W, resulting in a short increase in power demand caused by the startup of all components. After the initialization phase, the system maintained a stable power consumption of 3.3 W during continuous operation.

Table 3.4: Measured power consumption under startup and steady operating conditions.

Test Condition	Power (W)	h/day	Wh/day
Startup phase	4.6	24	110.4
Steady state	3.3	24	79.2

In order to obtain a more realistic estimation of the total energy demand, a mean value was calculated by combining the theoretical power derived from the component specifications with the measured startup and steady-state values. This approach, shown in eq. (3.3), provides a balanced estimate that accounts for both expected and observed system behavior.

$$P_{\text{avg}} = \frac{76.46 + 110.3 + 79.2}{3} = 88.69\,\text{Wh} \tag{3.3}$$

In order to account for system losses and avoid designing a system that underperforms, a correction factor must be applied to the average calculated daily consumption. Following the design methodology outlined in section 2.10.1, a correction factor of 1.3 was used, as shown in eq. (3.4).

$$E_r = 1.3 * \text{daily average energy consumption} = 1.3 * 88.69 = 115.29 \text{ Wh} \tag{3.4}$$

Following the correction of the estimated energy demand, the next step in the dimensioning process is to determine the required peak power output of the photovoltaic system. As described in the design methodology section 2.10.1, this is done by dividing the daily energy requirement by the average number of effective sunlight hours per day. This approach is expressed in eq. (2.14) and ensures that the selected solar panel can generate sufficient energy to meet the system's requirements.

The average number of effective sunlight hours per day for each month was estimated based on data published by the Royal Meteorological Institute of Belgium (Koninklijk Meteorologisch Instituut van België (KMI/IRM), 2020). The values presented in table 3.5 represent typical monthly solar exposure in Brussels throughout the year. These values were manually derived from official graphs available on the KMI's website.

Table 3.5: Monthly average peak sun hours at the installation site (Koninklijk Meteorologisch Instituut van België (KMI/IRM), 2020).

Month	**h/day**
January	1.9
February	2.7
March	4
April	5.6
May	6.4
June	6.6
July	6.6
August	6.4
September	5.1
October	3.6
November	2.2
December	1.5
Average	**4.38**

Based on the average of 4.38 h/day from table 3.5, the required peak power output is calculated using eq. (3.5). This parameter represents the nominal power rating that a solar panel must have in order to generate the corrected daily energy requirement under typical solar conditions.

$$P_p = \frac{E_r}{\text{average sun hours per day}} = \frac{115.29}{4.38} = 26.30 \text{ W} \tag{3.5}$$

3.1.6.1 Technical specifications of the PV module

Based on the calculated peak power requirement of 26.30 W, as established in section 3.1.6, a 30 W monocrystalline photovoltaic panel was selected, as shown in fig. 3.6. This selection includes a design margin to compensate for typical variations in performance and ensure sufficient power generation under realistic operating conditions. This solar panel allows the system to function independently, without the need for an external power supply, by continuously charging a 12 V battery that powers the setup.

The selected module consist of 182 mm monocrystalline PERC cells, offering a cell efficiency exceeding 23.1%. The design includes 10 multi-busbars to reduce the internal resistance, thereby minimizing energy loss during energy conversion. This design improvement enhances the overall efficiency of the panel.

The electrical specifications of the panel predict a power output of 30 W, a maximum voltage of 19 V and a corresponding maximum current of 1.58 A. At full power output the open-circuit voltage is 22.80 V, while the short-circuit current is 1.72 A. The module can operate reliably over a wide temperature range from −40 °C to +85 °C, ensuring a stable performance across a wide variety of environmental conditions. This is essential for accurate acoustic measurements in outdoor settings.

The construction of the panel features a tempered solar glass surface combined with a weather-proof coating and a corrosion-resistant anodized aluminum frame for outdoor use. It includes an integrated bypass diode to minimize power loss in case of partial shading and is certified according to the IP65 protection standards, indicating protection against dust and water intrusion. The dimensions of the panel are 465 × 400 × 25 mm, for a weight of 1.8 kg (Amazon, 2025b) (SolarV, 2025).

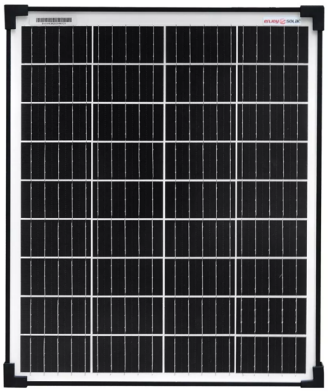


Figure 3.6: Monocrystalline solar panel 30W and 12V (Amazon, 2025b).

3.1.7 Sizing of the battery

To ensure continuous operation of the system during periods without direct sunlight, a battery storage unit was integrated into the design. The required battery capacity was determined using the sizing methodology outlined in section 2.10.3.

Given the compactness design requirements of the system, both the physical dimensions and the weight of the battery were key factors in the selection process. Lithium-based batteries and in particular $LiFePO_4$ batteries were identified as the most suitable storage technology. Compared to traditional lead-acid batteries, $LiFePO_4$ batteries allow for a more compact and lightweight solution, while still providing the required energy storage capacity.

Based on the daily energy requirement determined in eq. (3.4), the necessary battery capacity was calculated to ensure 24-hour autonomous operation under realistic conditions, using eq. (3.6).

$$\text{Battery capacity} = \frac{E_r * D}{MDOD * V_b} = \frac{115.29 * 1}{0.8 * 12.8} = 11.26Ah \tag{3.6}$$

As a result, a battery capacity of at least 11.26 Ah is required to meet the system's 24-hour energy demand under normal operating conditions.

3.1.7.1 Selection and specification of the battery

In line with the system's requirement for uninterrupted energy supply during periods without direct solar input, a lithium iron phosphate battery $LiFePO_4$ from Cyccle Volt was selected as the energy storage unit, as shown in fig. 3.7. This specific model offers a nominal voltage of 12.8 V and a nominal capacity of 12 Ah, corresponding to a total energy capacity of approximately 153 Wh. When compared to the minimum required battery capacity of 11.26 Ah, derived from eq. (3.6), the selected battery provides a slightly higher capacity than the minimum requirement, thereby offering a small safety margin that enhances the overall system reliability.

The selected battery features an integrated Battery Management System (BMS). This BMS ensures safe and stable operation by protection against overcharging, deep discharging, short circuit and high temperatures. According to the manufacturer specifications, the battery is rated for 4000 to 15000 charge cycles, offering a lifespan well suited to the system's long-term operational goals. From a mechanical perspective, the compact physical dimensions of the battery, measuring $152 \times 98 \times 96$ mm, supports the overall design requirements of the application without compromising functionality or endurance (Amazon, 2025a).



Figure 3.7: 12.8V and 12Ah $LiFePO_4$ battery selected for energy storage in the system (Amazon, 2025a).

3.1.8 Sizing of the solar charging controller

To ensure safe and efficient energy transfer from the photovoltaic panel to both the energy storage unit and the load, a solar charge controller is integrated into the system. This component regulates the charging process of the battery by controlling voltage and current levels, thereby preventing conditions such as overcharging, deep discharging and reverse current flow.

The solar panel used in this system has been previously detailed in section 3.1.6.1, it is a monocrystalline 30W module operating at 12V. Based on these characteristics and the system's 12.8 V battery, the required current rating of the charge controller is determined using eq. (3.7) (Errol Prowse, 2018).

$$\text{Charge control current} = \frac{P_{PV}}{V_b} = \frac{30}{12.8} = 2.34A \tag{3.7}$$

Although eq. (3.7) indicates that a charge controller with a current rating above 2.34 A would be sufficient to handle the charging requirements of the system, many compact controllers within this range are not designed to deliver the higher output currents demanded by the external loads. In this case the system includes a Raspberry Pi, which demands a stable current of approximately 2 A. Most compact charge controllers in this class are incapable of delivering this level of output current, making them unsuitable for a reliably powering of the Raspberry Pi under normal operating conditions.

3.1.8.1 Technical specifications of the solar charge controller

To ensure reliable system operation, including both continuous battery charging and power delivery to the load, a more robust solution was required. For this reason, a 30A MPPT solar charge controller was selected. In addition to its electrical capabilities, the choice was also influenced by the limited availability of charge controllers with a current capacity that is close to the system requirement on the consumer market. Selecting a 30A unit not only provides sufficient overhead for present system demands but also offers flexibility for potential future expansions.

The selected controller, as is shown in fig. 3.8, is equipped with Maximum Power Point Tracking (MPPT) technology, offering a high tracking efficiency of up to 98%, which maximizes the energy harvested from the solar panel under varying conditions. It supports both 12V and 24V batteries and automatically detects the battery voltage. The controller supports a wide range of battery chemistries, including gel, AGM, lead-acid and lithium batteries such as $LiFePO_4$.

Additionally, it features two integrated 5V/2A USB outputs. This is particularly useful for directly powering a Raspberry Pi, eliminating the need for a separate voltage regulator within the system.

To enhance safety and durability, the controller includes comprehensive electrical protection features, such as overcurrent protection, short circuit protection, overload protection, idle protection, undervoltage protection, reverse polarity protection and automatic recovery mode. Such protection mechanisms are essential to prevent hardware failure. To ensure thermal stability during operation, the controller is enclosed in a robust casing with an aluminum alloy heat sink panel that helps maintain a stable temperature using passive cooling.

Charging is managed through an intelligent 4-stage algorithm, consisting of fast charging, boost charging, floating charging and balanced charging. This algorithm is specifically designed to optimize battery performance, accelerate charging speed and extend the overall battery life (Amazon, 2025c).



Figure 3.8: 30A MPPT solar charge controller selected for battery regulation and system power delivery (Amazon, 2025c).

3.2 Methods

3.2.1 MEMS microphone hardware design and experimental evaluation

3.2.1.1 Microphone selection and evaluation board design

In order to develop a reliable system for measuring and analyzing urban noise, it is essential to select a microphone capable of capturing both low and high sound pressure levels, while maintaining low distortion and high sensitivity. Urban acoustic environments are characterized by a complex spectrum of sound sources, ranging from low-amplitude background noise to high intensity continuous sounds, distributed across a wide frequency spectrum. Accurately capturing this diversity in amplitude and frequency requires a sensor with a wide dynamic range to respond effectively to the full spectrum of urban sound conditions.

As outlined in section 2.3.6, multiple MEMS microphones were considered and evaluated. Based on this comparative evaluation, the ICS-40638 was identified as the most suitable option for the intended application. A comprehensive technical overview of the microphone is provided in section 3.1.1, where its suitability for integration into the measurement system is discussed in further detail.

To enable testing and implementation of the ICS-40638 MEMS microphone within a prototype system, a custom evaluation board was designed and fabricated. The design process was based on the technical documentation of the EV_ICS-40638-FX evaluation board (Inc., 2020). In this configuration, the microphone is connected to a 0.1 µF supply bypass capacitor placed between the power supply line (VDD) and ground (GND). This bypass capacitor is essential for stabilizing the power supply voltage delivered to the microphone. It functions by filtering high-frequency noise and transient voltage spikes present on the DC power line, providing a low-impedance path to ground for these disturbance. As a result, the capacitor helps maintain a clean and stable supply voltage. The schematic representation of this configuration is shown in fig. 3.9, illustrating the placement of the bypass capacitor relative to the microphone and power supply line.

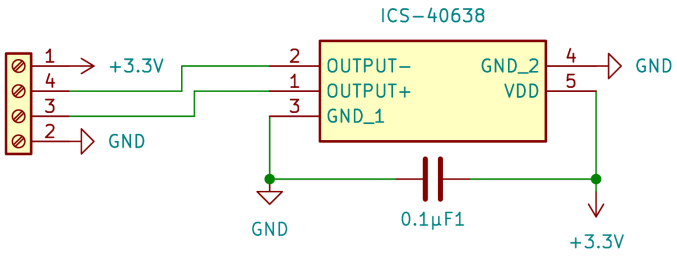


Figure 3.9: Schematic diagram of the ICS-40638 MEMS microphone evaluation board.

3.2.1.2 Implementation and testing of the evaluation board

After the design and fabrication of a custom PCB based on the EV_ICS-40638-FX evaluation board, the board was manually assembled. This involved soldering the ICS-40638 MEMS microphone and its supporting components. The microphone is a surface-mounted device (SMD) and due to the small footprint and thermal sensitivity of the microphone, the soldering process required significant precision. The soldering took place at the VUB Fablab, where the available tools were limited and professional equipment for controlled SMD soldering wasn't accessible. As a result, it was not possible to execute the SMD soldering with the level of control and reliability required for such delicate components.

Once the custom PCB has been assembled, its functionality was evaluated using an oscilloscope. To assess the microphone's response, multiple sinusoidal tones at different frequencies were played from a smartphone speaker positioned at a fixed, short distance from the microphone. The objective was to observe a corresponding sinusoidal waveform on the oscilloscope. However, the expected waveforms were not clearly visible. To further evaluate the microphone's responsiveness, we generated short, sudden sounds such as hand claps to check whether the microphone was responsive to acoustic impulses. In this case as well, the oscilloscope displayed little to no clear response, suggesting that the microphone failed to capture acoustic impulses.

These observations indicated that the microphone was not functioning as intended, leading us to suspect that the component may not have been soldered properly. Given the small footprint and thermal sensitivity of the MEMS microphone, it is likely that one or more pins were not fully connected or that the component was exposed to excessive heat during soldering. Since such problems are difficult to verify without advanced inspection tools and considering the limited equipment available in the VUB Fablab, we decided to replace the custom board with the official EV_ICS-40638-FX evaluation board, illustrated in fig. 3.10. This board includes a mounted MEMS microphone to eliminate uncertainties related to manual soldering.

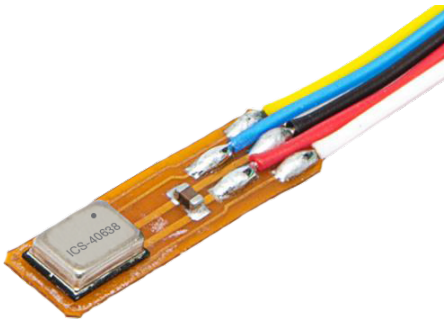


Figure 3.10: The ICS-40638 MEMS microphone evaluation board (TDK InvenSense, 2025).

3.2.1.3 Signal evaluation and voltage analysis using the EV_ICS-40638-FX evaluation board

After acquiring the official EV_ICS-40638-FX evaluation board, the acoustic tests described in section 3.2.1.2 were repeated in order to evaluate the performance of the board. A more consistent response was observed compared to the results from the custom-assembled board. Hand claps produced a measurable signal on the oscilloscope, illustrated in fig. 3.11, conforming the microphone's ability to detect sudden acoustic events.

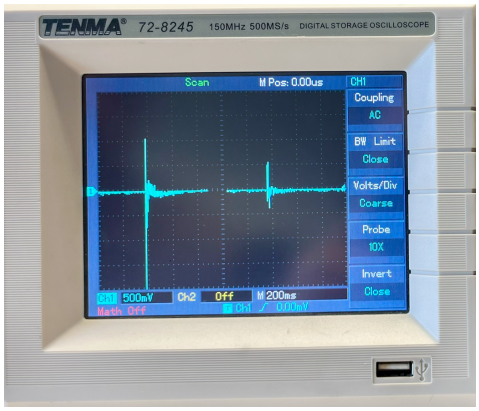


Figure 3.11: Time-domain signal of a hand clap captured by the ICS-40638 MEMS microphone evaluation board.

To evaluate the frequency response of the microphone in a more controlled setting, a signal generator was used to produce sinusoidal waveforms at specific frequencies and amplitudes. The generated signals included tones at 1 kHz, 5 kHz and 10 kHz covering a range of common audio frequencies. These signals were transmitted to an external loudspeaker, which was positioned at a fixed and short distance from the microphone (±5 cm).

To integrate the microphone into the full acquisition system, the evaluation board was connected to a Raspberry Pi equipped with a MCC 128 DAQ HAT. A Python script was developed to interface with the DAQ HAT and record the output voltages from the MEMS microphone. The resulting data, presented in table 3.6, indicated that the signal voltages were relatively low. As a consequence, the input signal utilized only a small fraction of the MCC 128 DAQ HAT's available input range (±1 V to ±10 V), thereby limiting the effective utilization of the resolution of the 16-bit analog-to-digital converter.

Table 3.6: Measured output voltages from the MEMS microphone.

Measurement number	Voltage (V)
1	0.0020339
2	0.0022767
3	0.0030014
4	0.0033499
5	0.0036375
6	0.0040908
7	0.0042265
8	0.0046838
9	0.0043013
10	0.0041525

To ensure accurate digitization and improve signal-to-noise ratio, an operational amplifier was used. It was designed to have an appropriate gain to scale the signal closer to the input range of the HAT. Several operational amplifiers were considered. Finally the INA129-EP differential instrumentation amplifier was selected, as it offers high precision, low input offset voltage and an adjustable gain. A detailed overview of the amplifier's characteristics is provided in section 3.1.2.

An additional key factor in the selection of the INA129-EP was its compatibility with single-supply operation. In contrast to many standard operational amplifiers that require both positive and negative supply voltages, the INA129-EP is capable of functioning reliably with a single positive supply. This characteristic was particularly important for the setup, as the Raspberry Pi does not provide a negative voltage. Eliminating the need for an additional negative power supply simplified the design, lowered the power consumption and contributed to a more compact and cost-efficient implementation.

To amplify the analog output of the microphone, a gain of 10 was selected. This value was chosen to bring the signal voltage within a range that optimally utilizes the resolution of the HAT. The gain of the amplifier is defined by an external resistor R_G and the relationship between the gain and the external resistor is given by the expression eq. (3.1), which is discussed in detail in section 3.1.2. To obtain a gain of 10, the necessary resistor value R_G can be calculated using eq. (3.8).

$$R_G = \frac{49.4\,\text{k}\Omega}{G-1} = \frac{49.4\,\text{k}\Omega}{10-1} = 5.49\text{k}\Omega \tag{3.8}$$

3.2.1.4 Design of the first PCB prototype

After determining the required gain and selecting the INA129-EP as the preferred instrumentation amplifier, the first PCB prototype was designed. The schematic of this prototype is shown in fig. 3.12. The acquisition process begins at the ICS-40638 MEMS microphone, which converts acoustic pressure variations into a differential analog output signal. To decouple the microphone from noise on the power supply, a 0.1 µF bypass capacitor is placed between the supply line (VDD) and ground (GND).

The differential output of the microphone is AC-coupled through a pair of 1.5 µF capacitors connected in parallel, resulting in an effective capacitance of 3 µF per signal path. In this configuration, the capacitor blocks any DC component present at the microphone output, while allowing the AC audio signal to pass. Together with the resistors in the following stage, these capacitors form a first-order RC high-pass filter that suppresses the DC component and alternates low-frequency noise. Such noise typically originates from mechanical vibrations. If not attenuated, it can degrade the accuracy and quality of the captured audio signal.

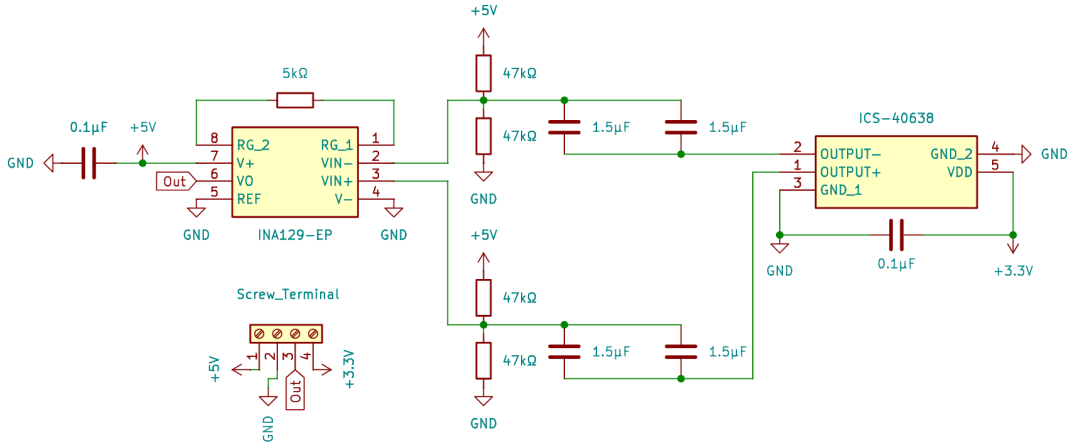


Figure 3.12: Schematic diagram of the first PCB design integrating the ICS-40638 MEMS microphone with the INA129-EP instrumentation amplifier.

In this design, for the DC analysis, two 47 kΩ resistors are connected in series between the +5 V supply and ground, forming a voltage divider. For the AC signal analysis, the +5 V supply can be modeled as an ideal voltage source with zero internal resistance. This means the +5 V is for AC signals, directly connected to ground. As a result, both resistors are in parallel, the resistance can be calculated with eq. (3.9).

$$R_t = \frac{R_1 * R_2}{R_1 + R_2} = \frac{47 * 10^3 * 47 * 10^3}{10^3 * (47 + 47)} = 23.5\,\text{k}\Omega \tag{3.9}$$

In combination with the capacitors, the -3 dB cutoff frequency of the filter can therefore be calculated using eq. (3.10).

$$f_c = \frac{1}{2 * \pi * C * R_t} = \frac{1}{2 * \pi * 3 * 10^{-6} * 23.5 * 10^3} = 2.26\,\text{Hz} \tag{3.10}$$

This cutoff frequency lies below the audible range, ensuring that the full bandwidth relevant to speech and environmental sounds is preserved while unwanted low-frequency noise is suppressed. Similar to the configuration at the microphone, a 0.1 µF capacitor is placed between the 5 V supply and ground to decouple the amplifier from high-frequency noise on the supply line.

The AC-coupled and biased differential signal is then fed into the INA129-EP instrumentation amplifier. As described in section 3.2.1.3, the amplifier is configured with a gain of 10, ensuring that the microphone's output amplitude is optimally scaled to fit in the available voltage range for subsequent signal processing stages. The REF pin is connected to ground, thereby setting the output reference potential to 0 V.

Within the INA129-EP, the incoming differential signal from the microphone is converted into a single-ended output. The resulting single-ended, amplified signal is routed to the screw terminal, which also distributes the 5 V supply, ground reference, a 3.3 V rail for powering the ICS-40638 microphone and a signal output pin that delivers the amplified signal to the Raspberry Pi for digitization and further processing. This complete hardware implementation is shown in Fig. 3.13.

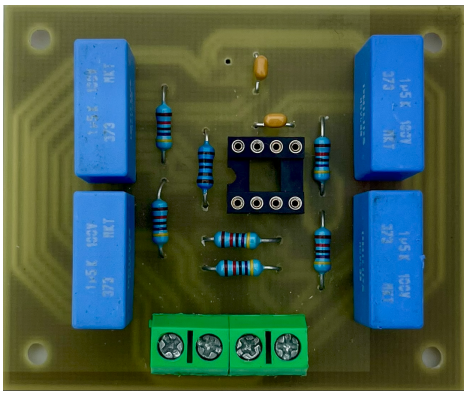


Figure 3.13: Assembled first custom PCB prototype designed for the ICS-40638 MEMS microphone.

3.2.2 Testing of the first PCB prototype

To verify the performance of the PCB, oscilloscope measurements were conducted. The microphone output was monitored while a continuous sinusoidal acoustic signal was applied. The oscilloscope measurements, as presented in fig. 3.14, revealed that the negative half cycles of the sinusoidal waveform were missing. Instead, the signal appeared clipped at 0 V, indicating that only the positive half of the waveform was being processed.

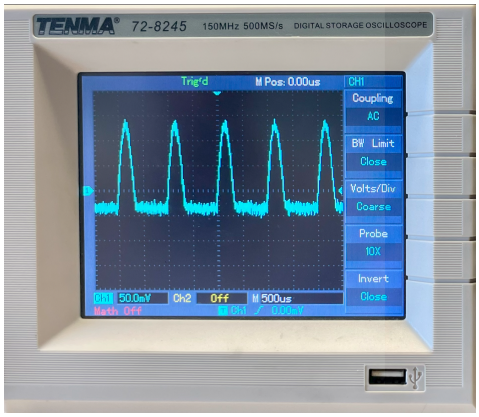


Figure 3.14: First PCB prototype evaluation: clipping phenomenon observed, with 0 V reference.

This behavior resulted from the configuration of the INA129-EP instrumentation amplifier. With the reference pin connected to 0 V, the output range was restricted to positive values, resulting in the clipping of the negative half cycles of the input waveform. As a result, a significant amount of the signal information was lost.

To address this issue, a voltage divider was implemented to set the reference pin to 2.5 V. With the reference set to half the supply voltage, the amplifier was able to reproduce both positive and negative portions of the waveform within the 0-5 V range. As shown in fig. 3.15, this adjustment restored the full sinusoidal signal shape, allowing the circuit to operate effectively within a single-supply configuration.

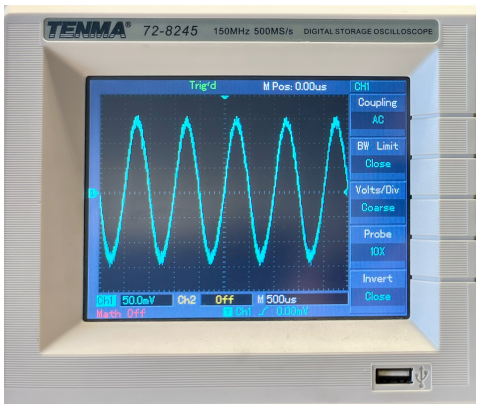


Figure 3.15: First PCB prototype evaluation: restored signal with 2.5 V reference.

