EE 506
Fundamentals of Wireless Communications

Summer 2013

http://courses.washington.edu/ee506

Design and Implementation of
a Wireless Acoustic Communication System

This project aims to design and implement an end-to-end digital wireless acoustic communication system, in which a speaker and a microphone are employed as the transmitting and receiving antennas respectively. You will use Matlab's built-in capabilities that let you emit a sound from the computer speaker, and record a sound incident upon the microphone (e.g. see functions `soundsc`, `sound`, `wavread`, `auread`, . . .).

Many sources of noise and distortion will be present in this system, among them, non-ideal and/or nonlinear speaker and microphone frequency responses, multipath propagation between the speaker and the microphone, propagation delay, and ambient noise.
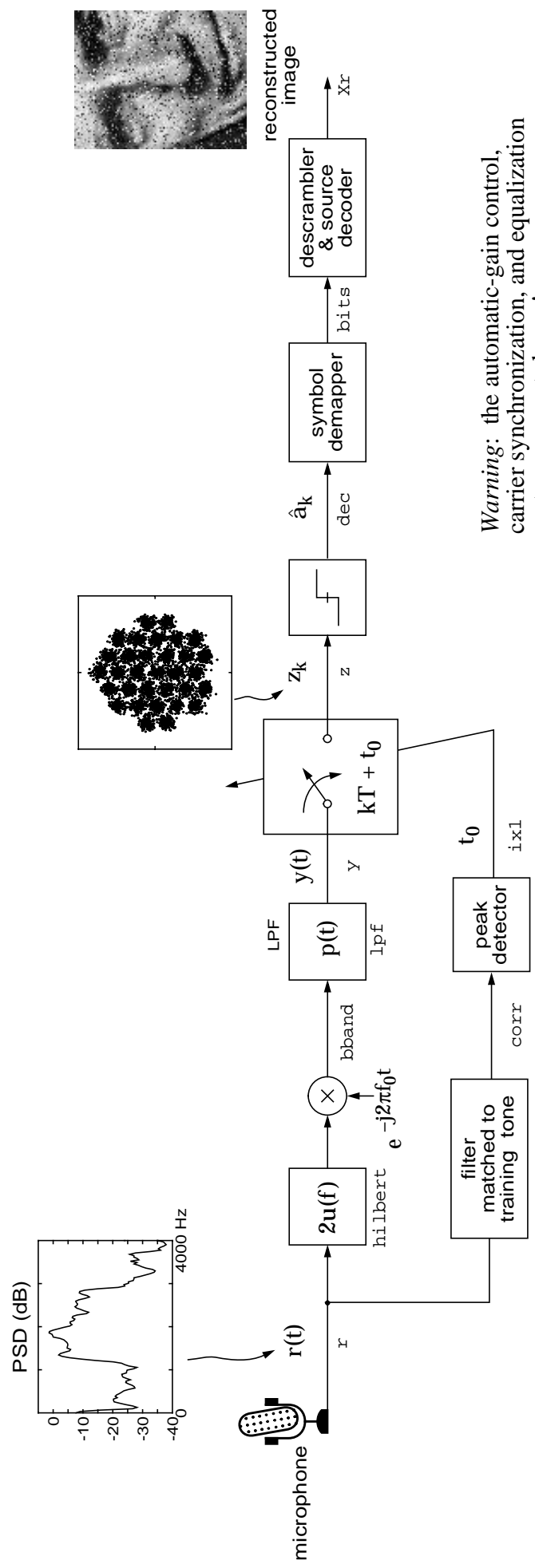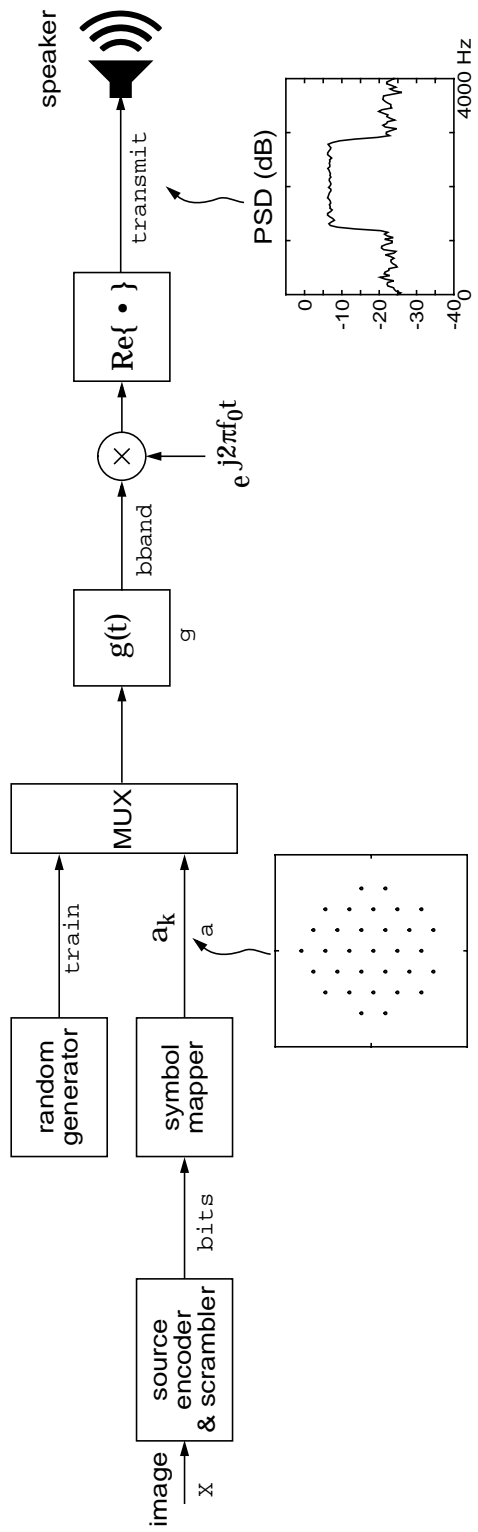
Your goal should be to overcome these challenges and design and implement a system that achieves high bit rate throughput and reliable transmission.[1] Specifically, your primary design objective should be the following: *Maximize the data bit rate R_b, while maintaining a BER of less than 2%, assuming that the microphone and speaker are separated by 15 cm.*

