

Open Research Institute Inner Circle Newsletter May 2025



Want more Inner Circle Newsletters?
Use the QR code at left or go to
http://eepurl.com/h_hYzL
and sign up.

What's Double Doppler Shift in Radar?	page 2
Open Source Cubesat Workshop 2025 Announced	page 4
Take This Job	page 5
An ORI Recommended Off-the-Shelf Photogrammetry Method for Dish Surface Mapping at DSES	page 6
Opulent Voice Protocol Development	page 8
Efficiently Using Transmitted Symbol Energy via Delay Doppler Channels	page 12
Inner Circle Sphere of Activity	back page

The Who What When Where Why

Open Research Institute is a non-profit dedicated to open source digital radio work. We do both technical and regulatory work. Our designs are intended for both space and terrestrial deployment. We're all volunteer and we work to use and protect the amateur radio bands.

You can get involved in our work by visiting <https://openresearch.institute/getting-started>

Membership is free. All work is published to the general public at no cost. Our work can be reviewed and designs downloaded at <https://github.com/OpenResearchInstitute>

We equally value ethical behavior and over-the-air demonstrations of innovative and relevant open source solutions. We offer remotely accessible lab benches for microwave band radio hardware and software development. We host meetups and events at least once a week. Members come from around the world.

What's Double Doppler Shift in Radar? Or, Why Do We Have a Factor of Two?

11 May 2025

As part of our ongoing work to provide an Earth-Venus-Earth (EVE) communications link budget for amateur radio and citizen science use, we have updated several sections of the Jupyter Lab notebook.

Link Budget Document Locations

A PDF version of the EVE link budget (lots of updates) is here: https://github.com/OpenResearchInstitute/documents/blob/master/Engineering/Link_Budget/Link_Budget_Modeling.pdf

The Jupyter Lab notebook is available right here: https://github.com/OpenResearchInstitute/documents/blob/master/Engineering/Link_Budget/Link_Budget_Modeling.ipynb

Explaining Doppler Spread

Doppler spread happens when our signal reflects off a rotating structure, like Venus. Part of Venus (the East limb, on the right side of Venus as viewed from Earth) is moving towards us. And, part of Venus is moving away from us (the West limb, on the left side of Venus as viewed from Earth). The parts moving towards us cause reflected frequencies to increase. In the radar community, when a target is moving towards you, it produces a negative Doppler shift. The parts moving away from us cause reflected frequencies to decrease. In the radar community, when a target is moving away you, it produces a positive Doppler shift. The signal reflected off the center of the planet is reflected back with little to no frequency change. Doppler spread is a quantification of how muddled up our reflected signal becomes due to the rotation of the reflector.

Gary K6MG writes "A rough calculation of venus limb-to-limb doppler spreading @ 1296MHz:
 $4 * \text{venus rotation velocity } 1.8 \text{ m/s} / 3e8 \text{ m/s}$

* $1.296e9 \text{ c/s} = 31 \text{ c/s}$. This calculation is the same as K1JT uses for EME in Frequency-Dependent Characteristics of the EME Path and is motivated from first principles. One edge of venus is approaching the earth at a 1.8m/s velocity relative to the center of venus, the other edge is receding at the same velocity giving one factor of 2. The other factor of 2 is due to the reflection, the wave is shortened or lengthened on both the approach and the retreat."

You may be curious as to how the wave can be shortened or lengthened on both the approach and the retreat.

The double Doppler shift happens because radar involves a two-way journey of the radio wave. For example, a station on Earth may transmit a radio wave at exactly 1296 MHz. This wave travels toward Venus. When the wave hits a moving point on Venus's surface (like the East or West limb), the wave's frequency as experienced by that point is different from what was transmitted.

If the point is moving away from Earth, it "sees" a lower frequency (in other words, a redshift). If the point is moving toward Earth, it "sees" a higher frequency (in other words, a blueshift). The wave bounces off part of Venus's surface. The wave is now re-emitted at the shifted frequency that the moving point "sees" or experiences. So the reflected wave already has one Doppler shift applied. As this already-shifted wave travels back to Earth, a second Doppler shift occurs. This is because the reflecting point is still moving, so the wave gets compressed or stretched again. The wave returns to Earth with two Doppler shifts applied. Therefore, there is a factor of two in the equation explained by Gary K6MG.

An example with real-world objects can help visualize what's going on. Imagine throwing a tennis ball at a person on a moving train, and the train is coming right at you. You throw the ball to the person on the train at 10 mph. To the person on the train, your 10 mph ball appears to be moving faster or slower depending on the

train's direction. Since they are moving towards you, your 10 mph ball's speed is added to the train's 30 mph speed, and the ball would arrive at what felt like 40 mph. From their perspective, they received a 40 mph ball. Now they are just going to "reflect" the ball back at you at the same relative force as they received it. Now the ball is coming back to you at that 40 mph plus the velocity of the train, which is 30 mph. So you are going to be catching a 70 mph fast ball! When you receive the ball, its speed has been affected twice by the train's motion.

This works in the other direction as well. Now that the train is moving away from you, it's a lot harder to catch up to. So, you get a baseball pro, your friend Shohei Ohtani, to throw the ball for you. The train is moving away from you at 30 mph. Ohtani throws the ball at 100 mph. The person on the train catches it. It arrives at what feels like 70 mph to the person on the train. He tosses it back to Ohtani at 70 mph, because that's how fast it arrived, relative to him. The train takes another 30 mph off the velocity of the ball, and Ohtani catches a ball

