Raytheon Preternship Final Project Report

Konrad Rozanski

Matthew Huebner

Table of Contents

Final Project Analysis and Overview

Our final project was a ground station satellite specifier incorporated into a tkinter GUI application. This project required applying the information learned in EE21225(Introduction to Electrical Engineering) and CSE20133(Introduction to Computing for Electrical Engineers). The main aspects incorporated from these classes are data structures, object oriented programming, standard coding practices, and the basics of signal transmission and amplification. The project came in two parts: computations and GUI application.

The computations involved characterizing an orbit and gain equations for the signal passed from the satellite to the ground station. Various assumptions were made for the computations in this project. We were unable to incorporate an API that had data about different locations into our project, this meant we couldn't compute average attenuation losses due to meteorological conditions. Therefore, we assumed the losses to be 6Db. Next, the reason for assuming an overhead pass on the ground station was that the matrix equations to incorporate complex orbits while they are accessible, there was not enough time to implement them. We assumed the ground station would transmit at 60% efficiency as this is considered a common value due to impedance mismatches and internal noise from station circuitry. Our final and most flaring assumption is that the signal to noise ratio that the circuit operates at is equivalent to the Shannon Limit. This is problematic in that it is only a theoretical limit for no loss transmission; however, by characterizing a satellite ground station pairing enough to compute its signal to noise ratio, our program would lose its simplicity and generality. Most of the computations use simple geometry. For instance in our asynchronous overhead pass orbit, we use angle side side triangles evaluated at every pointing angle above 5 degrees over the horizon to find the changing distance between the satellite and the ground station. This is then stored in a vector and the value of the vector is averaged. This is needed in order to accurately compute the path length for the space losses. Finally, we assumed a half wavelength dipole for our antenna length.

The second key component to our program is comprised of the GUI. It is using the GUI, or graphical user interface, that we are able to retrieve input from the user. As detailed in the following memorandums and project report, we took on the challenge of creating an antenna selector using Python, a high level interpreted programming language. We decided to do so mainly due to its rapid development time and vast array of third party libraries. In order to make the best use of our time, we sought out a GUI library that would allow us to construct a modern function user interface. In our search, we found tkinter. Accordingly, our GUI is built on tkinter. As are many modern GUI libraries, tkinter is best suited to object oriented programming. After undergoing the routine learning curve, we were able to use tkinter to produce a program that accomplished its goals both visually and computationally. Regarding the actual construction of the GUI itself, our program possesses three windows: a welcome window, a window that takes in the required inputs, and a final window that displays the output. Complying with tkinter's object oriented paradigm, every element within the GUI is an object. The whole GUI program is rooted within a root object. A class was created for each of the three needed windows. The root object is passed as a constructor parameter when the first window (i.e. the welcome window) is instantiated. As such, the welcome window object creates an instance of the input window when the user clicks on the required button. Likewise, the input window creates an instance of the final window (i.e. output window) when the user clicks the output button. The GUI becomes visible

when the mainloop function is called on the root object in which the whole program resides. In order to control application flow via the clicking of buttons, callback functions were created which are only called when the tkinter backend sense a user-triggered event. For instance, when the quit button is pressed, an event is generated which in turn calls the quit function that destroys the master (i.e. root). Seeking to improve the visual appearance of our application, we created a canvas background in each window containing an artist's rendition of space.

In light of the fact that the GUI and computations were the two main components of our application, the question of how the GUI interfaces with the computations remains. Initially, we were debating writing the GUI in one language (e.g. Python) and the computations in another (e.g. C++). The main reason for doing so was optimizing for speed as Python is not necessarily the fastest of languages due to its nature as an interpreted language. After becoming comfortable with the nature of the computations, it became apparent that Python should be sufficiently fast. In other words, there was no significant speed justification for introducing the complexity of interfacing two different languages into this project. Consequently, both the Computations and GUI Python files were located within the same directory. Being that Computations was structured as a class, the GUI was able to simply import Computations and create a Computations object with the specific attributes (i.e. inputs) that the user had specified in the input window. Passing in the frequency, data rate, orbit altitude, and transmission power, an instance of Computations could be created. As such, functions that return dish diameter, latency, and antenna length for both geostationary and non geostationary orbits can be called on this instance of Computations. Specifics aside, we found that utilizing an object oriented programming paradigm on this project allowed us better collaborate. It was relatively easy to define project roles and eventually interface our code.

