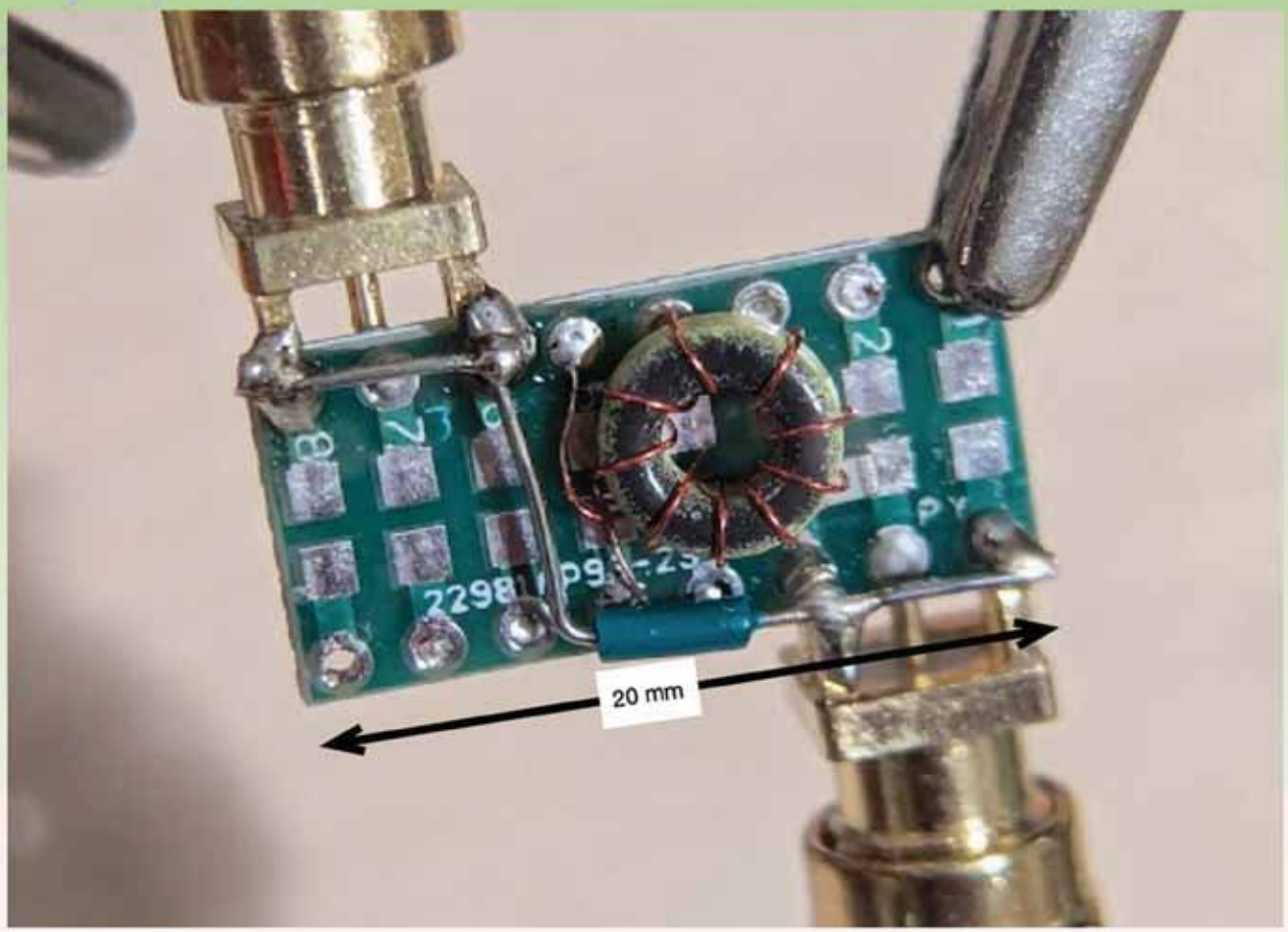
QEX
A Forum for
Communications Experimenters

September/October 2025
Issue No. 352 | www.arrl.org



Breadboard Implementation of a Simplified Filter

KENWOOD

Versatile and Flexible Radio Returns with a New Look and Soaring Functions That'll Thrill Amateur Radio Enthusiasts.

NEW TRIBANDER

TH-D75A

144 / 220 / 430 MHz TRIBANDER

Key Features

- **APRS® Protocol*1 compliant**
  To exchange GPS location data and messages in real-time.
- **D-STAR*2 with Simultaneous Reception on DV mode**
  Compatible for transferring voice and digital data over D-STAR networks.
- Reflector Terminal mode to access D-STAR Reflectors
- USB Type-C for Data Transfer and Charging
- Built-in Digipeater (a digital repeater) station to transmit received data
- Built-in GPS unit
- Easy-to-read Transflective Color TFT Display
- Call Sign Readout
- Tough & Robust – meets IP54/55 Standards
- Wide-band and multi-mode reception
- Built-in IF Filter for comfortable reception (SSB/CW)
- DSP-based Voice Processing and Reputable KENWOOD Custom Tuned Sound Quality
- Bluetooth®, microSD/SDHC Memory Card Slot for flexible link with a PC

*1: APRS® (The Automatic Packet Reporting System) is a registered trademark of TAPR (Tucson Amateur Packet Radio Corp.)
*2: D-STAR is a digital radio protocol developed by JARL (Japan Amateur Radio League).

JVCKENWOOD USA Corporation

Customer Support/Distribution Customer Support:
(310) 639-4200 Fax: (310) 537-8235 **www.kenwood.com/usa**
ADS#31823

QEX (ISSN: 0886-8093) is published bimonthly in January, March, May, July, September, and November by the American Radio Relay League, 225 Main St., Newington, CT 06111-1400. Periodicals postage paid at Hartford, CT and at additional mailing offices.

POSTMASTER: Send address changes to: QEX, 225 Main St., Newington, CT 06111-1400
Issue No. 352

Publisher
American Radio Relay League

Ronald Diehl, NQ8W
Editor

Lori Weinberg, KB1EIB
Assistant Editor

Publications Department
Becky R. Schoenfeld, W1BXY
Director of Publications and Editorial

Matt Ali
Layout & Production Specialist

David Pingree, N1NAS
Senior Technical Illustrator

Brian Washing
Technical Illustrator

Advertising Information
Janet L. Rocco, W1JLR
Business Services
860-594-0203 – Direct
800-243-7768 – ARRL
860-594-4285 – Fax

Circulation Department
Cathy Stepina
QEX Circulation

Offices
225 Main St., Newington, CT 06111-1400 USA
Telephone: 860-594-0200
Email: **qex@arrl.org**

Subscription rate for 6 print issues:
In the US: $29

US by First Class Mail: $40

International and Canada by Airmail: $35

ARRL members receive the digital edition of *QEX* as a member benefit.

In order to ensure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.

September/October 2025

About the Cover

Ted Dunning, K6BYT, uses modern tools to help you move beyond traditional formulas to create RF circuits that are fine-tuned for your specific needs in "Understanding and Using Circuit Optimization."

In This Issue:

Index of Advertisers

The American Radio Relay League

The American Radio Relay League, Inc., is a noncommercial association of radio amateurs, organized for the promotion of interest in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of the radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every three years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur," ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A *bona fide* interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters:

ARRL
225 Main St.
Newington, CT 06111 USA
Telephone: 860-594-0200
FAX: 860-594-0259 (24-hour direct line)

Officers

President: Rick Roderick, K5UR
P.O. Box 1463, Little Rock, AR 72203

The purpose of *QEX* is to:

1) provide a medium for the exchange of ideas and information among Amateur Radio experimenters,

2) document advanced technical work in the Amateur Radio field, and

3) support efforts to advance the state of the Amateur Radio art.