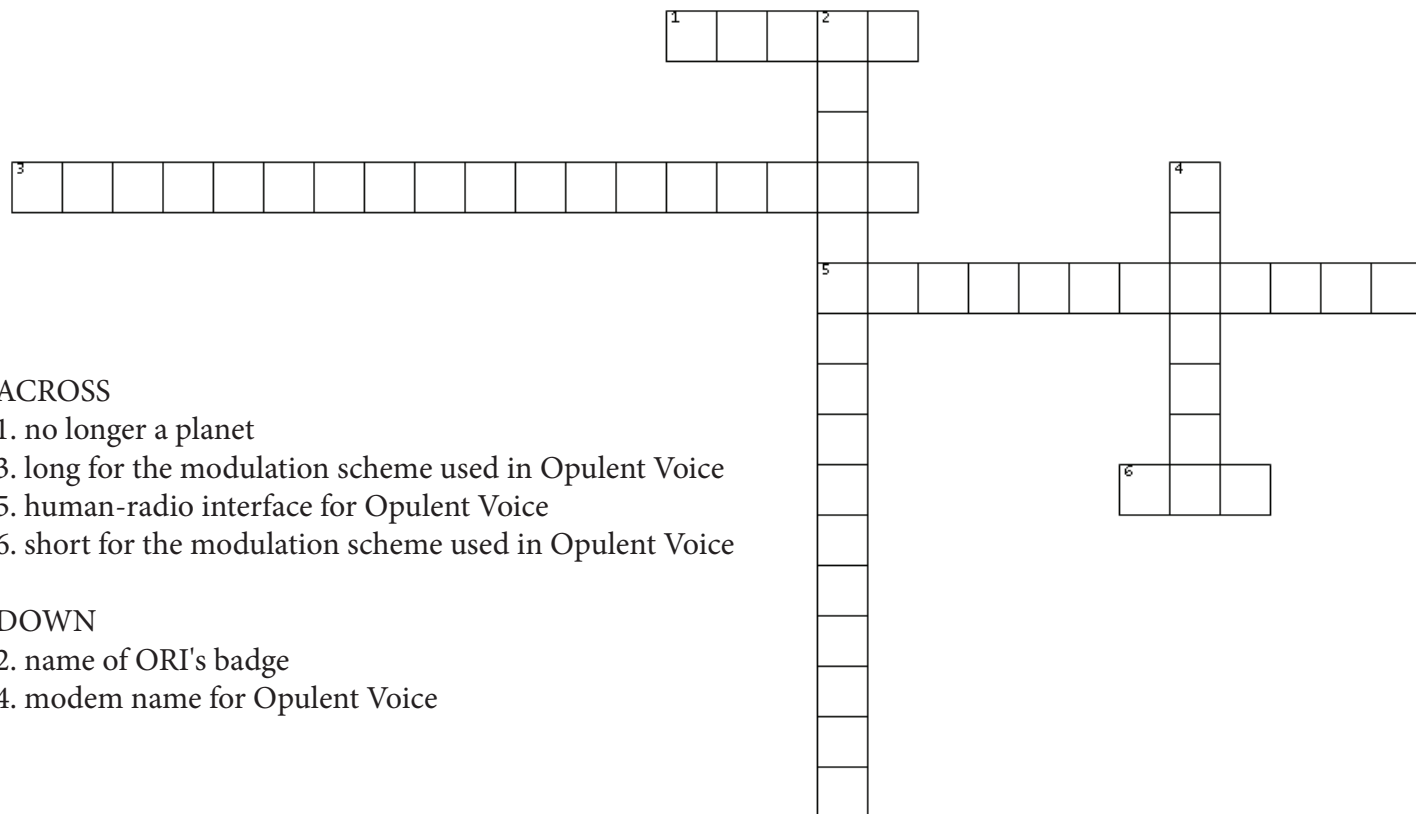
going 40 mph.

The important insight about radar returns off of moving objects is that the reflection does not simply "bounce back" the original frequency. The moving reflector actually re-emits the signal at the Doppler-shifted frequency it receives, and then this already-shifted frequency undergoes a second Doppler shift on its return journey.

Doppler Shift Challenge

While driving your car, you are stopped for running a red light. You tell the police officer that because of the Doppler shift, the red light (650 nm) was blueshifted to a green light (470 nm) as you drove towards the stop light. How fast would you have to be going in order for this to be true?

- A) 0.5 times the speed of light
- B) 0.3 times the speed of light
- C) 0.1 times the speed of light
- D) 0.01 times the speed of light
- E) 70 mph



ACROSS

- 1. no longer a planet
- 3. long for the modulation scheme used in Opulent Voice
- 5. human-radio interface for Opulent Voice
- 6. short for the modulation scheme used in Opulent Voice

DOWN

- 2. name of ORI's badge
- 4. modem name for Opulent Voice

Open Source Cubesat Workshop 2025 Announced

<https://events.libre.space/event/9/overview>

Want to represent ORI? Join our Slack workspace and participate in the process of applying to present a talk, workshop, or roundtable. Information from Libre Space Foundation about the workshop is below:

Reignite the Open Source CubeSat Workshop!

Join Libre Space Foundation again for a two-day workshop to see how the open source approach can be applied to CubeSat missions with a focus on innovative and state-of-the-art concepts!

Open source software and hardware is empowering and democratizing all areas of life, so why not apply it to space exploration? The Open Source Cubesat Workshop was created exactly for that: to promote the open source philosophy for CubeSat missions and beyond. The sixth edition of the workshop takes place in Athens, Greece, hosted by Libre Space Foundation.

CubeSats have proven to be an ideal tool for exploring new ways of doing space missions; therefore, let's remove the barrier of confidentiality and secrecy, and start to freely share knowledge and information about how to build and operate CubeSats. This workshop provides a forum for CubeSat developers and CubeSat mission operators to meet and join forces on open source projects to benefit from transparency, inclusivity, adaptability, collaboration and community.

The focus of this year's workshop is to develop and apply open source technologies for all aspects of a space mission. The target audience of this workshop is academia, research institutes, companies, and individuals.

Starts Oct 25, 2025, 10:00 AM

Ends Oct 26, 2025, 6:00 PM

Europe/Athens

Serafio of the Municipality of Athens

19 Echelidon Street & 144 Piraeus Street 11854, Athens Greece

The workshop is free of charge but limited to 200 people.

“Take This Job”

1 April 2025

Interested in Open Source software and hardware? Not sure how to get started? Here's some places to begin at Open Research Institute. If you would like to take on one of these tasks, please write hello@openresearch.institute and let us know which one. We will onboard you onto the team and get you started.