The development of our project can be seen below through our initial project presentation and all of our memorandums. The early stages of the project were primarily research based as we didn't understand signal transmission. Later as we made more progress the project began to take form. Our original computations and GUI classes weren't fully functional due to lapses in understanding in various electrical engineering topics. In our final stages of the project we learned a fair bit about how satellites and GUIs worked. The project remains to have flaws due to our various assumptions and limitations, which we would fix given more time.

Definitions

Satellite Ground Station: A building on the ground that is set up to take in data from a satellite and transmit data back to a satellite if necessary.

Downlink throughput: The amount of data being passed from the satellite to the ground station. This is closely related to bandwidth.

Duration of Ground Contact: The amount of time that a ground station can communicate with a satellite.

Satellite Orbit: The path a satellite traces in the sky.

Azimuth: The direction of a celestial object from an observer on the ground

Pointing Angle: The angle the antenna makes with the horizontal

Geosynchronous Orbit: an orbit in synch with the earth's rotation.

Shannon's Limit: Theoretical max information transfer rate for any broadcast link.

Gain: increasing the strength of a signal.

Altitude: The minimum distance between the satellite and ground station during its pass.

Design Considerations
(Assumptions, Project Requirements, Initial Risks and Alternatives, Description of Artifacts, Required Items)

Assumptions

- The project is done in Python with Tkinter library support
- The signal to noise ratio is equivalent to Shannon's limit(-1.6Db)
- The ground station operates at 60% efficiency
- The attenuation due to rain is 6Db
- The location of a satellite ground station is fixed.
- A satellites orbit is fixed with an overhead pass.

Project Requirements

- The projects inputs are frequency, which can be used to find Data Rate using lookup tables, altitude, and transmitter power
- Use gain and loss equations in order to find the dish diameter.
- Outputs are dish diameter, antenna length, and latency
- The antenna length is found using a half wavelength dipole.
- This project needs to be completed by December 19th, and reach objectives along the way.

Initial Risks and Alternatives

- Constructing ground stations and putting satellites into orbit are huge monetary commitments for Raytheon. Making mistakes is very costly.
- Our computations need to be precise.
- An alternative could be doing computations by hand or finding some sort of pre-existing software or libraries rather than starting from scratch.
- Mistakes aren't easy to undo after a satellite is in orbit and a ground station is constructed.

Specification Items
-Inputs used to describe the whole system

1.0 Satellite orbit
1.1 Data Rate
1.2 Frequency
1.3 altitude

1.4 Type of orbit

-Things needed to be established

2.0 Distance between satellite and station(changes during the pass)

2.1 Space(Path) Loss

2.2 Antenna Gain

2.3 Power in DB

2.4 Beamwidth

Requirement Items

1.0 Dish Diameter

1.1 Latency

1.2 antenna length

Description of Artifacts

-Scheduling

Matthew Huebner

- I am the athletics commissioner for my dorm(1 hrs weekly)
- I run my dorms signature event on November 3rd(1-4 hrs weekly until event)
- I have a large midterm project for Logic Design due early November(~10 hrs weekly expected)
- Logic Design Final Project(20 hours in final weeks of project)
- ND Rocket Team(3-4 hrs weekly)
- Other Classes(4-8 hrs weekly)

Konrad Rozankski

- SAE Hybrid → 3-4 hrs weekly
- Robotics Research → 4-6 hrs weekly
- Club Soccer → 2-3 hrs weekly
- Engineers Without Borders → 2-3 hrs weekly
- Student Manager for Soccer → 2-3 hrs weekly
- Other Classes → 4-8 hrs weekly

-Value

- This project brings value in that it allows Raytheon to easily compute what specifications they want to assign to their ground stations.
- Hopefully, this can be used to double check measurements so that no costly errors are made.
- Mr. Mallinger's time is deeply respected and the goal is to bring a project to him that shows the appreciation we have of the time he is giving to us.
- Raytheon isn't taking on too much risk for this project, the team is unpaid; however, Mr. Mallinger is offering his time so this project does need to produce results.