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main St., Newington, CT 06111 USA. Envelopes containing manuscripts and letters for publication in *QEX* should be marked Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be submitted in word-processor format, if possible. We can redraw any figures as long as their content is clear. Photos should be glossy, color or black-and-white prints of at least the size they are to appear in *QEX* or high-resolution digital images (300 dots per inch or higher at the printed size). Further information for authors can be found on the Web at **www.arrl.org/qex** or by e-mail to **qex@arrl.org**.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the Editor or the League. While we strive to ensure all material is technically correct, authors are expected to defend their own assertions. Products mentioned are included for your information only; no endorsement is implied. Readers are cautioned to verify the availability of products before sending money to vendors.

Ron Diehl, NQ8W

Perspectives

At the forefront of innovation, *QEX* remains committed to exploring the ever-expanding frontier of amateur radio science. In this issue, we present a dynamic mix of engineering challenges, cutting-edge technologies, and theoretical insights that reflect the vibrant and evolving landscape of RF experimentation.

As analog and digital RF designs continue to shrink in size while growing in precision, the need for accurate measurement of low-value surface mount components becomes increasingly critical. You can build a practical, workshop-ready method that delivers lab-grade accuracy—an invaluable resource for anyone looking to push the performance limits of their circuits.

Transitioning from hardware to digital modes, dive into a detailed examination of the Icom IC-9700's Digital Data functionality. With its SDR architecture and D-STAR capabilities, this popular VHF/UHF rig continues to generate excitement. Real-world testing puts its digital features through their paces to assess both strengths and limitations.

Artificial Intelligence is no longer just a buzzword in the RF world—it's becoming an active design partner. In this issue, you'll discover how AI can assist in iteratively optimizing circuit designs and opening new possibilities for adaptive, efficient experimentation.

On the propagation front, learn to analyze five key factors affecting HF performance using WSPR data from the WSJT-X software suite. By tapping into real-time, crowdsourced transmission reports and applying modern data visualization, you can reveal ionospheric trends and anomalies that can help you make informed and improved operating strategies.

Strong theoretical foundations are still essential. Examine a clear and accessible guide to real and complex signals—concepts every serious RF experimenter must understand, especially as digital modulation techniques grow in sophistication.

We hope this issue encourages fresh ideas, new experiments, and a deeper understanding of the principles that make amateur radio so rewarding. As always, *QEX* is your platform—keep building, keep testing, and keep sharing.

Writing for *QEX*

QEX is a forum for the free exchange of ideas among communications experimenters. *QEX* is published bimonthly.

Please send full-length *QEX* manuscripts, or share a Technical Note of several hundred words in length plus a figure or two, to **qex@arrl.org**. We pay $50 per published page for full articles and *QEX* Technical Notes. Get more information and an Author Guide at **www.arrl.org/qex-author-guide**. If you prefer postal mail, send a business-size self-addressed, stamped (US postage) envelope to: *QEX* Author Guide, c/o Maty Weinberg, ARRL, 225 Main St., Newington, CT 06111.

Ted Dunning, K6BYT
1070 Rosemont Ave., Los Altos, CA 94024 | **ted.dunning@gmail.com**

Understanding and Using Circuit Optimization

Modern tools let you move beyond rules of thumb and traditional design formulas to create RF circuits fine-tuned for your specific needs. In this article, you'll learn how to harness automated, application-specific optimization — using evolutionary algorithms and lightweight simulation — to dramatically improve circuit performance.

Introduction

Everybody who has designed and built an RF circuit has tweaked that design and coaxed a bit more performance out of it or compensated for non-ideal performance. That process of improvement through iterative design and building is a big part of the joy in amateur radio.

But what if you could make millions of adjustments instead of just a few to get your circuit to meet exactly your own idiosyncratic needs? The resulting improvements could be very dramatic, in part because they are customized to your specific needs.

How then, can you do that? One way is to automate the process of improvement.

This article describes just how to do that and shows how the improvement in performance can be dramatic. This approach is like some of the ambitious efforts in AI circuit design **[Ref.1]** that are beginning to show performance beyond what might be thought possible based on standard rules of thumb and expectations.

Fundamentally, this process works because of the combination of three major ideas. These are:

- Application-specific measurements of quality allow highly non-traditional design choices that can work much better than general-purpose designs.
- We can turn circuits into tiny computer programs. Circuits become executable simulation functions with component values as inputs and our measurement of quality as output.
- We can evolve better designs by evaluating these circuit functions for many different component values. We can then use the quality measure to pick new values to try.

Before we dig into the method and theory, let's see an example of the results.

A Better Filter

In a radio transmitter, it is typical to have an RF signal source (the exciter) that produces a low-level signal followed by one or more stages of power amplification. Some exciter designs, however, produce extraneous signals that would be amplified and result in undesired or even illegal transmissions unless removed. In modern designs, particularly with digital modes based on frequency-shift keying, it is common to use a frequency-agile clock generator such as the Si5351 as an exciter that produces substantial amounts of out-of-band signal.

To deal with this sort of issue, it is very common to use a low-pass or band-pass filter between the exciter and the power amplifiers. In the T41 software-defined transceiver **[Ref. 2]**, for instance, the filter for the 10 meter band is a 7th order LC network that can provide 30 dB of suppression at the second harmonic at 54-59.4 MHz. This filter is shown in **Figure 1** and is a conventional Butterworth filter that has very nice properties beyond what is needed for this application.

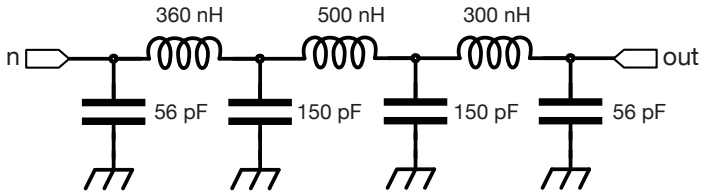


Figure 1 – The 7th order Butterworth-inspired transmit filter for the 10 meter band from the T41 transmitter. See https://github.com/peterbmarks/T41-EP-SDT for more information on the T41. Note that three custom-wound coils are needed in this design.

In contrast, however, I recently designed and tested a filter for a WSPR beacon operating in the 10m band that showed the same

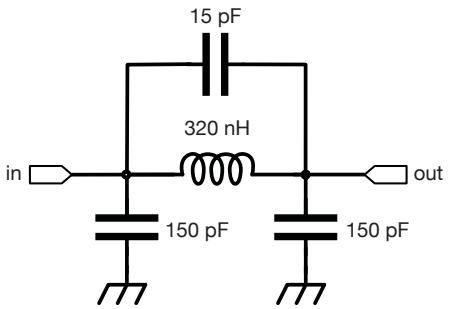


Figure 2 – A simpler filter design that exceeds the performance of a conventional design in the context of the specialized requirements for a 10m WSPR beacon.

30 dB of suppression at 50-60 MHz as the 7th order filter but only required a single inductor, as shown in **Figure 2.** This simpler filter design takes inspiration from a 3rd order inverse Chebyshev design, but the inverse Chebyshev design has less than 18 dB of suppression at the second harmonic. So, what is going on?

We know that Stephen Butterworth **[1]** was no fool, and his filter design methodology produces optimal filters given his very general design assumptions and goals. How is it that the dramatically simpler circuit in Figure 2 is possible without catastrophic (or even any) loss of performance? Did he miss something obvious?

The answer lies in the fact that in Butterworth's 1930's world of slide rules and nomograms, it was necessary to come up with closed form solutions for a few general-purpose designs that would fit a broad range of applications and require limited computation. Modern computers turn the assumptions of this design style upside down at almost every step. We no longer need to use only general-purpose designs, but we can instead take very specific design constraints into consideration. We don't have to reduce everything to simple formulas. Now we can throw a large amount of computation at the problem. The designs that result can be dramatically better.