Opulent Voice:

- Add a carrier sync lock detector in VHDL. After the receiver has successfully synchronized to the carrier, a signal needs to be presented to the application layer that indicates success. Work output is tested VHDL code.
- Bit Error Rate (BER) waterfall curves for Additive White Gaussian Noise (AWGN) channel.
- Bit Error Rate (BER) waterfall curves for Doppler shift.
- Bit Error Rate (BER) waterfall curves for other channels and impairments.
- Review Proportional-Integral Gain design document and provide feedback for improvement.
- Generate and write a pull request to include a Numerically Controlled Oscillator (NCO) design document for the repository located at <https://github.com/OpenResearchInstitute/nco>.
- Generate and write a pull request to include a Pseudo Random Binary Sequence (PRBS) design document for the repository located at <https://github.com/OpenResearchInstitute/prbs>.
- Generate and write a pull request to include a Minimum Shift Keying (MSK) Demodulator design document for the repository located at https://github.com/OpenResearchInstitute/msk_demodulator
- Generate and write a pull request to include a Minimum Shift Keying (MSK) Modulator design document for the repository located at https://github.com/OpenResearchInstitute/msk_modulator
- Evaluate loop stability with unscrambled data sequences of zeros or ones.
- Determine and implement E_b/N_0 /SNR/EVM measurement. Work product is tested VHDL code.
- Review implementation of Tx I/Q outputs to support mirror image cancellation at RF.

Haifuraiya:

- HTML5 radio interface requirements, specifications, and prototype. This is the primary user interface for the satellite downlink, which is DVB-S2/X and contains all of the uplink Opulent Voice channel data. Using HTML5 allows any device with a browser and enough processor to provide a useful user interface. What should that interface look like? What functions should be prioritized and provided? A paper and/or slide presentation would be the work product of this project.
- Default digital downlink requirements and specifications. This specifies what is transmitted on the downlink when no user data is present. Think of this as a modern test pattern, to help operators set up their stations quickly and efficiently. The data might rotate through all the modulation and coding, transmitting a short loop of known data. This would allow a receiver to calibrate their receiver performance against the modulation and coding signal to noise ratio (SNR) slope. A paper and/or slide presentation would be the work product of this project.

An ORI Recommended Off-the-Shelf Photogrammetry Method for Dish Surface Mapping at DSES

Equipment Needed

DSLR or Mirrorless Camera: Any modern camera with 20+ megapixels (Canon, Nikon, Sony, etc.) Higher resolution is better, but even a good smartphone camera can work in a pinch. Fixed focal length (prime) lenses are preferred for consistency.

Tripod: For stable, consistent images.

Reference Markers: Printed QR codes or ArUco markers. Reflective targets (can be made with reflective tape on cardboard). Tennis balls or ping pong balls (bright colors work well).

Measuring Tools: Laser distance meter for scale reference. Long tape measure. Spirit level

Software: Agisoft Metashape (\$179 for Standard edition) OR free alternatives like Meshroom, COLMAP, or even WebODM

Process

Prepare the Dish: Place reference markers in a grid pattern across the dish surface. For an 18m dish, aim for markers every 1-2 meters. Include some markers at known distances for scale reference.

Capture Photos: Take photos from multiple positions around the dish, high and low. Ensure 60-80% overlap between adjacent images. Capture both wide shots and detail shots. For best results, take like 100+ photos of the entire structure. Take photos under consistent lighting (overcast days are best).

Process the Images: Import photos into photogrammetry software. Align photos to create a sparse point cloud. Generate a dense point cloud. Create a mesh and texture if desired. Will be very nice in presentations! Scale the model using your known reference distances.

Analysis: Export the point cloud as a CSV or PLY file. Use MATLAB, Python (with NumPy/SciPy), or specialized surface analysis software and use appropriate fitting functions to “fit” the point cloud to an ideal parabola. Calculate root mean square (RMS) error from the ideal surface. Generate contour maps of deviation. More great images for presentations!

Special Considerations for Radio Dishes

Dish Position: Ideally position the dish at zenith or a consistent angle for the full survey.

Reference Frame: Establish the feed support point and dish center as key reference points!

Example Workday Plan

Morning Session: Set up markers across the dish (2-3 hours)

Midday Session: Photograph dish from multiple angles (2-3 hours)

Afternoon/Evening: Process initial results on-site to verify quality (1-2 hours) don't leave until the software workflow is confirmed!

Follow-up Analysis: Detailed surface error mapping and calculations (1-2 days or more)

This approach has been successfully used by amateur radio astronomy groups and small observatories to characterize dishes of similar size to DSES's 18m antenna. The resulting data can directly inform the noise temperature calculations and help verify the antenna efficiency value (I believe 69%) we are currently using in our EVE link budget.

For DSES specifically, this could be implemented as a weekend workshop activity that would not only produce valuable technical data but also serve as an educational opportunity for members.

Do you see a way to improve this proposed process? Comment and critique welcome and encouraged.