-Architecture for Implementing the Requirements

- Mr. Mallinger provided papers from industry on the constraints and inner workings of the problem we are trying to solve. Our initial plan is to go over these articles in more detail in order to understand the scope of the problem.

- The following links should be referenced when looking to recreate our code. For the tkinter application, consult its unique documentation
  https://www.getconnected.aero/2017/09/lkuka-band-satellites-mean/
  https://keisan.casio.com/exec/system/1224665242
  http://www.atmosp.physics.utoronto.ca/people/strong/phy499/section2_05.pdf
  https://www.eutelsat.com/files/contributed/support/pdf/azimuth-elevation-polarisation.pdf
  https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-851-satellite-engineering-fall-2003/lecture-notes/l21satelitecomm2_done.pdf

-Other

- There are no safety issues regarding our project; however, calculations should be double checked before implementation.
- The user interface is in a tkinter GUI application

Traceability Matrix

Requirement Items

Spec Items	**1.0**	**1.1**	**1.2**
1.0	**x**		
1.1	**x**		
1.2	**x**		**x**
1.3	**x**	**x**	
1.4	**x**		
2.0	**x**		
2.1	**x**		
2.2	**x**		
2.3	**x**		
2.5	**x**		

Goals and Timeline

Finish Reading and comprehending Mr. Mallinger's articles	**October 21st**
Write code for the first Review	**November 10th**
Code Review 1	**November 11-15**
Write code for the second Review	**November 17th**
Code Review 2	**November 18th-20th**
Develop Class Powerpoint	**November 20th-24th**
Submit PowerPoint Draft	**November 25th**
Prepare code for third Review	**December 1st**
Code Review 3	**December 2nd-6th**
Prepare final iteration of project	**December 10th**
Final Project Due	**December 16th**

Replicating Our Code

This project is highly technical and uses many assumptions. If you are recreating our code please consult the following links as they are integral to the understanding of the underlying math. Documentation for the tkinter GUI is available online. For altitude space(path) loss equations you need to sample the distance between the satellite and the ground station at many different angles as this value is not constant for an overhead pass. All the orbit calculations can be done with an introductory physics and geometry background. The gain and loss equations require some reading and potential correction, the following links should be consulted.

http://www.antenna-theory.com/basics/gain.php

https://www.getconnected.aero/2017/09/lkuka-band-satellites-mean/

https://keisan.casio.com/exec/system/1224665242

http://www.atmosp.physics.utoronto.ca/people/strong/phy499/section2_05.pdf

https://www.sciencedirect.com/topics/computer-science/noise-to-signal-ratio

https://www.eutelsat.com/files/contributed/support/pdf/azimuth-elevation-polarisation.pdf

https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-851-satellite-engineering-fall-2003/lecture-notes/l21satelitecomm2_done.pdf

http://news.mit.edu/2010/explained-shannon-0115

Memorandum 1

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger

Subject: Memorandum 1: Summary of Preliminary Discussion

What were the goals for this week?

→ Set up an interview with John Mallinger
→ Create a set of interview questions for John Mallinger

What was accomplished this week?

→ Emailed John Mallinger and set up an interview on Thursday, October 10 at 11:30am
→ Conducted the aforementioned interview

Preliminary Interview Summary (w/ Question and Answer Summary):

- In the project description there isn't a whole lot of clarification as to the direction that we can take with this project. Could you clarify a bit more about the project areas we can explore within the context of the JPSS project?