3.2.2.1 Final PCB design

Following the evaluation of the first prototype, several modifications were implemented to refine signal quality and to achieve a more compact layout. The resulting final PCB incorporates both electrical and structural optimizations over the initial design.

In the previous prototype, each differential output of the microphone was AC-coupled through two 1.5 µF capacitors connected in parallel, resulting in an effective capacitance of 3 µF per output. In the updated version, as shown in fig. 3.16, these parallel capacitors were replaced with single 3 µF SMD capacitors at each microphone output. This change maintains the required coupling capacitance allowing a more compact board layout, as shown in fig. 3.17.

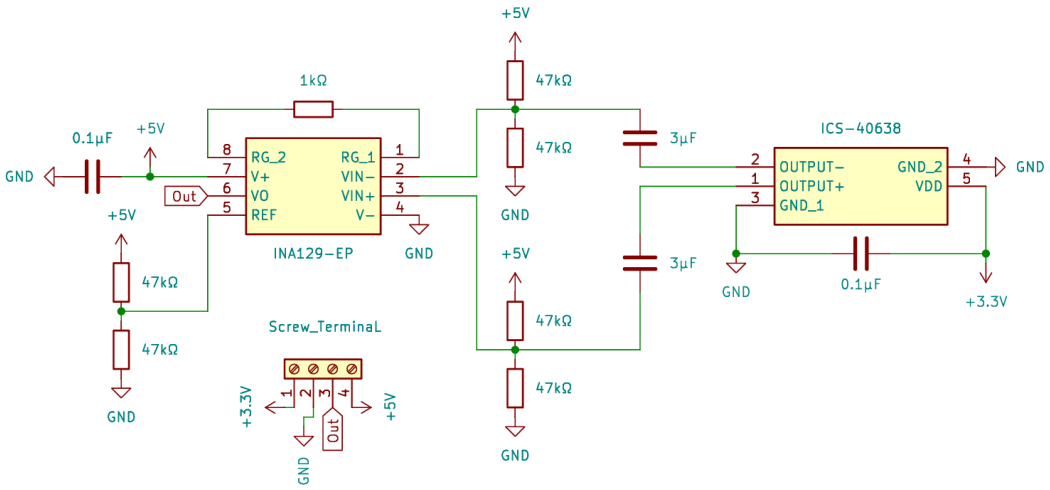


Figure 3.16: Schematic diagram of the final PCB design integrating the ICS-40638 MEMS microphone with the INA129-EP instrumentation amplifier.

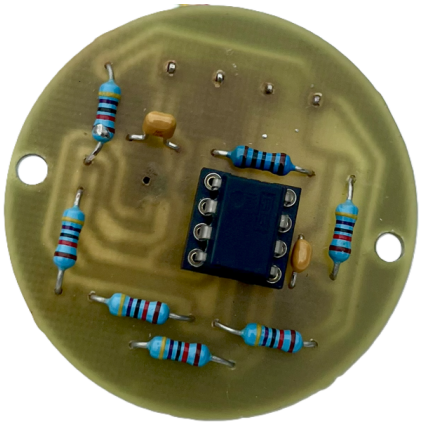


Figure 3.17: Assembled final custom PCB designed for the ICS-40638 MEMS microphone.

An additional modification in the final PCB design involved adjusting the gain setting resistor R_G of the instrumentation amplifier to achieve a higher amplification. In the initial prototype a 5 kΩ resistor was used, corresponding to a gain of approximately 10. This setting resulted in an output amplitude that was insufficient for optimal use of the available voltage range. To increase the signal strength, a target gain of 50 was established, the necessary resistor value R_G can be calculated using eq. (3.11).

$$R_G = \frac{49.4\,\text{k}\Omega}{G-1} = \frac{49.4\,\text{k}\Omega}{50-1} = 1.01\text{k}\Omega \tag{3.11}$$

Consequently, a resistor value of 1 kΩ was implemented. This adjustment substantially increased the output amplitude of the signal, improving the signal-to-noise ratio. The effectiveness of this modification is illustrated in fig. 3.18, which displays the amplified microphone signal captured with an oscilloscope. The waveform demonstrates a substantial increase in amplitude compared to the earlier configuration shown in fig. 3.15, while preserving its sinusoidal shape without noticeable distortion. This allows the circuit to deliver a clean and robust signal to the Raspberry Pi for further processing.

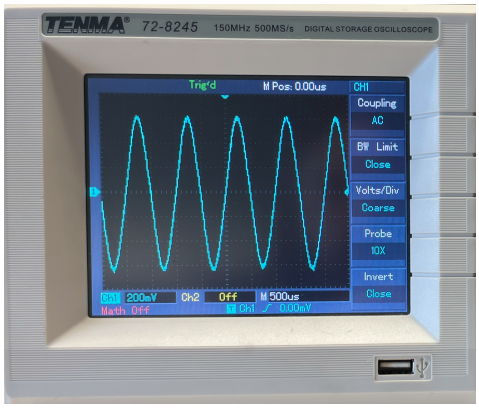


Figure 3.18: Measured output waveform of the final PCB design at a gain of 50.

3.2.3 Calibration of the Microphone

3.2.3.1 Goal of the calibration

Following the design and construction of our microphone, the next essential step is to assess its capability to accurately measure sound pressure levels in dB V. At this stage, no assumption can be made regarding its absolute precision. Therefore, to ensure reliable and reproducible measurement results, the accuracy of the microphone must be improved through calibration. For this purpose, the G.R.A.S PL40 reference microphone was used, as shown in fig. 3.19a.

The reference microphone itself is calibrated with a sound calibrator, illustrated in fig. 3.19b. This device produces a stable sinusoidal acoustic signal of 94 dB SPL at 1 kHz. Since this signal is standardized and precisely defined, it can be considered to be exact. This step provides the absolute calibration of the reference microphone, whereby its sensitivity is defined relative to a traceable acoustic pressure reference.

Calibration of our self-built microphone then consists of systematically transferring the established characteristics of the reference microphone to our microphone. This procedure, known as the relative calibration, aligns the test microphone with the fully characterized reference. Our device can therefore then be regarded as an accurate and reliable measurement instrument. A key assumption for this process is that the microphone to be calibrated remains stable over time, meaning that it responds consistently under identical conditions. Such stability eliminates the need for repeated calibration.



(a) Reference microphone: G.R.A.S. Type 40PL.



(b) Sound calibrator used to calibrate the reference microphone.

Figure 3.19: Reference microphone (a) and sound calibrator (b) used for calibration.

3.2.3.2 Setup of the calibration measurements

For the calibration of the test microphone, measurements were carried out in the anechoic chamber at the VUB campus. This environment ensures controlled acoustic conditions with minimal reflections and a fixed source position. The setup consisted of the uncalibrated ICS-40638 microphone and a calibrated reference microphone as mentioned in section 3.2.3.1.

The frequency response of the type 40PL reference microphone is shown in fig. 3.20. The upper curve represents the free-field response at 0°, while the lower dotted curve shows the pressure response. These curves demonstrate that the microphone exhibits an almost flat response across the majority of the audible frequency range. The response remains within approximately ±0.1 dB up to 10 kHz. This demonstrates the stability and precision of the G.R.A.S. type 40PL, confirming its suitability as a reference microphone for calibration purposes. The exact characteristics of the sound source (the loudspeaker) and its amplifier are not known and therefore cannot be directly taken into account during the calibration procedure. To overcome this limitation, the test and reference microphones were placed side by side and exposed simultaneously to the same acoustic signal.

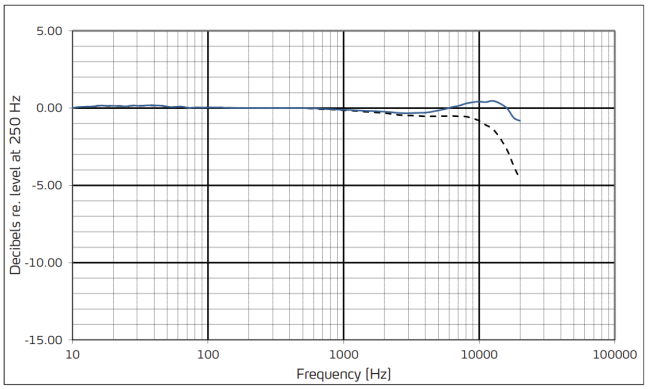


Figure 3.20: The frequency response of the G.R.A.S. Type 40PL microphone (GRAS Sound & Vibration, 2025)

.

Both microphones were connected to an HP 3562 FFT analyzer, which was used to determine the transfer function between the two microphones. The HP 3562A FFT analyzer, as illustrated in fig. 3.21, was selected because direct synchronization of the different acquisition systems for the test and reference microphone was not feasible. More specifically, the internal clock of the Raspberry Pi used for the measurement of the test microphone is not synchronized with the MyDAQ data acquisition system of the reference microphone. The absence of synchronization between the two devices results in a clock mismatch, which leads to timing errors and spectral leakage in the measurements. By employing the HP 3562A, which both generates the excitation signal and records the responses of both microphones simultaneously, the measurements are directly synchronized and free from these artifacts. In addition, the analyzer includes an analog anti-aliasing filter at 100 kHz and a anti-aliasing protected sub-sampling system, ensuring that the system is fully anti-alias protected.



Figure 3.21: The HP 3562A Dynamic Signal Analyzer which was used to define the transfer function during the calibration measurements.

The analyzer was configured to compute the frequency response function over the range of 0-5 kHz and 0-20 kHz using a chirp excitation signal. To improve the signal-to-noise ratio, the measurements were averaged over 32 acquisitions, while autoranging was enabled on both measurement channels to maximize the available dynamic range. Furthermore, both inputs were AC-coupled to remove the DC bias present on the channel that powers the reference microphone.

To determine the quality of the calibration procedure, a verification step was carried out. The spectrum measured by the test microphone was compensated and subsequently compared to the spectrum obtained with the reference microphone. For this verification, the same test signal was

used, since the FFT analyzer can only generate one type of periodic wide band excitation signal. The agreement between the two spectra confirms the validity of the calibration procedure.

3.2.3.3 Time-domain representation of the unprocessed microphone signal

After defining the calibration setup and instrumentation, the first step is to record the raw output of the ICS-40638 microphone using the Raspberry Pi. The signal is captured as a voltage expressed in Volts, as is shown in fig. 3.22. This figure displays an unprocessed 1 kHz sinusoidal signal measured with the ICS-40638. Each measurement has a duration of 20 seconds and was stored as a time series in CSV format with a sampling frequency of 51 200 Hz. This sampling frequency satisfies the Nyquist criterion for the audible frequency range that lies between 20 Hz and 20 000 Hz.

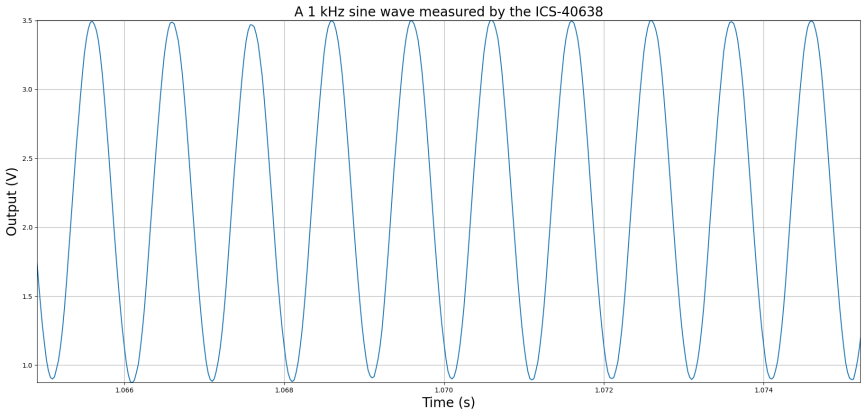


Figure 3.22: Example of an 1 kHz signal measured in the time-domain by the uncalibrated ICS-40638.

The raw data of the ICS-40638, as illustrated in fig. 3.22, not only contains the actual audio signal but also a DC component. This offset originates from the input stage of the acquisition hardware. Since the operational amplifier operates from a single supply of 0-5 V, it cannot process negative voltages. To accommodate the full waveform, the signal is biased around a reference of approximately 2.5 V. As a result, the recorded waveform is shifted upward by a constant DC level of 2.5 V. This value follows theoretically from a resistive divider between 5 V and ground, but in practice, this value may deviate slightly due to component tolerances and circuit conditions, as illustrated in fig. 3.22. To prevent this DC component from interfering with the signal analysis, the mean value of the raw signal is calculated and subtracted from each data point, as expressed in eq. (3.12), with N the number of acquired samples. This correction aligns the waveform around zero.

$$u_{\text{no\_dc}}(t) = u(t) - \frac{1}{N}\sum_{i=1}^{N} u(i) \tag{3.12}$$

After removal of the DC component, the signal is suitable for calibration. The primary objective is to determine the sound pressure level using the Raspberry Pi. To achieve this, the calibration must be applied to the recorded signal, which can be done using two possible approaches. The first approach is to implement the compensation directly in the time domain by designing a digital filter. This method is technically feasible, but it involves filter synthesis, which adds complexity and may lead to additional inaccuracies and was therefore considered to be outside the scope of this thesis. The second approach is to process the signal in the frequency domain, where the calibration can be applied directly to the spectrum. This method is valid provided that the spectrum is determined with minimal leakage, which ensures a reliable compensation.

3.2.3.4 Discrete Fourier Transform

Since the calibration is performed in the frequency domain, the DC-corrected signal must first be transformed using the Discrete Fourier Transform (DFT). As a result, the measured time-domain signal is represented as a spectrum. This provides direct insight into its spectral content in terms of amplitude and phase.

The DFT calculates a spectrum for each measurement. In the amplitude spectrum, all present spectral components are represented by their respective levels in dBV. This signal forms the basis for further processing. This frequency spectrum enables further signal analyses: the comparison of the frequency responses of the test and reference microphones, the detection of peaks at known test frequencies and the determination of the sensitivity curve of the test microphone as a function of the frequency.

3.2.3.5 Mitigating Spectral Leakage: Hanning Window

The accuracy of the Discrete Fourier Transform requires the assumption that the signal is exactly periodic within the chosen measurement window. In practice measured sound signals are generally not perfectly periodic within the measurement window. This results in discontinuities in the periodic repetition of the signal at the boundaries of the sampled window. These discontinuities introduce spectral leakage, causing energy to leak into neighboring frequency components.

To reduce this effect, a Hann window is applied to the time-domain signal prior to the Fourier transform. As shown in fig. 3.23, this window gradually tapers the amplitude of the signal to zero at both the beginning and the end of the measurement, thereby reducing the discontinuities that cause leakage.

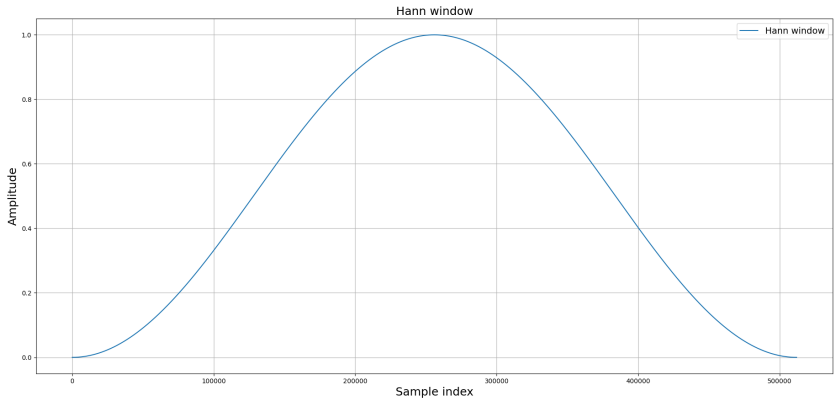


Figure 3.23: Shape of the Hann window function, which smoothly tapers the signal edges.

The Hann window is mathematically defined in eq. (3.13).

$$Window(n) = 0.5\left(1 - \cos\left(\frac{2\pi * n}{N-1}\right)\right) \tag{3.13}$$

The window is applied pointwise by multiplying it with the DC-corrected measured signal, as shown in eq. (3.14). The resulting effect is shown in fig. 3.24, where the original time-domain signal is compared with the windowed signal. The application of the Hann window attenuates the signal edges and reduces discontinuities, thereby minimizing spectral leakage.

$$u_{\text{windowed}}(nT_s) = u_{\text{no\_dc}}(nT_s) \cdot Window(n) \tag{3.14}$$

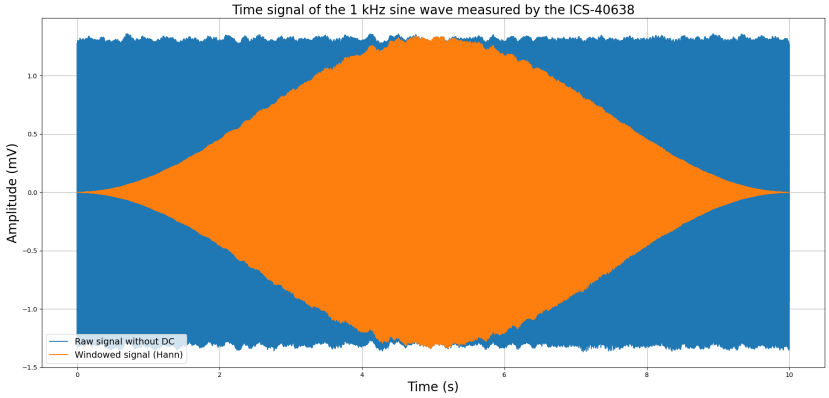


Figure 3.24: Comparison between the windowed time signal and the unprocessed time signal.

3.2.3.6 Normalization of the frequency spectrum

After performing the Fourier Transform, the resulting frequency spectrum represents the amplitudes of the spectral components. However, the raw mathematical FFT output, as implemented in software packages such as Python, cannot be directly interpreted in physical terms. A proper scaling procedure is therefore required to obtain meaningful quantitative values.

The need for scaling arises from the fact that the DFT is a numerical method that distributes the signal energy across a number of frequency bins, making the amplitudes dependent on both the number of samples and the applied windowing function. Without normalization, identical signals analyzed with different sample lengths would give inconsistent amplitudes, which would not correspond to the actual voltage levels or sound pressure levels. To correct for this effect, the spectrum is normalized by dividing it by the total number of samples N. This normalization renders the spectrum independent of signal length and provides a consistent foundation for further analysis.

Figure 3.25 shows the normalized frequency spectrum obtained from the time-domain signal presented in fig. 3.22. The measured signal corresponds to a sinusoidal waveform with a frequency of 1 kHz and a constant amplitude. After applying the Fourier Transform and normalization, the resulting spectrum reveals a peak at 1 kHz, consistent with the excitation signal. The normalization ensures that the amplitude of this spectral component is correctly scaled, allowing consistent comparison.

3.2.3.7 Improving frequency and amplitude estimation accuracy: The Grandke method

When analyzing periodic signals using a Discrete Fourier Transform, the primary objective is to extract specific signal properties such as frequency and amplitude. However, several issues may arise during this process. Two major challenges are spectral leakage and the frequency resolution of the DFT. Spectral leakage occurs when the signal does not contain an integer number of periods within the observation window. This causes part of the energy to spread into neighboring bins. In addition, because the DFT evaluates only a discrete set of frequency bins, the actual frequency may fall in between two bins, resulting in an error estimated of the frequency.

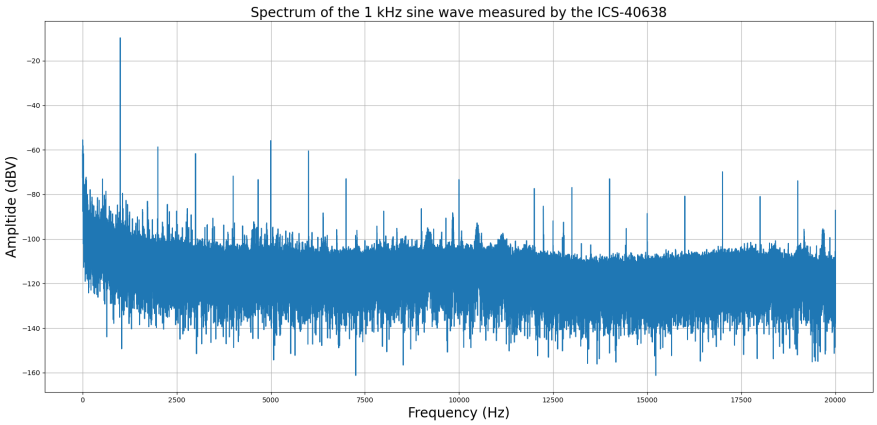


Figure 3.25: Spectrum from the 1 kHz sine wave signal shown in fig. 3.22.

To address these issues, Grandke (1983) proposed a method that suppresses leakage effects and improves the accurate estimation of both frequency and amplitude. The approach is particularly useful for multi-tone signals, such as the test signals used here, where precise identification of individual spectral components is required.

The method begins by applying a Hanning window to the time-domain signal in order to reduce discontinuities at the edges of the signal, as described in section 3.2.3.5. After performing the DFT, the algorithm identifies the local maxima in the tapered spectrum and applies interpolation around those peaks.

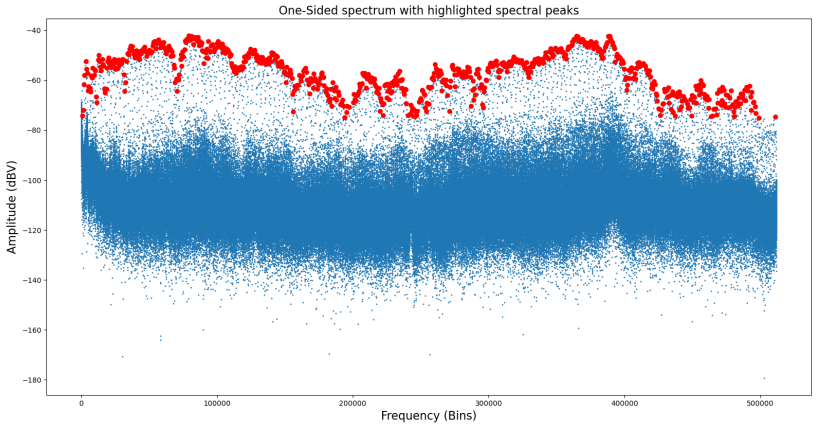


Figure 3.26: Visualization of local maxima in the frequency spectrum, with red points marking dominant frequency components.

Figure 3.26 illustrates the detection of the local maxima in the frequency spectrum, where the red markers highlight the identified peaks, which correspond to the dominant frequency components. Identifying these maxima is essential in Grandke's method, as it enables the interpolation required to determine the true frequency position.

The next step in Grandke's method consists of refining the estimated frequency positions through interpolation. Without this step, the frequencies are only located on the fixed DFT frequency grid, separated by a fixed spacing referred to as the frequency bin. Since the actual spectral components often fall in between these bins, Grandke (1983) introduced an interpolation method that refines the frequency estimate by taking into account the spectral leakage

values around each peak. This process begins by identifying a local maximum in the DFT spectrum, denoted by the index l_m. Then the ratio a, which is defined as the amplitude of the bin at position $l_m + 1$ with the amplitude of the maximum at l_m, as expressed in eq. (3.15).

$$a = \frac{|X(l_m + 1)|}{|X(l_m)|} \tag{3.15}$$

This ratio is then used to determine the interpolation factor x_m, defined in eq. (3.16), which indicates how far the actual frequency is offset from the center of the identified peak bin.

$$x_m = \frac{2a - 1}{a + 1} \tag{3.16}$$

The estimated frequency f_m is obtained using eq. (3.17), where x_m denotes the interpolation factor, l_m represents the index of the spectral maximum and Δf is the frequency resolution of the DFT

$$f_m = (l_m + x_m) \cdot \Delta f, \quad \text{where } 0 < x_m < 1 \tag{3.17}$$

When a frequency component does not align exactly with a DFT bin, errors occur in the estimated frequency as well as in the measured amplitude and phase. This occurs because the tone's energy is distributed across multiple DFT coefficients, which reduces the peak amplitude and distorts the phase. To compensate for this effect, the amplitude A_m is reconstructed through the DFT coefficients surrounding the local maximum. The appropriate interpolation formula depends on the bin containing the highest value. If the peak lies in bin l_m, eq. (3.18) is applied, while a peak located in bin $l_m + 1$ requires the use of eq. (3.19).

$$A_m = \frac{2\pi x_m(1 - x_m)}{\sin(\pi x_m)} \cdot e^{-j\pi x_m(x_m+1)} \cdot X(l_m) \tag{3.18}$$

$$A_m = \frac{2\pi x_m(1 - x_m)}{\sin(\pi x_m)} \cdot e^{-j\pi x_m(x_m-2)} \cdot X(l_m + 1) \tag{3.19}$$

Figure 3.27 illustrates the outcome of this interpolation method. The plot shows a comparison between the original one-side spectrum and the interpolated spectrum obtained with Grandke's method. The interpolation produces a smoother and more accurate representation of both frequency and amplitude estimation.

