Assignment #3: MATLAB GUI Design

Due Date: Wednesday, February 9 at 5:00 PM

Type of Assignment: Individual

Submission Procedure: Please submit your assignment via Canvas. We request you send your files in a single zip archive.

Instructions:

This assignment involves building a MATLAB GUI (Graphical User Interface) for GPS signal processing. The MATLAB GUI may be built using GUIDE, AppDesigner, or may be constructed directly from an m-file. You must submit both the m-file and fig-file associated with your MATLAB GUI as well as any associated m-files required to run the GUI.

Background:

On the course website under “Assignments/Homework Assignment/Assignment 3”, you will find a link to the file `GPS Code.zip`. This zip file contains a bundle of MATLAB m-files (and associated data files) which process measurements of GPS signals reflected from the Earth’s surface and collected on an airborne platform. These signals are obtained in the following manner. A modified GPS Delay Mapping Receiver (DMR) tracks and measures the direct, line-of-sight, right hand circulary polarized (RHCP) signal of a GPS satellite. It also simultaneously measures the delayed, Earth-reflected, near-specular, LHCP GPS signal. These measurements are a form of bistatic radar. The measurements can be used to estimate the surface scattering coefficient and path delays between the direct and reflected GPS signals. Over land, scattering coefficients can be used to estimate changes in surface soil moisture. Over water surfaces, the reflected pulses can be used to sense roughness and derive wind speeds. The bistatic range measurements can also be used to estimate the receiver height above the surface in a form of aircraft altimetry. An illustration of the GPS bistatic radar system is provided in Fig. 1.

The MATLAB m-file `findandtrack.m` is the primary driver for the bundle. It contains the basic functionality to do GPS bistatic reflection processing. It has four parameters that the user can specify at the top of the m-file: GPS satellite number, milliseconds of data to process, sampling frequency, and intermediate frequency. After running `findandtrack.m`, the user can then run either `plotresults.m` or `corrplot.m` for results visualization. The `plotresults.m` is a simple script for plotting the amplitude and frequency results of the bistatic processing of the direct channel. When a user runs `plotresults.m`, two figures are produced, Figure 901, which contains a collection of amplitude plots, and Figure 902, which contains two frequency plots. Two sample instances of Figures 901 and 902 are given in Fig. 2. The `corrplot.m` script plots the important values in the GPS bistatic processing for the direct and reflected channels. When a user runs `corrplot.m`, the user is asked for

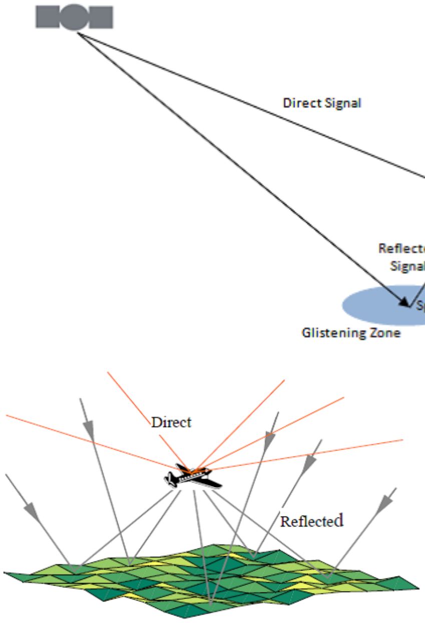


Figure 1: Illustration of the GPS bistatic radar system.

how many milliseconds should be averaged. A typical value is 200 milliseconds. After the user supplies this information, two figures are produced, Figure 100 and Figure 200. In Figure 100, correlation results are provided throughout the data processing time. There may be multiple instances of this data, equal to the milliseconds of data to process divided by the number of milliseconds the data is averaged over. These instances are provided as an "animation". In Figure 200, the estimated path delay for the reflected signal and the amplitude ratio (reflected max/direct max) are illustrated. These are the two most important results from the processing routine. Two sample instances of Figures 100 and 200 are given in Fig. 3.

GUI Design:

For this assignment, you will need to design a GUI to carry out the operations of `findandtrack.m`, `plotresults.m`, and `corrplot.m`. The GUI should allow the user to input the four primary parameters from `findandtrack.m`, namely:

- GPS satellite number (e.g., 22)
- milliseconds of data to process (e.g., 1000 milliseconds)
- sampling frequency (e.g., 4.096e6 Hz)
- intermediate frequency (e.g., 0e6 Hz)

The GUI then should allow the user to "process" this data. To do this, the GUI should include a "start" button to initiate operation once the parameters have been set. Upon completion, your GUI should create the Figures 901 and 902, leveraging `plotresults.m`. Both of these figures should be created *within* the GUI rather than outside the GUI in a separate figure. Note that this means the figures will no longer have figure numbers associated with them.

