Mini-Project #2: Filtering Out Satellite Signals

Ms. Alice Liu and Prof. Brian Evans

Assigned on Saturday, October 24, 2026
Due on Friday, October 30, 2026, by 11:59 pm via Canvas submission

Late submission is subject to a penalty of two points per minute late.

Reading: McClellan, Schafer and Yoder, *Signal Processing First*, 2003, Ch. 4-7. Errata.
Companion Web site with demos and other supplemental information: http://dspfirst.gatech.edu/
The Web site contains solutions to selected homework problems from *DSP First*.

E-mail address for Ms. Liu (TA) is aliceliu98@utexas.edu. Please post questions on the class Ed Discussion site, which can be answered by anyone in the class. You can post anonymously.

Time Slot	***Monday***	***Tuesday***	***Wednesday***	***Thursday***	***Friday***
11:00 am		Evans (ECJ 1.214)		Evans (ECJ 1.214)	
11:30 am		Evans (ECJ 1.214)		Evans (ECJ 1.214)	
12:00 pm		Evans (ECJ 1.214)		Evans (ECJ 1.214)	
12:30 pm - 1:30 pm					
2:00 pm		Evans (EER 6.882 and Zoom)		Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)
2:30 pm		Evans (EER 6.882 and Zoom)		Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)
3:00 pm		Evans (EER 6.882 and Zoom)		Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)
3:30 pm		Evans (EER 6.882 and Zoom)		Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)
4:00 pm				Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)
4:30 pm				Liu (EER 5.704 and Zoom)	Liu (EER 6.702 and Zoom)

1.0 Introduction

In this mini-project, we'll use a simplified example to explore wireless signal transmission and reception. Information is transmitted by modulating a low-frequency baseband signal onto a high-frequency sinusoidal carrier signal. To recover the original signal, the receiver filters out noise and unwanted signals and demodulates the desired signal. Although real-world systems are more complex, this simplified model illustrates the basic transmission and reception processes shown in the block diagrams below.

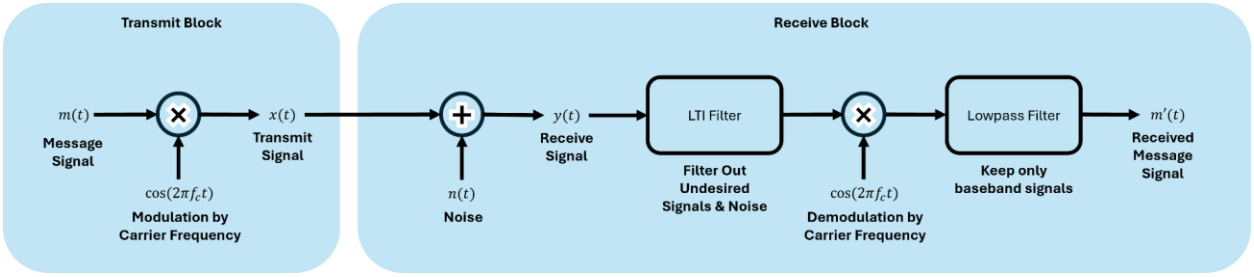
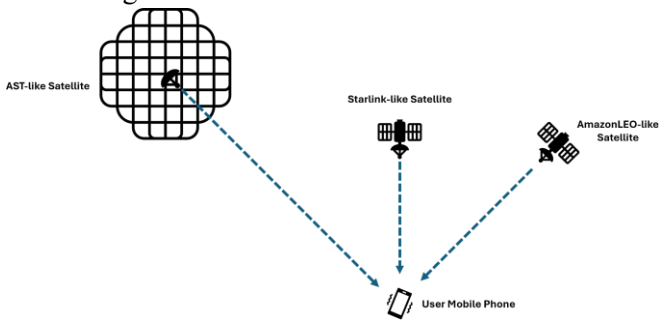


Figure 1: Signal processing blocks for the transmission and reception of a message signal.

For our mini-project communication example, let us pretend we are using our mobile phones to receive distinct signals from three satellites. This is referred to as a satellite direct-to-cellular (D2C) downlink (DL) system, and is shown in the figure below.



Each satellite sends down signals on their own unique carrier frequencies, listed in the table below.