3.2.3.8 Adjustment of the spectrum based on the transfer function

After applying Grandke's method to the measured signal from the uncalibrated microphone, the resulting spectrum is accurate with respect to both frequency and amplitude. However, the spectrum remains influenced by the characteristic response of the microphone itself. An uncalibrated microphone typically exhibits an unknown, frequency-dependent sensitivity, meaning that certain frequencies are over- or underrepresented relative to their actual levels. As a result, the spectrum obtained after applying Grandke's method is still subject to distortion caused by the microphone characteristics.

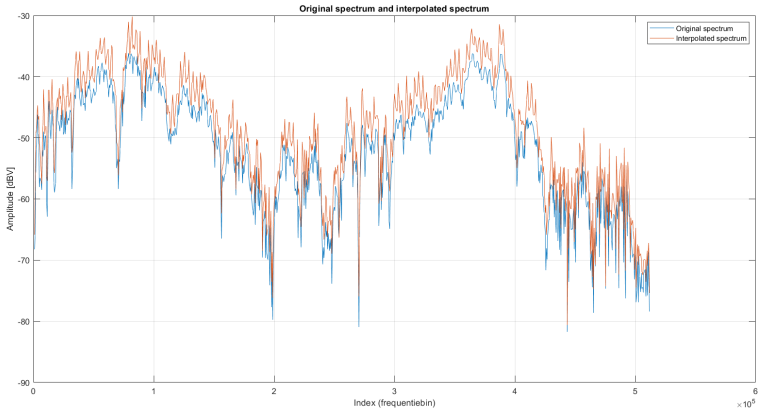


Figure 3.27: Example of an original spectrum (blue) and an interpolated spectrum (orange) after the method of Grandke.

To correct for this, the measured spectrum is adjusted using the transfer function between the uncalibrated test microphone and the reference microphone. This transfer function T(f), describes how the two microphones respond differently when exposed to the same acoustic signal and is defined in eq. (3.20).

$$T(f) = \frac{X_{\text{test}}(f)}{X_{\text{ref}}(f)} \tag{3.20}$$

Where $X_{\text{ref}}(f)$ represents the spectrum measured by the calibrated reference microphone, $X_{\text{test}}(f)$ the spectrum measured from the uncalibrated test microphone. In order to obtain the corrected spectrum, the inverse of the transfer function is applied to the measured signal of the uncalibrated microphone, as described in eq. (3.21).

$$X_{\text{ref}}(f) = \frac{1}{T(f)} * X_{\text{test}}(f) \tag{3.21}$$

As a result, the frequency-dependent deviations of the test microphone are compensated, yielding a more reliable and accurate representation of the acoustic signal. The measured spectrum of the test microphone is therefore adjusted to an equivalent spectrum, representing how the signal would have appeared if it had been captured with the calibrated reference microphone.

The transfer function $T(f)$ was measured with the aid of the HP 3562A Dynamic Signal Analyzer. This instrument measures the frequency response between both the calibrated and uncalibrated microphones subject to the same excitation signal and computes the corresponding transfer function.

The HP 3562A provides the frequency response at 801 discrete frequencies. This implies that the dataset contains noise and is only valid at the specific measurement frequencies. To obtain a smooth behavior over the entire frequency range, it is therefore useful to construct a mathematical model that accurately approximates the measured data. Using the code in listing A.4, the transfer function is represented as a ratio of polynomials in the Laplace variable and estimated from the mean data.

A critical aspect of this process is the selection of the model order. This is the number of poles and zeros in the transfer function. A model with an order that is too low is often not expressive enough and may fail to correctly capture key system characteristics. Conversely, a model with an order that is too high may lead to overfitting, where the model begins to represent not only the true system behavior but also noise and measurement artifacts (Pintelon and Schoukens, 2012).

Based on Pintelon and Schoukens (2012), the maximum likelihood cost function (ML cost) is used to select an appropriate order for both the numerator and denominator of the polynomial. This cost function quantifies how well the model fits the measured transfer function while accounting for measurement uncertainty. The ML cost takes into account the variance at each frequency point: measurement points with lower noise contribute more heavily than those with higher variance. In this way, each data point contributes to the total cost in proportion to its measurement quality. The ML cost is defined as the weighted deviation between the model and the measured values across all frequencies and can be derived using eq. (3.22). A lower value indicates a better model fit, as long as the model does not overfit.

$$\text{ML-cost} = \sum_{f} \left(\frac{\text{Measured}(f) - \text{Model}(f)}{\text{Standard Deviation}(f)} \right)^2 \tag{3.22}$$

To assess whether a model provides a good fit, the measured ML cost is compared to the expected ML cost under ideal conditions. The expected ML cost corresponds to the cost one would obtain if the model perfectly described the system and can be expressed as eq. (3.23), where N equals the amount of measured frequency lines, n_{num} and n_{dem} the orders of the numerator and denominator polynomials.

$$\text{Expected ML-cost} = N - \frac{(n_{\text{num}} + 1) + (n_{\text{den}} + 1) - 1}{2} \tag{3.23}$$

The ratio between the measured and expected ML cost serves as a measure of the remaining model error. Since the ML cost is a quadratic error metric, the square root of this ratio must be taken to interpret the error on the scale of the residuals. This square root indicates how many times larger the total deviation between the model and the measurements is, compared to the deviation caused by noise alone:

$$\text{Ratio} = \sqrt{\frac{\text{Measured ML-cost}}{\text{Expected ML-cost}}} \tag{3.24}$$

Thus, the higher this ratio, the greater the discrepancy between the model and the measured data. This indicates modeling errors when the model does not accurately represent the system. Conversely, the lower the ratio, the smaller the model error and the better the model describes the system.

3.2.3.9 Calibration to an absolute reference level

In the final step of the calibration process, the modeled frequency spectrum of the uncalibrated microphone is converted to an absolute scale. To achieve this, all individual modeled spectra, obtained from repeated measurements of the same acoustic signal, are first averaged. This averaged spectrum is more robust to small measurement variations and thus provides a reliable representation of the measured frequency response.

Subsequently, this averaged spectrum is scaled to an absolute level. For this purpose, individual measurements from the calibrated reference microphone are used. This reference microphone was previously calibrated using the Sound Calibrator at a frequency of 1 kHz and a sound pressure level of 94 dB SPL. The measured value of the calibrated microphone at 1 kHz serves as the reference point. The averaged spectrum of the uncalibrated microphone is then vertically shifted such that its value at 1 kHz matches that of the calibrated microphone. This results in a frequency response expressed on an absolute scale.

4.1 Hardware of the microphone setup

The aim of this project is to record urban noise using a microphone. Since the microphone will be placed outdoors, it must be protected against environmental factors such as rain, wind and heat. At the same time, portability is an important requirement. The case is designed to be lightweight enough for easy transport and practical use, allowing it to be deployed at different measurement locations without difficulty, while still remaining sturdy and robust.

To protect the Raspberry Pi and the associated electronics, a protective case was designed that is shown in fig. 4.1. Ideally, this case would be 3D-printed in ASA due to the material's excellent UV-resistance and high heat and water resistance. However, since ASA filament was not available at the time, PLA was chosen for the initial outdoor tests. In the future, this will be replaced by ASA.

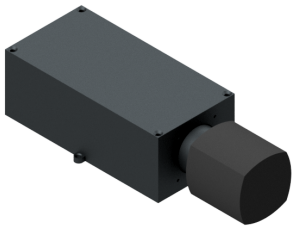


Figure 4.1: The printed box in which the microphone is placed to conduct its measurements.

Inside the case, the Raspberry Pi and the DAQ are securely mounted using screw holes at the bottom of the case. The PCB of the microphone is placed inside the protruding tube and is covered by a windscreen. This windscreen reduces wind noise and ensures that the recorded signals are not disturbed, making the measurements more reliable.

Since our project is powered by a solar panel and a battery, both the battery and the solar charge controller must also be protected against environmental influences. For this purpose, a separate case was designed, as shown in fig. 4.2. This case was initially 3D-printed in PLA but would also ideally be produced in ASA.

On top of the case, two push buttons were added to start and stop the measurements, along with a power switch to activate or deactivate the entire measurement setup. Mounting holes were included on the sides of the case, allowing it to be securely fixed to the plate on which the solar panel is mounted. This ensures a stable and robust configuration.

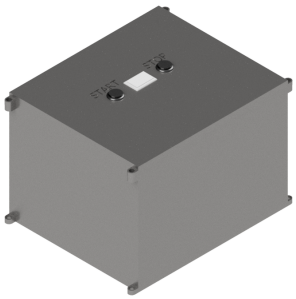


Figure 4.2: The printed case in which the battery and the solar charge controller are placed. An ON/OFF-switch and 2 buttons are placed on the case to start or end the measurements.

Table 4.1 provides an overview of the total expenses required to assemble the low-cost measurement setup. The table lists all individual hardware components, ranging from the data acquisition system and the microphone to the solar power supply system and mechanical housing. The total investment amounts to approximately €475. This demonstrates that the proposed setup can be realized at a fraction of the cost of reliable commercial sound level meters. Such professional devices typically cost several thousands of euros, highlighting the cost-effectiveness of the proposed setup.

The table clearly shows that the most significant costs are associated with the data acquisition system and the Raspberry Pi. Other key components, such as the solar panel, battery and charge controller, are relatively affordable. The remaining parts contribute only marginally to the overall budget, while still being crucial for the system's robustness and portability. This detailed cost overview highlights the cost-effectiveness of the setup, confirming that reliable acoustic measurements can be achieved without the need for expensive professional equipment.

Table 4.1: Overview of the costs of the components used in the measurement setup.

Object	**Unit Price**
Raspberry Pi Model 4B - 2GB Ram	€ 60.49
MCC128 DAQ HAT	€ 171
ICS-40638 MEMS Microphone	€ 2.32
PCB + Electronic Components	€ 15
USB 4G Dongle	€ 36.99
Solar Panel	€ 32.95
12V 12Ah LiFePO4 Battery	€ 44.90
Solar Charge Controller	€ 29.99
Solar panel cables 3 meter	€ 12.67
Windscreen	€ 8.49
Solar Panel Aluminum holder	€ 24.95
Prepaid SIM Card	€ 15
Generic PLA Filament	€ 17.50
Wooden base plate	€ 2.50
TOTAL COST	**€ 474.75**

5.1 Results of the calibration process

5.1.1 First calibration process

5.1.1.1 Setup for the first calibration

The initial calibration measurements were conducted at the Von Karman Institute. The experimental setup consisted of an electronic signal generator, an audio attenuator to control the sound level and a loudspeaker equipped with a conical horn to direct the acoustic signal. The reference microphone and the test microphone were placed side by side underneath the conical horn to ensure that both were exposed to the acoustic field as identically as possible. Since the measurements were carried out in a standard room rather than in an anechoic chamber, the results are inherently susceptible to reflections and standing waves.

Measurements were performed using three different types of acoustic signals: broadband noise, pure sine waves at various frequencies and frequency sweep signals. For each measurement, both the attenuation level and the duration of the signal were varied.

5.1.1.2 Results of the first calibration process

In fig. 5.1, a comparison is made between the time domain signals recorded by an uncalibrated and a calibrated microphone. Both signals were captured during a sine wave input signal at 2 kHz with an attenuation setting of 32.

The signal from the uncalibrated microphone exhibits a clear DC offset, with an average voltage of approximately 2.27 V. The AC component of the signal, which carries the actual acoustic information, shows only a limited amplitude of around 50 mV relative to this DC offset. Due to the small magnitude of this AC component, the signal-to-noise ratio is also low. The signal is insufficiently amplified to allow for reliable acoustic analysis.

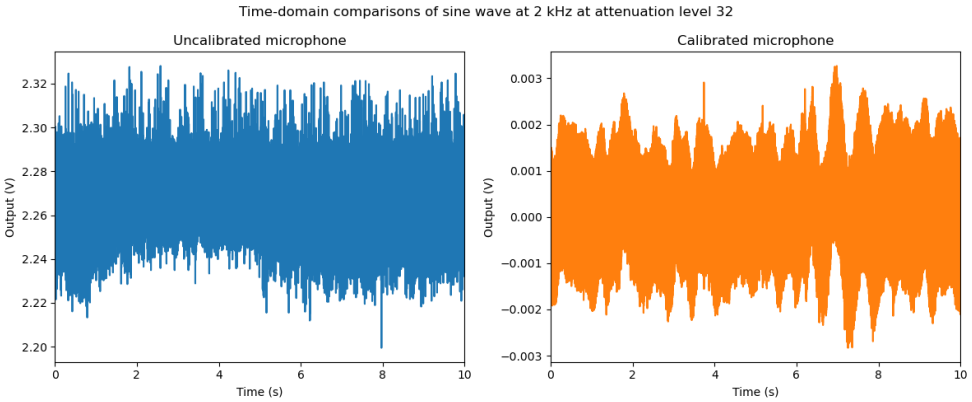


Figure 5.1: A 2 kHz sine wave signal in the time domain recorded by both microphones.

In fig. 5.2, the frequency spectrum of a noise signal is shown. The spectrum recorded by the uncalibrated microphone appears relatively flat, with energy distributed across the entire frequency band. This is the expected result when a noise signal is played.

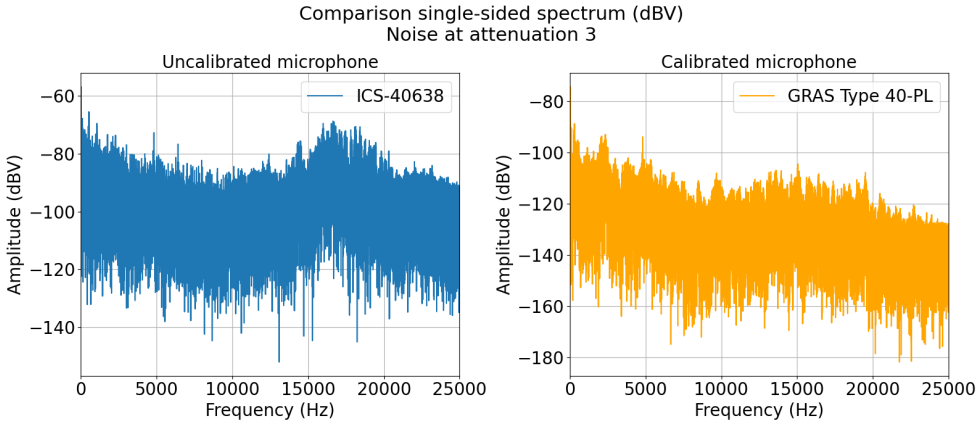


Figure 5.2: Frequency spectrum of both microphones. The excitation signal was a noise signal at attenuation level 3.

In fig. 5.3, the frequency spectrum of a sine wave signal at 2 kHz with an attenuation level of 32 obtained by the scaled DFT, is shown. In the uncalibrated spectrum, a prominent peak is visible at 2 kHz, with an amplitude exceeding −40 dB. The surrounding noise floor is low, reaching levels up to −80 dB. Several additional peaks are present, likely caused by harmonic distortion or artifacts introduced by the microphone itself.

The calibrated spectrum also displays a clear peak at 2 kHz on the same amount of samples, but with a lower amplitude of approximately −50 dB. The background noise level is significantly reduced. The spectral distribution is flatter, with fewer pronounced fluctuations outside the fundamental component. Nonetheless, a few harmonic peaks remain visible.

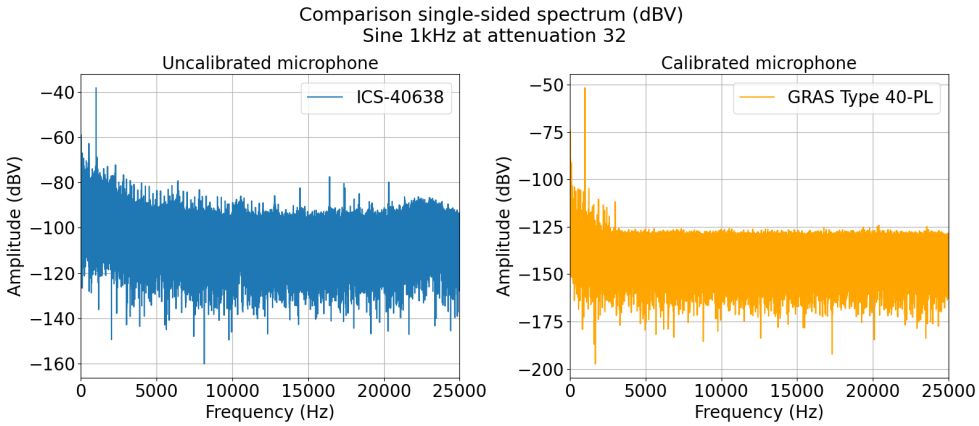


Figure 5.3: Frequency spectrum of both microphones. The excitation signal was a sine wave signal with a frequency of 1 kHz at attenuation 32.

Figure 5.4 shows the frequency spectrum of the microphones in response to a sweep signal that was logarithmically varied from 20 Hz to 10 kHz. The signal was attenuated with an attenuation setting of 39. In the spectrum of the uncalibrated microphone, a clearly distorted amplitude profile is observed. The energy distribution is inconsistent with the expected pattern of a sweep signal. Notably, the spectrum continues to show energy beyond 10 kHz even though the original sweep signal ends at that frequency. A part of this may come due to the fact

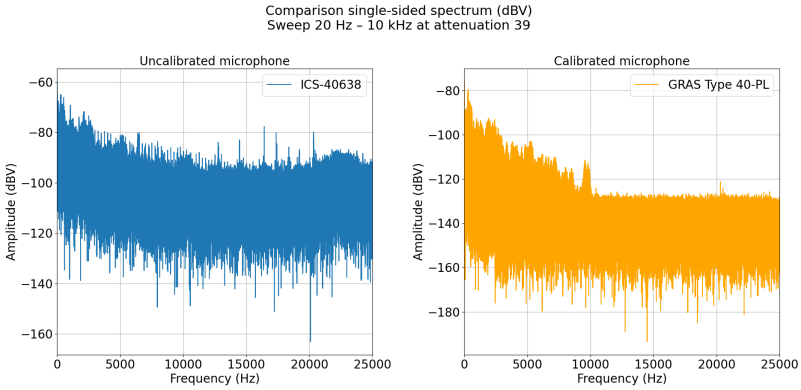


Figure 5.4: Frequency spectrum of both microphones. The excitation signal was a chirp signal between 20 Hz and 10 kHz at attenuation level 39.

that in the measurement setup the test microphone wasn't directed towards the sound source. However, the residual energy above the sweep's upper limit is more plausibly due to spectral leakage. In contrast, the calibrated microphone spectrum is much more consistent with the characteristics of the sweep signal: beyond 10 kHz, the spectrum levels off as expected.

Figure 5.5 shows the root mean square (RMS) output voltages of both microphones as a function of the attenuation level. A lower attenuation level corresponds to a louder acoustic signal. The sound used for these measurements was a sine wave with a frequency of 2 kHz.

It is observed that the RMS values of the uncalibrated microphone remain nearly constant across all attenuation levels. The variation is minimal, approximately 0.3 mV, indicating that the microphone output responds only weakly to changes in the applied sound pressure. This suggests that the microphone exhibits nonlinear behavior in this context. In contrast, the calibrated reference microphone shows a clear decreasing trend in RMS output voltage with increasing attenuation. The values decrease from approximately 0.61 mV at low attenuation to around 0.38 mV at the highest attenuation level. This decrease is consistent with the expected behavior of a linear microphone: lower acoustic energy results in a proportionally lower electrical output.

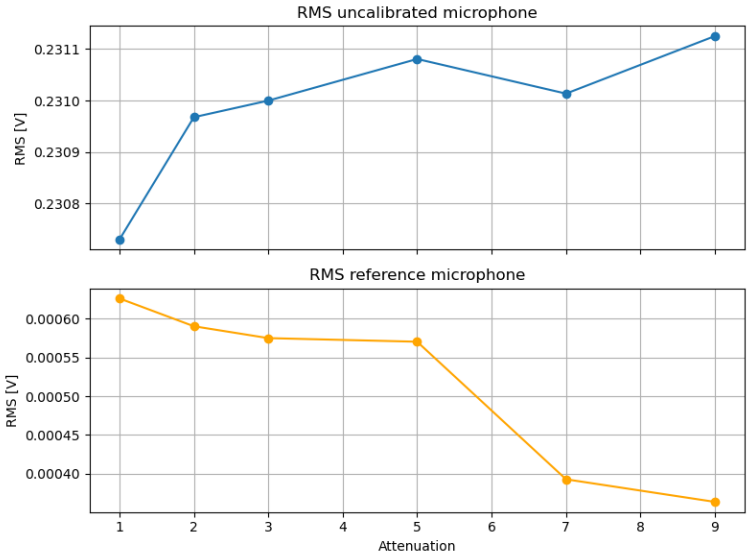


Figure 5.5: RMS values of the uncalibrated and the calibrated microphones at different attenuation levels. The lower the level of attenuation, the louder the sound is.

5.1.1.3 Conclusion after the first calibration

The measurement results imply that the electrical output signal of the test microphone is too small to allow for an accurate processing of the data. Additional amplification or preamplification of the test microphone's output is required to improve the resolution and linearity of the measurements.

5.1.2 Second calibration measurements

5.1.2.1 Setup of the measurements

The second series of calibration measurements was conducted in the anechoic chamber on the VUB campus. This environment was specifically chosen to minimize the influence of standing waves, reflections and external environmental factors on the measurements. Unlike the first set of measurements, no external signal generator was used. Instead, the excitation signal was generated by the HP 3562A Dynamic Signal Analyzer.

This measurement setup was selected to address a key practical issue: the test microphone and the reference microphone each rely on separate data acquisition system. As a consequence, each system operates with an independent clock. As a result, there is no guarantee of time synchronization between the signals when measured separately. By using the HP 3562A, the system simultaneously measures the frequency response of both microphones via synchronized inputs. This enables a reliable determination of the frequency-dependent transfer function between the two microphones, free from timing errors caused by clock drift or desynchronization.

5.1.2.2 Results of the second calibration process

A total of 11 measurements were carried out during this second calibration procedure. Each measurement was performed using a chirp excitation signal. This type of periodic signal was chosen because it covers a wide frequency range within a single measurement. This signal allows for an error-free representation of the frequency response across the entire audio frequency range. For these measurements, the gain of the test microphone's output signal was increased to a value of 50. This adjustment is clearly visible in the time domain signal shown in fig. 5.6. Compared to the signal obtained during the first calibration measurements, a significantly stronger output signal can be observed.

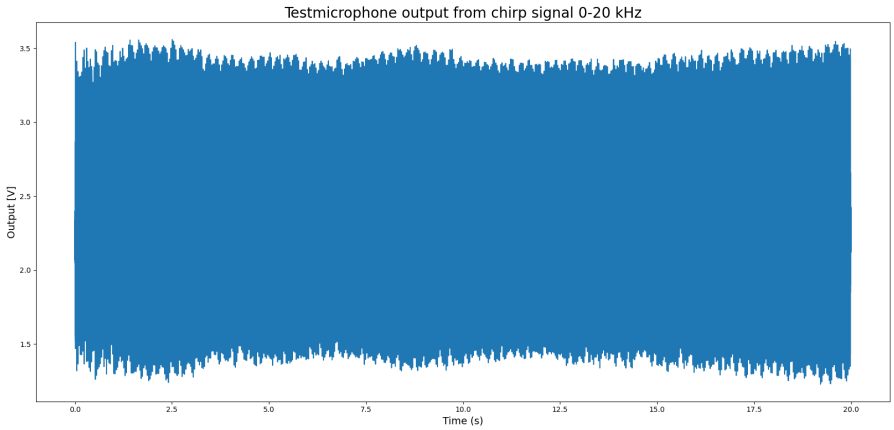


Figure 5.6: A chirp signal from 0-20 kHz in the time domain measured by the test microphone.

Figure 5.7 shows the frequency response of the test microphone for the same signal as the one shown in fig. 5.6 after applying a Hann window. It is clearly visible that the frequency response of the test microphone is not perfectly flat. This suggests that the microphone exhibits frequency-dependent behavior. This makes the microphone unsuitable for accurate measurements without a proper correction, called the calibration.

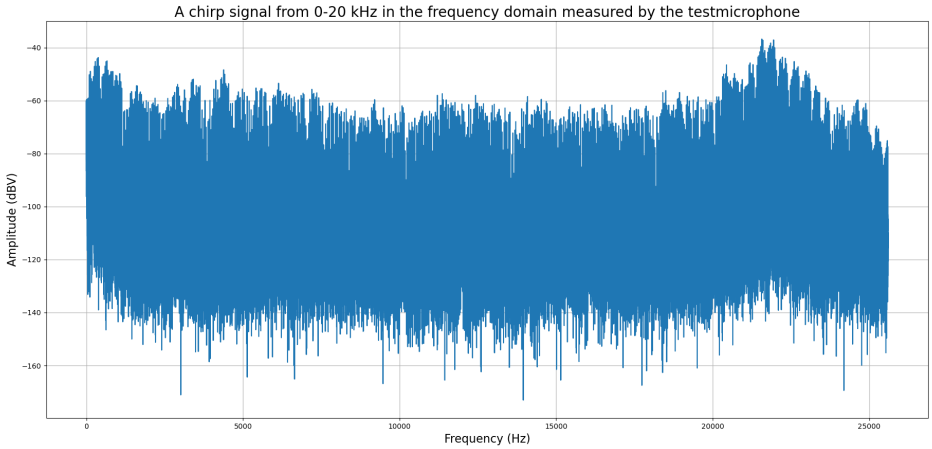


Figure 5.7: A chirp signal from 0-20 kHz in the frequency domain measured by the test microphone.

