Digital Signal Processing for Sound and Music
Final lab, December 2025

Each of you has a sound assigned. You should edit your sound to change the format to: .wav, mono, 44.1KHz, 16 bits, not longer than 5 seconds, and process it to adapt the sound to your needs. Save the modified sound and only use this one in your code. The lab must be submitted as a python notebook. Unless you have some justified reason, the only other content to submit with the exam is your edited sound.

List of assigned sounds:

u269248	Garuda1982/sounds/536638/	u269855	johntrap/sounds/529655/
u254630	AJlekceu/sounds/257418/	u269132	joepayne/sounds/413208/
u269279	AndreAngelo/sounds/246219/	u269153	xserra/sounds/162092/
u269241	AlmightyPsyche/sounds/751609/	u269288	xserra/sounds/162102/
u267547	skylarjulian/sounds/558999/	u269190	xserra/sounds/162045/
u269423	universildo/sounds/415627/	u269294	xserra/sounds/162044/
u269672	Wood_Flutes/sounds/333996/	u269768	xserra/sounds/161907/
u269148	elmomo/sounds/760/	u269309	xserra/sounds/126099/
u235590	elmomo/sounds/748/	u269278	emirdemirel/sounds/416026/
u269263	Alsterrr/sounds/672090/	u269775	kirbydx/sounds/175409/
u269230	elvenvoice/sounds/796436/	u269130	cms4f/sounds/159114/
u269602	Ninad_P/sounds/796398/	u269416	GRD-music-/sounds/413884/
u269234	Duisterwho/sounds/735878/	u269357	Flynn413/sounds/618175/
u269264	GregorQuendel/sounds/743335/	u244194	CarlosCarty/sounds/429060/
u269919	WelvynZPorterSamples/sounds/621852/		

The lab has two parts. Part 1 is required, and you should choose one of the 3 optional exercises of Part 2.

Part 1
Download from freesound.org the sound assigned to you (for the sound url include before the text given http://www.freesound.org/people/) and read all the information describing it. If the sound is too long, select a representative fragment of less than five seconds in length (2 seconds would be ideal). If the sound does not have the recommended format, change it using any sound editor. You can also perform some pre-processing to create a more adequate sound. Save the edited sound in your working directory.

Create the python notebook to be used for the whole exam and perform the following tasks: (1) read the downloaded/edited sound and display it, (2) compute its STFT and show the magnitude spectrogram, (3) explain the edits you did on the original sound given and explain why you choose this sound fragment, (4) describe the sound by using its temporal and spectral representation, focusing on what information should be useful for processing the sound with spectral models, thus for deciding the spectral analysis parameters, and (5) show any partial representation, temporal or spectral, of the sound to support your description.

Part 2 (option a): Enhanced sound modeling
The aim of this exercise is to modify the code of one of the spectral models presented in class, adapting it to the sound assigned and, hopefully, obtaining a better result for your sound and for a particular chosen application. As a result, you should compare the existing model with your modification and you should explain why your modification is better, according to your own defined criteria. Examples of possible code improvements are: modify f0 detection, perform a multiresolution analysis, perform a pitch-synchronous analysis, ….. Examples of possible applications are: sound transformation, compression, perceptual analysis, …

1. Choose a model that performs an adequate analysis of the sound, thinking on the application you want to focus on. Perform the analysis and check that the synthesis is good. Explain the choices made and the result obtained.
2. Choose the type of improvements you want to make. Explain them and include any code that can help explain the core software component that you want to modify, and the improvements proposed.
3. Include the modified spectral model in the notebook. Comment the parts of the code you have modified. Perform the analysis/synthesis of the sound with your modified code and explain the result.

Part 2 (option b): Theme and variations
The aim of this exercise is to extend what you did in Assignment 8 using the assigned sound and having no limitations on the analysis models to use, the transformations to apply, and the number of transformed sounds to combine. The result should be a short "music piece" (maximum 30 seconds long) combining various transformations using the tools explained in class. Everything must be done from the python notebook and there should be no need to attach any sound other than your original one (unless clearly justified, for example for doing a morph).

1. Choose a model that would allow to perform an adequate analysis of the assigned sound, thinking on the transformation application. Perform the analysis and check that the synthesis without transformations is good. Explain the choices made and the result obtained.
2. Perform a variety of single transformations (one single analysis/transformation/synthesis sequence for each one), using any analysis model (not just the one used in the previous part and without the need to focus on quality of imitation), and covering a wide variety of effects (natural and unnatural). Explain each transformation, both technically and musically.
3. Create your final sound mix by post-processing and mixing the transformations obtained in the previous section (all done in python). Explain your intention, the process done, and the result obtained.

Part 2 (option c): Sound classification
The aim of this exercise is to extend what you did in Assignment 9 using the assigned sound and focusing on the topic of automatic sound classification. The goal is to perform a classification of your sound into one of 5 sound classes that you define, extracting and using low-level Essentia descriptors and a k-NN classifier. The sound classes should be collections of sounds from freesound, but you should not attach the sound collections, your code should load the sounds directly from freesound.

1. Analyze with Essentia your sound, selecting a maximum of 10 descriptors from the ones used in the freesound extractor, that you believe are most adequate. Use only statistical aggregations of each descriptor. Display the analysis data and explain your choices and your results.
2. Create 5 sound collections manually, using sounds from freesound that you believe can exemplify your classification task. Each collection should be a list of the freesound urls, thus sounds should not be downloaded and edited from the code. Your code should access directly the sounds using the freesound api. Each collection should not be larger than 10 sounds each. No collection should include any sound that is "very" similar to your sound, and each collection should have a coherence (there should be a clear similarity between them). You can use, and explain, your own definition of similarity and coherence. Analyze the 5 sound collections with the descriptors chosen in question 1. Display the analysis data and explain your choices and your results.
3. Perform a classification of your sound using a k-NN classifier, choosing your own k value. Display the result and explain the classification obtained.