For this design, the constraints, assumptions, and goals were:

- This filter is to be used in a single-band WSPR beacon, so we only need to consider a very narrow band of transmitted frequencies. This limited scope makes conventional design goals like phase linearity and passband flatness irrelevant.
- The key figure of merit is the power (after filtering) of the strongest harmonic relative to the carrier. Suppression of non-harmonic frequencies is irrelevant since they aren't present.
- This tuned circuit is used between gain stages and does not need to present a well-behaved output impedance to an antenna.
- The harmonics produced by the Si5351 are already somewhat suppressed with even harmonics more suppressed than neighboring odd harmonics. This means that we can trade off suppression of some harmonics to get better suppression of others.
- Component values should be those already found in my parts bin.
- Performance should be robust with respect to expected component variation.
- Insertion loss of up to 10 dB is acceptable.

These constraints and goals are very different from the ones used in general-purpose filter designs. As an old saying goes, you choose the answer you get by the question you ask.

A breadboard version of the resulting filter together with the measured performance is shown in **Figure 3**. The locations of the first through fourth harmonics are marked.

When applied to output from the actual signal source, the results are dramatic as shown in **Figure 4**. Only the odd harmonics can even peek above the noise floor and the strongest remaining harmonic is the 11th at 300MHz. In any case, all harmonics are at least 50 dB below the carrier.

The filter works by putting the filter zero between the second and third harmonic to get balanced suppression of these two harmonics. That would be unacceptable for general use because this positioning causes ripple and phase irregularities in the passband, but it works very well in this design because we get very strong suppression of the second and third harmonics. By the time we get to the fourth and fifth harmonics at 112 and 140 MHz, the normal

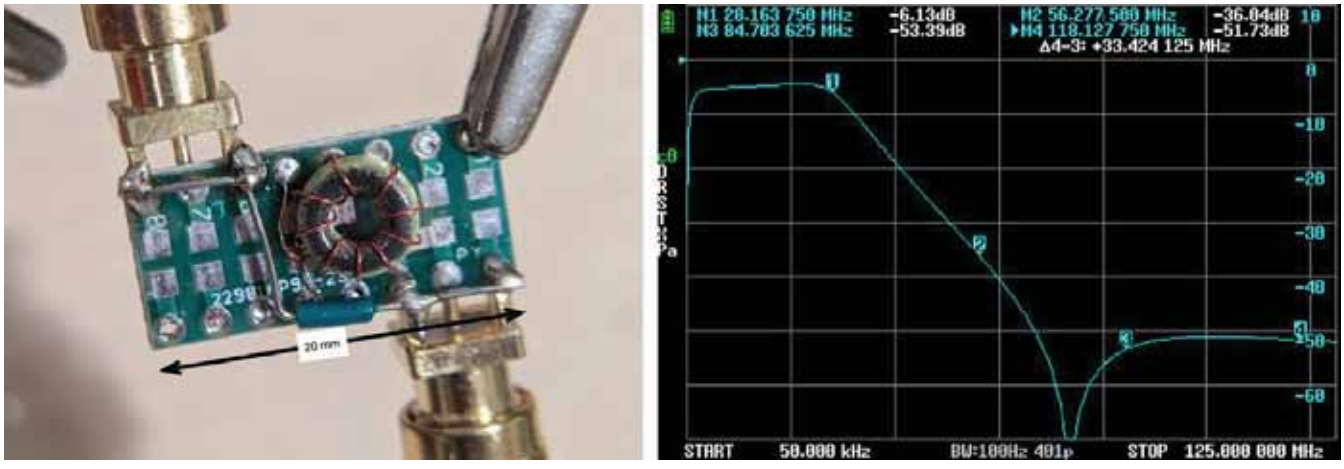


Figure 3 – A breadboard implementation of the simplified filter and the measured response of the filter. The response has the first through fourth harmonics marked and shows -30 dB response for the second harmonic compared to the first and -50 dB for the third. Note how the response below the first harmonic is not flat and the insertion gain is significant.

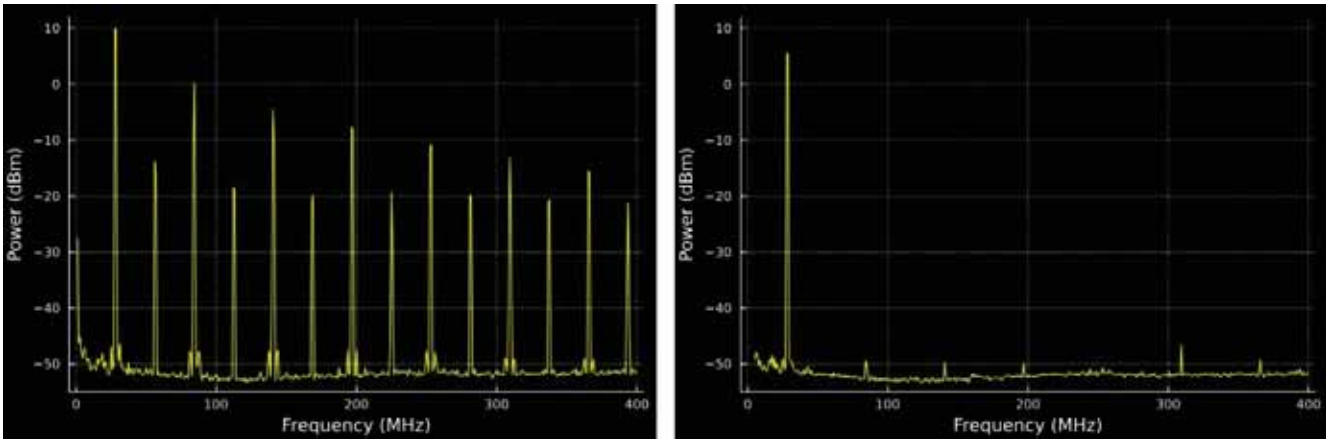


Figure 4 – Dramatic suppression of unwanted signals: The measured output spectrum from the Si5351 signal source is shown on the left and the filtered output on the right. The even harmonics are not completely suppressed in the input because the waveform is not exactly symmetrical. Note the near strong suppression of all harmonics.

attenuation of a third order filter is sufficient, particularly since these harmonics are weaker in any case.

So how did the software come up with these component values? That's the topic of the next section.

Evolution in Action

The circuit described in the previous section was designed by taking the basic form of the circuit as fixed and using an evolutionary algorithm to find good component values. There are three steps in this process — defining the figure of merit to optimize, encoding the circuit as a callable function, and the optimization process itself. Each step builds on the previous one.

Figure of Merit

The figure of merit for a circuit should mirror its actual use. For the filter in this article, the primary goal is that the combination of exciter and filter produces minimal unwanted signal energy while allowing as much of the desired signal through. Further, when we say minimal, we really mean that the strongest unwanted signal in the final output should be as small as possible, and we also want to consider the reasonable range of values each component might have given the precision of each value.

To compute the signal levels at the output, we need to know the levels at the input and the response of the filter. The datasheet for the Si5351 **[2]** indicates that the worst-case asymmetry of the output on the device is 55% duty cycle and that the rise time on the signal at full drive is about 1ns. We can use these numbers to simulate **[3]** what the output spectrum should look like to get something like **Figure 5**.

This simulated spectrum lets us see that the second harmonic is already attenuated nearly 20 dB while the third harmonic is only just a bit more than 10 dB down. We can take the amplitude of each harmonic compared to the fundamental and add the filter response at that frequency to get an estimated output after the filter to get the core of our figure of merit. The corrections that I used are shown in **Table 1.**

It is useful to add the response at the fundamental frequency with a weaker weight as secondary correction to this figure of merit to avoid the optimization algorithm from finding very strange filters that appear to have good harmonic attenuation at the cost of insertion losses in the range of 50-100 dB. Not only would it be expensive in practice to recover that much gain, but such filters probably depend on round-off effects to appear useful.