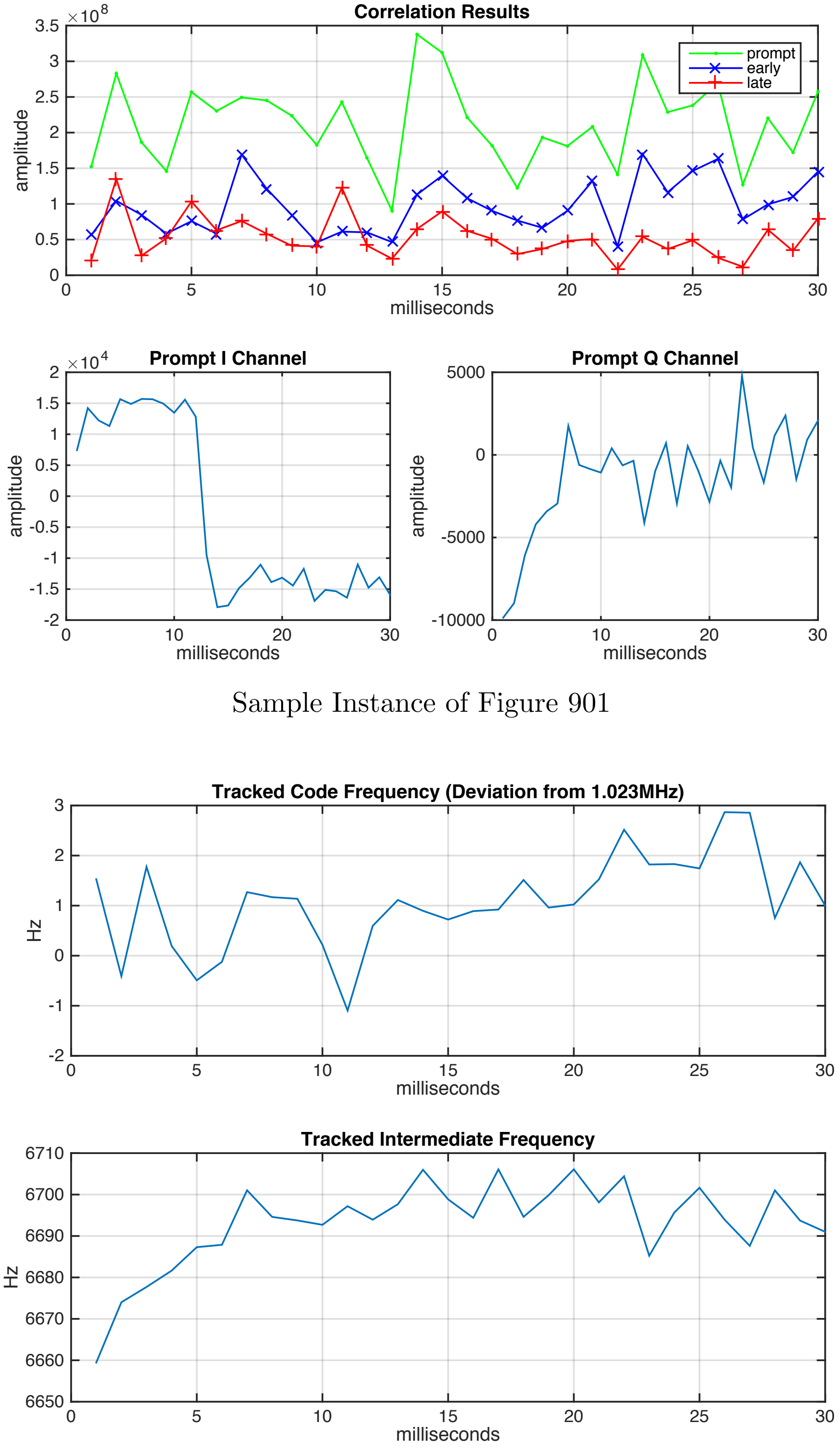
After the GUI has processed the data for the chosen four primary parameters, the GUI should allow the user to input the number of milliseconds to average. The GUI then should allow the user to process the data using this chosen average through a “start” button. This button should also initiate the animation in Figure 100, leveraging `corrplot.m`. As with Figures 901 and 902, this figure should be created *within* the GUI rather than outside the GUI in a separate figure. Finally, the “start” button should create the final plot (Figure 200) from `corrplot.m`. This final plot may be created outside the GUI in a separate figure.

An illustration of a resulting GUI satisfying the aforementioned requirements is shown in Fig. 4.