You have full control over what to choose for the transmit filter, modulator, receive filter, and equalizer, and you are encouraged to try a number of different combinations of these. It is expected that you will implement the types of equalizers discussed in class and compare their performance. It is also expected that you will make smart choices about the type(s) of modulation to choose, and compare their performances as well.

Important points:

- Do not forget practical necessities such as carrier and symbol synchronization, and automatic gain control. For instance, it will be helpful if you use a "training" (known) sequence or tone, to help in symbol timing and also with equalization (see the figure, which does not show carrier synchronization or AGC blocks).

- It is expected that you will generate with eye diagrams for your system, both before and after equalization.

- Once you meet the above design objective generate a plot of data rate R_b vs. distance, for different values of total (average) received energy per bit, i.e., set the speaker volume at low, medium, and high; devise a mechanism for accurately (important!) computing the average received energy per bit for each case; vary the distance for each volume setting, and measure the maximum bit rate achievable. Analyze tradeoffs.

- You can use one or two computers; and you may work in groups of 2, or by yourself. One person from each group must inform me about the names of team members prior to start.

- On TBD, each team will be asked to demo the system on a computer of your choice. You will be asked to transmit an image of your choice across the link (see the class web site for image samples).

- You are also required to submit a detailed 8-10 page report, outlining strategies used, your design, results, insights, and conclusions, via email only, to me. The report, in compressed postscript, PDF, or Word, should contain an appendix, not included in the page limit of 8-10 pages, consisting of a *very well documented* version of your computer code. The report - email only - is due by 5:00 pm on TBD. **This is a hard deadline.**

- The project will be graded based on the quality of your design, how well you satisfy or exceed the design objective, and the organization, clarity, and level of detail of your report.

---

[1]To make everything fair for all students, you are required *not* to use data compression or error correction/detection coding in this system.

Transmitter

image — x → source encoder & scrambler → bits → symbol mapper → a_k (a) → MUX → bband → g(t) (g) → ⊗ $e^{j2\pi f_0 t}$ → Re{ • } → transmit → speaker

random generator → train → MUX

PSD (dB): 0, -10, -20, -30, -40 ; 4000 Hz

Receiver

microphone → $r(t)$ (r)

PSD (dB): 0, -10, -20, -30, -40 ; 4000 Hz

$r(t)$ → 2u(f) (hilbert) → ⊗ $e^{-j2\pi f_0 t}$ → bband → p(t) (lpf) LPF → $y(t)$ (y) → $kT + t_0$ → z_k (z) → symbol demapper

$r(t)$ → filter matched to training tone (corr) → peak detector (ix1) → t_0

z_k → [threshold] → $\hat{a}_k$ (dec) → symbol demapper → bits → descrambler & source decoder → reconstructed image — X_r

Warning: the automatic-gain control, carrier synchronization, and equalization systems are not shown!