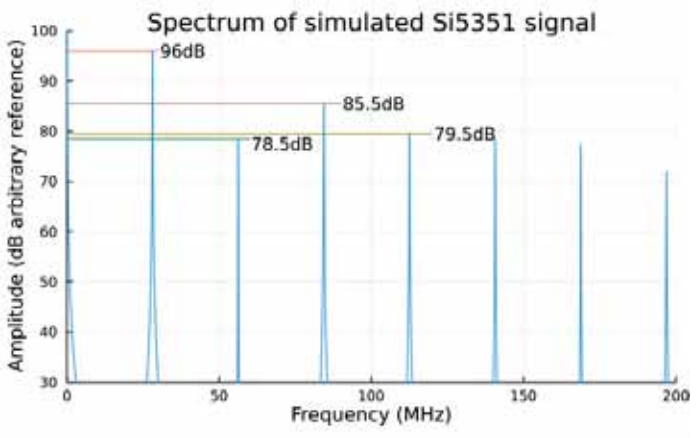


Figure 5 – The simulated spectrum of a signal with 55% duty cycle and 1ns rise/fall times shows how large the even harmonics can be. The level of the odd harmonics is not changed very much with duty cycles in the 45-55% range even though duty cycle strongly affects the even harmonics.

Harmonic	Frequency (MHz)	Input Level (dBc)
Fundamental	28.13	0
2	56.25	–19.0
3	84.38	–14.5
4	112.50	–24.0
5	140.63	–27.0

Table 1 – Simulated levels for filter input. These values will be added to the filter response to get the estimated output levels. These values are computed with a 55% duty cycle. For a duty cycle closer to 50%, the even harmonics will decrease while odd harmonics will stay nearly constant. Thus, a filter that works for these levels will work at least as well for more symmetric inputs.

Finally, to be sure that a solution is robust to component variations, we can sample the merit of 100 different solutions with random variations in component values and take the worst-case as the final figure of merit for a particular design.

Circuits as Functions

To compute the figure of merit, we need to be able to compute the filter response for specific frequencies and component values. There are several full-featured open-source circuit simulators available such as *LTSpice* **[4]**, *NGSpice* **[5]** or *Xyce* **[6]** but these systems are large complex programs optimized for the interactive simulation of large circuits. They can be made to work for the problem at hand, but integration with such large systems tends to be very cumbersome. In particular, the turn-around on the simulation of a small circuit is likely to be dominated by startup overheads. To see why this can become a problem, consider an evolutionary process that runs for 100 generations, considers 100 configurations for each generation, and runs 100 simulations on variants of each configuration. If the setup for a simulation is 10ms, then this will turn into hours of overhead.

To avoid this problem, I wrote an open-source circuit simulator in Julia called *MicroSpice* **[7]** that supports AC small signal simulations of circuits with only passive components. Using just a laptop, *MicroSpice* can run tens to hundreds of thousands of simulations per second which is ideal for evolving circuit designs. To use the simulator, you write a Spice-compatible netlist for the circuit (typically done using a schematic entry tool such as *LT Spice*) and use that to create a simulator as shown in **Listing 1**.

````
```julia
nl6 = MicroSpice.Netlist(raw"""
R1 in N001    85
L1 N001 out   $L1
C1 N001 gnd   $C1
C2 N001 out   $C2
C3 out gnd    $C3
R2 out gnd    50
""", [:L1, :C1, :C2, :C3], [:in,
:gnd], [:out])
```
````

Listing 1 – A simulator can be created in *MicroSpice* by specifying the Spice netlist of the circuit and identifying the parameters, inputs and outputs of the circuit.

This simulator is a function that accepts frequency, input voltages and a list of parameter values. The result is a list of the outputs of the circuit. We need to do just a bit more, however, since we want the maximum response for several frequencies as adjusted by input levels. We also want to vary that over many components. This added complexity is handled by the code (named *Julia*) in **Listing 2**. The quality function handles the variation of component values with a fixed 5% tolerance for all components while the q0 function handles the computation of the largest output harmonic.

````
```julia
"""
Finds a worst case value of
performance over variation of
component values
"""
function quality(sim, parameters,
frequencies, penalties)
    let n = 100
        maximum(i -> begin
          p = parameters .* (1 .+ 0.05
*randn(size(parameters)))
                    q0(sim, p,
frequencies, penalties)
         end, 1:n)
    end
end
"""
Internal function that returns the
fitness of a solution prioritizing
the net attenuation at frequencies of
interest adjusted by penalties.
There is a secondary goal the
encourages low attenuation at the
primary
frequency.
"""
function q0(sim, parameters,
frequencies, penalties)
    response = [20 *
log10(abs(only(sim(f, [2,0],
parameters)))) for f in frequencies]
    net =  (response[2:end] .-
response[1]) .+ penalties
    return maximum(net) -
response[1]/2
end
```
````

Listing 2 – The basic simulator is wrapped in code to vary component values (in quality) and compute the strongest harmonic (in q0).

Julia is very useful for this kind of computation, partly because it allows easy multi-threading, vectorized computations using functional programming (`i -> begin ... end` introduces an inline function), list comprehensions and vectorization of primitive operations (`x.+ y` is a vector sum and `f.(v)` is a vectorized application of the function `f` to vector `v`). All these expressions are compiled down to native code which is important in performance sensitive computations like this. The effect is that we get the expressivity of a language like Python, but with performance as good or better than C++. In many cases, these vectorized expressions can be compiled down to code that never even has to allocate vectors to hold intermediate values giving even higher performance.

So now we have a function to compute the figure of merit for the actual filter we are building. The next step is to find the best component values to optimize the performance of the filter.

Evolutionary Optimization

Given a simulation function that computes the quality of a design given component parameter values, we are left with the

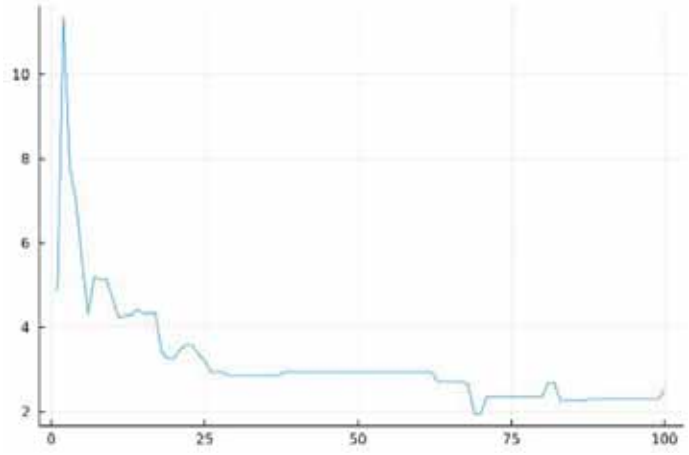


Figure 6 – The scale of mutation (vertical axis) jumps up at the beginning of an optimization and then drops down as fine-tuning becomes important. The generation number is on the horizontal axis.

problem of finding the best values. To start, I inventoried my parts bin to get all the values of the surface mount resistors, capacitors, and inductors I have on hand. In addition, I built a small function that can compute the inductance of coils built on the cores that I have and calibrated that function against observed values.

As my first optimization strategy, I wrote a program that simply iterated over all possible combinations of components. My parts box isn't as big as some people have, but this still took a long time to run. I wanted something that gets a design in less than a minute so that I can try alternative circuit configurations quickly.

To reduce the time to optimize a circuit, I switched to an evolutionary algorithm. The basic idea here is simple: you randomly select a wide range of random circuits and evaluate them. Keep the best few and iterate by generating a new population of potential solutions by mutating each of the surviving circuits by a small amount to get new candidates. In each generation, the candidate circuits are based on the best-known circuits from the previous generation, so you eventually find the best solution.

The problem with this simplest form of evolutionary algorithm is that if you create new candidates by very small mutations, it can take a very long time to find the best answer if you start with candidates that are far from the best. On the other hand, if you use large mutations, you may find the neighborhood of the best circuit quickly, but it will be hard to make the small changes needed to find the very best answer. What is needed is a way to start with big mutations and then, over time, reduce the scale of mutation to allow fine-tuning.