Before the measured spectrum can be corrected by multiplying it with the transfer function, it must first be more accurately determined using the method of Grandke as the signal is periodic, the spectrum is discrete. This is explained in section 3.2.3.7. To apply this method, it is necessary to extract the local maxima from the frequency spectrum along with their corresponding index numbers, as shown in fig. 5.8.

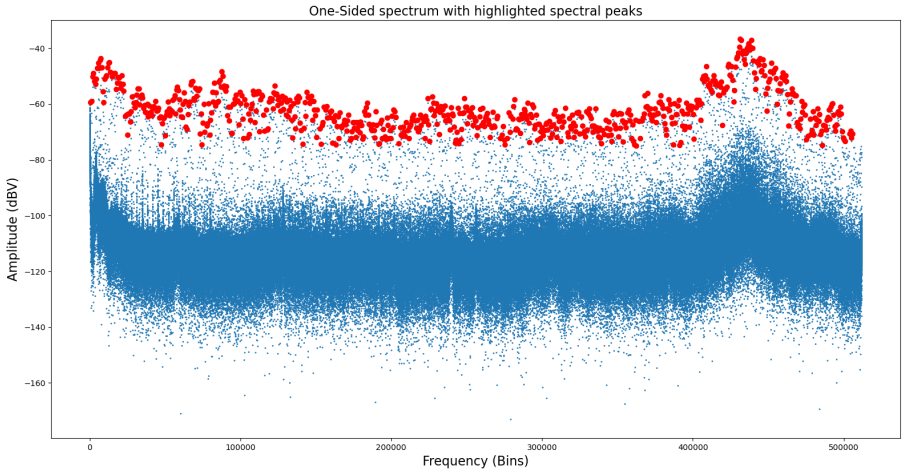


Figure 5.8: One-Sided Spectrum with Highlighted Spectral Peaks

Once the local spectral maxima and their corresponding indices are known, the Grandke method can be applied to correct the measured spectrum to obtain the exact spectrum. Figure 5.9 shows the interpolated spectrum obtained after applying the Grandke method. The original spectrum is clearly recognizable within the interpolated version. The interpolated spectrum appears to be slightly vertically shifted relative to the original spectrum.

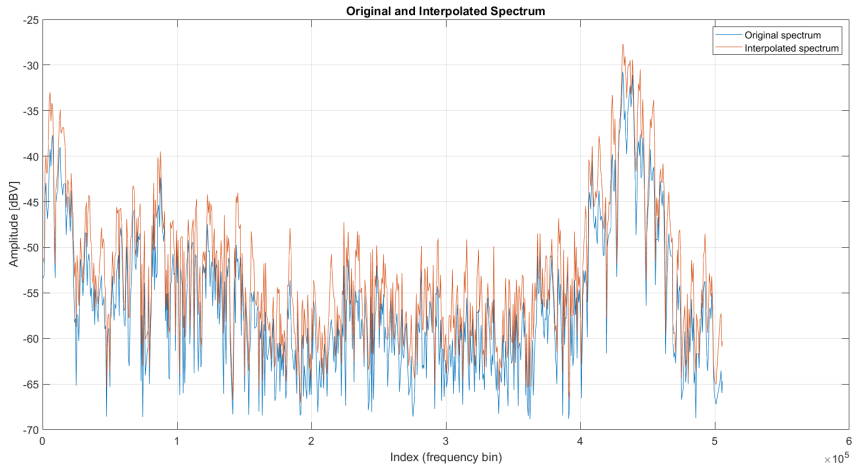


Figure 5.9: Original spectrum of the measurement of the test microphone versus the interpolated spectrum after the Grandke method.

Figure 5.10 shows the double-sided spectrum as a verification step to check whether the spectrum remains symmetric. The analysis continues using only the single-sided spectrum as the negative part is redundant for real-valued signals.

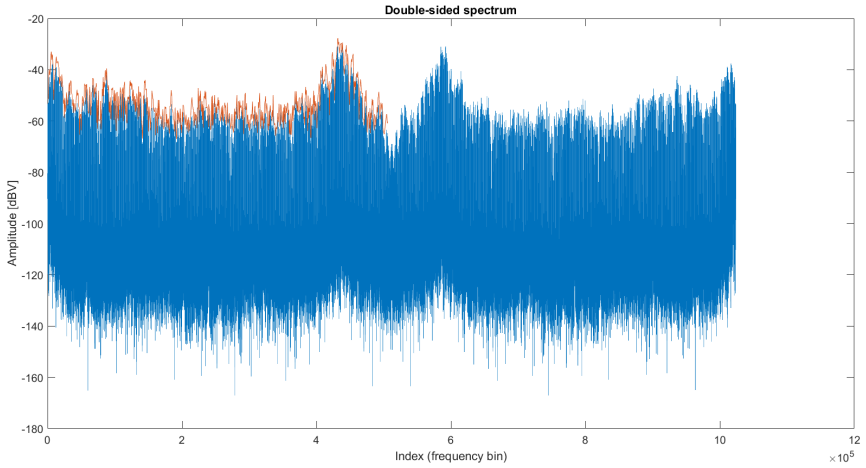


Figure 5.10: Original double-sided spectrum with the interpolated spectrum on the positive side.

Before the spectrum can be corrected by multiplying it with the transfer function between the two microphones, a model of this transfer function must first be created. Figure 5.11 shows the estimated model of the transfer function. It can be observed that the modeled transfer function closely follows the behavior of the measurements. The selected model order represents an appropriate compromise between accuracy and complexity.

In the frequency range below 1.5 kHz, it is noticeable that the transfer function does not exhibit the same shape as at higher frequencies. In this low-frequency region, the measurement uncertainty is higher and the deviation between the measured and modeled transfer function is greater. This observation is confirmed when directly examining the measurements from the HP 3562A, as shown in fig. 5.12.

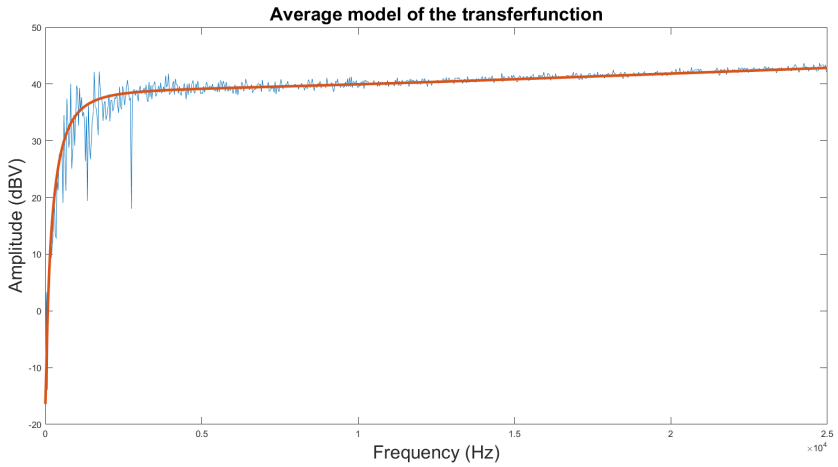


Figure 5.11: The estimated model of the transfer function.

The values in the low-frequency region reach amplitudes up to 30 dB higher than the rest of the model. Referring to the transfer function formula given in section 3.2.3.8:

$$T(f) = \frac{X_{\text{test}}(f)}{X_{\text{ref}}(f)} \tag{5.1}$$

this implies that the test microphone exhibits a lower response at low frequencies compared to the reference microphone.

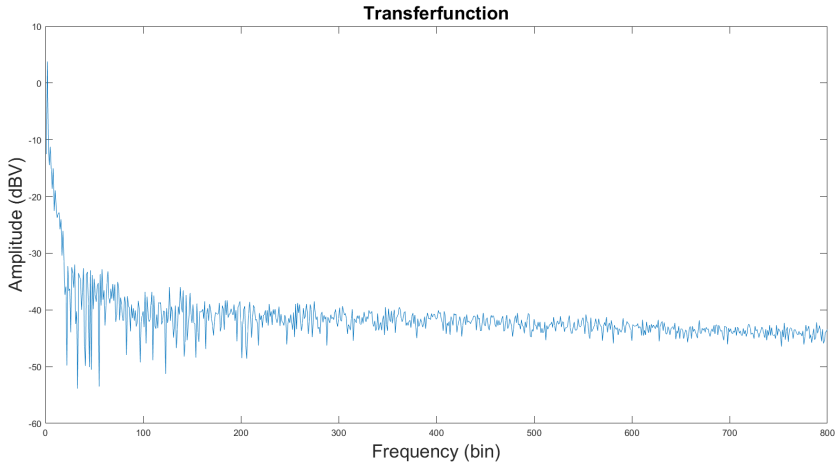


Figure 5.12: The inversed unprocessed transfer function. This graph has been inversed to get a more clear view of the lower frequency region.

5.1.2.3 Conclusion of the second calibration process

The second calibration measurements confirm that the improved measurement setup and the increased amplification of the test microphone result in a more reliable and more usable output signal. However the results also show that the test microphone, when compared to the reference microphone, exhibits a less accurate frequency response in the low-frequency range. As a consequence, it is not possible to correctly adjust the spectrum to the reference value at 1 kHz. This prevents absolute calibration. Additional measurements are therefore required to achieve a complete and accurate calibration (GRAS Sound & Vibration, 2025).

5.1.3 Final calibration measurements

5.1.3.1 Setup of the final measurements

The final calibration measurements were conducted at the VUB campus in an enclosed room, outside an anechoic chamber. The anechoic chamber was unavailable at the time due to occupancy and it was not possible to perform these measurements in that facility at a later date. As a result, potential disturbances caused by unwanted reflections must be taken into account when interpreting the results.

The measurement setup was largely identical to that of the second calibration measurements (section 5.1.2.1), with the main addition being the use of a separate DAQ system for the reference microphone. This reference microphone is of the IEPE (Integrated Electronics Piezo-Electric) type and requires a dedicated power supply for proper operation.

5.1.3.2 Results of the final calibration process

A total of 10 measurements was performed: 5 chirp signals from 0–5 kHz and 5 chirp signals from 0–20 kHz. The 0–5 kHz measurements primarily target the low- and mid-frequency range. The 0–20 kHz measurements cover the entire audible spectrum, including the high frequencies where a microphone's response often shows significant variation.

Figure 5.13 shows the output signal of ICS-40638 test microphone in the time domain for a 0–5 kHz chirp signal. The signal amplitude is sufficiently high, which aids in verifying whether a measurement was successful. For the subsequent steps, this excitation signal will be used.

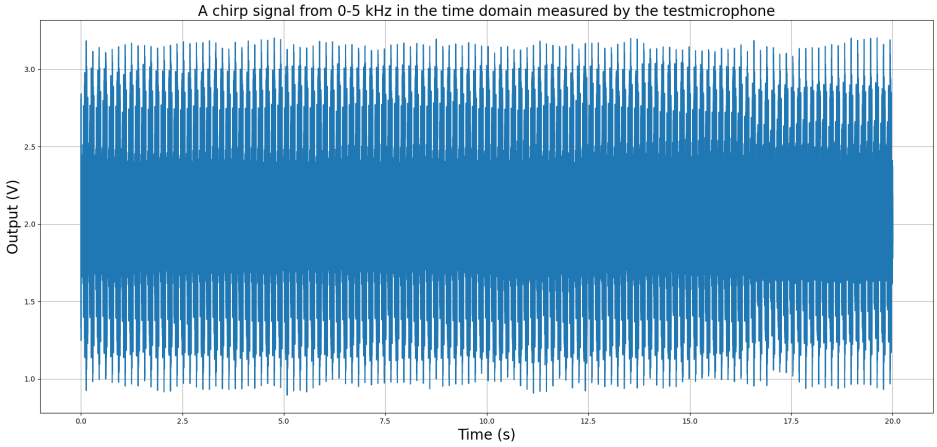


Figure 5.13: An unprocessed chirp signal from 0-5 kHz in the time domain measured by the test microphone.

To apply the calibration process, the measured spectrum must first be compensated for leakage using the Grandke method as previously done in section 5.1.2. For this purpose, all local spectral maxima are required, as shown in fig. 5.14. The refined interpolated spectrum is presented in fig. 5.15.We notice a systematic offset between spectra, this is caused by an over-correction of the Hann window attenuation due to a double factor of 2, which was subsequently corrected in later processing. The double-sided spectrum is shown in fig. 5.16.

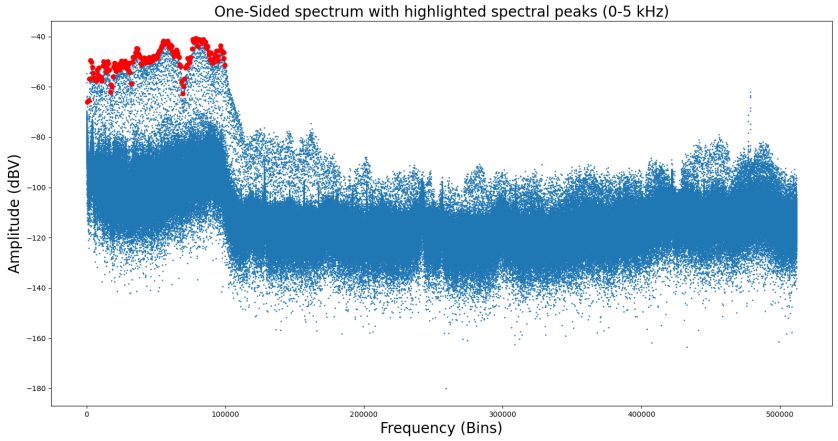


Figure 5.14: The local spectral maxima in the one-sided spectrum. These maxima are used to correct the spectrum using Grandke's method.

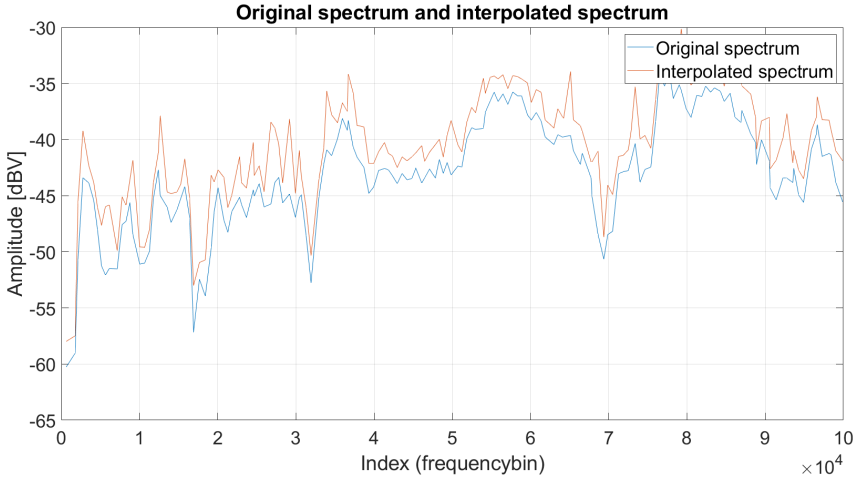


Figure 5.15: Original spectrum of the measurement of the test microphone versus the interpolated spectrum of the same microphone after the Grandke method.

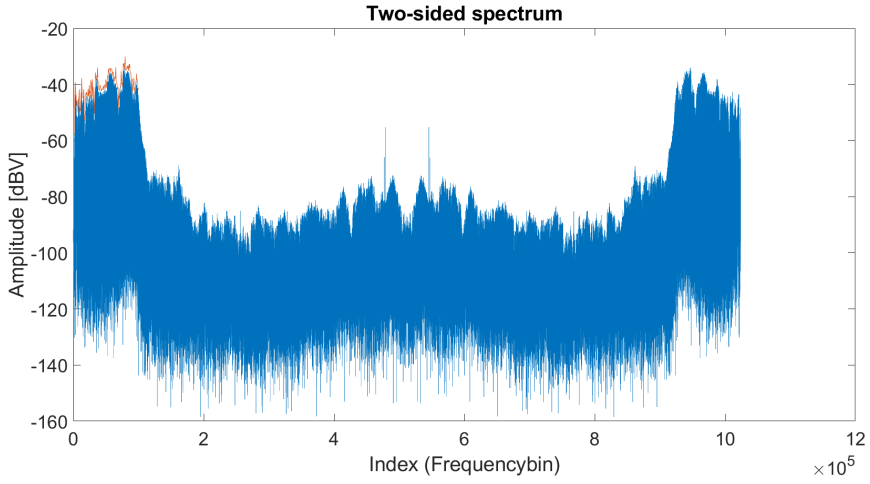


Figure 5.16: Original double-sided spectrum with the interpolated spectrum until 5 kHz on the positive side.

To calibrate the refined interpolated spectrum, it must be multiplied with the inverse transfer function between the reference microphone and the test microphone. Figure 5.17 shows the transfer function model based on the measurements from the HP 3562A. Overall, there is good agreement between the two curves, indicating that the model is capable of accurately representing the frequency response characteristics of the system. A noticeable dip appears in the model at a frequency of 3.5 kHz, which is also present in the measurements themselves. The dip observed around 3.5 kHz represents an anti-resonance in the measured transfer function. Although the exact origin cannot be determined with certainty, it is most likely related to the measurement setup. Therefore, this feature is considered an artifact of the setup rather than a true characteristic of the device under test. The measurement uncertainty in the low-frequency range is relatively small, indicating a high signal-to-noise ratio and stable measurement conditions. As the frequency increases, the uncertainty becomes more significant.

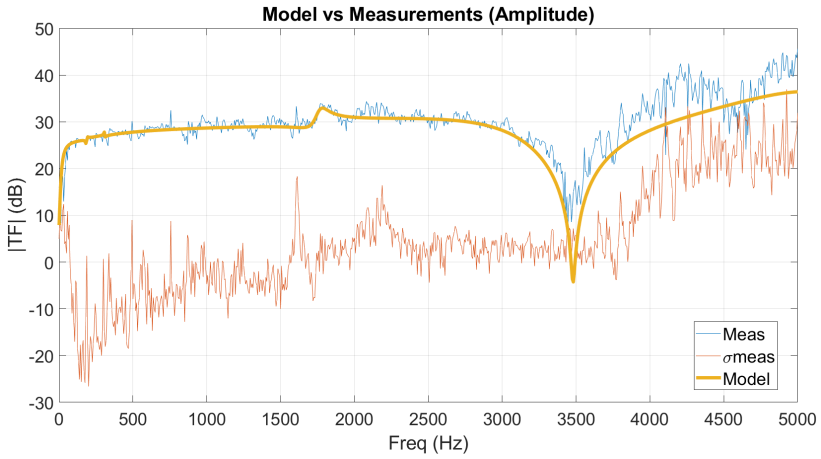


Figure 5.17: Measured transfer function amplitude compared with the analytical model, including measurement uncertainty.

Figure 5.18 shows the coherence of the measurements. It indicates to what extent the signal of the test microphone can be explained by the signal of the reference microphone, thereby providing an assessment of the reliability of the transfer function. The values always fall between 0 and 1, with 1 meaning that both signals fully correspond to each other and that the transfer function is reliable in that frequency region.

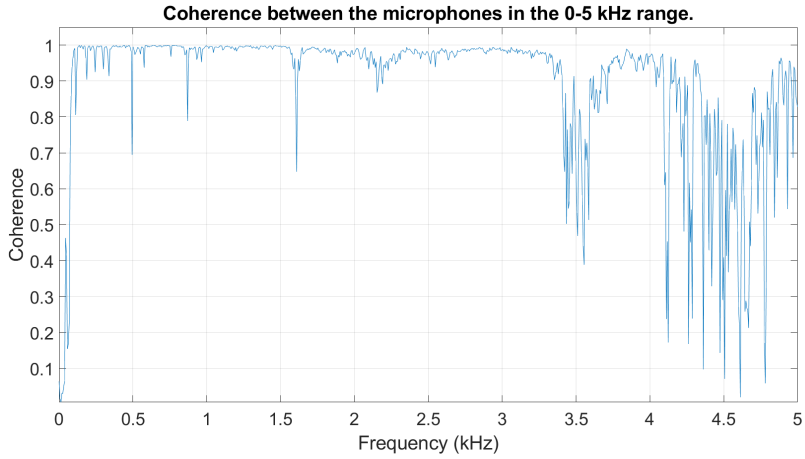


Figure 5.18: The coherence between the test microphone and the reference microphone. This plot shows a good coherence between 500-3000 Hz, but a lower coherence at the other frequencies.

From the coherence plot it can be observed that the coherence remains high in the range from approximately 0.1 to 3.5 kHz. This means that the transfer function can be interpreted with confidence in this frequency region. Around 3.5 kHz the coherence drops sharply, coinciding with the anti-resonance observed in the transfer function. This indicates that the measurement in this region is less reliable and most likely affected by an artifact of the measurement setup, such as interference or reflections. Above 4 kHz the coherence decreases further and becomes highly irregular, showing that the signal-to-noise ratio is insufficient there and the results are less useful.

When we examine the phase plot of the measurements compared to the transfer function model in fig. 5.19, we observe good agreement between the model and the measurements. Beyond a frequency of 3.5 kHz, the measurement phase exhibits large jumps of 360°, or 2π radians due to phase wrapping. The phase in this region becomes more and more random, due to the low signal-to-noise ratio. At 3.5 kHz, an abrupt phase shift of 180° is observed. This corresponds to the amplitude dip shown in fig. 5.17. This shift is characteristic of a transmission zero and indicates destructive interference, in which the output signal is significantly attenuated.

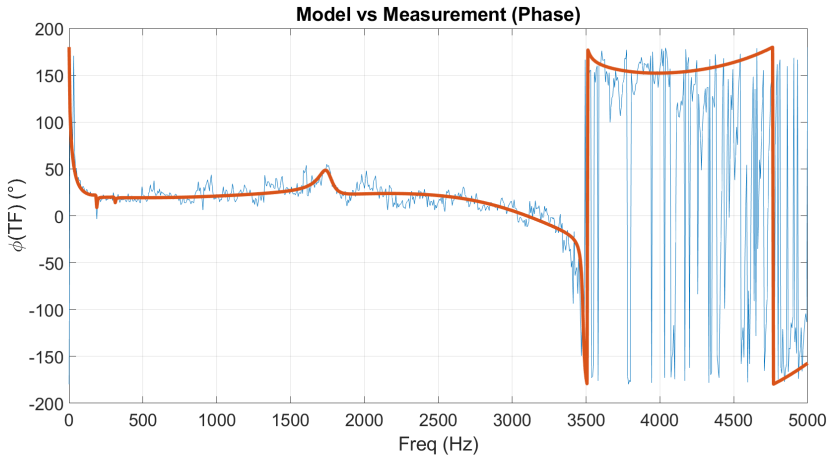


Figure 5.19: Measured transfer function phase compared with the analytical model as a function of frequency.

To finally assess how well this model describes the measured transfer function, the ML-cost function of the model is examined. The ML-cost function evaluates how well the model represents the measured data by computing a probabilistic measure of the differences between the model and the measurements. The expected ML-cost value, the theoretical value if the model was perfect, at a chosen order of 10 for the numerator and 14 for the denominator is:

$$\text{Expected ML-cost} = 801 - \frac{(10+1)+(14+1)-1}{2} = 788.5 \tag{5.2}$$

We obtain an actual ML-cost with a value of 26,220.9. At first sight, this appears to be an extremely high value compared to the expected cost. However, since the ML-cost is a quadratic error metric, the ratio between the two is:

$$\text{Ratio} = \sqrt{\frac{26220.9}{788.5}} \approx 5.77 \tag{5.3}$$

This means that the total deviation between the model and the measurements is approximately 5.8 times greater than the error that would be caused solely by measurement noise if the model were perfect. Given that a low-cost sensor is used and the measurements were not conducted in an anechoic chamber, this error factor is considered acceptable for further calibration.

Figure 5.20 shows the frequency-dependent residuals that collectively determine the ML-cost. Ideally, these residuals should average around one, since a well-fitting model would deviate only within the limits of the measurement noise. In this case the values are higher, with peaks exceeding twenty. The residuals are particularly large in the low-frequency region. Pronounced peaks are observed below approximately 1 kHz. This suggests that in this domain the model does not correctly capture the system behavior. As the frequency increases, the residuals decrease and stabilize at values between three and six. The graph therefore clearly indicates that the main contribution to the high ML-cost originates from the low-frequency region.

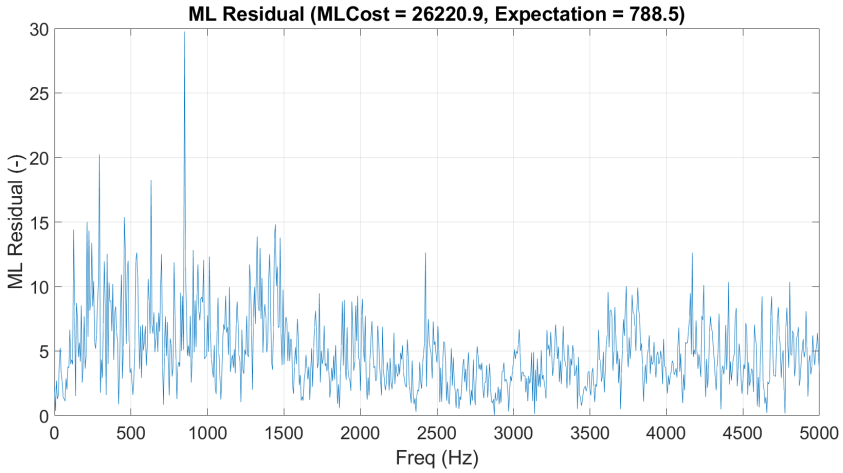


Figure 5.20: Maximum likelihood residuals as a function of frequency.

