

Engineering Tripos Part IIA Project, GF3: Audio Modem, 2018-19

Leader

[Dr J Sayir](#) [1]

Timing and Structure

Thursdays 11-1pm and Mondays 9-11am plus afternoons

Prerequisites

IB Paper 6 assumed, 3F7 and/or 3F4 useful

Aims

The aims of the course are to:

- design a software modem that transmits files from one computer to another over the audio system (loudspeaker, microphone)
- gain a practical understanding of modulation and coding by implementing them on real-world channels and data
- gain a comprehensive view of all elements involved in communication systems, including several not taught in our lectures, e.g., synchronisation and channel estimation
- deepen your understanding of the discrete Fourier transform (DFT) through the implementation of a Basband-OFDM / DMT modulation system
- learn to design software using a combination of Python and C, where the latter is only used for time-critical functions that are slow in Python

Content

This is a new project this year; some of the details below may evolve as the project content is developed further during the Michaelmas and Lent Terms.

Communications engineering can be a pain to convey to the general public because the border between abstract concepts and concrete products is hard to bridge: you design methods using math, sometimes test them in simulations, but they cannot become concrete without specialised hardware. Once the product is made, it's hard to point your finger and tell your friends "Here: I made this part of the product!". This project is different: you are to design a full modem for data transmission over audio, where files will be transmitted from one computer over its loudspeaker and received by another through its microphone. In a way, we are going back to the good old days of telephone modems where users could hear a painful string of beeps and squeaks when they transferred files, except that we will use state-of-the-art modulation and coding techniques that did not exist back then, and achieve data rates and reliability no one has ever achieved before.

The end product should be a working piece of software with the suggested name "Cambridge Audio Transfer" (cat logos and cat "meeows" to announce the start of transmission would be a nice finishing touch!) The software should use a variant of Orthogonal Frequency Division Multiplexing (OFDM) as its modulation (taught in 3F4), and Low-Density Parity-Check (LDPC) codes as part of its coding (taught in 3F7). If you have not taken these modules, you will be able to learn the techniques during the project as they are easy to grasp, but you won't get the more in-depth understanding provided in the lectures. In addition, you will have the opportunity to learn about synchronisation, channel estimation, and other modem components during the project.

Support for the project is based on a Python implementation, with the LDPC decoder written in C, that works on DPO computers. It should not be a problem to do the work on your own computer if you wish to do so (but we may not be able to support you as we don't know all platforms). You can also do the work in a programming language of your choice if you prefer, though again our ability to support you may be more limited.

FORMAT

You will work in groups of 3. Initially, you will be guided as a group through multiple levels, from simple operations that can be tried in the command line to more complex composite operations. Later it will be part of your assignment to find an appropriate task distribution that mixes parallel integration (you work on this part, I work on that part of the system) with vertical integration (how to dimension the overall system).

Week 1: *Quick & Dirty implementation phase*

Complete the missing parts in a framework software. Students will be guided step by step through the process of implementing a DMT modulator and demodulator, and a LDPC encoder and decoder (with existing matrices) to work with provided basic channel estimation, equalisation, synchronisation to produce a first working modem. Aim to achieve successful transmission of a JPEG at low data rates with offline (recorded) transmission/reception

Week 2: *Optimisation phase*

Perform channel measurements and capacity computations. Learn about unequal loading of constellations in OFDM, waterfilling optimisation, LDPC code design using density evolution and/or EXIT charts, irregular Repeat-Accumulate IRA codes, and/or Turbo codes.

Week 3: *Advanced implementation*

Refine the DMT implementation, adapt channel equalisation, implement techniques to optimise the loading and power profiles. Design an LDPC weight polynomial and generate corresponding encoder and parity-check matrices. Implement an IRA encoder / decoder. Implement a Turbo encoder / decoder using convolutional codes. Start planning report.

Week 4: *Final refinements*

Improve synchronisation, implement real-time channel estimation using pilot symbols, test transmission under various conditions (simulated channel, anechoic quiet outdoor scenario, indoor with echo, indoor with echo and interference.) Measure capacity, data rate and error rates achieved. Test transmission in restricted frequency ranges to emulate the regulatory masks that dictate the design of wireless systems. Write report.

Coursework

| Coursework | Due Date | Marks |
|-----------------------------|--------------------------|------------------------|
| Group demo and presentation | Thursday, 16 May, 11 am | 10 (group) |
| Interim report | Tuesday, 21 May, 4 pm | 20 (individual) |
| Final demo and presentation | Monday, 3 June, 9:15 am | 20 (group) |
| Final report | Thursday, 6 June at 4 pm | 30 (all individual) |

Examination Guidelines

Please refer to [Form & conduct of the examinations](#) [2].

Last modified: 03/10/2018 10:16

Source URL (modified on 03-10-18): <https://teaching.eng.cam.ac.uk/content/engineering-tripos-part-ii-a-project-gf3-audio-modem-2018-19>

Links

[1] <mailto:js851@cam.ac.uk>

[2] <https://teaching.eng.cam.ac.uk/content/form-conduct-examinations>