The algorithm I used does this adjustment of scale by including the scale (and direction) of earlier mutations as part of each candidate solution. This allows our program to evolve the mutation scale at the same time as it is adjusting the solution. The effect of this "meta-mutation" is that the scale of mutation starts out big because big changes help when we don't have very good solutions, but the scale decreases as the solutions converge to the best possible value. **Figure 6** shows an example of how this self-adjustment in action.

We can see the process of evolution of optimal circuits in the panels of **Figure 7** (animated version is available online **[8]**).

By generation 14, the system has learned that moving the zero response near the 3rd harmonic is a good idea both because the 3rd harmonic is attenuated, but also because the resulting ripple decreases insertion loss. The level of ripple in these circuits would be unacceptable for general use but is fine for this application if component variation is not a problem.

By generation 29, the system has learned to straddle the 2nd and 3rd harmonics with the response zero but hasn't learned quite how to balance the responses. In this diagram, the green trace has a zero near the 2nd harmonic and is good at suppressing the 2nd harmonic, but poor on the 3rd. The light blue trace, on the other hand, has the opposite problem because zero is closer to the 3rd harmonic.

Considerably later, by generation 100, all the top solutions are very close to the same answer and all the nearby alternatives have been tested and rejected. The performance is relatively well balanced with the worst-case harmonic output predicted to be at least 50 dB below the fundamental. As built, the filter produces a slightly different response, probably due to parasitic effects or because the input has a better symmetry than we have modeled here. The ripple in the pass band does not appear (probably due to parasitic resistance) and the output level of the 3rd harmonic is bigger than the second. Referring to Figure 4, we can see that the 2nd harmonic is about 6-7 dB lower than we accounted for. This indicates that the duty cycle is probably about 52% instead of the worst case 55% used in the optimization.

One fun hack in this evolutionary optimization is that all the component values were encoded by an index into an array of all in-stock components sorted by value. This representation has the benefit of proximity in that similar indexes usually have similar component values and provides a very rough logarithmic transformation of parameter values simply due to the way that component values happen naturally. The resulting designs are inherently buildable because they can only express available parts, but the representation keeps topological similarity to the underlying physics which helps the evolutionary optimization converge.

Conclusion

The method described here can produce surprisingly good designs that perform much better than would be expected from normal rules of thumb. The method is also much simpler and does not require sophisticated computers to run, in contrast to other AI-based algorithms.

The components of this method can also be applied separately if desired. For instance, the way that the simulator offers a view of circuit designs as lightweight functions has applicability to a broad range of problems and techniques.

Similarly, the idea that design goals and constraints can be specifically targeted to a particular application is useful whenever optimization techniques are available.

Finally, the meta-evolutionary algorithm and design encoding method used here are widely applicable to many kinds of simulation-based design efforts. Taken together or separately, all these components are freely available for use or extension to whatever

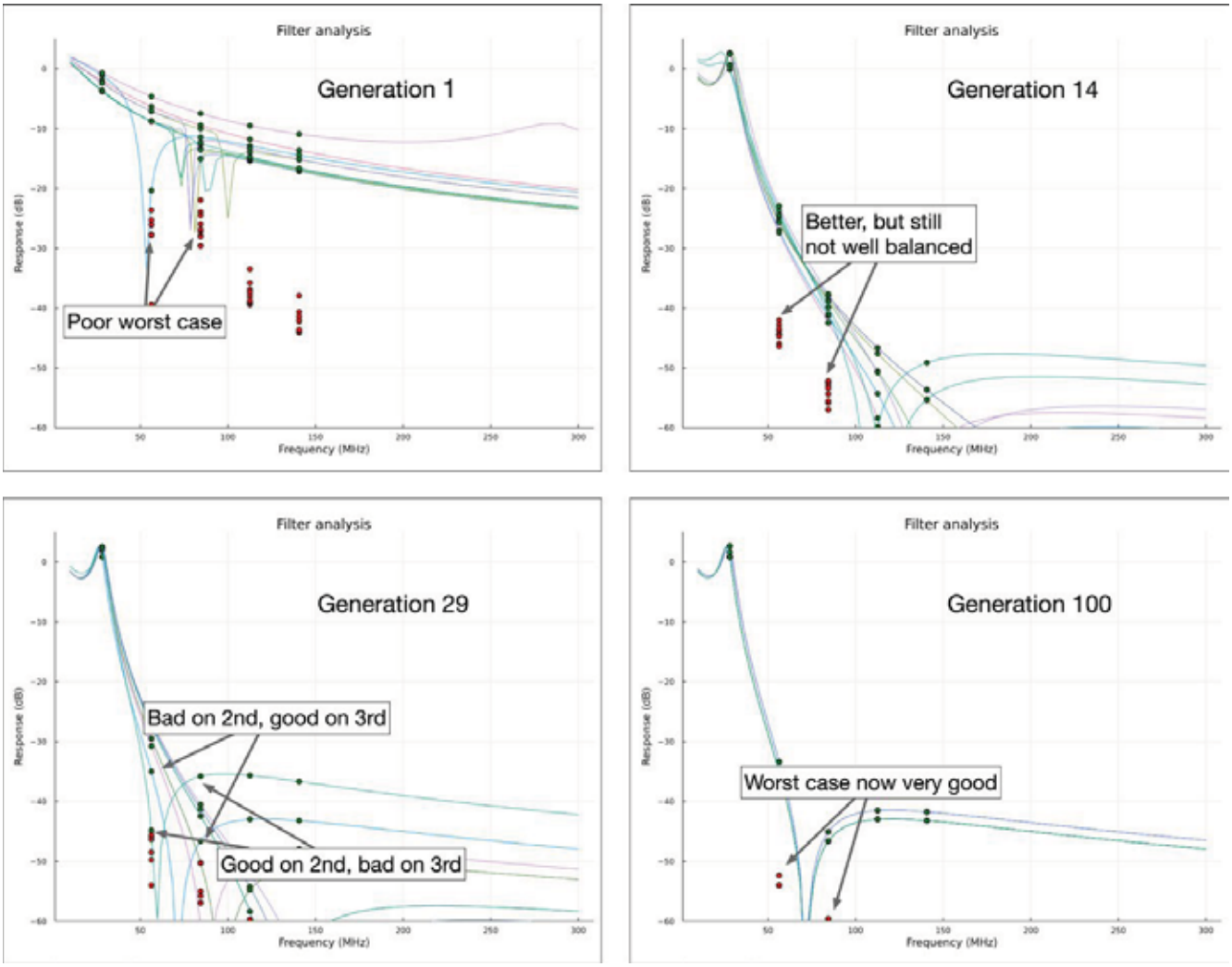


Figure 7 – The evolutionary algorithm at different stages of convergence. In each panel, each curve is a different design and the green dots on the curves are the filter response at a harmonic frequency. The red dots are output amplitude after accounting for input level. Initially, all solutions produce poor results. By generation 14, the system has learned to drive the 3rd harmonic very low, but this leaves the 2nd at too high a level. By generation 29, the system has a number of alternatives that focus either on the 2nd or 3rd harmonic, but none with good balance. By generation 100, all solutions attenuate all harmonics very well. The residual imbalance is probably due to the system hedging against component variations. From generation 14, all filters attenuate the 4th and higher harmonics sufficiently that they are no longer in the frame of view.

problems you want to solve and contributions back to the software are welcome.

Ted Dunning, K6BYT, has been fascinated by electronics and computing from an early age, but never pulled the trigger on getting an amateur license until very recently (in 2024). This article captures his love of frontiers, in this case between evolutionary computing and analog circuit design. He has worked in several areas of advanced computing for decades and currently works at HPE designing new systems to enhance computer security. He has a bachelor's degree from the University of Colorado and a doctorate from Sheffield University.