Now that a reliable model of the transfer function has been established, the response of the test microphone can be corrected by multiplying the measured spectrum with the inverse of this transfer function. In this way, the frequency spectrum of the test microphone is adjusted to align with that of the reference microphone. This correction compensates for system deviations and enables the measurements of the test microphone to be interpreted in an equivalent and reliable manner. The 30 dB shift between the original and corrected spectra, as shown in fig. 5.21, is a direct reflection of the 30 dB amplitude difference described by the transfer function model.

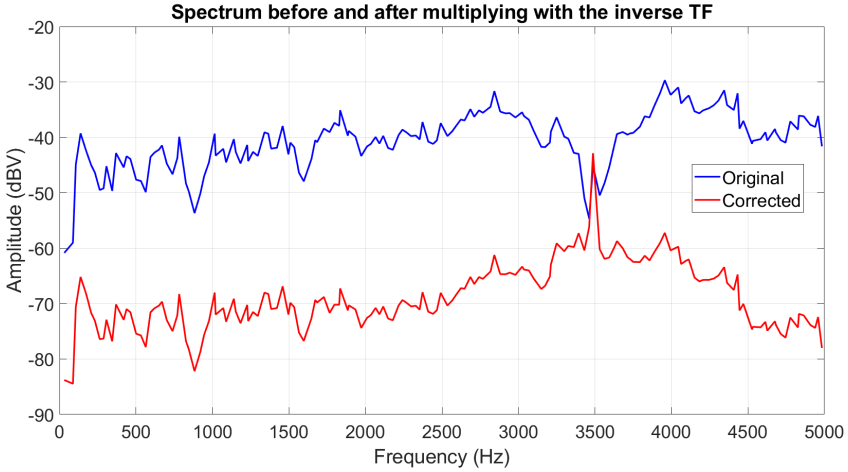


Figure 5.21: The measured spectrum from the test microphone compared to the corrected spectrum. The corrected spectrum is the original spectrum multiplied with the inverse transfer function.

This correction procedure was carried out for a single measurement but must be systematically repeated for all measurements performed with the same excitation signal. From the collection of corrected spectra, the average spectrum can then be computed. This average spectrum serves as the representative input for the subsequent steps of the calibration process. In fig. 5.22, this average spectrum is shown together with the corresponding standard deviation. The resulting spectrum can be regarded as the equivalent of the average spectrum of the reference microphone during the measurements with a sweep signal from 0–5 kHz.

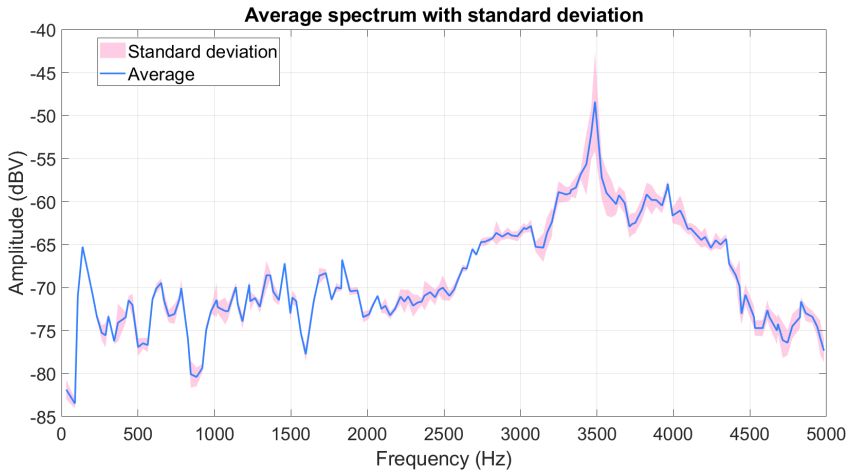


Figure 5.22: Average corrected spectrum of the test microphone together with the standard deviation.

Prior to the absolute calibration of the test microphone, the response of the reference microphone was characterized using a sound calibrator. This device generates a pure sinusoidal tone at 1 kHz with a fixed sound pressure level of 94 dB SPL, corresponding to an effective pressure of 1 Pa. A total of eight measurements were performed with the reference microphone placed in the calibrator. The average output level, expressed in dBV, together with the standard deviation, is shown in fig. 5.23. These measurements indicate that the reference microphone has a -38 dBV output at a sine wave sound of 94 dB SPL at 1 kHz.

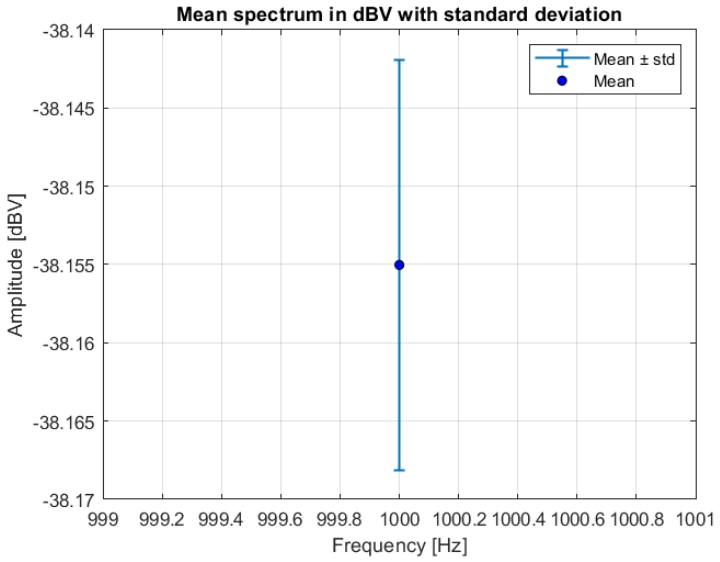


Figure 5.23: Mean spectral amplitude of the test microphone when it has been placed in the Sound Calibrator.

Figure 5.24 shows the difference between the corrected spectrum of the test microphone and the spectral value of the reference microphone at 1 kHz obtained with the sound calibrator. A clear discrepancy can be observed between the spectral values of both microphones at this frequency. To achieve absolute calibration of the test microphone, the spectrum must be vertically shifted so that the values of both spectra are equal at 1 kHz.

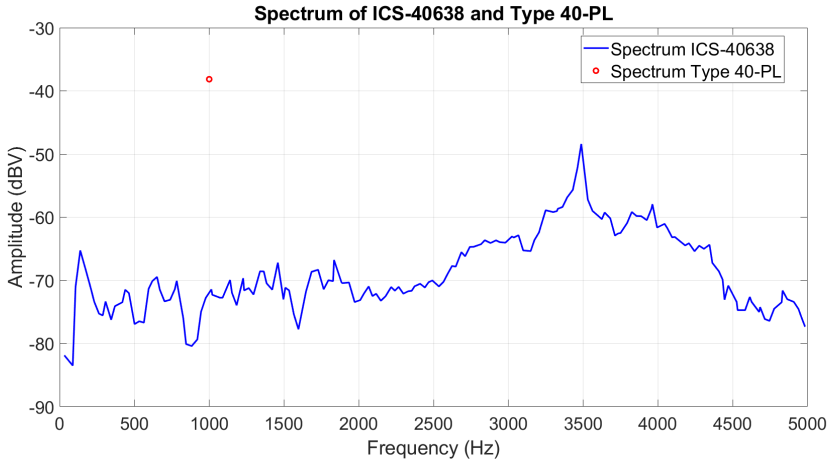


Figure 5.24: The corrected spectrum of the ICS-40638 and the spectral value at 1 kHz of the Type 40PL.

Figure 5.25 shows the vertically shifted spectrum of the ICS-40638. The spectrum has been shifted by 33.2 dB. By first correcting the spectrum of the test microphone using the transfer function, a relative calibration of the microphone is obtained. Since this result is then compared with a fixed reference point from the sound calibrator, it can be concluded that the microphone is now absolutely calibrated.

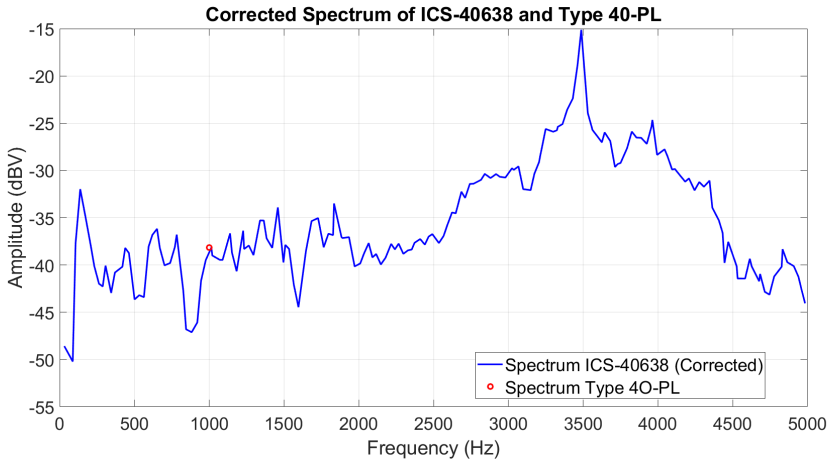


Figure 5.25: The corrected spectrum of the ICS-40638 vertically shifted to equal the spectral value of the Type 40PL at 1 kHz.

When examining the measurements obtained from the 0–20 kHz sweep signal in fig. 5.26, the transfer function exhibits numerous fluctuations. Several anti-resonance dips are clearly visible in the measurements and are also reproduced by the model, indicating that these features are part of the physical frequency response rather than measurement noise. The coherence plot in fig. 5.27 supports this interpretation, as it reveals significant variations in coherence across the frequency range. In particular, between 4 and 16 kHz the coherence drops substantially and fluctuates strongly. This suggests that the reference and test microphones do not correlate

well in this region. Consequently, the reliability of the measurements above 4 kHz is limited. For this reason, the analysis is restricted to the frequency range up to 5 kHz, where both the transfer function and coherence demonstrate consistent and trustworthy behavior.

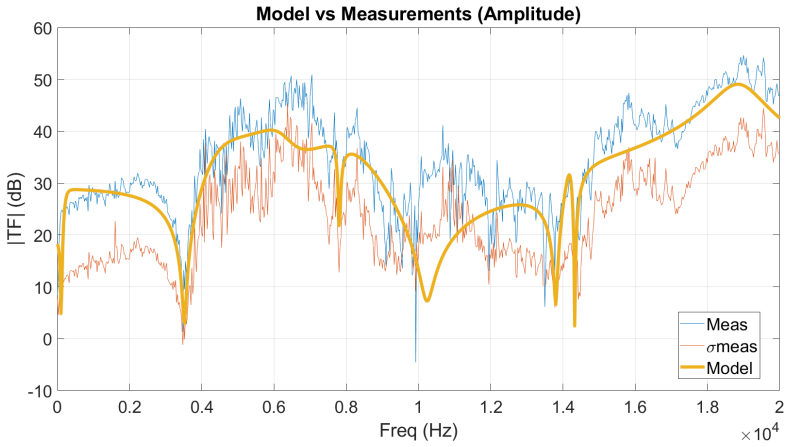


Figure 5.26: The model of the transfer function from the 0-20 kHz range. The model shows many distortions making it not useful for further measurements.

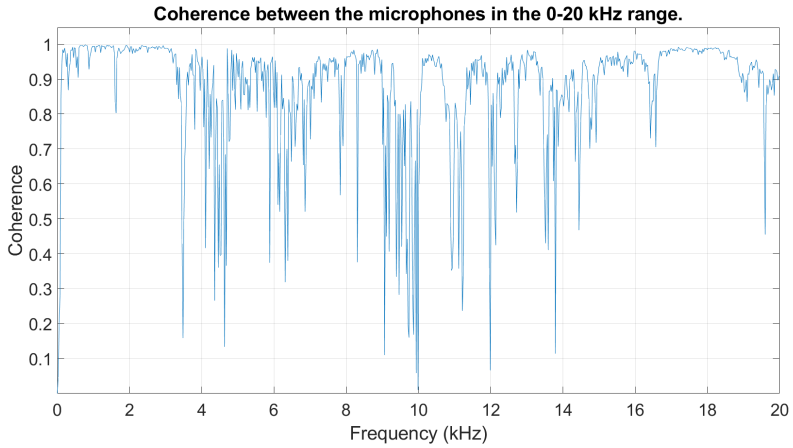


Figure 5.27: Coherence between the reference and test microphones for the 0–20 kHz sweep. High coherence is observed up to about 3–4 kHz, while strong fluctuations between 4 and 16 kHz indicate reduced reliability of the measurements in this range.

5.2 Measurements of urban noise

Now that the microphone has been absolutely calibrated, we can proceed to measure the sound pressure in a more realistic setting. In listing A.2, the code is shown that calculates the sound pressure level (dB SPL) along with the corresponding frequency spectrum. The results are compiled into a CSV file and automatically uploaded to Google Drive to ensure continuous data availability. Since this process generates large amounts of data and we wish to limit the number of CSV files, the system is configured to perform 4 measurements of 5 seconds each within every 10-minute interval. This approach provides a representative snapshot of the urban sound pressure approximately every 2.5 minutes.

The measurement procedure is as follows. First, the raw time signal is recorded at a sampling frequency of 51 200 Hz for a duration of 5 seconds. Next, the DC offset is removed, since it carries no information about the actual sound and may interfere with further analysis. Importantly, no Hann window is applied. A Hann window forces the edge samples toward zero,

which is problematic because the signals are first corrected in the frequency domain and then transformed back into the time domain via an inverse FFT for the calculation of the sound pressure. Applying a window would disrupt this process and introduce numerical inaccuracies.

Since no Hann window is applied, we accept the occurrence of spectral leakage when the time signal is transformed into the frequency domain via an FFT and the corresponding scaling. To obtain calibrated measurements, this spectrum must then be multiplied by the inverse transfer function as defined in section 5.1.3.2. However, the initial results obtained using the full inverse transfer function (see fig. 5.28) proved to be unreliable. In particular, the pronounced anti-resonance around 3.5 kHz and the strong amplification below 500 Hz significantly distorted the reconstructed time signal and consequently the final sound pressure levels. In these frequency ranges, the inverse transfer function imposes excessive corrections. This leads to a substantial amplification of noise and measurement uncertainties. The presence of these may be a consequence of the absence of anechoic measurements during calibration, due to the unavailability of the anechoic chamber.

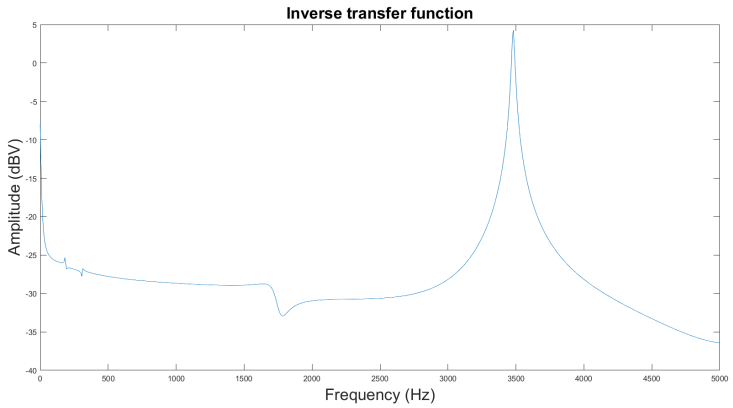


Figure 5.28: The inverse transfer function obtained from the final calibration process.

To avoid this effect, the analysis was restricted to the frequency range between 500 Hz and 3000 Hz. Within this band, the transfer function is stable and reliable, ensuring that the inverse correction yields physically meaningful results. Moreover, this frequency range is particularly relevant to human perception, as the human ear is most sensitive here.

To analyze only this part of the spectrum, frequencies outside the 500 Hz–3000 Hz range are filtered out using a brick-wall band-pass. In this process, all FFT bins outside the band are set exactly to zero, as illustrated in fig. 5.29. The resulting values appear in the plot as an amplitude of –240 dBV. This is due to the introduction of a numerical floor of $\epsilon = 10^{-12}$, ensuring that spectral components are never exactly zero.

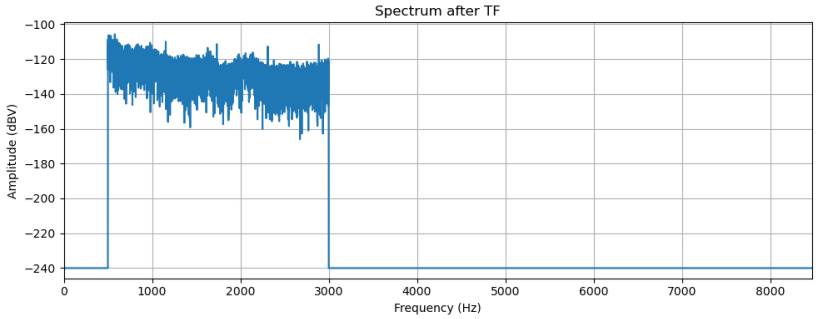


Figure 5.29: The filtered frequency spectrum of a random measurement. This graph clearly shows the brick-wall band-pass filter.

We now have the calibrated spectrum. In principle, the effective sound pressure can be obtained directly in the frequency domain by applying Parseval's theorem (Rolain, 2023). However, this would only yield a single average value over the entire analysis block. Since we are interested in the temporal variations of the sound pressure, we return from the frequency domain to the time domain using the inverse FFT. In this process, the spectrum is first multiplied by the number of samples N, since the forward FFT was already scaled by a factor of $\frac{1}{N}$. This correction ensures that the reconstructed time signal has the correct amplitude scale.

The result is a time signal that, on the one hand, is corrected for the characteristics of the measurement chain and, on the other hand, contains only the frequency components within the selected analysis range of 500–3000 Hz. This reconstructed signal serves as the basis for the subsequent calculation of the effective RMS value in successive time windows.

The corrected time signal obtained after the IFFT spans 5 seconds. Instead of calculating a single RMS value, which would merely provide an average over the entire period and obscure temporal variations, we choose to work with time windows of 100 ms. This duration is sufficiently long to yield a stable RMS estimate, yet short enough to capture fluctuations in sound pressure. To avoid gaps between successive windows, a 50% overlap is applied, resulting in a new RMS value being computed every 50 ms. In this way, a time series of RMS values is obtained, which is then converted into sound pressure levels in dB SPL. This is done by transforming the RMS voltage values into a logarithmic scale using eq. (5.4).

$$L_{\mathrm{dBV}} = 20 * \log_{10}\left(\frac{V_{\mathrm{rms}}}{1\ \mathrm{V}}\right) \tag{5.4}$$

This yields a value in dBV for each window. However, a voltage expressed in dBV does not provide information about the corresponding acoustic pressure. To establish this relation, a calibration was performed using a reference tone, as described in section 5.1.3.2. This reference tone had a known sound pressure level of 94 dB SPL at 1 kHz. From this calibration, a fixed offset was obtained:

$$L_{\mathrm{SPL}} = L_{\mathrm{dBV}} + 127.2 \tag{5.5}$$

Using this formula, a time series of dB SPL values is obtained with respect to the reference pressure $p_{\mathrm{ref}} = 20\ \mu$Pa. Figure 5.30 shows an example of one of the sound pressure measurements in the time domain. Each plot represents a 5-second recording and clearly shows the variation of sound pressure over time, with values ranging between 40 and 50 dB SPL. This is consistent with what can be expected from distant traffic noise.

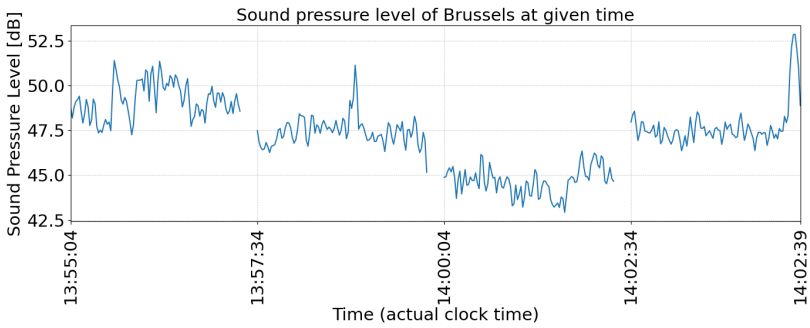


Figure 5.30: SPL in Brussels at a certain time. Each graph represents 5 seconds of measurements.

5.2.1 Setup and acoustic characterization of the measurement site

For the urban noise measurements, the experimental setup was installed on the roof of building L of the Vrije Universiteit Brussel (VUB), as shown in fig. 5.31, indicated by the red marker on the campus map. This building is situated along the Generaal Jacqueslaan. The installation site is situated at an approximate distance of 20 m from the street. Due to the limited availability of suitable measurement locations, it was not possible to perform measurements closer to the traffic flow. As a consequence, the measurement device operates from an elevated position, without a direct line to the main road. This configuration affects the measurements, since both the horizontal offset and the rooftop elevation are likely to reduce the recorded sound pressure levels compared to measurements at street level.



Figure 5.31: Map of the VUB Etterbeek campus with indication of the measurement site (Vrije Universiteit Brussel, 2025).

The test setup, as illustrated in fig. 5.32, consists of a MEMS-microphone integrated with a Raspberry PI, which records and processes acoustic signals before transmitting them via a 4G connection to a cloud server. To ensure autonomous operation under outdoor conditions, the system is powered by a solar panel mounted at 45° angle towards the sun. The solar panel is connected to a solar charger and battery pack that provides reliable energy storage. In addition, the system is configured to perform four measurements of 5 seconds each within every 10 minute interval. This approach provides a representative measurement of the urban sound pressure approximately every 2.5 minutes.



Figure 5.32: Noise monitoring system with solar panel, solar charger and battery pack.

The surrounding acoustic environment is characterized by a variety of urban noise sources. The Generaal Jacqueslaan functions as a main roadway, with constant flow of cars, buses, trams and trucks. This traffic generates a consistent background noise, which is particularly relevant for understanding how additional sound sources, such as eVTOL aircraft, might be perceived by residents.

In addition to road traffic, the proximity of two large hospitals, the CHIREC Delta hospital and the Etterbeek-Ixelles hospital, significantly influences the acoustic environment. The proximity of these healthcare facilities results in frequent ambulance traffic, often accompanied by active sirens. These introduce sharp peaks in sound pressure levels. Furthermore, the federal police is located near building L. They introduce another source of irregular but high intensity noise, as police vehicles regularly pass the site with or without sirens. The chosen location is therefore characterized by continuous noise from traffic and trams, combined with high intensity disturbance such as ambulance and police sirens. Although the rooftop position results in some attenuation of absolute noise levels, the diversity and variability of the sources ensure that the measurement site still captures the characteristic of an urban sound environment.

5.2.2 Results of the measurements

5.2.2.1 The sound pressure levels

The rooftop sound measurements reveal clear temporal variations in the urban acoustic environment of Brussels. During the first measurement series, conducted between 13:26 and 14:45, the sound pressure level ranged from 42 to 52 dB SPL, as shown in fig. 5.33. The average sound pressure level varied between 46 and 48 dB SPL. In this period, the environment was characterized by a relatively stable background level, with occasional sharp peaks. Given the short duration of these peaks, it is likely that they were caused by nearby sound sources, such as a window being closed, a passing bird or a short burst of localized activity. Overall, the sound level in this interval may be described as a continuously audible yet non-dominant urban acoustic environment, reflecting a typical mid-day sound climate in the city.

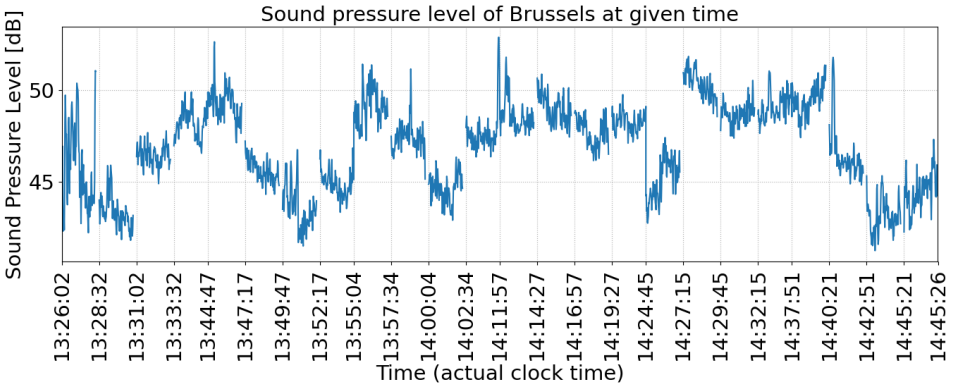


Figure 5.33: Graph of the sound pressure levels from 13:26 until 14:45.

The second measurement series, shown in fig. 5.34, was recorded between 15:43 and 16:48. In this case, the sound pressure level increased noticeably, ranging from 44 to 55 dB SPL with an average of around 49 dB SPL. Compared to the first series, this period exhibited more frequent and more pronounced rises in sound levels, indicating a more intense traffic pattern. This period corresponds with the start of the afternoon rush hour, when traffic density increases and the acoustic environment becomes more dynamic.