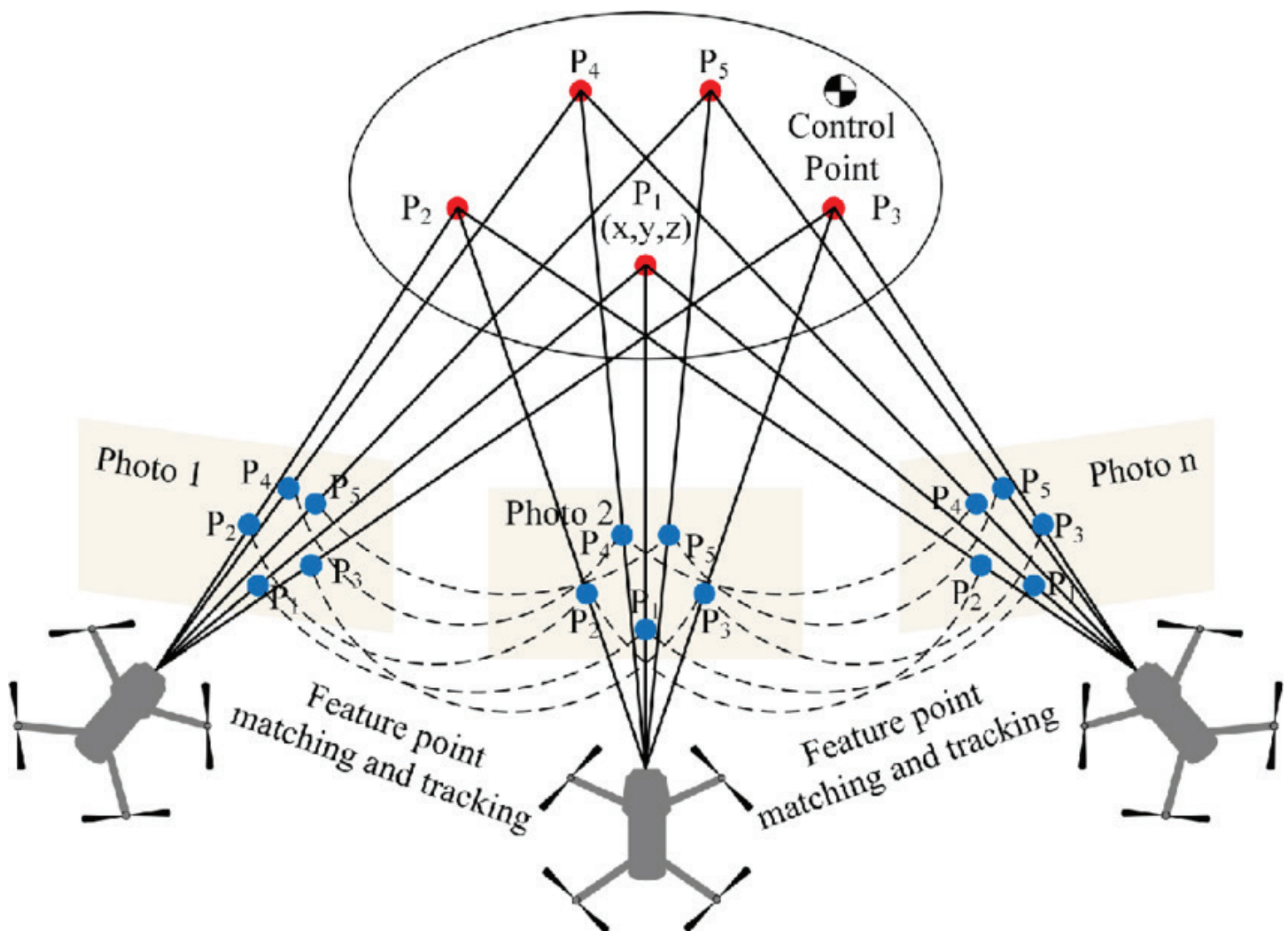


Illustration-of-3D-spatial-reconstruction-based-on-SfM-algorithm.png from ResearchGate. This is an illustration of how a drone might assist with photogrammetry of a large dish surface.

Opulent Voice Protocol Development

Opulent Voice is an open source high bitrate digital voice (and data) protocol developed by ORI. It's designed as the native digital uplink protocol for ORI's broadband microwave digital satellite transponder project. Opulent Voice (OPV) can also be used terrestrially.

Locutus

The focus of most of the recent work has been on the minimum shift keying (MSK) modem, called Locutus. The target hardware for implementation is the PLUTO SDR from Analog Devices. See

https://github.com/OpenResearchInstitute/pluto_msk for source code, documentation, and installation instructions.

With the modem in an excellent "rough draft" state, with positive results in over-the-air testing, work on the human-radio interface has begun.

Introducing "Interlocutor"

The human-radio interface is the part of the design that takes input audio, text, and keyboard chat and processes this data into frames for the modem. It also handles received audio, text, and data. The target hardware for this part of the design is a Raspberry Pi version 4, and the project name for this part of the radio system is called Interlocutor.

Why Not Just Use the PLUTO for Everything?

Why don't we use the PLUTO SDR for handling the input audio, text, and data? For two reasons.

First, the field programmable gate array (FPGA) on the device, a Zynq 7010, can handle the modem design, but there isn't a lot of space left over after it's programmed to be an Opulent Voice transceiver. The ARM processor on the Zynq 7010 is a dual-core ARM Cortex-A9. This processor is fully up to the task of running many

of the functions of an SDR, but experiments with encoding with Opus on the A9 caused some performance concerns.

Second, there are no dedicated microphone or headset jacks on the PLUTO SDR. Everything, a keyboard, monitor, and headphones, would need to be connected through USB. While this is a potential solution, we decided to try separating the two parts of this radio system at the natural dividing line. Namely, the stream of protocol frames that are ready to process into symbols. We do quite a lot of work to prepare the frames, as the voice, control, text, and data are multiplexed in priority order. Voice is encoded with Opus, and there are two different types of forward error correction. By doing all this work on a Raspberry Pi, the frames can then be sent out over Ethernet to the radio. This enables remote operation as well as providing a useful modularity in the design. As long as the modem knows what to do with the received frames, an Opulent Voice signal is created and sent out over the air. The same Raspberry Pi codebase can interface to any Opulent Voice modem, as long as it has Ethernet. This division of labor also easily enables Internet-only use.

No Replacement for Demonstration

It doesn't work until it works "over the air", and Opulent Voice protocol development is no exception. Turning good ideas into actually physically working implementations is the real value of an open source effort. There are always things that are discovered when drawings and sequence diagrams become code and that code is run on real devices.

A Raspberry Pi was set up with Raspian OS. Python was chosen as the language to develop an Interlocutor. A momentary (push) switch was connected to general purpose input and output (GPIO) pin 23 and a light emitting diode (LED) was connected to GPIO pin 17.

The first order of business was to get GPIO input (push to talk button) and output (push to talk indicator light) working. After this, audio

functionality was sorted out. A USB headset microphone was connected, the pyaudio library was used, and after some back-and-forth, receive and transmit audio was confirmed present.

Next was installing of opuslib, which handles the encoding of audio with the Opus voice encoder. This was the point where a Python environment was necessary to have, as library management became increasingly complex. Python environments are software "containers" for the all of the custom configurations needed to make interesting things happen in Python. Environments are the recommended way to handle things. After the environment was set up, the next challenge was to restrict Opus to constant bit rate of 16 kbs and to fix the frame length at 80 bytes. This was not as straightforward as one might hope, as direct access to the constant-vs-variable bit rate settings in opuslib did not appear to be available. Eventually, the correct method was found, voice data was sent to a desktop computer across the Remote Labs LAN, and a receiver script in Python played the audio sent from the Raspberry Pi. This was a great "lab call" milestone.