References

[Ref. 1] E.A. Karahan, Z. Liu, A. Gupta, *et al.* Deep-learning enabled generalized inverse design of multi-port radiofrequency and sub-terahertz passives and integrated circuits. *Nat Commun* 15, 10734 (2024). **https://doi.org/10.1038/s41467-024-54178-1**

[Ref. 2] Source and design files for the open-source Software Defined Transceiver. GitHub repository. **https://github.com/peterbmarks/T41-EP-SDT**

Notes

[1] https://en.wikipedia.org/wiki/Stephen_Butterworth

[2] https://cdn-shop.adafruit.com/datasheets/Si5351.pdf

[3] https://gist.github.com/tdunning/e551ae973422609f031c1bdca39ff5b4

[4] https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html

[5] https://ngspice.sourceforge.io/

[6] https://xyce.sandia.gov/

[7] https://github.com/tdunning/MicroSpice.jl

[8] https://github.com/tdunning/MicroSpice.jl/blob/main/examples/xmit-filter/images/evolution.gif

John Clark, K2AOP
6226 E. Carolina Dr., Scottsdale, AZ 85254 | **k2aop@arrl.net**

Measure Low Value Surface Mount Components Accurately

Achieving precise resistance measurements for low-value surface mount components demands more than just standard test equipment — it requires the true four-terminal Kelvin connection. This article presents a simple, cost-effective method for accurate measurements.

Introduction

To measure a surface mount resistor's value accurately, a current is applied, then the voltage across it is measured with connections that are different from where the current connections are made. This is necessary so that the contact resistance at the current connections does not add to the value of the resistor being measured. This high accuracy technique is called "The 4-terminal Kelvin Connection".

A Simple, Accurate Solution

Accurate measurements can be obtained by placing a surface mount resistor across a narrow slit that separates a circuit board into two halves. When a precision source of constant current is applied across the two halves, current will flow through the surface mount part. Your voltmeter probes then measure the voltage across the top of the part's end connections. This would be the "4-terminal Kelvin

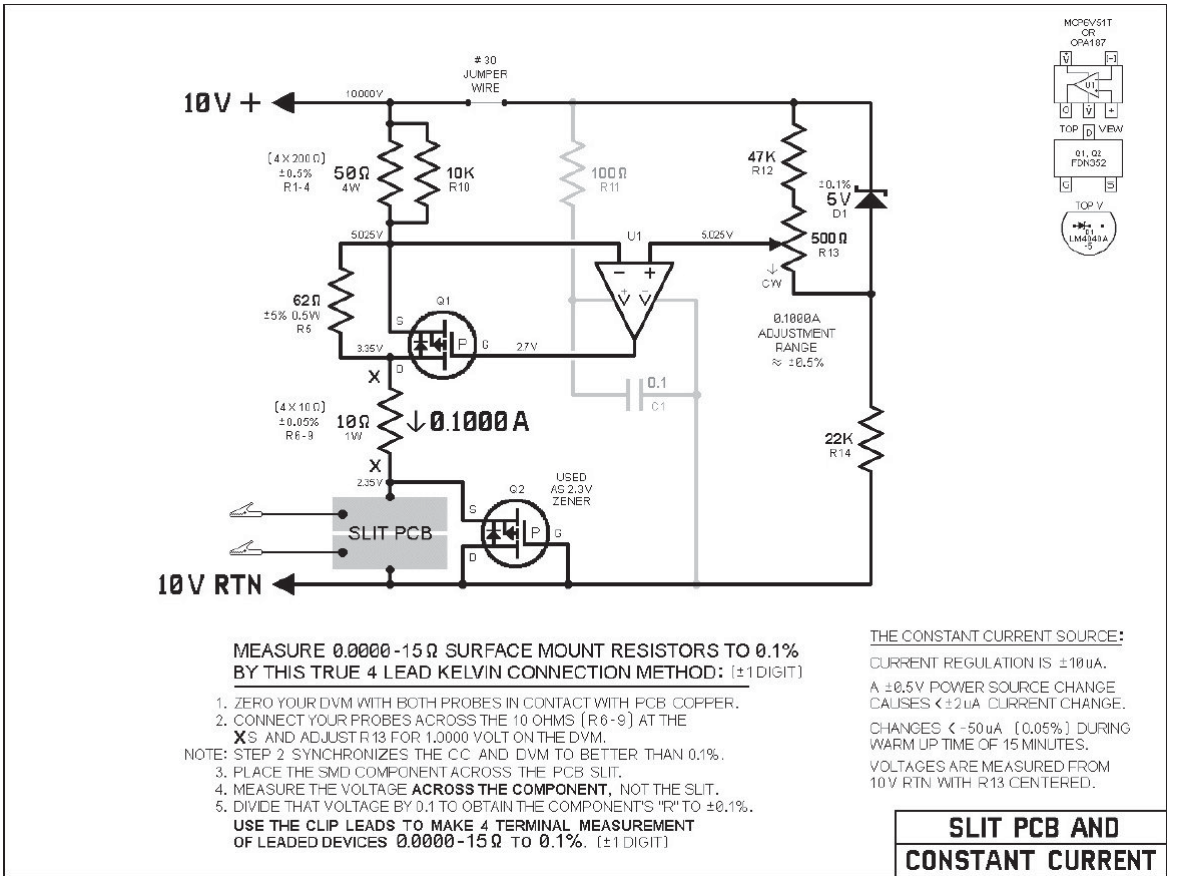


Figure 1 – Schematic of the Slit and Constant Current Source board

Connection". The part's calculated value, R = E divided by I, is as accurate as the current and DVM.

Expensive DVMs that provide a way to measure surface mount resistors have manuals that state that four wires are used. The four wires don't connect to the part. Those four wires end at a probe that clamps onto the ends of the part and make only two connections to the part. This is not the 4-terminal, high accuracy, Kelvin Connection! If you take the trouble to review the DVM specifications, you will find a total tolerance as high as 8 milliohms because the probe's two connections can add that much resistance to the measurement. When measuring 100 milliohms with that clamp probe, the measurement accuracy could be as poor as plus or minus 8%. At 10 milliohms it could be an 80% error.

Constant Current and Slit PCB

The design provides a stable and accurate measuring tool for low resistance surface mount and through-hole parts. The constant current circuit, **Figure 1**, and layout, **Figure 2,** will provide 0.1% accuracy from under 10 milliohms to 15 ohms and 1% at 1 milliohm. (plus/minus one least significant digit). The graphics on the website, and parts, are the result of considerable testing and design to minimize calibration change from the one watt that heats the powered circuit board. The back side copper provides only a slight improvement in temperature stability and can be omitted. Check between all pads for leakage or shorts before attaching components.

The graphics of the PCB and parts list are available at **www.arrl.org/QEXfiles**.

John Clark, K2AOP, was first licensed in 1952. In December 1965 his 432 MHz converter was the first using transistors to be published in QST. *It was low noise, low cost and easy to build. It was so well received, that from 1968 through 1971 it was included in the* Radio Amateur's Handbook *along with ARRL awarding John the first ever author credit in the* Handbook. *Beyond these easily checked details, his extensive accomplishments would not be believed.*