Satellite Direct-to-Cellular System	**Downlink Carrier Frequency*, f_c**
AST-like	7630 Hz
AmazonLEO-like**	15000 Hz
Starlink	19911 Hz

*For our mini-project, we use carrier frequencies in the kHz range. In reality, these satellite systems communicate down to receivers on earth in much higher frequencies, on the order of 1 GHz. For example, AST satellites downlink at 0.763 GHz, AmazonLEO's proposed satellites downlink in the 1.5 GHz range, and SpaceX's Starlink Gen2 D2C satellites can downlink at 1.9911 GHz.

**For our mini-project, to enable students to more clearly distinguish each signal and also enable cleaner filtering, we have adjusted the AmazonLEO-like signal to operate at a lower 15 kHz. A more correct frequency range would be frequencies in the range 15.180 kHz to 15.250 kHz.

Each satellite transmits a distinct audio signal that is modulated by its specified carrier frequency to bring the signal from baseband up to higher transmit frequencies. Modulation is implemented by multiplying the desired baseband signal with a high-frequency sinusoid. This should be familiar to us, as it is implemented the same way we implemented beat frequencies in the Tuneup #2 assignment.

So, a user receives the three satellite signals simultaneously. In our simplified example, we are interested in deciphering each of the three audio signals, which we do through filtering and demodulation.

1.1 Filtering Prior to Demodulation

We must first filter the received signal to separate the desired satellite signal from the others. We can do so by cascading the input signal through multiple filters. Each filter removes an undesired frequency component from the signal until we are left with only the desired components.

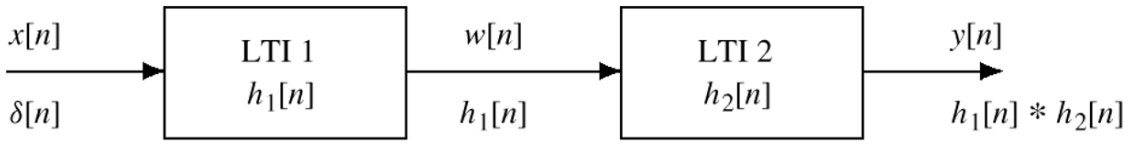


Figure 2: Cascade of two linear time-invariant (LTI) systems.

In Figure 2, our input signal, $x[n]$, passes through LTI 1 which attenuates frequency components around frequency $\hat{\omega}_1$ and their negative frequency counterparts, and outputs signal $w[n]$. This resulting signal is then passed as input to LTI 2 which attenuates frequency components around $\hat{\omega}_2$ and their negative frequency counterparts, and outputs signal $y[n]$. These "Nulling Filters" remove a specified frequency and attenuate the frequencies around it. Recall that we can convert a continuous-time frequency f_1 to a discrete-time frequency using $\hat{\omega}_1 = 2\,\pi\, f_1 \,/\, f_s$ where f_s is the sampling rate.

A Finite Impulse Response (FIR) nulling filter to remove frequency ω_0 has three filter coefficients

$$\{\, 1, -2\cos(\hat{\omega}_0), 1 \,\}$$

The difference equation relating the input signal $x[n]$ to the output signal $w[n]$ is

$$w[n] = x[n] - 2\cos(\hat{\omega}_0)\, x[n-1] + x[n-2]$$

We will verify that the filter nulls frequency ω_0 and its negative counterpart by computing the frequency response, which is the Fourier transform of the impulse response $h[n]$. To compute the impulse response $h[n]$, we input the discrete-time impulse $\delta[n]$ and compute the response $h[n]$:

$$h[n] = \delta[n] - 2\cos(\hat{\omega}_0)\, \delta[n-1] + \delta[n-2]$$

The Fourier transform of $h[n]$ is

$$H(e^{j\hat{\omega}}) = 1 - 2\cos(\hat{\omega}_0)\, e^{-j\hat{\omega}} + e^{-j2\hat{\omega}}$$

We can factor the nulling filter's frequency response into an amplitude and phase:

$$H(e^{j\hat{\omega}}) = e^{-j\hat{\omega}} \left(e^{j\hat{\omega}} - 2\cos(\hat{\omega}_0) + e^{-j\hat{\omega}}\right) = 2\,(\cos(\hat{\omega}) - \cos(\hat{\omega}_0))\, e^{-j\hat{\omega}}$$

So, when $\hat{\omega} = \hat{\omega}_0$, the frequency response is zero. Same is true when $\hat{\omega} = -\hat{\omega}_0$.