There are a lot of things that can be done with satellite ground systems: orbital tracking of satellites, managing telemetry streams, etc. We will spend some time evaluating the different programs at JPSS to find something that aligns with our interests and skills.

- Do you think that Raytheon does a good job of working through various iterations of a project?

Raytheon highly emphasizes the engineering process and evaluating their requirements to drive their engineering. Getting the right answer is important, but learning things on how to implement solutions over the course of the project is also highly emphasized.

- What are you specifically working on within the JPSS program?

The project that John Mallinger is currently working on concerns the integration of the software components for the JPSS. Most of the work that he does is software oriented.

- You mentioned that JPSS is the next generation of satellites -- what separates it from the previous generation?

This new satellite system is supposed to be far more accurate and have more civilian interaction. There will be higher fidelity signals → more integrity in the signals. The size of the constellation will stay the same. An example of the benefits includes the possible automation of civilian airplane takeoff and landings.

- Do Raytheon engineers split time between programming and hardware, or are these tasks mostly separated?

Raytheon does the ground systems for the satellites. Mostly telemetry and data processing. The computers on the satellites end up being super small due to mass constraints on orbiting objects. Therefore, it is important to do the computation work for the satellites on the ground instead of on the satellites.

- How does Raytheon account for space trash?

These satellites are in medium earth orbit (i.e. around 20,000 km above the earth's surface). Space trash really isn't that big of a deal in this orbit due to the fact that space trash mostly inhabits lower earth orbit.

- What is your favorite project while working at Raytheon?

Mallinger has worked on various projects, but has spent the vast majority of his time on the JPSS because satellites have a long term more purposeful development cycle → a single satellite can cost upwards of a billion dollars during its development cycle when accounting for labor costs and what not. He believes that he spent more time than most other employees on this project. That being said, he believes this project to be both frustrating and meaningful.

- Is signal latency (i.e. between the satellites and ground systems) a big problem?

A lot of design work needs to be focused around this. As a result, Raytheon ends up having relatively large ground systems scattered across the world. Furthermore, different satellites have different rates of data transmission. For example, weather satellites have quite high rates of data transfer. Conversely, GPS satellites have relatively little data transmission.

- Is there anything in your job that could be improved or automated by coding?

John Mallinger believes that an area that too much time is spent on is simplifying the time to change something in a factory and then rerunning tests in order to implement changes. Being able to automate the processes behind making changes would drastically improve product development time.

What are the goals for next week?

→ Continue the brainstorming process for the project proposal

Are there any critical dependencies, open problems, or other things to be aware of for the next week?

→ None of the above have yet been encountered.

How many hours were spent on each goal noted above?

→ Konrad: 1.5 hours
→ Matthew: 1.5 hours

Very Respectfully,
Matthew Huebner
Konrad Rozanski

Memorandum 2

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger

Subject: Memorandum 2: Further Development of Initial Project Proposal

What were the goals for this week?

→ Set up a meeting with Pascal Phoa (i.e. assigned TA) regarding initial project proposal
→ Review articles provided by John Mallinger regarding background knowledge
→ Research languages for creating a simple GUI (i.e. graphical user interface)
→ Finalize and submit project proposal

What was accomplished this week?

→ Emailed Pascal Phoa and set up a meeting on Thursday, October 31 at 5:15pm
→ Met with Pascal Phoa and discussed initial project proposal
→ Brainstormed and researched several viable languages for creating GUI

Summary of Meeting with Pascal:

→ Implementing a clean graphical interface will help us present our project effectively at the end of the semester
→C/C++ are languages that have long development cycles when used to create graphical interfaces
→ It is important to remember that we are not limited to just C/C++
→ A goal of this project is to experience the software development cycle, a part of which is identifying the best tool for a given task and learning to use it.
→ We should look for languages that have significant graphical interface libraries.
→ Pascal suggested using Python, Java, or C#
→ It may be possible to complete the heavy calculations using C/C++ while using another language better suited to graphics for the GUI

Changes in Approach:

→ We may have to abstract away some of the theoretical complexity through the use of libraries as the time allotted for the project may not be adequate.