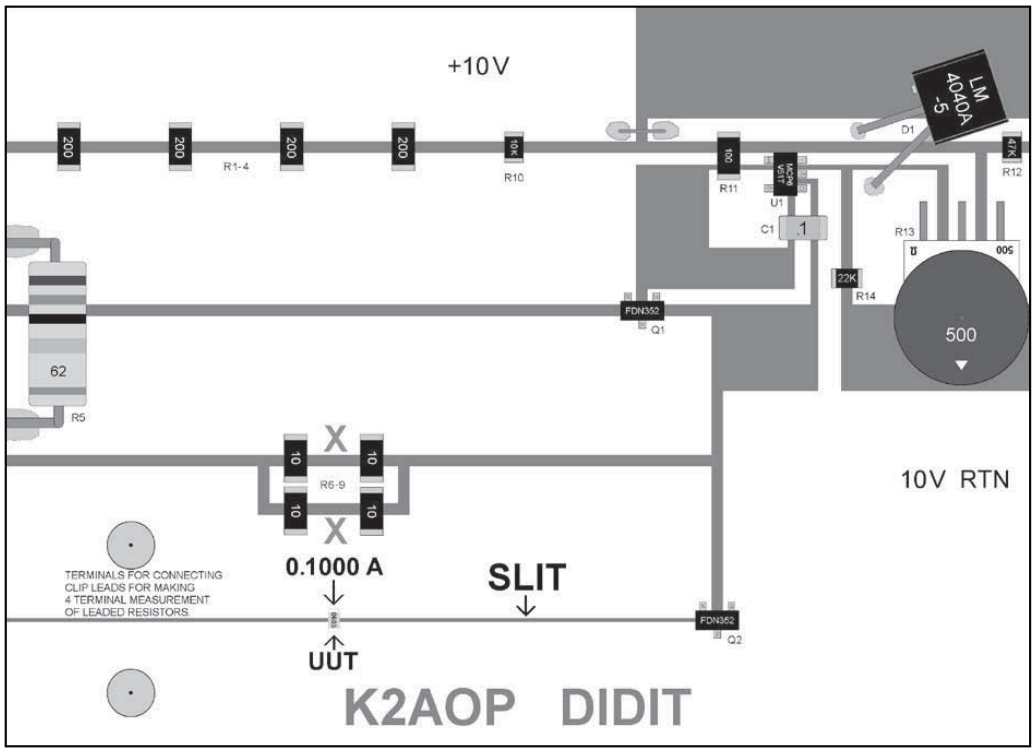


Figure 2 – Assembled circuit board.

Michael M., PC7MM | **pc7mm@pc7mm.nl**
Richard J., PD3RFR | **pd3rfr@pd3rfr.nl**

A Deep Dive into the Digital Data Function of the Icom IC-9700

This article explores the little-known DD mode of the Icom IC-9700, enabling 128 kbps data communication over 23 cm. A performance analysis and a remotely operated radio demo reveal its strengths, limitations, and potential for further advanced experimentation.

This article is translated from the January 2025 issue of Electron.

Introduction

The IC-9700 transceiver has a number of features that are often underutilized by most radio amateurs. One of these features might be the Digital Data (DD) function **[1]**. This function allows data communication via connected equipment using the LAN port of the IC-9700, offering a speed of 128 kbps on the 23 cm band, either with or without the involvement of a repeater. See **Figure 1**.

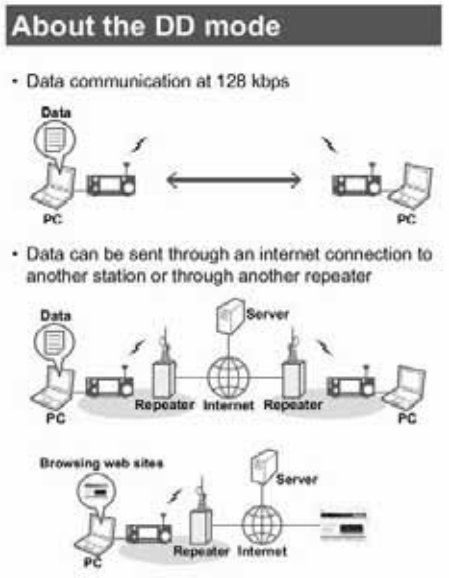


Figure 1 – Diagram from the Advanced Manual of the IC-9700.

Chapter 13 of the IC-9700 Advanced Manual **[2]** titled, DD Mode Operations, discusses the existence of the DD function of the IC-9700 but does not provide an in-depth explanation of how this function works or what its practical applications are. This gave rise to the idea to experiment with this DD function, after which the idea emerged to write an article about the findings.

Part 1 of this article describes the results of an analysis of the characteristics and performance of a DD connection. Part 2, describes the nature and results of a practical experiment conducted with the DD connection. The article concludes with Part 3, where several conclusions and recommendations regarding the usage of the DD function are listed. A video link **[3]** illustrating the experiment appears at the end of this article.

Part 1: Analysis of the characteristics and performance of a DD connection

After having performed several analyses with the Wireshark software **[4]**, it appeared that the IC-9700 operates as a simple bridge when it is in DD mode. This means that all data that a device sends to the LAN port of the IC-9700, is (re)transmitted by the IC-9700 on the 23 cm band. Similarly, any data received on the 23 cm band by the IC-9700 in DD mode is forwarded via the LAN port to a device connected to that LAN port.

Figure 2 – Setup for performance measurements with iperf3 software.

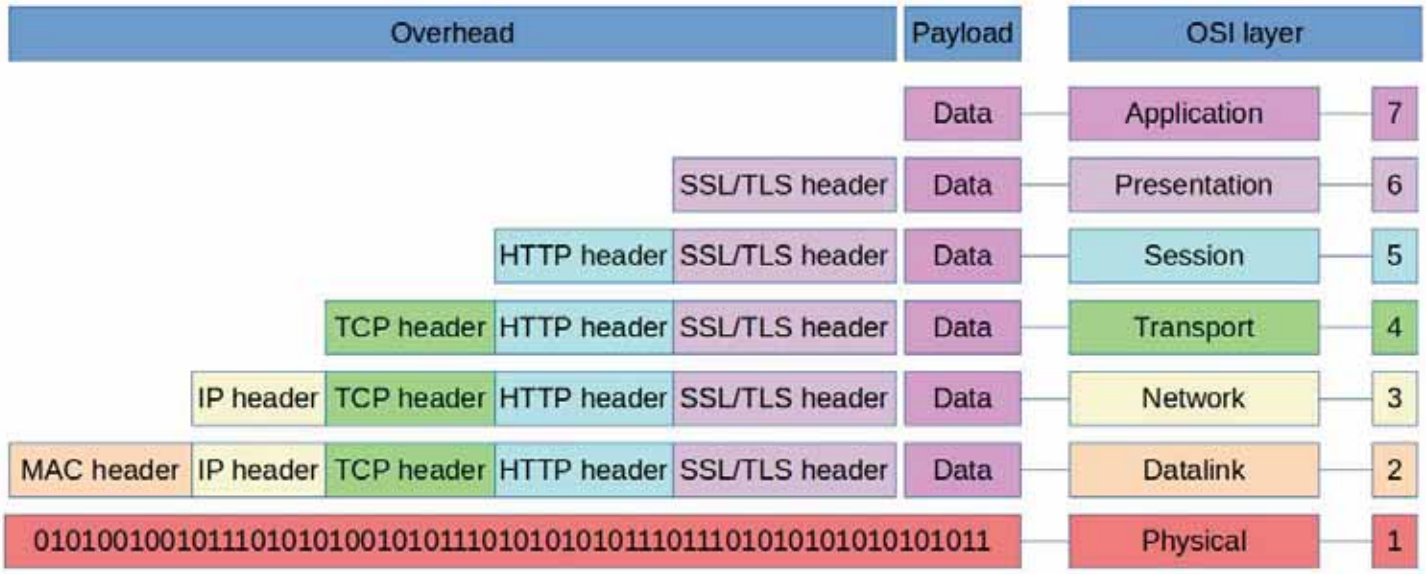


Figure 3 – Overview of OSI layers with visually separated Payload and Overhead, indicating that only a part of the actual data transfer concerns the Payload itself.

The DD function is therefore generic, meaning it could, in theory, be used for any application that requires a wireless bridge between two devices or networks. [*FCC rules regarding non-amateur or commercial content apply to the data in the Payload layer — Ed.*] To evaluate its practical applications, the actual throughput and latency of the DD connection was measured. The following setup was used with two IC-9700 transceivers connected to a dummy load as shown in **Figure 2**.

The first question to be answered was whether the 128 kbps throughput claimed by Icom is actually achievable in a practical application, considering that this speed most likely refers to throughput on OSI layer 1. Taking into account that lower OSI layers progressively reduce the bandwidth available for actual data due to overhead building up layer by layer, bandwidth that is effectively available for software application is most likely less than 128 kbps. **Figure 3** visualizes this and shows that only a portion of the 0s and 1s being transmitted corresponds to actual data, or the payload.