Getting the voice stream established allowed for iterative development. With each new function, voice transmission could be tested. If it stopped working, then it was highly likely that the most recent change had "broken" the radio.

Chat was added and iteratively improved until it was successfully integrated with voice. Voice has a higher priority than text. Text chat messages will not interrupt a voice call. They will be queued up and sent as soon as the push to talk button is released. One can also have just keyboard-to-keyboard chat over the air, similar to RTTY. Having the option to use voice or chat with the same radio and the same user interface is one of the goals of Interlocutor.

Current codebase can be found at <https://github.com/Abraxas3d/interlocutor>

Reducing Uncertainty

The protocol specification had control messages, for assembly, acquisition, access, authorization, and authentication, or A5 for short, as the highest priority traffic. These messages could therefore interrupt voice, text, or data communications.

A5 messages can be quite large. If these messages are very infrequent, then voice dropouts might be considered tolerable. However, we don't really know how often a satellite will do authorization or authentication checks. That is up to the satellite or repeater operator. They may have a policy of anyone being able to use the resource at any time without any checking. Or, they may enforce any number of restrictive policies. There may be policies for emergency communications use, or a lot of A5 traffic for safety or shutdown reasons during mode changes. Since A5 traffic is the highest priority traffic, and since we really don't know how often these messages will arrive, then we don't know how much damage to voice calls will happen. Interlocutor development raised this issue. The solution currently being tested reverses the priority level of voice and A5 messages.

This means that an amateur operator could access the communications resource at most once with a voice call, before an A5 message could inform the station that it was not authorized or authenticated. This situation would occur if the satellite or repeater was permissive about access (which is something we do want, because we want people using their radios and talking to other hams) and let the amateur operator transmit first and "asked questions" later.

For example, an operator has been behaving badly on the air or harassing other operators. Their station ID may be added to a block list at the repeater. The repeater checks authentication and authorization, but only after people successfully access the resource. This would catch any blocked operators, but one voice

transmission could potentially get through before the second-highest-priority A5 traffic was processed.

An obvious solution would be to do a full system authentication and authorization before any transmission is allowed at all. The downside to doing this is that it adds latency to accessing the satellite or repeater. We should try our very best to not have any worse performance in terms of ease of use when compared to a traditional FM repeater. Access latency is something that we are carefully tracking and monitoring during development, so any time the protocol can be modified to improve either the appearance or the actuality of latency, that is the choice we will make.

Accessibility

Using a Raspberry Pi (or similar computer) for Interlocutor also allows co-development for accessibility requirements. Opulent Voice must be as easy to use as possible. The Python script is currently a terminal program, but is designed to also be able to use an HTML5 interface. This is the second major modularity in the radio system.

The HTML5 interface is not in the code yet. Full functionality through a command line terminal access will be completed first, then the HTML5 interface written to expose all of the configuration and input and output from the terminal program. The goal is exactly the same behavior whether you are using Interlocutor from the command line or from a web browser.

Being able to run a radio using a browser means that the radio functions can be available on a very large number of devices, from ordinary computers using a keyboard or mouse, to screen readers, audio browsers, devices with limited bandwidth, old browsers and computers, and mobile phones and touch devices. It can be used by those with disabilities, senior citizens, and people with low literacy levels or temporary illness.

Our four areas of accessibility for Interlocutor are hearing, mobility, cognitive, and visual. HTML5 interface provides an enormous quantity of leverage in all four of these categories. While it may not be possible to design a perfectly accessible radio, we are committed to designing one as close to perfect as we can get. Accessibility improves the experience for everyone, so it's more than worth doing.

Hearing

If you can't hear, then how do you use our radio? First, Interlocutor has keyboard to keyboard or text chat mode built in. As long as you can see the screen and use a keyboard, you can use Opulent Voice. What about the voice traffic? There are several options. An operator may already have a trusted third party speech to text going on. However, since Opulent Voice uses Opus voice encoder, and since the voice quality is quite good, we can include speech-to-text in Interlocutor. There are cloud and local options for the Raspberry Pi, and we're going to test Whisper <https://github.com/openai/whisper> as well as Faster-Whisper (fwhisper) <https://github.com/SYSTRAN/faster-whisper>

If the performance can't be done with on-board solutions, then one of the cloud-based options will be considered. Speech-to-text would be one of the run-time options for Interlocutor.

All of the controls must be made easy to use without hearing. This means that the HTML5 layout will prioritize simplicity and accessibility and give clear cues about audio status. We will need specific advice here on how to lay out the page.

Mobility

If you have limited mobility, but you can use a browser, then you can operate Opulent Voice. The HTML5 interface will be simple and easy to use for people with limited movement. We understand that we need to support as wide a variety of inputs as possible and not get in the way of or prevent the use of any trusted

third-party solutions that the radio operator already has up and running on their computer or device. We will not have keyboard traps in the interface, where you can "tab" in to a configuration setting, but then can't "tab" back out. Want to take Interlocutor with you? The entire radio system, even as a prototype build, is not heavy or excessively bulky. The parts are all available off the shelf and can be set up all in the same location or in different locations, connected by Ethernet. Interlocutor can be with you wherever you are in your house, and the radio can be installed in a radio room. The push to talk button is on a GPIO pin in the current code base, it provides a hardware interrupt with minimal latency to the Interlocutor code, and it is very simple design to have a physical button on the radio case. However, this does mean you have to press down on the button as long as you want to talk. Someone with limited mobility may find this not easy or not even possible. A USB foot switch, to give just one example, is a very common way to activate a radio and may be a more accessible inclusion.

Cognitive

Configuration options will stay open as long as necessary for a radio operator to understand and select them. While Opulent Voice is an advanced and technical mode, it should not require the operator to be advanced or technical to fully enjoy it. There will not be any flashing lights or complicated graphics. The vocabulary for configuration will be consistent and simple. Since the code is open source, additional complexity, or an even simpler interface, can be exposed if desired with alternate HTML5.