For additional insight into the nulling filter, we can compute the transfer function in the z-domain:

$$H(z) = 1 - 2\cos(\hat{\omega}_0)\, z^{-1} + z^{-2} \quad for\ z \neq 0$$

The transfer function $H(z)$ has two zero locations at $\{ e^{j\hat{\omega}_0}, e^{-j\hat{\omega}_0} \}$. In the complex z-domain, the zeros are located on the unit circle at angles with respect to the horizontal positive axis of $\hat{\omega}_0$ and $-\hat{\omega}_0$.

1.2 Demodulation

Once we have filtered out the undesired signals, we must bring our high-frequency signal down to baseband (low frequencies), which can be done through demodulation. Demodulation is modulation followed by lowpass filtering. Modulation multiplies the input signal by a cosine signal at frequency f_c which produces an estimate of the original message signal $m(t)$ as well as a high-frequency version of the original message signal $m(t)$ modulated by a cosine at frequency $2f_c$. We reduce the latter by applying a lowpass filter.

2.0 Filtering Out Unwanted Signals and Noise

In this section, students will be required to identify and filter out the various signals present in the provided mystery_satelliteSignal audio file. The sampling rate is 48 kHz.

(a) Write MATLAB code to…
- (i) Implement Nulling Filter function
- (ii) For **each** identified satellite signal…
  - (1) Design Nulling Filter for each frequency component to remove
  - (2) Filter the signal
  - (3) Analyze the filter's frequency response
  - (4) Determine if the filtered signal has removed enough frequency components. Students may find it helpful to perform frequency domain analysis to determine if the filtered output is "good enough".
  - (5) Continue to design & apply Nulling Filters to the signal (this is a CASCADE OF FILTERS) until the final filtered signal contains only the desired individual satellite components.

(b) For analysis, please discuss the following…
- (i) Identify the signals present in the mysterySignal
  - (1) Specify each signal's center frequency and the range of frequencies that signal occupies. Explain your choice of values.
- (ii) Provide the general transfer function equation of the Nulling Filter
- (iii) For **each unique** filter…
  - (1) Analyze the performance of the filters implemented to obtain each signal
    - a) Frequency response of filter
    - b) Frequency Domain Analysis/Spectrogram of filtered signals

3.0 Demodulating the Desired Bandpass Signal

Demodulation brings the desired transmitted bandpass signal to baseband (centered at 0 Hz). This can be performed by multiplying by a sinusoid at the carrier frequency followed by lowpass filtering. The modulation step creates two copies of the bandpass transmission in the frequency domain– one centered at 0 Hz and one centered at $2f_c$. In some cases, the band centered at $2f_c$ will alias. The lowpass filter attenuates the band of frequencies centered at $2f_c$ and passes the baseband frequencies.

3.1 Modulation

a) For **each** identified satellite signal, write MATLAB code to…
   i) Generate carrier sinusoid signals for the "satellite" signal
   ii) Demodulate the "satellite" signal with its corresponding carrier sinusoid
   iii) Analyze the demodulated signal
   iv) Playback individual signal audio for comparison

b) For analysis, please discuss the following…
   i) For **each** identified satellite signal,
      (1) How many copies were generated as a result of demodulation? Where are these copies located?
   ii) The audio signal sent by the satellites are the beginnings of the following pieces listed below. Please match each satellite signal to its corresponding piece.
      (1) Nocturnes, Op. 9 No. 2 (1832), by Frederich Chopin
      (2) The Entertainer (1902), by Scott Joplin
      (3) Swan Lake Op. 20 Act 2 No. 10 (1876), by Pytor Tchaikovsky

3.2 Lowpass Filtering

a) For **each** desired satellite signal, write MATLAB code to…
   i) Filter out duplicate/aliased copies resulting from demodulation
      (1) For this filtering step, students are encouraged to try available MATLAB filtering functions, and compare against their filter implementations in **Section 2.0**
      (2) For example:
         (a) Using designfilt to obtain FIR filter coefficients
         (b) Utilizing MATLAB lowpass/bandpass/highpass filters
         (c) etc.
   ii) Analyze the implemented filter and the filtered signal (frequency responses, spectrogram, frequency domain plots, etc).
   iii) Playback individual signal audio for comparison
   iv) Save final signals as audio files and include in submission

b) For analysis, please discuss the following…
   i) Discuss how you implemented filtering this time. What are some differences in performance between our implemented null filter and your current method? What are some implementation differences?
      (1) It may be helpful for students to …
         (a) Use MATLAB's "doc function_name" command to better understand the filtering functions used
         (b) Examine filter/filter coefficient lengths
         (c) Time the different implementations of filter design/generation
         (d) Compare the frequency response performance
   ii) Compare and contrast the audio after this last filtering step