Approved Project Proposal

Memorandum 3

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger

Subject: Memorandum 3: Finalizing Project Tools

What were the goals for this week?

→ Set up an phone interview with John Mallinger
→ Start making design decisions for what project tools we plan on using.

What was accomplished this week?

→ Emailed John Mallinger and set up an interview on Thursday, November 7th
→ Conducted the aforementioned interview.
→ Made design decisions for our programming language and scope of project.

After the phone call with John Mallinger, we believe that this project will be best completed using Python to do both the computation and GUI. Initially, we thought that using C/C++ to complete the computations and Python to complete the GUI would be most practical. That said, John Mallinger suggested that doing so might in fact be counterproductive in the way that it could introduce unnecessary complexity into the project. In the case that we believe it to be absolutely necessary to complete a computation in C/C++, we can always wrap it in Python without too much trouble.We also discussed the viability of creating a visual demo for the class presentation of the project at the end of the term. In order to accomplish this, John Mallinger suggested using Worldwind (developed by NASA) and Google Earth. It was also suggested that we could look to introduce other parameters into the function that determines the optimal antenna if we succeed in creating one that uses the basic parameters (e.g. throughput and duration of contact). It appears that Tkinter would be best Python library for the development of a quick, easy to use GUI. A few useful Python libraries that could be used to abstract away computational complexity are Orbit-Predictor and Beyond. Such parameters could include environmental factors such as atmospheric conditions (e.g. overcast day vs clear day). The conversation finished with us suggesting that all the code be documented and put on GitHub so that it could be best reviewed by our mentor.

Wrapping C in Python:
https://intermediate-and-advanced-software-carpentry.readthedocs.io/en/latest/c++-wrapping.html

Orbit-Predictor:
https://pypi.org/project/orbit-predictor/

Beyond:
https://pypi.org/project/beyond/
Worldwind:
https://worldwind.arc.nasa.gov/

What are the goals for next week?

→ Continue the brainstorming process for the project proposal.
→ Begin coding on our project.
→ Researching documentation on various libraries we intend to use.

Are there any critical dependencies, open problems, or other things to be aware of for the next week?

→ None of the above were encountered this week.

How many hours were spent on each goal noted above?

→ Konrad: 1.5 hours
→ Matthew: 1.5 hours

Very Respectfully,
Matthew Huebner
Konrad Rozanski

Memorandum 4

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger, Pascal Phoa

Subject: Memorandum 4: First Project Spiral

What were the goals for this week?

→ Begin coding the input window for the GUI
→ Begin experimenting with how to perform the optimal antenna computations in Python using the libraries Orbit-Predictor, Beyond and NumPy
→ Learn the basics of Python 3
→ Compare and contrast Python with C++ and C
→ Finalize the inputs that will be used to determine optimal ground station specifications

What was accomplished this week?

→ Began coding the aforementioned part of the GUI using the Python library Tkinter
→ Learned the basics of OOP in Python
→ Applied OOP to the creation of the input window
→ Created a remote repository for the project on Github

What are the goals for next week?

→ Continue reevaluating the scope of the project
→ Make more progress on computations for the GUI and clean it up.
→ Researching documentation on various libraries that could be used to further simplify the computational and graphical aspects of this project.
→ Spend time determining how to interface the GUI with the computations.
→ Send John Mallinger the link to the project Github

Are there any critical dependencies, open problems, or other things to be aware of for the next week?

→ Currently we are slightly behind schedule in regard to code due to the fact that we needed to learn Python during a demanding exam week
→ Still getting used to using a new IDE in the form of PyCharm
→ Git and Github may cause unforeseen problems in ways similar to what was seen with the in-class assignment utilizing version control systems

How many hours were spent on each goal noted above?