Remark: Note that completion of this assignment will require modification of the files `findandtrack.m`, `plotresults.m`, and `corrplot.m`.

Bonus:

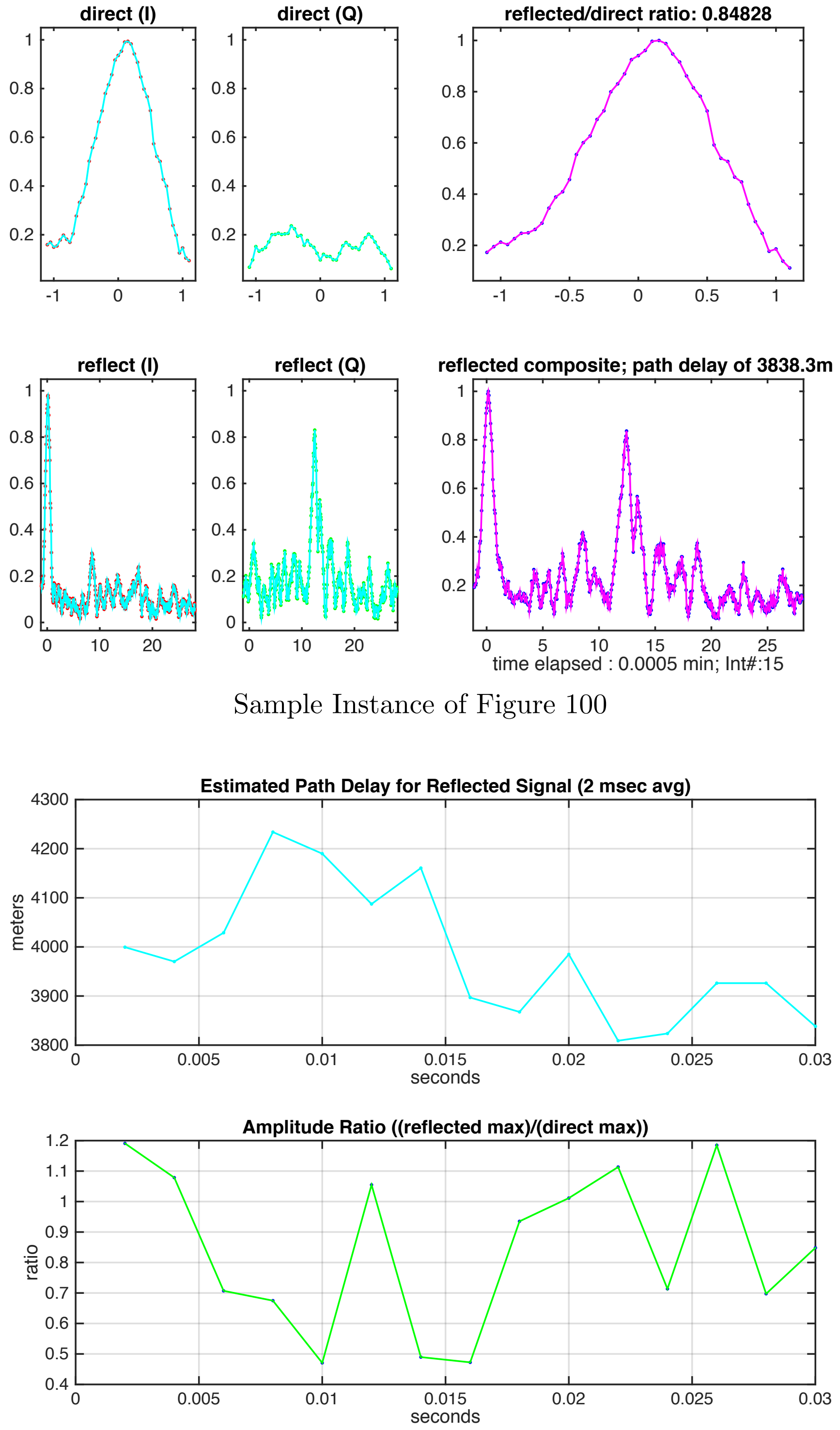
The individual who produces the most visually appealing GUI will receive five bonus points. The instructor will determine whose GUI is most visually appealing (and nobody is necessarily arguing the instructor has good taste).