Visual

Interlocutor must not get in the way or prevent the use of any third-party accessibility technologies that are used for visual impairments. It also must be as accessible as possible on its own. For visual accessibility, high-contrast displays, large buttons, and braille labels are commonly used. A Raspberry Pi in a case doesn't have a lot of room for Braille labels,

but incorporating them into a 3d printed case would be a good start. Some radios include voice prompts for navigation and settings, or integration with screen readers. There are radios that have incorporated braille keys. Part of the process of developing Interlocutor will be identifying what methods can be easily integrated into the code base.

What is Next?

What comes next? The current codebase has quite a bit of functionality. It delivers audio and chat frames, with priority Opulent Voice encoding, over Ethernet. Opus frames are the lowest layer of the protocol stack, and the Opulent Voice headers are the highest. In between, we intended to put RTP, UDP, and IP layers. This means that the stream of frames can be handled by a very large number of existing applications. Those layers are being added and tested now.

A decision will be made whether this additional overhead is worth doing or not. The only way we can make that decision with any confidence is to test the assumptions with real-world use.

Error correction needs to be fully implemented, and a COBS encoder/decoder https://github.com/OpenResearchInstitute/cobs_decoder for distinguishing the frame boundaries of the non-Opus data.

Want to Help?

If you would like to help make these and the many other things that we do happen more quickly or better, then you are welcome at ORI.

Read over our code of conduct and developer and participant policies on our website at <https://www.openresearch.institute/developer-and-participant-policies/>, and then visit our getting started page at <https://www.openresearch.institute/getting-started/>

Efficiently Using Transmitted Symbol Energy via Delay-Doppler Channels – Part I

Pete Wyckoff, KA3WCA

Introduction

There are few DSP avenues to improve link margin for extremely weak signal amateur radio communications. Prior research into modulation and coding already efficiently uses energy that actually reaches the demodulator. This is not the only problem though. A separate and important problem via linear time-variant (LTV) multi-path channels is making efficient use of the constrained energy-per-symbol transmitted by an emitter.

This paper introduces an efficiency measure as the ratio of energy-per-symbol reaching the receiver (E_s) vs. energy-per-symbol leaving the transmitter antenna (E_{tx}). This is used to quantify advantages of DSP waveform design. For a single path line-of-sight channel, this metric in dB equals the receiver antenna gain minus the path-loss. For a single path channel, the specific transmit waveform shape does not matter. For a channel with more than one path, the efficiency metric becomes a function of the specific transmit waveform. As a result, some DSP waveform designs more efficiently transfer energy from the transmitter to the receiver — *exploiting rather than mitigating multi-path*. The goal is developing theory to inform future over-the-air amateur radio experiments possibly via the LTV Earth-Moon-Earth (EME) channel.

General Problem Statement & Performance Metric Definition

Figure 1 shows the model for this problem. The transmitter antenna emits $x(t)$ as a function of time. This signal undergoes path loss and passes through the channel with M-paths as represented by impulse response $h_M(t)$. After the receiver antenna gain, the signal is $y(t)$.

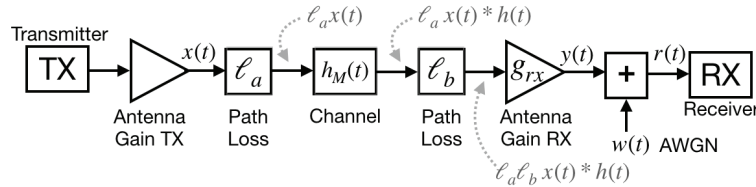


FIGURE 1: This system model for the problem is formulated generally as a moon-bounce (EME) channel yet it can be applied to many problems including direct line-of-sight.

The transmit symbol occurs during $0 \leq t < T_s$ where T_s is the symbol duration. Transmit symbol energy equals E_{tx} as the power integrated over time:

$$E_{tx} = \int_0^{T_s} |x(t)|^2 dt. \quad (\text{Equation 1})$$

Suppose the channel impulse response $h_M(t)$ paths deliver minimum path-delay τ_{min} and maximum τ_{max} during $0 \leq t < T_s$. Received symbol energy E_s equals:

$$E_s = \int_{\tau_{min}}^{T_s + \tau_{max}} |y(t)|^2 dt. \quad (\text{Equation 2})$$

Let the efficiency metric for the M-path LTV channel equal ξ_M :

$$\xi_M = 10 \log_{10} \left(\frac{E_s}{E_{tx}} \right). \quad (\text{Equation 3})$$

This metric ξ_M is useful to analyze waveform efficiency for single-path and multi-path LTV channels.

Single-Path Efficiency Metric Analysis

The main point of this section shows waveform design does not impact the efficiency metric for a single-path channel. The single path channel has $M=1$ and therefore the channel impulse response exhibits path-delay τ_1 and path-coefficient α_1 :

$$h_1(t) = \alpha_1 \delta(t - \tau_1). \quad (\text{Equation 4})$$

From Figure 1, the receive antenna output equals:

$$y(t) = \ell_a \ell_b g_{rx} x(t) * h_1(t) = \ell_a \ell_b g_{rx} \alpha_1 x(t - \tau_1). \quad (\text{Equation 5})$$

Suppose we summarize the path loss terms using $\ell = \ell_a \ell_b$. The received energy-per-symbol equals:

$$E_s = |\ell \alpha_1 g_{rx}|^2 \int_{\tau_1}^{T_s + \tau_1} |x(t - \tau_1)|^2 dt = |\ell \alpha_1 g_{rx}|^2 \int_0^{T_s} |x(t)|^2 dt. \quad (\text{Equation 6})$$

The efficiency metric for a single path equals:

$$\xi_1 = 10 \log_{10} \left(\frac{E_s}{E_{tx}} \right) = 10 \log_{10} |\ell \alpha_1 g_{rx}|^2. \quad (\text{Equation 7})$$

Example: Suppose that for a single-path channel $\ell_a = \ell_b = 1 \times 10^{-3}$. The path loss terms alone equal $-10 \log_{10} |\ell|^2 = 120$ dB. Suppose we further set $g_{rx} = 1$, and $\alpha_1 = 1$. Therefore $|\ell \alpha_1 g_{rx}|^2 = 1 \times 10^{-12}$ and the efficiency factor equals -1 times the path loss as $\xi_1 = 10 \log_{10} (1 \times 10^{-12}) = -120$ dB.