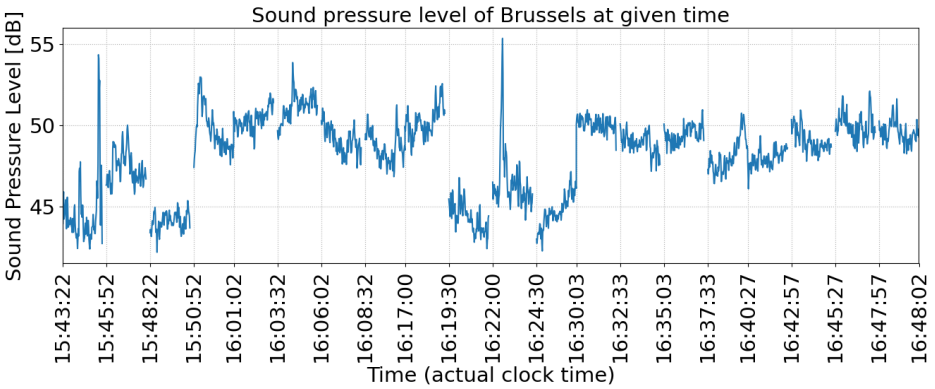


Figure 5.34: Graph of the sound pressure levels from 15:45 until 16:48.

The third measurement series, recorded between 16:51 and 17:43, coincides with the evening rush hour and displays the greatest variability of all three, as illustrated in fig. 5.35. The measured values range from 43 to 57 dB SPL, with an average again around 49 dB SPL. Particularly notable are the pronounced peaks up to 57 dB SPL, which are likely associated with loud and incidental sound sources such as motorcycles, emergency vehicle sirens or heavier traffic. This increased dynamic range reflects a more complex acoustic environment, in which both the higher traffic volume and the diversity of vehicle types contribute to the fluctuations.

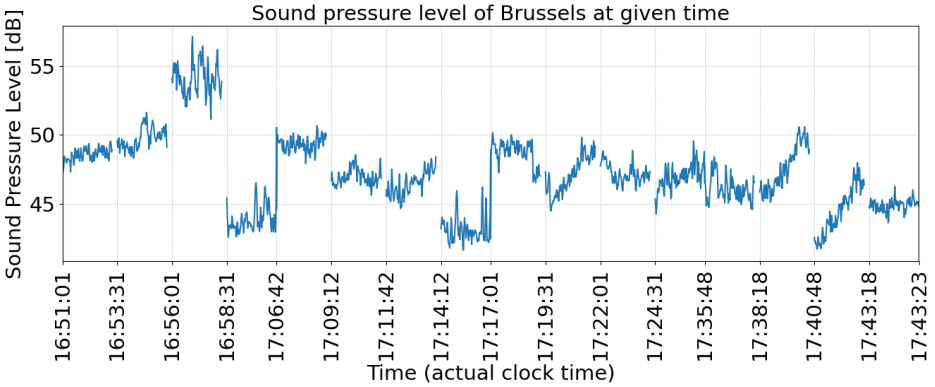


Figure 5.35: Graph of the sound pressure levels from 16:51 until 17:43.

When considered together, the three measurement series illustrate a clear variation in the urban soundscape of Brussels. In the early afternoon, the sound levels are lower and relatively stable. As the day progresses into the late afternoon both the average level and the degree of variability increase. Since this analysis is based on only three measurement series of an hour, the findings can be regarded as indicative rather than conclusive.

For a more robust conclusion about the overall acoustic climate of Brussels, long-term and more systematic measurements would be necessary. Such extended monitoring campaigns would allow the characterization of monthly and weekly variations in the acoustic environment. Still, the results presented here confirm that the developed measurement device is capable of providing reliable and detailed information on temporal variations in urban sound levels.

5.2.2.2 The frequency spectra

The frequency spectrum of the first measurement series, shown in fig. 5.36, exhibits a distinct peak around 1 kHz and a smaller peak near 2 kHz. The narrow bandwidth of these peaks indicates a tonal sound rather than the broadband characteristics typically associated with traffic noise. The origin of these peaks cannot be readily explained.

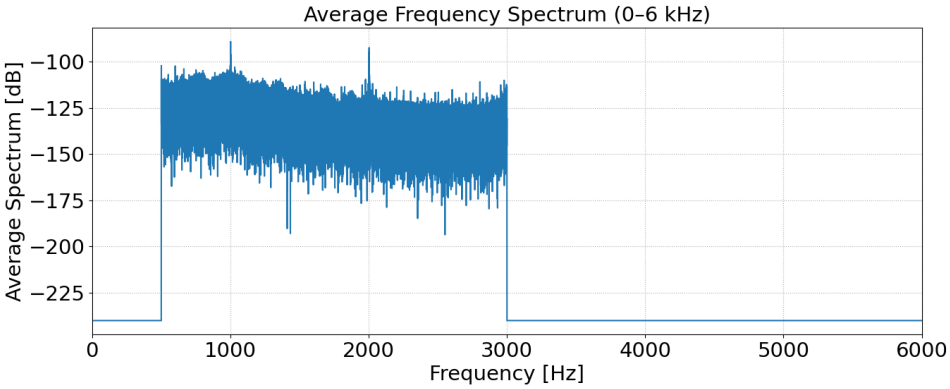


Figure 5.36: The frequency spectrum of the measurements from 13:26 until 14:45.

The frequency analysis of the second measurement series is shown in fig. 5.37. It reveals an increase in broadband energy between 0.5 and 1.5 kHz, combined with a pronounced narrow component around 2 kHz. A distinct peak near 1 kHz is absent in this series. This suggests that the 2 kHz component is not a fixed second harmonic of a 1 kHz tone, but rather an independent tonal contribution.

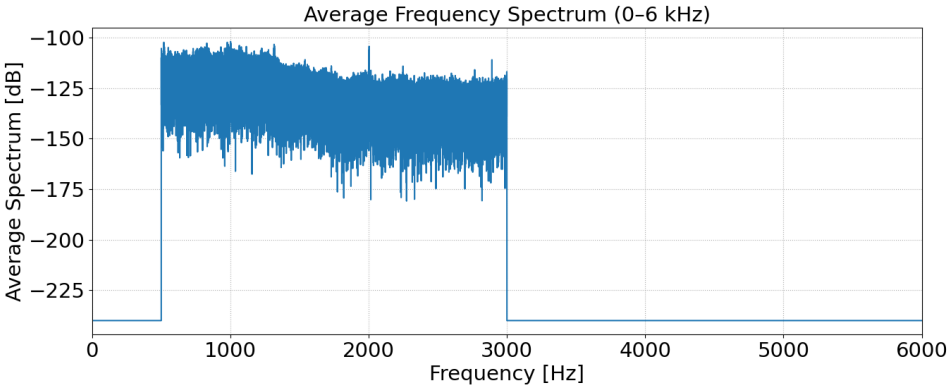


Figure 5.37: The frequency spectrum of the measurements from 15:45 until 16:48.

The spectrum of the last set of measurements is shown in fig. 5.38. This spectrum clearly displays multiple peaks at different frequencies. These peaks also appear to originate from tonal contributions. At present, however, the source of these peaks remains unclear.

When the three frequency spectra are considered together, several patterns emerge. First, the majority of the acoustic energy is distributed relatively evenly across the available frequency range, which is consistent with typical urban traffic noise. Second, narrowband components appear in all spectra, although their exact position and strength vary. This variation suggests that the tonal contribution is source- and time-dependent and therefore cannot be attributed solely to broadband traffic noise. Finally, the spectral complexity increases as the day progresses, which is consistent with the observations made in the time domain.

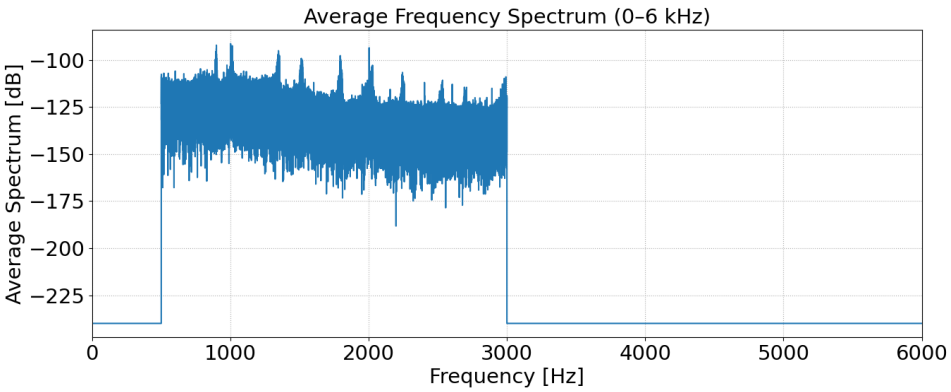


Figure 5.38: The frequency spectrum of the measurements from 16h51 until 17h43.

Given the limited dataset and the use of non-windowed spectra, some spectral leakage artifacts may influence the shape of the peaks. These findings should therefore be interpreted as indicative. Taken together, the observations demonstrate that the measurement setup is capable of distinguishing both broadband trends and tonal signatures in the urban soundscape. However, longer and more detailed analyses are required for reliable source identification and generalizable conclusions.

5.2.2.3 Solar power performance during measurements

An essential aspect of the developed measurement setup concerns its capacity to operate autonomously in outdoor environments. Throughout the entire four-hour measurement campaign, the solar panel provided a stable and uninterrupted power supply to the system. No power-related interruptions or performance drops were observed, confirming that the system can run continuously under normal outdoor conditions with the available solar power and storage.

In addition to the main test series, supplementary experiments were conducted under less favorable conditions. During these trials, the solar panel and battery pack successfully powered the measurement setup for both a full morning and an entire afternoon. This demonstrates that the system can reliably operate not only under clear skies but also under reduced solar irradiance, allowing the setup to function effectively in varying outdoor conditions.

5.2.2.4 eVTOL noise in context of urban background

In addition to the characterization of the urban soundscape, these findings must be interpreted in the context of eVTOL operations. Recent work represented by VKI and DLR within the eVTOLUTION project had demonstrated that multirotor systems generate pronounced tonal noise components. These tones originate from self noise and tip-tip interactions between nearby rotors. These tones are strongly directional, meaning their perceived loudness depends on the observer's position relative to the aircraft. It should be noted that these results are derived from aeroacoustic analyses rather than real flight tests. Another important factor that was not yet included in this analysis, is the altitude of the aircraft relative to the observer (Lade et al., 2025).

The result, illustrated in fig. 5.39, presents the directivity of tonal noise for the first three harmonics of the blade-passing frequency (BPF) as well as for their combined effect. The upper row represents the downward-facing polar, which corresponds to the noise radiated towards observers located beneath the vehicle. The lower row represents the sideways-facing polar, which corresponds to the noise radiated towards observers located laterally.

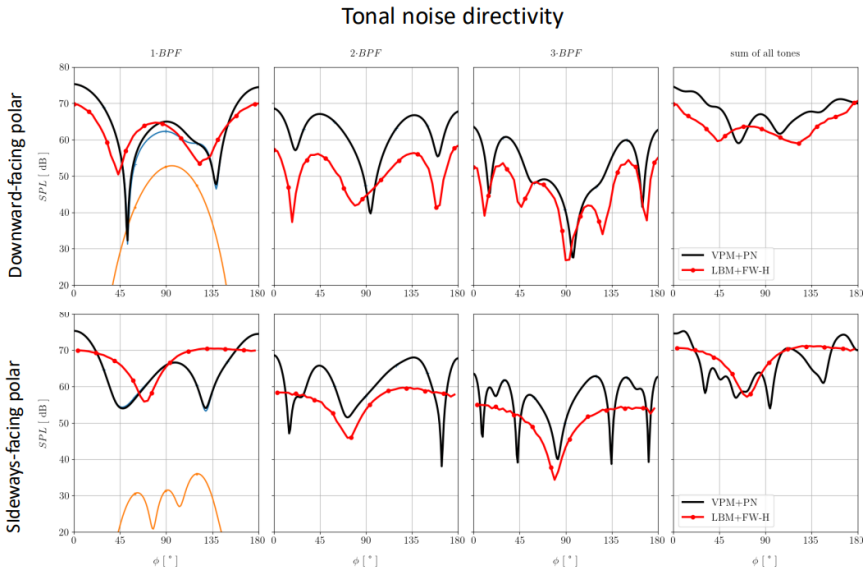


Figure 5.39: Multirotor tonal noise at different blade-passing frequency harmonics, shown for downward-facing (top) and sideways-facing (bottom) directions (Lade et al., 2025).

In these plots, the black curves show the analytical PropNoise model prediction, while the red curves represent the numerical results of the FW-H simulations. The additional curve in blue and orange indicate the separate contribution of interaction noise and self noise respectively. Together, these components form the total tonal noise signature of the multirotor system (Lade et al., 2025).

The comparison between upward- and sideways-facing polar shows that the tonal noise levels are generally higher in the downward direction, with values reaching on the order of 60–75 dB SPL. In contrast, the sideways-facing polar results in somewhat lower levels, typically around 55–75 dB SPL. This confirms that the acoustic signature of multirotor systems is not only dominated by tonal peaks at multiples of the blade-passing frequency but also by its strong dependence on observer position.

The urban background noise levels recorded at the VUB campus ranged between 42-57 dB SPL. The predicted tonal noise of the multirotor therefore exceeds the urban background by more than 10 dB SPL when measured beneath the vehicle. This indicates that eVTOL operations will be distinctly audible to observers directly underneath the flight path. In sideway directions, the audible levels are closer to the measured background. This suggests that in these positions the tonal components may sometimes be masked by urban noise. The analysis shows that eVTOL detectability in cities will be the highest for observers directly underneath the vehicle, while sideway observers may experience partial masking by urban noise. Both this positional dependence and the influence of flight altitude should be considered in future studies of urban air mobility.

Chapter 6

Conclusions

The central research question of this thesis was: "How can a low-cost and autonomous acoustic noise measurement device be designed, constructed and calibrated to reliably capture urban background noise and evaluate the acoustic impact of eVTOL aircraft?".

The results of this thesis confirm that a low-cost and autonomous measurement device can be successfully designed and deployed to capture urban background noise and assess the potential acoustic impact of eVTOL aircraft. The measurement system, equipped with a MEMS microphone, a Raspberry Pi, a 4G dongle and solar powered autonomy, proved to be both robust and reliable in outdoor conditions.

A critical step in the validation process was the comparison of the ICS-40638 with a reference microphone in order to establish a transfer function, using the HP 3562 analyzer. Since the calibration measurements were not performed in an anechoic chamber, the transfer function was affected by distortions. When applying the inverse transfer function to the measured signal, it became apparent that frequencies below 500 Hz and above 3000 Hz could not be considered reliable. Within these regions the inverse correction introduced artifacts, such as pronounced anti-resonance around 3.5 kHz and excessive amplification at lower frequencies. These effects led to distortions in the reconstructed time signal and consequently in the derived SPL values. To ensure the accuracy of the measurements, only the frequency range between 500 and 3000 Hz was considered valid.

The measurements that took place at the VUB campus confirmed the system's ability to capture both the daily variations in noise levels and the spectral characteristics of an urban environment. Sound pressure levels ranged from 42 to 57 dB SPL, with average values between 46 to 49 dB SPL, depending on the time of the day. During the early afternoon, the sound levels were relatively stable, whereas during the late afternoon and evening, the average increased and the variability became more pronounced. These results show that the system is capable of capturing both the gradual changes in noise level throughout the day and the short transient peaks generated by events such as traffic surges or emergency sirens. In addition spectral analyses revealed that the urban background is not purely broadband. Distinct tonal peaks were consistently observed, most notably around 1 kHz and 2 kHz, with their frequency and intensity varying across three measurement series. This increasing spectral complexity, especially during the evening, illustrates the capability of the system to detect both broadband noise contribution and narrow band tonal components within the restricted 500 and 3000 Hz.

In addition to these field measurements, aeroacoustic analyses were performed within the eVTOLUTION project. These analysis demonstrate that multirotor eVTOL systems produce tonal noise components originating from rotor self noise and tip-tip interactions. These tonal components are highly directional. Noise levels were estimated at 60–75 dB SPL for observers located beneath the vehicle, whereas sideway observers were exposed to slightly lower levels of 55–75 dB SPL. When compared to the measured urban background, the simulations suggest that eVTOL aircraft will be clearly audible for people located under the flight path, while for observers positioned at the side, the tonal noise may be partly masked by the existing city background. It should be noted that these results are based on simulations and that the flight

altitude relative to the observer is an important parameter that has not yet been included in this analysis.

In conclusion, this thesis demonstrates that a low-cost and autonomous noise measurement system can be designed, constructed, calibrated and deployed effectively under real-world conditions. The system was able to accurately captured urban SPL levels, revealed both gradual variations and transient peaks and detected tonal components within the 500 and 3000 Hz range. By combining field measurements with aeroacoustic simulations of multirotor eVTOL aircraft noise, this work provides a first step towards understanding how such vehicles will acoustically interact with urban environments. Although longer-term measurement campaigns and broader-band calibration would further strengthen the results. Nevertheless, this work establishes a reliable framework that answers the central research question but also delivers a valuable contribution on the ongoing debate on the acoustic integration and societal acceptance of urban air mobility.

Future work

Although this thesis has demonstrated the feasibility of a low-cost, autonomous measurement setup for urban noise monitoring, the work presented here should be regarded as a starting point rather than a final product. To keep the system functional within the scope of the project, certain design choices and assumptions were made, which inevitably impose limitations on its performance, accuracy and applicability. At the same time, the results open up several promising directions for further development. In the following, we outline several areas where the system can be improved and its functionality further expanded.

A first improvement would be to repeat the calibration measurements in an anechoic chamber. Since the current calibration was not performed under fully controlled acoustic conditions, the usable frequency range was limited to 500–3000 Hz. This represents a clear restriction, as only this part of the spectrum could be reliably analysed. By repeating the calibration in an anechoic chamber, a broader and more stable frequency range can be obtained. This is important because urban sound does not only contain information in the mid-frequency range. It also includes relevant components at lower frequencies, such as traffic vibrations and engine hum, and at higher frequencies, such as braking noise or sirens. Extending the usable frequency range would therefore allow for a more comprehensive representation of the tonal components present in urban noise and contribute to a better understanding of the acoustic environment.

A second improvement would be to apply the calibration directly in the time domain by implementing an appropriate digital filter, such as an Infinite Impulse Response (IIR) filter. In the current approach, the signal is transformed to the frequency domain to apply the inverse transfer function and then converted back to the time domain. By designing an equivalent filter that represents the calibration, this additional transformation step could be avoided, resulting in a more efficient signal processing chain.

In addition, applying an A-weighting filter to the calibrated signal would provide a frequency weighting that reflects the sensitivity of the human ear. This results in a more perceptually meaningful representation of the measured sound levels. This would make the results more comparable to conventional noise measurements and offer clearer insight into how urban sounds are actually perceived. Furthermore, future versions of the system could be equipped with a second microphone or even a small microphone array, to capture spatial information about the sound field. By replicating the binaural perception of the human auditory system, such a setup would enable estimation of the direction of arrival of sound sources. This can add valuable context to the analysis of urban noise.

A third improvement concerns the reduction of the system's power consumption. In its current form, the setup relies on a Raspberry Pi which requires approximately 50.4 Wh per day. This relatively high energy demand necessitates the use of a large solar panel and battery, making the installation less practical and portable. An interesting direction for future work would therefore be to replace the Raspberry Pi with a more energy-efficient platform that still provides the essential functionality: acquiring the data, performing the required signal processing and uploading the results to a cloud. Such a modification would greatly reduce the system's power requirements and improve its practicality.

Finally, an important step would be to conduct additional measurement campaigns in order to obtain more extensive and reliable results. The current work demonstrates the feasibility of the setup, but broader validation under different conditions and over longer periods is necessary to fully assess its accuracy and robustness. By increasing both the quantity and diversity of the measurements, the conclusions drawn from this study can be strengthened and the system's applicability to real-world urban noise monitoring can be more convincingly demonstrated.

A.1 Python

Listing A.1: Signal processing in Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
from scipy.signal import find_peaks



# CSV-bestanden
csv_files = [
    r"C:\Users\Brent VG\Documents\School\1e Master\Masterproef\Calibratietesten\
    Microtesten_2805\Calibratie\2boxmeting9.csv"

]

index = []
# Parameters
sampling_rate = 51200  # in Hz
duration = 20          # in seconden
N = sampling_rate * duration
epsilon = 1e-12

for path in csv_files:
    filename = os.path.basename(path)
    raw_data = pd.read_csv(path)["mV"][:N]
    raw_no_dc = raw_data - np.mean(raw_data)
    time = np.arange(len(raw_data)) / sampling_rate


    # Hann-venster aanmaken en toepassen
    hann_window = 0.5 * (1 - np.cos(2 * np.pi * np.arange(N) / (N - 1)))
    windowed_data = raw_no_dc.values * hann_window

    # FFT
    fft_data = np.fft.fft(windowed_data)
    fft_magnitude = np.abs(fft_data) / N
    fft_magnitude = fft_magnitude[:N // 2]
    fft_magnitude_db = 20 * np.log10(fft_magnitude + epsilon)

    index_x = np.arange(len(fft_magnitude_db))
    #plt.scatter(index_x, fft_magnitude_db, s=1, label=filename)

```

```
    # Pieken zoeken
    window_size = 600
    start_index = 600
    max_index = len(fft_magnitude_db)

    for window_start in range(start_index, max_index, window_size):
        window_end = min(window_start + window_size, max_index)
        segment = fft_magnitude_db[window_start:window_end]
        peaks, _ = find_peaks(segment, prominence=5)

        if len(peaks) > 0:
            peak_values = segment[peaks]
            valid_peaks = peaks[peak_values > -105]

            if len(valid_peaks) > 0:
                best_peak = valid_peaks[np.argmax(segment[valid_peaks])]
                absolute_index = window_start + best_peak
                #plt.scatter(absolute_index, fft_magnitude_db[absolute_index], color='
    red', zorder=2)
                index.append(absolute_index)
```

Listing A.2: Code that measures the dB SPL and uploads it to a Google Drive

```
import time
import os
import csv
from datetime import datetime, timedelta
import numpy as np
from daqhats import mcc128, OptionFlags, AnalogInputRange, AnalogInputMode
from scipy.io import loadmat
import RPi.GPIO as GPIO



# Instellingen
MAT_FILE = "/home/Brent/Downloads/TF_InverseModel.mat"
ADDRESS = 0
FS = 51200
MEETDUUR_S = 5          # duur van 1 meting (s)
INTERVAL_S = 150         # elke 15 seconden starten
WINDOW_S_TOTAAL = 600   # nieuwe CSV elke 1 minuut
NUM_SAMPLES = MEETDUUR_S * FS
OPTIONS = OptionFlags.DEFAULT
EPS = 1e-12

# Short-time vensters voor CSV
ST_WINDOW_S = 0.100     # 100 ms venster
ST_OVERLAP = 0.5        # 50% overlap
FREQ_DECIMATE = 1       # schrijf elke N-de frequentiebin (1 = allemaal)

# Kalibratie
CAL_DBV = -33.2
CAL_SPL = 94.0
DBV_TO_SPL_OFFSET = CAL_SPL - CAL_DBV

BUTTON_GPIO = 18        # fysieke pin 12

```

```
#Onedrive
UITVOER_MAP = "/home/Brent/Documents/Finale_metingen"   # lokale map voor CSV
ONEDRIVE_REMOTE = "Thesis"                               # rclone remote naam
ONEDRIVE_FOLDER = "Thesis"                               # map op de OneDrive remote



# Uploaden naar OneDrive
def upload_naar_onedrive(pad):
    try:
        os.makedirs(os.path.dirname(pad), exist_ok=True)
    except Exception:
        pass
    cmd = f'rclone copy "{pad}" {ONEDRIVE_REMOTE}:"{ONEDRIVE_FOLDER}"'
    print(f"{cmd}")

# Bestandsnaam aanmaken
def maak_csv_pad(prefix="SPL_FFT_ST_1min", ext="csv"):
    dt = datetime.now().strftime("%Y%m%d_%H%M%S")
    bestandsnaam = f"{prefix}_{dt}.{ext}"
    return os.path.join(UITVOER_MAP, bestandsnaam)


# Indrukken van de button
def wait_for_button_press(pin=BUTTON_GPIO):
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    try:
        # Zorg dat de knop niet al ingedrukt is bij opstart
        while GPIO.input(pin) == GPIO.LOW:
            time.sleep(0.01)
        print("Druk op de knop om te starten.")
        # Wachten tot indrukken
        while GPIO.input(pin) == GPIO.HIGH:
            time.sleep(0.005)
        # Debounce
        time.sleep(0.05)
        # Wachten tot loslaten
        while GPIO.input(pin) == GPIO.HIGH:
            time.sleep(0.005)
        print("Knop gedetecteerd, start meting.")

        while GPIO.input(pin) == GPIO.LOW:
            time.sleep(0.005)
    finally:
        GPIO.cleanup()

# Timing helper
def wacht_tot_moment_monotonic(t0_mono, offset_sec):

    doel = t0_mono + offset_sec
    while True:
        nu = time.monotonic()
        rest = doel - nu
        if rest <= 0:
            break
```