→ Konrad: 3 hours
→ Matthew: 3 hours

Very Respectfully,
Matthew Huebner
Konrad Rozanski

Memorandum 5

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger, Pascal Phoa

Subject: Memorandum 5: Second Project Spiral

What were the goals for this week?

→ Continue coding the input window for the GUI
→ Begin coding the output window for the GUI
→ Explore the API for Google Earth and Whirlwind
→ Convert the computations for a basic overhead pass of a satellite from paper calculations to Python code
→ Finalize the inputs that will be used to determine optimal ground station specifications for a case in which the satellite passes directly overhead

What was accomplished this week?

→ Began coding the aforementioned part of the GUI using the Python library Tkinter
→ Applied OOP to the creation of the input and output windows
→ Added to the remote repository for the project on Github
→ Created a Python computations class to model the satellite's movement
→ Began design work on the form we want the GUI to take
→ Discovered that Google Earth no longer possesses a supported API
→ NASA Whirlwind only possesses an API for Java (i.e. incompatible with Python)
→ We decided we will probably use Matplotlib along with Basemap to create a 3D representation of the Earth on which we can plot the ground station and satellite

What are the goals for next week?

→ Finish the Computations class by finding the necessary groundlink throughput and data amounts to transfer as this is necessary to determine antenna size.
→ Further research the Tkinter library more to find out the best way to embed Matplotlib images and renderings
→ Spend time determining how to interface the GUI with the computations.
→ Determine how to package dependencies with a project in Python as we have several libraries upon which our code depends (i.e. we want our project to ship as one package without the user having to download the necessary libraries)
→ Send John Mallinger an email with questions regarding the specifics of the output
→ Share Github repo with John Mallinger

Are there any critical dependencies, open problems, or other things to be aware of for the next week?

→ Depending on how well the reading of the matrix computations goes for the satellite the next iteration of satellite computations could take some time to complete.
→ Being that next week is Thanksgiving break, we do not anticipate making much progress until the week after break

How many hours were spent on each goal noted above?

→ Konrad: 4 hours
→ Matthew: 4 hours

Very Respectfully,
Matthew Huebner
Konrad Rozanski

Memorandum 6

From: Konrad Rozanski, Matthew Huebner

To: Matthew Morrison

CC: John Mallinger, Pascal Phoa

Subject: Memorandum 6: Completing Project

What were the goals for this week?

→ Finish coding the input window for the GUI
→ Continue coding the output window for the GUI
→ Create Cesium account and begin exploring how to embed it within our application
→ Finish converting the computations for a basic overhead pass of a satellite from paper calculations to Python code
→ Finalize the appearance of the output
→ Schedule meeting with Pascal Phoa
→ Schedule phone call with John Mallinger
→ Create powerpoint for in-class presentation

What was accomplished this week?

→ Finished the introduction and input windows of the GUI
→ Added Pascal Phoa and John Mallinger to the remote repository on Github
→ Finished implementing functions to model base satellite case with an overhead pass
→ Finalized the number of windows in the GUI
→ Determined that Cesium will be used for 3D visualization
→ Created account on Cesium
→ Met with Pascal Phoa last Thursday
→ Checked in with John Mallinger last Thursday

What are the goals for next week?

→ Finish the Computations class by generalizing the path of the satellite in relation to the ground station
→ Further research the Tkinter library more to find out the best way to embed Cesium using the REST API
→ Finish the window in which Cesium will be embedded
→ Finish the output window
→ Add more capabilities to the program in order to account for different frequencies.

Are there any critical dependencies, open problems, or other things to be aware of for the next week?

→ Depending on how well the reading of the matrix computations goes for the satellite the next iteration of satellite computations could take some time to complete.
→ We are unsure of whether or not we will have time to account for environmental variables through the utilization of Google Earth Engine ⇒ the decision will be made sometime this week

How many hours were spent on each goal noted above?

→ Konrad: 4 hours
→ Matthew: 4 hours

Very Respectfully,
Matthew Huebner
Konrad Rozanski