EE 340 End-semester Examination

Name:
Roll no.:

Read the following instructions carefully before starting the exam.

1. *Fill in your name and roll number at the top of this question paper. You have to return this sheet to your TA after the exam.*
2. *You have until 4.30 pm to complete this exam.*
3. *You are not allowed access to any notes, labsheets, or older GNU Radio files during the exam.*
4. *Save important snapshots and your GNU Radio source files in a zip archive (the file name being your roll number).*
5. *You have to upload the above zip file on the moodle assignment by the name 'End-sem'* **by 4.35 pm**. *The system will not allow an upload after this deadline; students who do not make the upload will be awarded zero marks.*
6. *Access to the internet, except for downloading the two data files required for the exam, and for uploading your final results, is strictly prohibited.*

---

1. **[10 marks]** The goal of this problem is to recover a binary message string transmitted via differential BPSK signalling.

   Specifically, we have a (binary) message signal $x[\cdot]$. However, instead of modulating this message directly, we modulate the differential signal $y[\cdot]$, defined as follows:

   $$y[n] = x[n] \oplus y[n-1]$$

   Here, $\oplus$ denotes the XOR operation. Note that the message signal is encoded in the differences of the $y[\cdot]$ signal. In particular, if $x[n] = 0$, then $y[n] = y[n-1]$, and if $x[n] = 1$, then $y[n] = \overline{y[n-1]}$.

   The above defined $y[\cdot]$ signal is transmitted using BPSK, at a symbol rate of $64kSymbols/sec$, with carrier frequency $f_c = 500kHz$. Pulse shaping is performed using the raised cosine pulse, at $\text{sps} = 4$, using the following GNURadio command.

   ```
   firdes.root_raised_cosine(32, 32*input_sps, 1.0, 0.35, 1408)
   ```

   The received (complex) signal, post I-Q demodulation, is provided to you, at $\text{sps} = 4$, as `File1.dat` on moodle.

   (a) Recover the message signal $x[\cdot]$. Show the recovered signal to your TA on a time scope (QT GUI Time Sink or WX GUI Scope Sink), and also save a snapshot of the same in your submission archive.

TA comments

(b) What is the purpose of using differential encoding/decoding?

Your response

Useful information: You can use the "File Source" block to read the data file into GNU Radio. Remember to set the output type to **complex**, and to configure the block to repeat the data in a loop so that you have a continuous data stream to work with.

Note: You are not allowed to use the built-in differential decoding block in GNURadio.

2. **[10 marks]** A message signal is transmitted using QPSK modulation with a symbol rate of 20 kSymbols/sec, using a carrier frequency of 100 kHz. Pulse shaping is performed, at $\mathrm{sps} = 4$, using the raised cosine pulse using the following GNURadio command.

```
firdes.root_raised_cosine(32, 32*input_sps, 1.0, 0.35, 1408)
```

The transmitted (passband) signal $x(t)$ suffers multipath reflection, such that the received (passband) signal $y(t) = x(t) + 0.65x(t - \tau)$, where $\tau = 75\mu s$. This received (**real** passband) signal $y(t)$, sampled at 400 kHz, is provided to you as `File2.dat` on moodle.

Your task is to recover the QPSK constellation *without using built-in equalization blocks*. You have to design and implement the equalization filter yourself. The filter design should be analytically sound; trial and error solutions will receive less credit.

Useful information: Use the "File Source" block to read the data file into GNU Radio. Remember to set the output type to **real**, and to configure the block to repeat the data in a loop so that you have a continuous data stream to work with.

Describe your equalization filter design here:

Your response

Show your constellation output (using QT GUI Constellation Sink or WX GUI Scope Sink with XY Mode ON) to your TA and also save a snapshot in your submission archive.

TA comments