```
        time.sleep(0.1 if rest < 1 else min(1.0, rest / 10))


# Data-acquisitie

def acquire_data_with_hat(hat, fs=FS, time0=MEETDUUR_S, num_samples=NUM_SAMPLES,
    options=OPTIONS):
    print("Start meting.")
    collected_data_chunks = []
    start_time = time.time()
    try:
        hat.a_in_scan_start(0b1, num_samples, fs, options)
        while time.time() - start_time < time0:
            read = hat.a_in_scan_read(num_samples, timeout=5.0)
            data = np.array(read.data, dtype=float).flatten()
            if data.size:
                collected_data_chunks.append(data)
    finally:
        try:
            hat.a_in_scan_stop()
            hat.a_in_scan_cleanup()
        except Exception:
            pass
    print("Meting is gestopt.")

    x = (np.concatenate(collected_data_chunks) if collected_data_chunks
         else np.array([], dtype=float))
    t = np.linspace(0, len(x) / fs, num=len(x), endpoint=False)
    return x, t


# Transferfunctie
def load_transfer_function_linear(mat_path=MAT_FILE):
    mat = loadmat(mat_path)
    freq_TF = np.squeeze(mat['freq_cut']).astype(float)
    TF_linear = np.squeeze(mat['TFinv_abs']).astype(float)
    TF_linear = np.nan_to_num(TF_linear, nan=1.0, posinf=1.0, neginf=1.0)
    TF_linear = np.maximum(TF_linear, EPS)
    return freq_TF, TF_linear


def build_fullband_linear_gain(frequencies_all, freq_TF, TF_linear):
    TF_lin_all = np.interp(np.abs(frequencies_all), freq_TF, TF_linear, left=0, right
    =0)
    return TF_lin_all


def apply_transfer_function_to_spectrum(U_complex, G_linear):
    return U_complex * G_linear


# RMS / dB conversies

def vrms(x):
```

```
145
146     return float(np.sqrt(np.mean(np.square(x)))) / 2
147
148
149 def to_dbv(v_rms):
150     return 20.0 * np.log10(max(v_rms, EPS))
151
152
153 def dbv_to_spl(dBV):
154     return dBV + DBV_TO_SPL_OFFSET
155
156 # SPL-tijdreeks
157 def compute_spl_time_series(x, fs, window_s=ST_WINDOW_S, overlap=ST_OVERLAP):
158     N = len(x)
159     win = max(1, int(round(window_s * fs)))
160     step = max(1, int(round(win * (1.0 - overlap))))
161     if step <= 0:
162         step = 1
163
164     times = []
165     spl_values = []
166     for start in range(0, N - win + 1, step):
167         seg = x[start:start + win]
168         v_rms = vrms(seg)
169         dBV = to_dbv(v_rms)
170         spl = dbv_to_spl(dBV)
171         t_center = (start + win / 2) / fs
172         times.append(t_center)
173         spl_values.append(spl)
174     return np.array(times), np.array(spl_values)
175
176
177
178 # Short-time rijen voor CSV (SPL + per-bin rFFT na TF)
179 def st_rows_after_tf(x_after_tf, fs, window_s=ST_WINDOW_S, overlap=ST_OVERLAP, freq_
      decimate=FREQ_DECIMATE):
180     """
181     Generator die per short-time venster:
182       - SPL_dB (tijdsignaal na TF, RMS)
183       - rFFT na TF (1/N geschaald), omgezet naar dBV per-bin
184
185     Yield: (t_center_s, spl_db, freq_hz, fft_dbv)
186     """
187     N = len(x_after_tf)
188     win = max(1, int(round(window_s * fs)))
189     step = max(1, int(round(win * (1.0 - overlap))))
190     if step <= 0:
191         step = 1
192
193     # Frequenties voor rFFT
194     freqs = np.fft.rfftfreq(win, d=1.0 / fs)
195
196     for start in range(0, N - win + 1, step):
197         seg = x_after_tf[start:start + win].astype(float)
198
199         # SPL (short-time)
```

```
200         v_rms = vrms(seg)
201         spl_db = dbv_to_spl(to_dbv(v_rms))
202 
203         # rFFT met 1/N schaling (zoals code 2)
204         U = np.fft.rfft(seg) / win
205         mag = np.abs(U)
206         fft_dbv = 20.0 * np.log10(mag + EPS)
207 
208         # Tijdstempel = midden van het venster
209         t_center_s = (start + win / 2.0) / fs
210 
211         # Frequentie decimatie
212         if freq_decimate > 1:
213             idxs = np.arange(0, len(freqs), freq_decimate)
214         else:
215             idxs = np.arange(len(freqs))
216 
217         for k in idxs:
218             yield (t_center_s, spl_db, float(freqs[k]), float(fft_dbv[k]))
219 
220 
221 
222 # 1-minuut venster uitvoeren
223 
224 def run_1min_venster(csv_pad, hat, freq_TF, TF_linear):
225     os.makedirs(os.path.dirname(csv_pad), exist_ok=True)
226     t0_wall = datetime.now()
227     t0_mono = time.monotonic()
228 
229     print(f"Start (wall): {t0_wall.strftime('%H:%M:%S')}")
230     print(f"CSV: {csv_pad}")
231 
232     with open(csv_pad, mode='w', newline='') as f:
233         w = csv.writer(f)
234         w.writerow(["Tijd", "SPL_dB", "Frequentie_Hz", "FFT_TF_DBV"])
235 
236         offsets = [0, 150, 300, 450]
237         for i, off in enumerate(offsets, start=1):
238             print(f"offset {off}s voor meting {i}...")
239             wacht_tot_moment_monotonic(t0_mono, off)
240 
241             #METING OPNEMEN
242             x, _t = acquire_data_with_hat(hat, fs=FS, time0=MEETDUUR_S, num_samples=NUM
    _SAMPLES, options=OPTIONS)
243             if x.size == 0:
244                 print("[WAARSCHUWING] Geen data ontvangen in deze meting; ga door naar
    volgende.")
245                 continue
246 
247             # DC-offset verwijderen
248             x = x - np.mean(x)
249 
250             #TF toepassen (frequentiedomein) op deze 5s meting
251             N = len(x)
252             U_complex = np.fft.fft(x) / N               # 1/N schaal
253             freqs_all = np.fft.fftfreq(N, d=1.0 / FS)
```

```
254             G_all = build_fullband_linear_gain(freqs_all, freq_TF, TF_linear)
255             U_after_TF = apply_transfer_function_to_spectrum(U_complex, G_all)
256             x_after_tf = np.fft.ifft(U_after_TF * N).real   # herstel 1/N
257 
258             #Short-time SPL + FFT per venster binnen de 5s meting
259             for t_center_s, spl_db, freq_hz, fft_dbv in st_rows_after_tf(
260                 x_after_tf, FS, window_s=ST_WINDOW_S, overlap=ST_OVERLAP, freq_decimate
    =FREQ_DECIMATE
261             ):
262                 # Absolute tijd = start van venster + offset + t_center_s
263                 tijd_dt = t0_wall + timedelta(seconds=float(off + t_center_s))
264                 # Alleen tijd (uur:minuut:seconde.millis), geen datum
265                 tijd_str = tijd_dt.strftime("%H:%M:%S.%f")[:-3]
266                 w.writerow([tijd_str, f"{spl_db:.2f}", f"{freq_hz:.6f}", f"{fft_dbv:.6f
    }"])
267 
268         print("Wachten tot einde 1-minuut venster.")
269         wacht_tot_moment_monotonic(t0_mono, WINDOW_S_TOTAAL)
270 
271     print(f" Data opgeslagen in: {csv_pad}")
272     upload_naar_onedrive(csv_pad)
273 
274 
275 
276 # main
277 
278 def main():
279     # Wachten op hardware-knop
280     wait_for_button_press(BUTTON_GPIO)
281 
282     # MCC 128 initialiseren
283     hat = mcc128(ADDRESS)
284     hat.a_in_range_write(AnalogInputRange.BIP_5V)
285     hat.a_in_mode_write(AnalogInputMode.SE)
286 
287     # Transferfunctie inladen
288     freq_TF, TF_linear = load_transfer_function_linear(MAT_FILE)
289 
290     venster_teller = 1
291     try:
292         while True:
293             csv_pad = maak_csv_pad(prefix=f"SPL_FFT_ST_1min_{venster_teller:03d}")
294             print(f"Nieuw 1-min venster => {csv_pad}")
295             run_1min_venster(csv_pad, hat, freq_TF, TF_linear)
296             venster_teller += 1
297     except KeyboardInterrupt:
298         print("Gestopt door gebruiker (Ctrl+C)")
299     except Exception as e:
300         print(f"FOUT: {e}")
301 
302 
303 if __name__ == "__main__":
304     main()
```

Listing A.3: Code that processes the measurement data and plots the SPL over time together with the spectrum

```

import glob
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
from collections import OrderedDict

plt.rcParams.update({
    "font.size": 21,
    "axes.titlesize": 21,
    "axes.labelsize": 21,
    "xtick.labelsize": 21,
    "ytick.labelsize": 21,
})

CSV_GLOB = r"C:\Users\Brent VG\Documents\School\1e Master\Masterproef\Resultaten\2208_1651_1735\*.csv"
DB_FACTOR = 20               # 10 for power-dB, 20 for amplitude-dB
WHITE_GAP = 0.5              # seconds of white space between bursts (visual)

def parse_time_series(tijd_series: pd.Series) -> pd.Series:
    """Parse 'Tijd' (HH:MM:SS.mmm) to pandas datetime (date-less)."""
    dt = pd.to_datetime(tijd_series, format="%H:%M:%S.%f", errors="coerce")
    if dt.isna().any():
        dt2 = pd.to_datetime(tijd_series, errors="coerce")
        dt = dt.fillna(dt2)
    if dt.isna().any():
        raise ValueError("Could not parse some 'Tijd' values (expected HH:MM:SS.mmm).")
    return dt

def load_and_prepare(csv_path: str, file_idx: int) -> pd.DataFrame:
    df = pd.read_csv(csv_path)
    required = {"Tijd","Type","SPL_dB","Frequentie_Hz","FFT_TF_DBV"}
    missing = required - set(df.columns)
    if missing:
        raise ValueError(f"{os.path.basename(csv_path)} is missing columns: {missing}")

    dt = parse_time_series(df["Tijd"])
    # Correct potential midnight rollovers
    diffs = dt.diff().dt.total_seconds().fillna(0)
    rollovers = diffs < -43200  # >12h backwards => next day
    day_offsets = rollovers.cumsum() * 24*3600
    dt_corrected = dt + pd.to_timedelta(day_offsets, unit="s")

    t_rel = (dt_corrected - dt_corrected.iloc[0]).dt.total_seconds()
    df = df.copy()
    df["__file_idx"] = file_idx
    df["__t_rel"] = t_rel
    df["__dt_corr"] = dt_corrected
    return df

def build_continuous_time(df_all: pd.DataFrame) -> pd.DataFrame:
    """Glue multiple files back-to-back to form a continuous time base."""
```

```
    max_per_file = df_all.groupby("__file_idx")["__t_rel"].max().sort_index()
    offsets = max_per_file.cumsum().shift(fill_value=0)
    df_all["__t_cont"] = df_all["__t_rel"] + df_all["__file_idx"].map(offsets.to_dict()
    )
    df_all["__t_cont"] = df_all["__t_cont"] - df_all["__t_cont"].min()
    return df_all

def to_num_with_comma(s: pd.Series) -> pd.Series:
    """Convert numbers with comma decimals to float."""
    return pd.to_numeric(s.astype(str).str.replace(",", ".", regex=False), errors="
    coerce")

paths = sorted(glob.glob(CSV_GLOB))
if not paths:
    raise SystemExit(f"No CSV files found with pattern: {CSV_GLOB}")

frames = [load_and_prepare(p, i) for i, p in enumerate(paths)]
df = pd.concat(frames, ignore_index=True)
df = build_continuous_time(df)


# 1) SPL: concatenate all bursts with white gaps

spl = df[df["Type"].astype(str).str.strip().str.upper() == "SPL_ST"].copy()
spl["SPL_dB"] = to_num_with_comma(spl["SPL_dB"])
spl = spl.dropna(subset=["SPL_dB", "__t_cont", "__dt_corr"]).sort_values("__t_cont")

# Detect bursts via time gaps (within SPL rows)
dt = spl["__t_cont"].diff()
typical_step = dt[dt > 0].median()
if not pd.notnull(typical_step) or typical_step <= 0:
    typical_step = 0.01  # fallback
gap_thr = max(1.0, 10 * typical_step)  # anything > gap_thr separates bursts
spl["__burst_id"] = (dt > gap_thr).cumsum()

# Build stitched time axis with a small white gap between bursts
stitched_times = []
cum_offset = 0.0
burst_bounds = []  # (xs, xe, start_datetime, end_datetime)
for _, burst in spl.groupby("__burst_id", sort=True):
    burst = burst.sort_values("__t_cont")
    t0 = burst["__t_cont"].iloc[0]
    t_rel = burst["__t_cont"] - t0
    t_stitched = cum_offset + t_rel
    stitched_times.append(t_stitched)

    xs = float(t_stitched.iloc[0])
    xe = float(t_stitched.iloc[-1])
    burst_bounds.append((xs, xe, burst["__dt_corr"].iloc[0], burst["__dt_corr"].iloc
    [-1]))

    burst_duration = float(t_rel.iloc[-1]) if len(t_rel) else 0.0
    cum_offset += burst_duration + WHITE_GAP

spl["t_stitched"] = pd.concat(stitched_times).sort_index()

```

```
# ---- Build tick set: burst starts + CSV starts + last burst end ----
# Burst-start ticks
start_positions = [xs for (xs, xe, dts, dte) in burst_bounds]
start_labels    = [dts.strftime("%H:%M:%S") for (xs, xe, dts, dte) in burst_bounds]

# CSV-start ticks = first SPL timestamp per file (on the stitched axis)
first_spl_per_file = (
    spl.sort_values("__t_cont")
       .groupby("__file_idx", as_index=False)
       .first()[["t_stitched", "__dt_corr"]]
)
csv_start_positions = first_spl_per_file["t_stitched"].astype(float).tolist()
csv_start_labels    = first_spl_per_file["__dt_corr"].dt.strftime("%H:%M:%S").tolist()

# Combine and de-duplicate while preserving order by x-position
pairs = start_positions + csv_start_positions + [burst_bounds[-1][1]]
labels = start_labels    + csv_start_labels    + [burst_bounds[-1][3].strftime("%H:%M:%
    S")]

# Sort by position and drop duplicates of the same x (keep first label)
sorted_pairs = sorted(zip(pairs, labels), key=lambda x: x[0])
unique_pairs = list(OrderedDict((round(p, 9), lab) for p, lab in sorted_pairs).items())
tick_pos = [p for p, _ in unique_pairs]
tick_lab = [l for _, l in unique_pairs]

# ---- SPL Figure ----
fig, ax = plt.subplots(figsize=(14, 6))
# draw each burst separately so the gaps remain white
for _, burst in spl.groupby("__burst_id", sort=True):
    ax.plot(burst["t_stitched"], burst["SPL_dB"], color="C0")

ax.set_xlabel("Time (actual clock time)")
ax.set_ylabel("Sound Pressure Level [dB]")
ax.set_title("Sound pressure level of Brussels at given time")
ax.set_xlim(spl["t_stitched"].min(), spl["t_stitched"].max())
ax.grid(True, which="both", ls=":")

# Force exactly our ticks
ax.xaxis.set_major_locator(mticker.FixedLocator(tick_pos))
ax.xaxis.set_major_formatter(mticker.FixedFormatter(tick_lab))
plt.setp(ax.get_xticklabels(), rotation=90, ha="center")

plt.tight_layout()
plt.show()

# 2) Average frequency spectrum (0 to6 kHz), all files
spec = df[df["Type"].astype(str).str.strip().str.upper().str.startswith("SPECTRUM")].
    copy()
spec["Frequentie_Hz"] = to_num_with_comma(spec["Frequentie_Hz"])
spec["FFT_TF_DBV"] = to_num_with_comma(spec["FFT_TF_DBV"])
spec = spec.dropna(subset=["Frequentie_Hz", "FFT_TF_DBV"])
spec = spec[(spec["Frequentie_Hz"] >= 0) & (spec["Frequentie_Hz"] <= 6000)]

# Average in linear domain -> back to dB
avg_lin = spec.groupby("Frequentie_Hz", as_index=True)["FFT_TF_DBV"].apply(
    lambda s: (10 ** (s / DB_FACTOR)).mean()
```

```
)
avg_db = DB_FACTOR * np.log10(avg_lin)
avg_db = avg_db.sort_index()

plt.figure(figsize=(14, 6))
plt.plot(avg_db.index.values, avg_db.values)
plt.xlim(0, 6000)
plt.xlabel("Frequency [Hz]")
plt.ylabel("Average Spectrum [dB]")
plt.title("Average Frequency Spectrum (0 to 6 kHz)")
plt.grid(True, which="both", ls=":")
plt.tight_layout()
plt.show()
```

A.2 MATLAB

Listing A.4: Code that calculates a rational transfer model.

```
clc
close all
clear variables

addpath(genpath("@RationalModel/estimators"))
addpath(genpath("@RationalModel/helper"))
%% Load data

TF = zeros(5,801);
for ii = 1:5
    load("FRF_5KHz_MS" + ii + ".mat")

    TF(ii,:) = (UntitledTF_Real + 1j * UntitledTF_Imag);
    freq = UntitledTF_Real_Time;
end
F = numel(freq);

% figure
% plot(freq, 20*log10(abs(TF)))
%
% figure
% plot(freq, unwrap(angle(TF)))
%
%% Compute the average

meanTF = mean(TF, 1);
varTF = var(TF, 0, 1)/5;

figure(1)
plot(freq, 20*log10(abs(meanTF)))
hold on
plot(freq, 20*log10(abs(sqrt(varTF))))

%% Fit a model

rm          = RationalModel;
rm.Input    = ones(F, 1);
rm.Output   = meanTF.';
```

```
rm.OutputCovariance = varTF.';
rm.add_variable('Frequency', 2*pi*1j*freq, 1);
rm.NumOrder = 10;
rm.DenOrder = 14;

% Estimate the model
rm.btls(1000,1e-15,1e-15)

% Evaluate the model
TFmodel = rm.evaluate_tf().';

% Plot the model
figure(2)
plot(freq, 20*log10(abs(meanTF)))
hold on
plot(freq, 20*log10(abs(sqrt(varTF))))
plot(freq, 20*log10(abs(TFmodel)), 'LineWidth', 4)
ylabel('|TF| (dB)')
xlabel('Freq (Hz)')
legend({'Meas','\sigma{meas}','Model'})

figure(3)
plot(freq, (angle(meanTF))/pi*180)
hold on
plot(freq, angle(TFmodel)/pi*180, 'LineWidth', 4)
ylabel('\phi(TF) (o)')
xlabel('Freq (Hz)')


figure(4)
MLResidual = abs(meanTF - TFmodel)./sqrt(varTF);

plot(freq, MLResidual)
MLCost    = real(MLResidual*MLResidual');
EstMLCost = F - (rm.NumOrder+rm.DenOrder-1)/2;
fprintf(' ML Cost = %g Expected Cost = %g',MLCost,EstMLCost)
title(sprintf('ML Residual(MLCost = %g)',MLCost))

%save('TF_ModelResult.mat', 'freq', 'TFmodel')
```

Listing A.5: Correction of the spectra with the inverse TF.

```
%   PARAMETERS
Fs = 51200;           % Samplingfrequentie in Hz
N_fft = 1024000;      % Aantal FFT-punten in originele berekening
N = Fs * 20;

%   STAP 1: Laad de gegevens
load('Grandke_5kHz_1.mat'); % bevat AllFreqs, AllLines
load('TF_ModelResult.mat'); % bevat TF , freq

%   STAP 2: Zet AllFreqs om van bins naar Hz
freqs_Hz = (AllFreqs / N_fft) * Fs;

%   STAP 3: Bereken inverse van de complexe transferfunctie
epsilon = 1e-12;
Inverse_TF = 1 ./ (TFmodel + epsilon);
```

```
16
17 %   STAP 4: Interpoleer inverse TF naar de frequenties in Hz
18 Inverse_TF_interp = interp1(freq, Inverse_TF, freqs_Hz, 'linear', 'extrap');
19
20 %   STAP 5: Pas inverse TF toe op het spectrum
21 AllLines_corrected   = AllLines .* Inverse_TF_interp;
22 AllLines_corr_mag_dB = 20*log10(abs(AllLines_corrected) / N);
23
24 %   STAP 6: Bereken het gemiddelde magnitude-spectrum in dB
25 corr_mean_dB = mean(AllLines_corr_mag_dB, 2);
26 orig_mean_dB = 20*log10(mean(abs(AllLines) / N, 2));
27
28 %   STAP 7: Plot resultaat
29 figure;
30 plot(freqs_Hz, orig_mean_dB, 'b', 'LineWidth', 2); hold on;
31 plot(freqs_Hz, corr_mean_dB, 'r', 'LineWidth', 2);
32 legend('Original', 'Corrected', 'FontSize', 21, 'Location', 'best');
33 xlabel('Frequency (Hz)', 'FontSize', 21);
34 ylabel('Amplitude (dBV)', 'FontSize', 21);
35 title('Spectrum before and after multiplying with the inverse TF', 'FontSize', 21);
36 ax = gca;
37 ax.FontSize = 21;
38 grid on;
39
40 %   STAP 8: Opslaan van het resultaat (alleen dB waarden)
41 % save('TF_spec_5kHz_1.mat', 'freqs_Hz', 'AllLines_corr_mag_dB')
```

Listing A.6: Average of the Grandke spectra of the pistonphone.

```
1  n_files = 5;
2  start_idx = 1;
3  n_used = n_files - start_idx + 1;
4
5  Fs = 51200;
6  N = Fs*20;
7  N_fft = 1024000;
8
9  for ii = start_idx:n_files
10     data = load("TF_spec_5kHz_" + ii + ".mat");
11
12     % Gebruik de aangeleverde waarden rechtstreeks (al in dB/dBV)
13     freqs_local    = data.freqs_Hz(:);                % frequenties in Hz
14     amplitude_dB   = data.AllLines_corr_mag_dB(:);    % al in dB (geen extra log/
       normalisatie)
15
16     if ii == start_idx
17         % Basis-frequentierooster (referentie)
18         freqs_ref     = freqs_local;
19         spectra_dB_all = zeros(n_used, numel(freqs_ref));
20     end
21
22     % Interpoleer naar gemeenschappelijk rooster (frequenties kunnen afwijken)
23     interp_amplitude = interp1(freqs_local, amplitude_dB, freqs_ref, 'linear', 'extrap'
       );
24     spectra_dB_all(ii - start_idx + 1, :) = interp_amplitude.';
25 end
26
```

```
% Statistiek
mean_spectrum_dB = mean(spectra_dB_all, 1);
std_spectrum_dB  = std(spectra_dB_all, 0, 1);

% Roze wolk (1 sigma)
upper  = mean_spectrum_dB + std_spectrum_dB;
lower  = mean_spectrum_dB - std_spectrum_dB;
x_poly = [freqs_ref.', fliplr(freqs_ref.')];
y_poly = [upper, fliplr(lower)];

% Plot
figure
fill(x_poly, y_poly, [1.0 0.6 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5)
hold on
plot(freqs_ref, mean_spectrum_dB, 'Color', [0.2 0.5 1], 'LineWidth', 2)

legend('Standard deviation', 'Average', 'FontSize', 21, 'Location', 'best')
xlabel('Frequency (Hz)', 'FontSize', 21)
ylabel('Amplitude (dBV)', 'FontSize', 21)
title('Average spectrum with standard deviation', 'FontSize', 21)
ax = gca;
ax.FontSize = 21;
grid on
hold off

% Opslaan
%save('Average_Spectrum.mat', 'freqs_ref', 'mean_spectrum_dB');
```

Listing A.7: Absolute calibration of the microphones.