In summary, efficiency shows that less path loss improves efficiency. As well, greater receiver antenna gain improves efficiency. If the path is reflected — *moon-bounce for instance* — then a stronger reflection $|\alpha_1|$ also improves efficiency. The key-point of Equation 7 though is the efficiency for a single path is not a function of the transmitted waveform shape $x(t)$. Provided two different alternatives for the shape of $x(t)$ have the same E_{tx} , then either alternative delivers the same E_s for the single-path channel formulation of Figure 1.

Multi-path Efficiency Metric Analysis

The main point of this section shows that with more than one path ($M > 1$), the efficiency metric is impacted by the shape of the transmitted waveform $x(t)$. For an LTV channel with M paths, the channel impulse response equals $h_M(t)$:

$$h_M(t) = \sum_{m=1}^M \alpha_m \delta(t - \tau_m - \dot{\tau}_m t) \quad (\text{Equation 8})$$

The m^{th} path has magnitude and phase α_m , path delay τ_m at time $t = 0$, and rate-of-change in path delay $\dot{\tau}_m$. If the channel were linear time-invariant (LTI), then it would have $\dot{\tau}_m = 0$ for all paths. Since the channel scattering — *amateur radio moon bounce for instance* — has relative motion, the linear time-variant (LTV) channel is a more general formulation. It allows for motion. The path delay $\tau_m = r_m/c$ for a path with range r_m meters and wave propagation speed c meters/second. The rate-of-change $\dot{\tau}_m = v_m/c$ for a path with range-rate v_m meters/second.

Using this channel, the received energy-per-symbol equals:

$$E_s = \int_{\tau_{min}}^{T_s + \tau_{max}} |y(t)|^2 dt = |\ell_{g_{rx}}|^2 \int_{\tau_{min}}^{T_s + \tau_{max}} |x(t) * h_M(t)|^2 dt. \quad (\text{Equation 9})$$

The efficiency metric equals for $M > 1$ paths equals ξ_M :

$$\xi_M = 10 \log_{10} \left(|\ell_{g_{rx}}|^2 \frac{\int_{\tau_{min}}^{T_s + \tau_{max}} |x(t) * h_M(t)|^2 dt}{E_{tx}} \right). \quad (\text{Equation 10})$$

For the multi-path LTV channel, the shape of the waveform $x(t)$ impacts the efficiency metric. This characteristic makes the multi-path channel different than the single-path channel.

Using Time-Lens Effect of a Known LTI Multi-Path Channel to Boost Efficiency Metric

As a baseline, Figure 2 passes a pulsed RF signal through a multi-path LTI channel without range-rate — *in other words* $\dot{\tau}_m = 0$. For this example, $E_{tx} = 0.0$ dB and $E_s = -119.0$ dB as shown in the figure. The efficiency metric is therefore -119.0 dB for this first waveform.

For comparison, Figure 3 passes a different signal using two RF pulses through the same channel. Whereas $E_{tx} = 0.0$ dB still, now $E_s = -117.6$ dB as shown in the figure. For this second waveform, the efficiency metric is boosted to -117.6 dB. This waveform, using two pulses, is 1.4 dB more efficient than the single pulse waveform for this LTI channel.

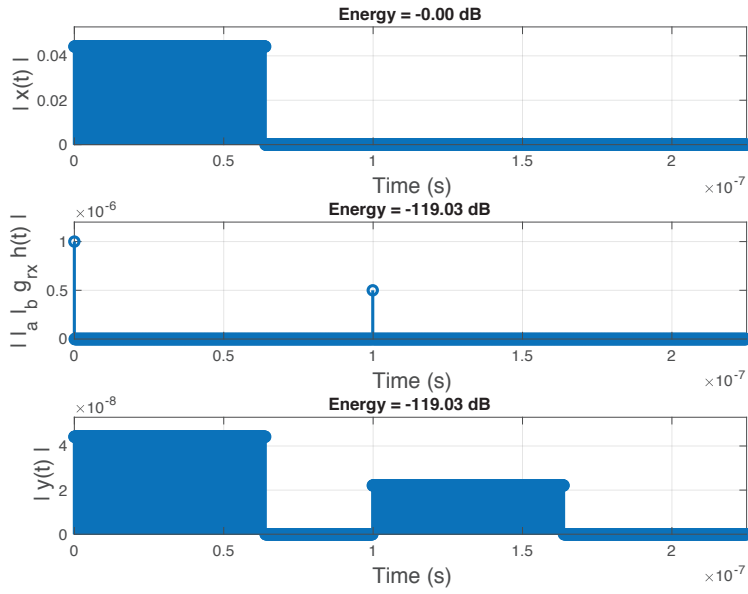


FIGURE 2: A single pulse waveform yields efficiency metric -119.03 dB.

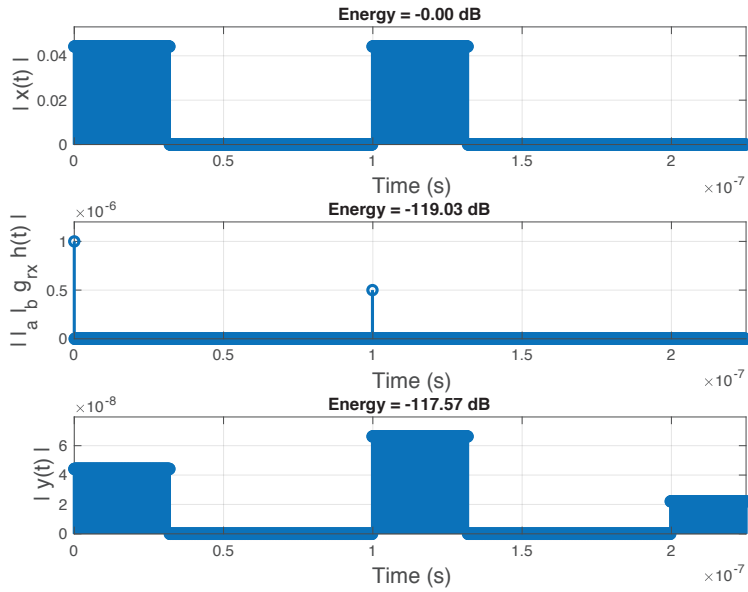


FIGURE 3: A two pulse waveform yields efficiency metric -117.57 dB.

