GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Signal Processing - laboratory, PG_00047791		
Field of study	Biomedical Engineering, Biomedical Engineering, Biomedical Engineering		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2025/2026
Education level	first-cycle studies	Subject group	Obligatory subject group in the field of study Subject group related to scientific research in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	4	ECTS credits	1.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department of Multimedia Systems -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Daniel Węsierski	
	Teachers	dr inż. Daniel Węsierski	

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	0.0	0.0	15.0	0.0	0.0	15

E-learning hours included: 0.0

eNauczanie source addresses:

Moodle ID: 47792 Przetwarzanie sygnałów - laboratorium 2026
https://enauczanie.pg.edu.pl/moodle/course/view.php?id=47792

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	15	1.0	9.0	25

Subject objectives

The student applies Python tools to implement discrete-time algorithms for real signal processing; to analyze real signals and simulated systems in the time and frequency domains (DFT); and to design algorithms for basic discrete-time processing systems.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	- student uses PYTHON tools to implement discrete-time signal processing algorithms - student analyzes real signals and simulated systems in the domains of time and frequency (DFT) - student designs algorithms for basic discrete-time signal processing systems	[SW1] Assessment of factual knowledge [SW2] Assessment of knowledge contained in presentation [SW3] Assessment of knowledge contained in written work and projects
[K6_U05] can plan and conduct experiments related to the field of study, including computer simulations and measurements; interpret obtained results and draw conclusions	- student uses PYTHON tools to implement discrete-time signal processing algorithms - student analyzes real signals and simulated systems in the domains of time and frequency (DFT) - student designs algorithms for basic discrete-time signal processing systems	[SU1] Assessment of task fulfilment [SU5] Assessment of ability to present the results of task

Subject contents

Course content – laboratory
Noise corruption of real audio signals. Filtering and analysis of real audio signals. The Z-transform in practice - signal and system diagnostics. Signals under transformations - windowing, sampling, matching, synchronization. Image and filter as 2D signals - spatial processing. Signal-based methods for compression and information hiding.

Prerequisites and co-requisites

Passed exam on Signal Processing from semester 3

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Practical exercise	51.0%	100.0%

Recommended reading

Basic literature	1. Allan V. Oppenheim, Ronald W. Schafer "Discrete-Time Signal Processing - Third Ediction", Prentice-Hall Signal Processing Series, 2014 2. T.P. Zieliński "Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań", WKŁ Warszawa 2005. 3. Instrukcje laboratoryjne zawierające opracowania teoretyczne zagadnień.
Supplementary literature	Presentations connected with Signal Processing lectures.
eResources addresses	

Example issues/ example questions/ tasks being completed

1. Add white Gaussian noise with zero mean and appropriately selected noise power to the randomly chosen signal z(t) in order to obtain a user-defined signal-to-noise ratio (SNR). The SNR determines the quality of the signal relative to the noise level. The higher the SNR value, the clearer the signal and the less audible the noise. The lower the SNR value, the more strongly the signal is distorted by noise.
The signal power should be computed as the average value of the squared samples. Based on the chosen SNR value, determine the required noise power by rearranging the relationship that defines SNR. In the report, describe the algorithm for adding noise for a given SNR level (function add_gaussian_noise()), and present the waveform of the original signal, the generated noise, and the noisy signal. Also explain whether the signal has been significantly distorted and whether the amplitudes of the noisy signal remain within the amplitude resolution range of the original signal.
Next, compute the actual signal-to-noise ratio for the signal z(t) and the generated noise n(t). In the report, describe the SNR calculation algorithm (function calculate_snr()) and state whether the computed SNR matches the previously specified SNR value.
2. Normalize the selected colored noise from Task 2 with respect to the maximum amplitude of the signal x(t) from Task 1, and then reduce its level by multiplying it by a factor in the range 0.4 to 0.8. Store the scaled noise in a new variable. Add it to the signal x(t) to obtain a noisy signal.
Then attempt to denoise the resulting signal. To do this, design a practical Butterworth filter in one of the frequency bands (low-pass or high-pass), appropriately matched to the frequency characteristics of the selected noise. When choosing the filter, analyze the amplitude spectrum of the signal x(t) as well as the frequency characteristics of the noise.
A key part of this task is the proper selection of the filter type based on the noise spectrum. If the noise contains mainly high-frequency components, use a low-pass filter. If low-frequency components dominate, use a high-pass filter. Note that different types of colored noise from Task 2 differ in terms of dominant frequency bands.
Based on your assessment of the noise frequency characteristics, select an appropriate passband for the filter and apply the designed filter to the noisy signal.
3. Many biological signals, such as EEG, exhibit characteristic rhythms and oscillations that can be described using mathematical models. Randomly select a single-channel EEG signal recorded from an electrode. Model the EEG signal using an autoregressive (AR) model of a chosen order p. Use the Yule-Walker method to estimate the AR model coefficients.
Next, determine the transfer function of the AR model in the Z-domain. Compute the poles of this transfer function and plot them in the complex plane. For each pole, determine its angle and convert this angle into the corresponding frequency in hertz using the sampling frequency of the EEG signal.
Compare the obtained frequencies with the amplitude spectrum of the EEG signal. State whether the AR model has identified dominant or typical EEG oscillations. To do this, experiment with different model orders p and analyze how changing the order affects the pole locations and the corresponding frequencies.
4. Add Gaussian noise to the selected image. Then perform image denoising by convolving the image with a Gaussian filter of different kernel sizes, for example 3x3, 5x5, 7x7, and larger. Implement the filtering in Python in two ways: first using standard functions from the SciPy library, and second using a custom module written in C++ and integrated with Python via the PyBind11 interface.

Practical activites within the subject

Not applicable