```
n_files = 5;
Fs = 51200;
N_fft = 1024000;


for ii = 1:n_files
    data = load("Grandke_pist_" + ii + ".mat");
    if ii == 1
        freq_bins = data.AllFreqs;                    % Bin-nummers
        freqs_Hz = (freq_bins / N_fft) * Fs;          % Omrekenen naar Hz
        spectra_dB_all = zeros(n_files, numel(freqs_Hz));
    end
    epsilon = 1e-12;
    amplitude_dB = 20 * log10((abs(data.AllLines(:) + epsilon))/(N)) ;
    spectra_dB_all(ii, :) = amplitude_dB;
end

% Statistieken
mean_spectrum_dB = mean(spectra_dB_all, 1);
std_spectrum_dB  = std(spectra_dB_all, 0, 1);

% Plot
errorbar(freqs_Hz, mean_spectrum_dB, std_spectrum_dB, 'LineWidth', 1.2)
hold on
plot(freqs_Hz, mean_spectrum_dB, 'o', 'MarkerSize', 5, 'MarkerFaceColor', 'b', '
    MarkerEdgeColor', 'k')

```

```
xlabel('Frequency [Hz]')
ylabel('Amplitude [dBV]')
title('Mean spectrum in dBV with standard deviation')
grid on
legend('Mean +- std', 'Mean')
hold off

% Opslaan
%save('Average_Piston_fin.mat', 'freqs_Hz', 'mean_spectrum_dB');

sensitivity_ICS  = -43;   % dBV/Pa
sensitivity_GRAS = 0.01;  % V/Pa
ref_pressure = 20e-6;     % Referentiedruk: 20 x 10e-6 Pa

% Laad de eerste .mat file
data1 = load('Average_Spectrum.mat');  % bevat: freqs_ref, mean_spectrum_dB
frequenties1 = data1.freqs_ref(:);
spectrum1     = data1.mean_spectrum_dB(:);

% Laad de tweede .mat file
data2 = load('Average_Piston_fin.mat'); % bevat: freqs_Hz, mean_spectrum_dB
frequenties2 = data2.freqs_Hz(:);
spectrum2     = data2.mean_spectrum_dB(:);

% Zoek dichtstbijzijnde frequentie bij 1000 Hz in beide datasets
[~, idx1] = min(abs(frequenties1 - 1000));
waarde1 = spectrum1(idx1);

[~, idx2] = min(abs(frequenties2 - 1000));
waarde2 = spectrum2(idx2);

% Toon de waarden
fprintf('Waarde spectrum 1 bij 1 kHz: %.4f dBV\n', waarde1);
fprintf('Waarde spectrum 2 bij 1 kHz: %.4f dBV\n', waarde2);
verschil = waarde2 - waarde1;

% Omzetten naar dB SPL
V_measured  = 10.^(spectrum2/20);          % V (vector)
pressure_Pa = V_measured / sensitivity_GRAS;   % Pa (vector)
dB_SPL      = 20*log10(pressure_Pa / ref_pressure);  % dB SPL (vector)

db_SPL_meting                 = spectrum1 + verschil + 43 + 94;
db_SPL_meting_zonder_verschil = spectrum1 + 43 + 94;

waarde4        = db_SPL_meting_zonder_verschil(idx2);
verschil_SPL   = 94 - dB_SPL;                   % vector
verschil_ICS   = 94 - waarde4;                  % scalar
Totaal_verschil = verschil_SPL + verschil_ICS;  % vector

SPL_Piston = dB_SPL + verschil_SPL;  % vector
SPL_ICS    = waarde4 + verschil_ICS; % scalar

% Resultaat tonen (let op: sommige variabelen zijn vectoren)
fprintf('Gemeten spanning (eerste punt): %.4f V\n', V_measured(1));
fprintf('Geluidsdruk (eerste punt): %.3f Pa\n', pressure_Pa(1));
fprintf('dB SPL (eerste punt): %.2f dB\n', dB_SPL(1));
```

```
fprintf('Waarde piston bij 1 kHz: %.3f dB SPL\n', SPL_Piston(idx2));
fprintf('Waarde ICS bij 1 kHz: %.3f dB SPL\n', SPL_ICS);
fprintf('Verschil piston tot 94 dB (1 kHz): %.3f dB SPL\n', verschil_SPL(idx2));
fprintf('Verschil ICS tot 94 dB (1 kHz): %.3f dB SPL\n', verschil);
fprintf('Totaal verschil voor RPI (1 kHz): %.3f dB SPL\n', Totaal_verschil(idx2));

%% Plot 1: spectrum vs piston (dBV)
figure;
plot(frequenties1, spectrum1, 'b-', 'LineWidth', 2); hold on;
plot(frequenties2, spectrum2, 'ro', 'LineWidth', 2, 'MarkerSize', 6);
legend('Spectrum ICS-40638', 'Spectrum Type 40-PL', 'FontSize', 21, 'Location', 'best')
    ;
xlabel('Frequency (Hz)', 'FontSize', 21);
ylabel('Amplitude (dBV)', 'FontSize', 21);
title('Spectrum of ICS-40638 and Type 40-PL', 'FontSize', 21);
ax = gca; ax.FontSize = 21; grid on; hold off;

%% Plot 2: gecorrigeerd spectrum vs piston (dBV)
figure;
plot(frequenties1, spectrum1 + verschil, 'b-', 'LineWidth', 2); hold on;
plot(frequenties2, spectrum2, 'ro', 'LineWidth', 2, 'MarkerSize', 6);
legend('Spectrum ICS-40638 (Corrected)', 'Spectrum Type 40-PL', 'FontSize', 21, '
    Location', 'best');
xlabel('Frequency (Hz)', 'FontSize', 21);
ylabel('Amplitude (dBV)', 'FontSize', 21);
title('Corrected Spectrum of ICS-40638 and Type 40-PL', 'FontSize', 21);
ax = gca; ax.FontSize = 21; grid on; hold off;

%% Plot 3: SPL vergelijking op 1 kHz (punten)
figure;
plot(1000, SPL_Piston(idx2), 'ro', 'LineWidth', 2, 'MarkerSize', 8); hold on;
plot(1000, SPL_ICS,          'bo', 'LineWidth', 2, 'MarkerSize', 8);
legend('Spectrum pistonphone', 'Spectrum ICS-40638', 'FontSize', 21, 'Location', 'best'
    );
xlabel('Frequentie (Hz)', 'FontSize', 21);
ylabel('Amplitude (dB SPL)', 'FontSize', 21);
title('SPL-vergelijking bij 1 kHz', 'FontSize', 21);
ax = gca; ax.FontSize = 21; grid on; hold off;
```

Abdallah, M. (2009). "A Survey on Data Acquistion systems DAQ". In: *International Conference on Computing, Engineering and Information*.

Adafruit (2025). *Adafruit Data Logging Shield for Arduino*. Accessed: 15 March 2025. URL: https://www.adafruit.com/product/1064.

Amazon (2025a). *CYCCLEVOLT 12 V 12 Ah LifePo4 Battery Deep Cycles, Integrated BMS, 5000+ Deep Cycles 12 V Lithium Battery for Home Lighting, Small UPS, Power Wheels, Fish Finder, 12 V Children's Electric Car and Toys*. https://www.amazon.de/dp/B0F134VR23/?coliid=ICHVWJYJHCPVV&colid=3C3WXMWLPP4W1&ref_=list_c_wl_lv_ov_lig_dp_it&th=1. Accesed: 2025-07-14.

– (2025b). *enjoy solar® 30 W 12 V monocrystalline solar module, 182 mm solar cells, 10 bus bars, solar panel, ideal for caravans, balcony systems, garden sheds, boats*. https://www.amazon.de/dp/B0DD4HFNWS/?coliid=I104VTNXGL9NAJ&colid=3C3WXMWLPP4W1&ref_=list_c_wl_lv_ov_lig_dp_it&th=1. Accesed: 2025-07-14.

– (2025c). *MPPT 30A Solar Charge Controller 12V 24V Solar Controller, MPPT Solar Charge Controller 30A with Temperature Sensor 2 USB Ports LCD Display, Suitable for Solar Panel 12V 24V Lead Acid and Lithium*. https://www.amazon.de/-/nl/dp/B0CM5VHH48/?coliid=I2IQNZWP49S5TU&colid=3C3WXMWLPP4W1&ref_=list_c_wl_lv_ov_lig_dp_it_im&th=1. Accesed: 2025-07-14.

– (2025d). *ZTE MF79U Wingle CAT4-4G Unlocked Wi-Fi USB Modem Low Cost Travel WiFi 150Mbps External Antenna Ports White*. https://www.amazon.de/-/en/MF79U-CAT4-4G-Unlocked-150Mbps-External/dp/B08WPXRTRL?th=1. Accesed: 2025-04-28.

Bakhoum, E. G. and Cheng, M. H. M. (Apr. 2011). "Novel Electret Microphone". en. In: *IEEE Sensors Journal* 11.4, pp. 988–994. ISSN: 1530-437X, 1558-1748. DOI: 10.1109/JSEN.2010.2077276. URL: http://ieeexplore.ieee.org/document/5585649/ (visited on 12/18/2024).

Bilgin, B., Magne, P., Malysz, P., Yang, Y., Pantelic, V., Preindl, M., Korobkine, A., Jiang, W., Lawford, M., and Emadi, A. (June 2015). "Making the Case for Electrified Transportation". In: *IEEE Transactions on Transportation Electrification*. DOI: 10.1109/TTE.2015.2437338.

Boxwell, M. (2017). *Solar electricity handbook: A simple, practical guide to solar energy-designing and installing photovoltaic solar electric systems*. Greenstream publishing.

Bracewell, R. N. (June 1989). "The Fourier Transform". In: *Scientific American* 260.6, pp. 86–95. URL: http://www.jstor.org/stable/24987290.

Chen, R. (2022). "Battery pack design of cylindrical lithium-ion cells and modelling of prismatic lithium-ion battery based on characterization tests". PhD thesis.

Cho, Y. (Apr. 2018). "Noise Source Visualization Using a Digital Voice Recorder and Low-Cost Sensors". en. In: *Sensors* 18.4, p. 1076. ISSN: 1424-8220. DOI: 10.3390/s18041076. URL: https://www.mdpi.com/1424-8220/18/4/1076 (visited on 12/18/2024).

Cook, G., Billman, L., and Adcock, R. (1991). *Photovoltaic fundamentals*. Solar Energy Research Institute.

Corporation, M. C. (2020). *MCC 128 - 16-bit Voltage Measurement DAQ HAT for Raspberry Pi*. Rev 1. Datasheet, December 2020. Measurement Computing Corporation. URL: https://github.com/mccdaq/daqhats.

Dekoninck, L., Thomas, P., Botteldooren, D., Van Laer, J., Duerinckx, A., Van Campenhout, K., Verlaek, M., Van Haver, P., Geens, R., and Van Renterghem, T. (2023). "Region-wide Environmental Noise Monitoring in Flanders through Citizen Science: Protocol to Integrate Surveys and Measurements". In: *Forum Acusticum 2023*.

Dekoninck, L., Wolters, I., Thomas, P., Devuyst, B., Botteldooren, D., Verlaek, M., Van Haver, P., Pe, G., Den Hond, E., and Van Renterghem, T. (2023). "Development of a sleep disturbance protocol collecting simultaneous indoor and outdoor noise, motility and heart rate". In: *14th International Congress on Noise as a Public Health Problem (ICBEN 2023)*, pp. 1–9.

Dileepan, V., Madhavan, K., and Jayakumar, J. (2017). "Performance analysis of lithium polymer battery and super capacitor". In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, pp. 2029–2033.

Dixit, A., Saxena, A., Sharma, R., Behera, D., and Mukherjee, S. (2023). "Solar Photovoltaic Principles". In: *Solar PV Panels-Recent Advances and Future Prospects*. IntechOpen. Chap. 1. DOI: 10.5772/intechopen.109730.

Dmochowski, J. (2009). "On Spatial Aliasing in Microphone Arrays". In: *IEEE TRANSACTIONS ON SIGNAL PROCESSING*.

Errol Prowse, W. (2018). *Mobile Solar Power - Made Easy: Do-it-yourself Guidebook to Vehicle Mounted Solar System Design and Installation*. en. Mobile-SolarPower. URL: http://www.mobile-solarpower.com/ (visited on 08/05/2025).

European Commission (2025). *eVTOLUTION — EVTOL multi-fidelity hybrid design and optimization for low noise and high aerodynamic performance*. European Commission, CORDIS. URL: https://cordis.europa.eu/project/id/101138209 (visited on 08/23/2025).

European Commission, J. R. C. ((2023). *PVGIS Photovoltaic Geographical Information System*. Accessed: 2025-04-16. URL: https://re.jrc.ec.europa.eu/pvg_tools/en/.

European Union Aviation Safety Agency (EASA) (2025). *What is UAM*. European Union Aviation Safety Agency. URL: https://www.easa.europa.eu/en/what-is-uam (visited on 08/23/2025).

eVTOLUTION (2025). *eVTOLUTION – Objectives and Ambitions*. eVTOLUTION. URL: https://evtolution.eu/about/objectives (visited on 08/23/2025).

Franklin, E. (2017). "Solar photovoltaic (PV) site assessment". In: *Retrieved from Cooperative extention: https://extension. arizona. edu/sites/extension. Arizona. edu/files/pubs/az1697-2017. pdf*.

Giancoli, D. C. (2016). *Natuurkunde 1: Mechanica en thermodynamica*. Dutch. 4th edition. Pearson Benelux B.V. ISBN: 978-90-430-2865-3.

Ginzburg, D. (2023). "Evaluating Plant Drought Resistance with a Raspberry Pi and Timelapse Photography". In: *BIO$_P$ROTOCOL*.

Grandke, T. (1983). "Interpolation Algorithms for Discrete Fourier Transforms of Weighted Signals". In: *IEEE Transactions on Instrumentation and Measurement* 32.2, pp. 350–355. DOI: 10.1109/TIM.1983.6313398.

GRAS Sound & Vibration (Aug. 2025). *GRAS 40PL-10 CCP Free-field Array Microphone, High Pressure Technology*. Product datasheet (PDF). GRAS Sound & Vibration. Skovlytoften 33, 2840 Holte, Denmark. URL: https://www.grasacoustics.com/?eID=1709645390%5C&product=341.

Grindstaff, B. (2019). "Affordable remote monitoring of plant growth in facilities using Raspberry Pi computers". In: *Applications in Plant Sciences*.

Hansen, P. C. H. (1951). *FUNDAMENTALS OF ACOUSTICS*. en. Vol. 1. South Australia. (Visited on 12/18/2024).

Horlings, J. (May 2024). *Het Batterijboek*. Uitgeverij Sycorax. URL: http://www.sycorax.nl.

IMILINK (2020). *IM920D 4G WiFi Dongle*. Geraadpleegd op 21 maart 2025. URL: https://imi-link.com/portfolio/im920d-4g-wifi-dongle/.

Inc., I. (July 2020). *AN-000012: Differential Analog Output MEMS Microphone Flex Evaluation Board User Guide*. San Jose, CA, USA. URL: https://www.invensense.com.

InvenSense (June 2016). "Microphone Specifications Explained". In: Application Note AN-1112, Revision 1.1. URL: https://www.invensense.com.

InvenSense, T. (2024). *ICS-40638 High-AOP Analog MEMS Microphone with Differential Output*. Accessed April 2025. URL: https://invensense.tdk.com/products/analog/ics-40638/.

Karlina, F., Waruwu, M., Wijaya, R., et al. (2021). "Study of Several Types of Lithium-polymer Batteries With 3s Battery Management System". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 927. 1. IOP Publishing, p. 012023.

Keizer, A. de, Alsema, E. A., Groeneveld, P., et al. (2008). *Batterijen*. Kennispunt Bètawetenschappen, Universiteit Utrecht.

King, C. B. and Monti, C. (2022). "Microphone vibration sensitivity: what it is, why it is important, and how to measure it". en. In: Nashville, Tennessee, p. 065001. DOI: 10.1121/2.0001702. URL: https://pubs.aip.org/asa/poma/article/2870770 (visited on 12/18/2024).

King, J. P. and Underbrink, J. (May 2008). "Characterization of a Microelectromechanical Systems (MEMS) Microphone". en. In: *14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference)*. Vancouver, British Columbia, Canada: American Institute of Aeronautics and Astronautics. ISBN: 978-1-60086-983-9. DOI: 10.2514/6.2008-2912. URL: https://arc.aiaa.org/doi/10.2514/6.2008-2912 (visited on 12/18/2024).

Koninklijk Meteorologisch Instituut van België (KMI/IRM) (2020). *Klimaatatlas – Dagelijkse waarde van de jaarlijks gemiddelde zonneschijnduur*. https://www.meteo.be/nl/klimaat/klimaat-van-belgie/klimaatatlas/klimaatkaarten/zonnestraling/zonneschijnduur/jaarlijks. Accessed: 2025-04-23.

Lade, T., Guérin, S., and Guignard, T. (Feb. 6, 2025). *Qualification of the Vortex Particle Method for Multi-Rotor Configurations*. eVTOLUTION project. Deutsches Zentrum für Luft- und Raumfahrt (DLR). URL: https://evtolution.eu/eVTOLUTION_M12_WS_Lade.pdf (visited on 08/24/2025).

Lewis, J. (Dec. 2013a). "Analog and Digital MEMS Microphone Design Considerations". en. In: *InvenSense*.

– (Dec. 2013b). "Understanding Microphone Sensitivity". In: White Paper, Revision 1.0. URL: https://www.invensense.com.

Malcoci, I. (Oct. 2014). "Sound Reasearch in Precessional Transmission". en. In: *Applied Mechanics and Materials* 657, pp. 584–588. ISSN: 1662-7482. DOI: 10.4028/www.scientific.net/AMM.657.584. URL: https://www.scientific.net/AMM.657.584 (visited on 12/18/2024).

Mamat, K. and Basiran, M. Z. M. (2022). "Enabling Multi-homed Computer using 4G Dongle in Restricted LAN". In: *Unpublished Technical Report*. URL: https://www.researchgate.net/publication/364404374.

Marin-Garcia, G., Vazquez-Guzman, G., Sosa, J., Lopez, A. R., Martinez-Rodriguez, P., and Langarica, D. (2020). "Battery types and electrical models: A review". In: *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Vol. 4. IEEE, pp. 1–6.

Measurement Computing Corporation (2023). *DAQHATS: Overview*. Accessed: 2025-04-23. URL: https://mccdaq.github.io/daqhats/overview.html.

Moyano, A. P. (2022). *Solar System by Raspberry Pi*. Tech. rep. Master in Renewable Energies.

Muzaffar, C. B. (May 2021). "Design and analysis of off-grid solar system for DC load of a house in Pakistan". In: Memorial University of Newfoundland. URL: https://research.library.mun.ca/14976/.

Mydlarz, C. (2015). "The Design and Calibration of Low Cost Urban Acoustic Sensing Devices". en. In: *NAG*.

Nurjannah, I., Harijanto, A., and Supriadi, B. (2017). "Sound Intensity Measuring Instrument Based on Arduino Board with Data Logger System". In: *International Journal of Advanced Engineering Research and Science (IJAERS)* 4.9. ISSN: 2349-6495 (P), 2456-1908 (O), pp. 27–35. DOI: 10.22161/ijaers.4.9.7. URL: https://dx.doi.org/10.22161/ijaers.4.9.7.

Picaut, J., Can, A., Fortin, N., and Ardouin, J. (2020). "Low-cost sensors for noise monitoring networks: a review". en. In: Artwork Size: 8 pages Publisher: e-Forum Acusticum 2020, 8 pages. DOI: 10.48465/FA.2020.0747. URL: https://hal.archives-ouvertes.fr/FA2020/hal-03233750 (visited on 12/18/2024).

Picaut, J., Can, A., Fortin, N., Ardouin, J., and Lagrange, M. (Apr. 2020). "Low-Cost Sensors for Urban Noise Monitoring Networks—A Literature Review". en. In: *Sensors* 20.8, p. 2256. ISSN: 1424-8220. DOI: 10.3390/s20082256. URL: https://www.mdpi.com/1424-8220/20/8/2256 (visited on 12/18/2024).

Pintelon, R. and Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

Pop, L. and Avram, D. (2019). *Solar Power Demystified: The Beginners Guide To Solar Power, Energy Independence And Lower Bills*. 3rd. Digital Publishing Ltd.

Por, E. (2019). "Nyquist-Shannong sampling theorem". In: *Leiden University*.

Rahn, C. D. and Wang, C.-Y. (2013). *Battery systems engineering*. John Wiley & Sons.

Ramrakhiani, M., Dubey, S., Waxar, H., Kushwaha, K. K., and Singh, P. (2017). "Basic principles and theory of the photovoltaic effect". In: *Recent Advances in Photovoltaics* 17, p. 37.

Ramrakhiani, M. (2017). *Recent Advances in Photovoltaics*. Materials Research Foundations. Materials Research Forum LLC. ISBN: 978-1-945291-37-1. URL: https://books.google.be/books?id=rpVLDwAAQBAJ.

Raspberry Pi (Trading) Ltd (Mar. 2024). *Raspberry Pi 4 Model B Datasheet*. Version 1.1. Accessed April 2025. URL: https://www.raspberrypi.org/documentation/hardware/.

Reddy, K. N., Chandini, T., Reddy, V. P., and Reddy, B. S. (2022). "Monitoring and Controlling of Mobile Robot using 4G-Dongle & Raspberry Pi". In: *Journal of Information and Computational Science* 12.6. ISSN: 1548-7741. URL: https://www.researchgate.net/publication/361407527.

Rolain, Y. (2023). *Systeemtheorie*. VUB.

Rumsey, F. (2009). *Sound and recording*. en. 6th ed. Oxford, UK: Focal Press. ISBN: 978-0-240-52163-3.

Science News (2025). *Cross-Curricular Discussion Q&A*. URL: https://www.sciencenews.org/learning/guide/component/cross-curricular-discussion-qa-4.

Al-Shamani, A. N., Othman, M. Y. H., Mat, S., Ruslan, M., Abed, A. M., and Sopian, K. (2015). "Design & sizing of stand-alone solar power systems a house Iraq". In: *Design & Sizing of Stand-alone Solar Power Systems A house Iraq Ali*, pp. 145–150.

Silverman, H. F. (Dec. 1987). "Some Analysis of Microphone Arrays for Speech Data Acquisition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-35.12, pp. 1699–1711. DOI: 10.1109/TASSP.1987.1165087.

Sliepenbeek, J. (n.d.). "MEMS microphone design." en. In: ().

Solanki, C. S. (2015). *Solar photovoltaics: fundamentals, technologies and applications*. Phi learning pvt. Ltd.

SolarV (2025). *enjoy solar® Monokristallin 182mm 10BB 30W/12V*. https://solarv.de/?a=294&lang=eng. Accesed: 2025-07-14.

Soloveichik, G. L. (2011). "Battery technologies for large-scale stationary energy storage". In: *Annual review of chemical and biomolecular engineering* 2.1, pp. 503–527.

TDK InvenSense (2014). "Analog Output MEMS Microphone Flex Evaluation Board User Guidet". In: *InvenSense*. URL: https://invensense.tdk.com/wp-content/uploads/2015/02/Analog-Output-MEMS-Microphone-Flex-Evaluation-Board-User-Guide8.pdf.

– (2018). *High SPL Analog Microphone with Extended Low Frequency Response*. URL: https://invensense.tdk.com/wp-content/uploads/2019/02/DS-ICS-40300-00-v1.3.pdf.

– (2019a). *High AOP Analog MEMS Microphone with Differential Output*. URL: https://invensense.tdk.com/wp-content/uploads/2020/05/DS-000281-ICS-40638-v1.1.pdf.

– (2019b). *Ultra-Low Noise Microphone with Differential Output*. URL: https://invensense.tdk.com/wp-content/uploads/2020/06/DS-000045-ICS-40720-v1.4.pdf.

– (2025). *ICS-40638 Analog MEMS Microphone Evaluation Board*. https://invensense.tdk.com/products/analog/ics-40638-evb/. Accessed: 2025-08-01.

Texas Instruments (Dec. 2015). *INA129: Precision, Low Power Instrumentation Amplifier*. URL: https://www.ti.com/lit/ds/symlink/ina129-ep.pdf?ts=1743514537424&ref_url=https%253A%252F%252Fwww.google.com%252F.

– (2025). *INA129 Precision Instrumentation Amplifier Datasheet*. Accessed: 2025-05-03. URL: https://www.digchip.com/datasheets/parts/datasheet/477/INA129.php.

Tiwari, G. N. and Dubey, S. (2009). *Fundamentals of photovoltaic modules and their applications*. Royal Society of Chemistry.

Vastav, B. K. S., Nema, S., Swarnkar, P., and Rajesh, D. (2016). "Automatic solar tracking system using DELTA PLC". In: *2016 International Conference on Electrical Power and Energy Systems (ICEPES)*. IEEE, pp. 16–21.

VectorStock (2024). *Wave amplitude and wavelength diagram*. Accessed: 2025-04-16. URL: https://www.vectorstock.com/royalty-free-vector/wave-amplitude-and-wavelength-diagram-vector-49058648.

Vinck, B. and Degeest, S. (2024). *Basisbegrippen Audiologie 1*. Universiteit Gent.

Vrije Universiteit Brussel (2025). *Campusplan VUB Etterbeek*. Accessed: 23 August 2025. URL: https://www.vub.be/nl/over-de-vub/faculteiten-instituten-en-campussen/onze-campussen/vub-main-campus-brussel/campusplan.

Waveshare (2025). *Waveshare SIM7600G-H 4G HAT voor Raspberry Pi*. Geraadpleegd op 21 maart 2025. URL: https://www.tinytronics.nl/nl/communicatie-en-signalen/draadloos/gps/modules/waveshare-sim7600g-h-4g-hat-voor-raspberry-pi.

Wessling, R., Penert, P., and Esser, B. (2025). "How Do Organic Batteries Work? Theoretical and Design Principles of Electrode Materials for All-Organic Batteries". In: *Advanced Energy Materials*, p. 2500150.

You, H., Kim, D., Kim, J., Park, K., and Maeng, S. (2024). "Detection of Damage on Inner and Outer Races of Ball Bearings Using a Low-Cost Monitoring System and Deep Convolution Neutral Networks". In: *Machines*.

Zeman, M. (2003). "Introduction to photovoltaic solar energy". In: *Delft University of Technology*.

Zhu, S., Zhang, G., Wu, D., Jia, L., Zhang, Y., Geng, Y., Liu, Y., Ren, W., and Zhang, W. (2023). "High Signal-to-Noise Ratio MEMS Noise Listener for Ship Noise Detection". In: *Remote Sensing* 15.3, p. 777. DOI: 10.3390/rs15030777. URL: https://www.mdpi.com/2072-4292/15/3/777.