For this channel, each input pulse produces two output pulses. In Figure 3, two input pulses delivered only three output pulses. During the middle pulse, two pulses that were transmitted at different times constructively combined to boost the field-strength. Since power is proportional to the magnitude of field-strength squared, the received waveform delivers more energy to the receiver than some other transmit waveform that does not combine pulses in this manner.

Figure 4 boosts the efficiency further by using 8 pulses. $E_{tx} = 0.0$ dB still, and $E_s = -117.22$ dB as shown in the figure. The efficiency metric is now -117.22 dB. For this channel, using these 8 pulses is 1.78 dB more efficient than the single pulse from Figure 2.

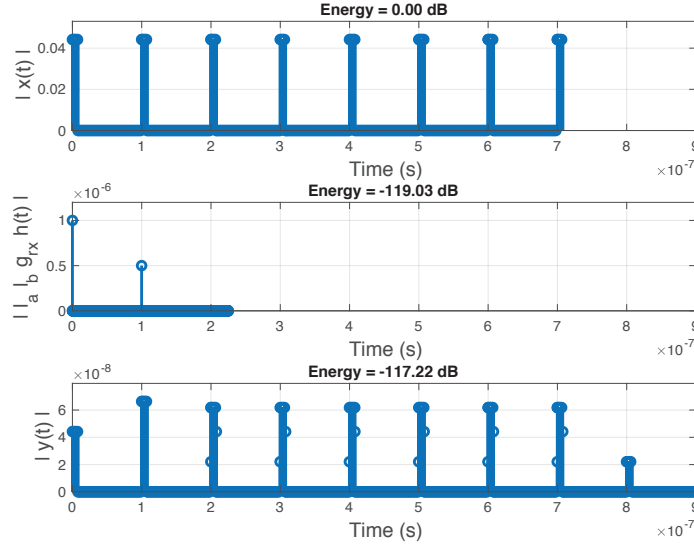


FIGURE 4: An eight pulse waveform yields efficiency metric -117.22 dB.

Since this example uses $\alpha_1 \ell g_{rx} = 1 \times 10^{-6}$ and $\alpha_2 \ell g_{rx} = 0.5 \times 10^{-6}$, the efficiency of the single short pulse from Figure 2 will be the sum of the energy delivered by each path:

$$10 \log_{10}(|1 \times 10^{-6}|^2 + |0.5 \times 10^{-6}|^2 / 1) = -119 \text{ dB.} \quad (\text{Equation 11})$$

This represents the gain from multi-path with a signal that adds in power only. The efficiency from a train of many designed pulses will be the square of the magnitude of the sum of the path coefficients, which is larger:

$$10 \log_{10}(|1 \times 10^{-6} + 0.5 \times 10^{-6}|^2 / 1) = -116.4 \text{ dB.} \quad (\text{Equation 12})$$

This represents the gain from multi-path with a signal that adds in field-strength. Thus, a very long train of many designed pulses will be up to 2.6 dB more efficient than the single pulse for this channel. Of course if there are errors in estimating the channel response, the efficiency gain will be less — *more to follow in later Part*.

Summary of Part I

Part 1 presented a waveform efficiency metric. Given a known LTI multi-path channel, waveform design examples were shown that improved the waveform efficiency metric. The particular waveforms were designed as pulsed waveforms because these serve as a foundation to build waveforms for the LTV channel in Part 2. Those waveforms use techniques from Orthogonal Time Frequency Space (OTFS) to take similar advantage of delay-Doppler channels. The goal for this series of papers is developing theory to inform future over-the-air amateur radio experiments possibly via the LTV Earth-Moon-Earth (EME) channel.

About the Author

For more than 20 years, Peter S. Wyckoff has designed and tested a wide variety of digital communications systems, modems, and antenna arrays, particularly for the satellite communications industry. He graduated from Penn State University with an MSEE in 2000 and graduated from Pitt with a BSEE in 1997. Since graduation, he has been awarded seven U.S. patents, which are mostly about co-channel interference mitigation, antenna array signal processing, and digital communications.

**Where will we go next?
Find out!**

<https://openresearch.institute/>



<https://www.youtube.com/@OpenResearchInstituteInc>

RFBitBanger

Get your kit today
while supplies last!

HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP
HF Digital QRP

<https://www.ebay.com/usr/openresearchinstitute>

\$175

Shipping
Calculated
Separately

The Inner Circle Sphere of Activity

If you know of an event that would welcome ORI, please let your favorite board member know at our hello at [openresearch dot institute](mailto:openresearch@openresearchinstitute.org) email address.

28 May 2025 - ORI was represented by two volunteers at KiCon 2025, held at University of California at San Diego. The conference is the major North American event for KiCad, a free and open-source software suite for Electronic Design Automation (EDA).

13-23 June 2025 - Michelle Thompson is on the road in Arkansas, USA and will have some free time to meet up with volunteers and supporters.

30 June 2025 - Future Amateur Geostationary Payload Definitions comment deadline.

20 Jul 2025 - Submission deadline for Open Source Cubesat Workshop, to be held 25–26 October 2025. Location is Serafio of the Municipality of Athens, Greece.

5 August 2025 - Final Technological Advisory Council meeting at the US Federal Communications Commission (FCC) in Washington, DC. The current charter concludes 5 September 2025.

7-10 August 2025 - DEFCON 33 in Las Vegas, Nevada, USA. ORI plans an Open Source Digital Radio exhibit in RF Village, which is hosted by Radio Frequency Hackers Sanctuary.

5 September 2025 - Charter for the current Technological Advisory Council of the US Federal Communications Commission concludes.

25-26 October 2025 - Open Source Cubesat Workshop, Athens, Greece.

Thank you to all who support our work! We certainly couldn't do it without you.

Anshul Makkar, Director ORI
Frank Brickley, Director ORI (SK)
Keith Wheeler, Secretary ORI
Steve Conklin, CFO ORI
Michelle Thompson, CEO ORI
Matthew Wishek, Director ORI