Sample Instance of Figure 901

Sample Instance of Figure 902

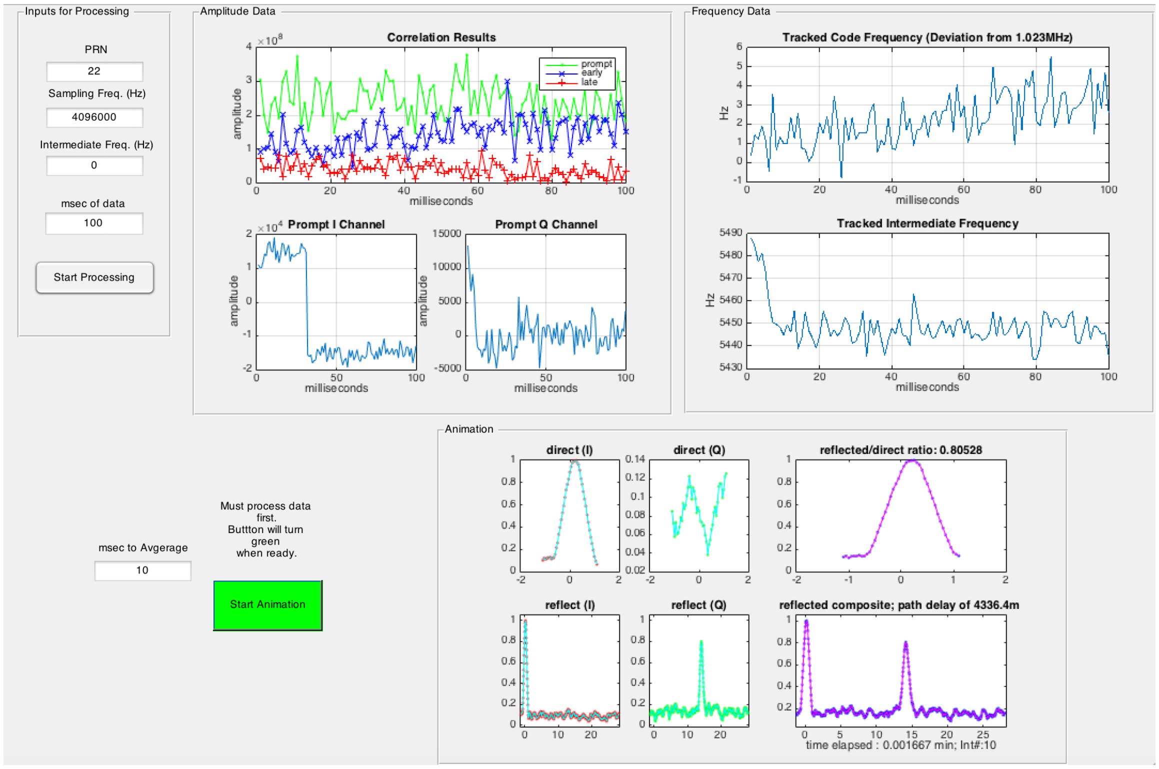
Figure 2: Sample plots produced by the plotresults script.



Sample Instance of Figure 100

Sample Instance of Figure 200

Figure 3: Sample plots produced by the corrplot script.



Inputs for Processing

PRN: 22

Sampling Freq. (Hz): 4096000

Intermediate Freq. (Hz): 0

msec of data: 100

Start Processing

msec to Avgerage: 10

Must process data first.
Buttton will turn green when ready.

Start Animation

Amplitude Data
Correlation Results
×10^8
amplitude
milliseconds
prompt
early
late
Prompt I Channel
×10^4
Prompt Q Channel
Frequency Data
Tracked Code Frequency (Deviation from 1.023MHz)
Hz
Tracked Intermediate Frequency
Animation
direct (I)
direct (Q)
reflected/direct ratio: 0.80528
reflect (I)
reflect (Q)
reflected composite; path delay of 4336.4m
time elapsed : 0.001667 min; Int#:10

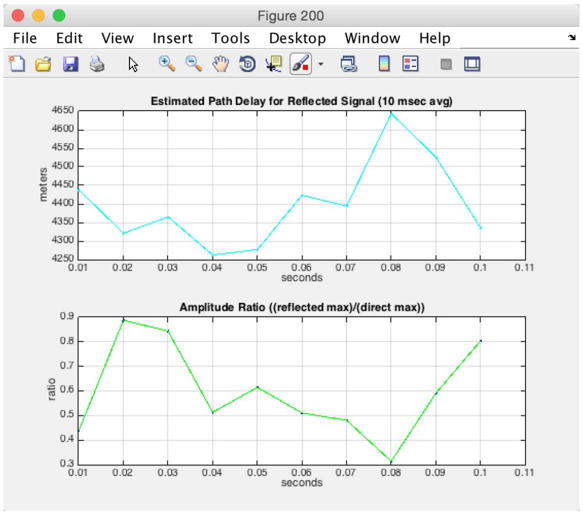


Figure 4: Illustration of sample GUI satisfying the assignment requirements.