4.0 Deliverables

1. Self-contained written narrative report
    c) Formatting: Given personalized feedback and suggestions on mini project 1, students will be held to higher standards on presentation of their written report. See the Grading Rubic in Section 5.0 below. Please take care to format your responses.
    d) As it is a self-contained report, students should take care to place **all** their responses into the written report.
    e) Please do not include code snippets/chunks in your narrative report unless you are explicitly using the **code** to support your answers/analysis. If you would like to include code as a part of your written report, please leave it in the appendix.
2. Zipped Folder containing…
    a. Code (MATLAB .m files)
    b. Recovered audio files in ".wav" format

5.0 Grading Rubric

We are interested in the critical thinking process you use when solving a problem. Please provide rationale and justification. For this mini-project, we'll use the following rubric that are used on homework assignments and exams:

- 3/3 points. Correct rationale/justification and correct results
- 2/3 points. Correct rationale/justification but incorrect results.
- 1/3 points. Some progress; errors in rationale/justification even if results are correct.
- 0/3 points. Answer is blank, or only repeated/reworded the question, or gave an incorrect answer without justification, or never really "got going" towards a solution.

Appendix: Homework and Mini-Project Guidelines

Here are some things you should follow for all assignments.

Amount of work to show:

1. An explanation should be given for every single answer. Answers written without explanation will lose two-thirds of the points allotted for that part.
2. Only "standard" formulas (like Euler's formula, trigonometric formulas, etc.) can be used without a reference. If you're using something non-standard, then please put a reference to the formula number in the book, or whatever source you got it from. Just using the final result of a similar problem done in the class, and omitting the intermediate steps, is not okay. You have to show your work.
3. There shouldn't be big jumps in logic from one step to the next.
4. For everything, expect to show at least one intermediate step between the first line and the answer. Even if it seems unnecessary to you, please err on the side of caution. Things that seem obvious to you when you're writing the solution are not quite so obvious for someone reading it.
5. If you're in any doubt about how much work to show, please ask the instructor or the teaching assistant.

MATLAB source code guidelines:

1. Put a comment before the solution of each part, telling the question number of the solution.
2. If you're using complicated logic, leave a comment telling what that block of code is supposed to do.
3. Use variable names that relate to their meaning/use.
4. Avoid using two different variables for the same thing.
5. Try to avoid using "magic numbers" in the code. If you're using a number, write a comment telling me how you derived it.
6. Make sure that your code will compile & run in a clean workspace; i.e., one without any variables present. Use a clear all; at least once before submitting it.
7. No marks will be deducted based on the efficiency of the code unless the problem asks you to write efficient code.

Technical points:

1. Merge all the files together into one PDF file.
2. Please adjust the contrast, exposure etc., to get a good scan quality so that the TA can easily read what you write. Take extra care to get a good scan for parts written in pencil.
3. For the MATLAB code you write for an assignment, please copy the code into Word or include a screenshot showing the code. Do not submit handwritten code.

Other things:

1. All plots must have axis labels, with units.
2. Final answers must be boxed, or underlined or otherwise differentiated from the rest of the solution.
3. All final answers must have units, if they exist.
4. Read the questions carefully.

5. Try to answer all parts of a question together. If the solution to some parts of a question is written elsewhere, then leave a note telling the reader where to find it.

Organization of a mini-project report:

Please write a **self-contained** narrative report. The audience is someone who has taken the equivalent of this class. The report should provide references to the textbook and other sources as needed. Please refer to the hints above, which apply to homework assignments and mini-project reports, as well as the following additional guidelines for the mini-project.

Here are example mini-project #1 reports written by the instructors:

- "FM Synthesis for Musical Instruments" (2018)
- "Sinusoidal Speech Synthesis" (2021)
- "Music Synthesis" (2023)

Please see the homework hints page for specific guidelines for this project.