The *iperf3* software **[5]** was used for determinig the practical bandwidth available for software applications on OSI layers 3 and 4. This software allows exchanging UDP datagrams between a server and client over a network connection and it allows measuring practical throughput of UDP datagrams through that network connection. Conclusions regarding both unidirectional and bidirectional data communication are shown in **Figure 4** and **Figure 6**.

Figure 4 presents measurement results for unidirectional communication and shows that the UDP datagrams sent by laptop 1 are successfully received by laptop 2 at speeds up to about 79 kbps. From 78-79 kbps until about 91 kbps, no UDP datagrams are lost, but some form of queuing occurs: the datagrams sent at speeds of 80-91 kbps are received at a speed of around 78-79 kbps. The maximum length of the queue was not determined, but jitter of up to 1-2 ms appears to occur during queueing.

At speeds above 79 kbps, jitter increases, reaching up to 10 ms. As shown in **Figure 5**, at speeds above 91 kbps, actual UDP datagrams are lost, and jitter rises to around 40 ms.

Figure 6 presents measurement results for bidirectional communication, showing that UDP datagrams exchanged simultaneously in both directions between laptop 1 and laptop 2 can be reliably transmitted at speeds up to around 44 kbps. Speeds above 44 kbps result in the loss of UDP datagrams. Note that the bidirectional analysis was limited to two measurements and all the tests performed in the unidirectional measurement were not repeated.

Rated/Set unidirectional UDP data stream bandwidth From laptop 1 to laptop 2	Real actual UDP data stream sent by Laptop 1	Real actual UDP data stream received By laptop 2	Lost UDP datagrams in communication channel
50kbps	50.2kbps	49.5kbps	0.0%
60kbps	60.7kbps	59.5kbps	0.0%
70kbps	70.1kbps	69.2kbps	0.0%
80kbps	80.5kbps	78.9kbps	0.0%
85kbps	85.2kbps	77.9kbps	0.0%
90kbps	91.0kbps	77.9kbps	0.0%
95kbps	95.7kbps	76.2kbps	4.9%
100kbps	100kbps	78.6kbps	7.0%
150kbps	150kbps	79.3kbps	37.0%
1000kbps	1000kbps	79.0kbps	91.0%

Figure 4 – Overview of iperf3 measurements regarding a unidirectional data stream.

From the measurement results, one can conclude that at OSI layer 4, unidirectional communication is reliable at speeds up to about 80 kbps, and bidirectional communication is reliable at speeds up to about 45 kbps. Speeds above 80 kbps (unidirectional) or 45 kbps (bidirectional) result in lost datagrams in the case of UDP streams, and result in retransmits in the case of TCP connections. This leads to further delays and may even cause the application software using the DD connection to crash due to a congested connection.

Given the fact that the maximum speeds at OSI layer 4 are lower than the 128 kbps as claimed by Icom, we conclude that Icom most likely measured the 128 kbps on OSI layer 1. In each subsequent OSI layer, the payload to be transmitted is supplemented with an additional header (i.e., overhead), which means that for every higher OSI layer, relatively more overhead is sent with the payload, reducing the effective bandwidth available for the actual payload.

In addition to the *iperf3* tests described above, a simple ICMP ping test was conducted with no other network traffic through the DD connection. As visible in **Figure 7**, the conclusion from this test is that, under ideal conditions, the ICMP ping time is around 62 ms. If the connection is simultaneously burdened with large amounts of other network traffic, the ping time increases to 500-1500 ms, and if the connection becomes congested, ICMP timeouts occur, making it unreliable.

Part 2: A practical experiment with the DD Connection

Taking into account the conclusions from the performance tests, a practical experiment utilizing the wireless DD connection was performed.

Inspired by the question whether an AAC audio stream could be transmitted via a DD connection, it was decided to remotely operate an IC-7300 using the *RemAud* software **[6]** made by DF3CB, making use of the wireless DD connection as well as two laptops and two IC-9700 transceivers. A video link **[3]** shows the practical implementation of this experiment.

Setup Components

Figure 8 shows the setup of the experiment and the components that were used. From left to right, the components are:

The first laptop (IP address = 192.168.10.3), which is directly connected to the first IC-9700 (IP address = 192.168.10.4) using a UTP (cross) cable, with the IC-9700 connected to a dummy load.

The second IC-9700 (IP address = 192.168.10.4), connected to a dummy load and directly connected to the second laptop (IP address = 192.168.10.2) via a second UTP (cross) cable.

An IC-7300 connected to a dummy load, with a USB cable connecting it to the second laptop, allowing the speaker, microphone, and PTT of the IC-7300 to be controlled via the second laptop.

A President Randy III handheld radio, which communicates with the IC-7300 via the 10-meter and/or 11-meter band, connected to a dummy load. **Figure 9** shows a schematic of the test setup.

Configuration of network settings on the laptops

As previously described, the IC-9700 in DD mode functions as a simple bridge. To preserve as much valuable bandwidth as possible for the application needed in the experiment, the laptops were directly connectted to the IC-9700 transceivers using crossover UTP cables. Static IP addresses were used and no default gateway was configured on the laptops. As **Figure 11** shows, DNS servers were not set, and all network services except IPv4 were disabled. This helped avoid unnecessary and unpredictable network traffic preventing disruption of the UDP data streams necessary for the experiment.

The network settings of the second laptop are shown in **Figure 10**. The IP addresses that were used are already men-

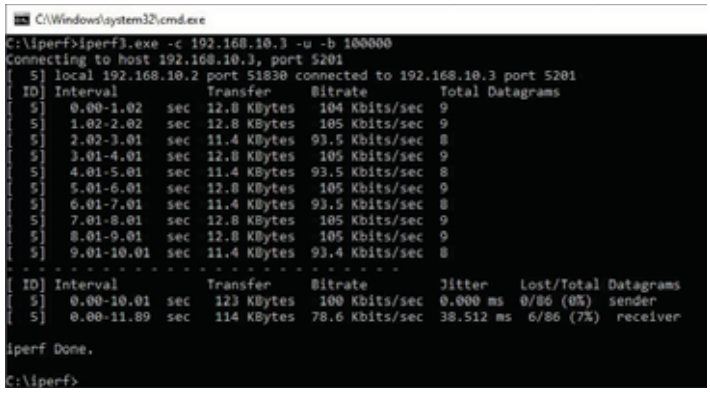


Figure 5 – Unidirectional iperf3 UDP stream with supply of 100 kbps, resulting in a percentage of lost datagrams and a significant jitter.

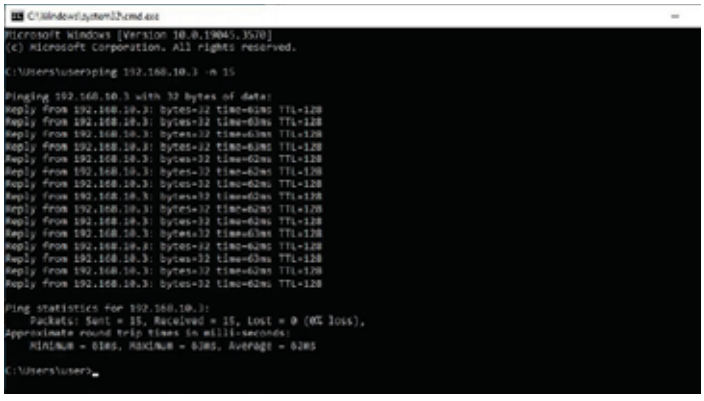


Figure 7 – Results of ICMP ping test from laptop 1 to laptop 2 via an unused DD connection.

Rated/Set bidirectional UDP data stream bandwidth From laptop 1 to laptop 2	Real actual UDP data stream sent by Laptop 1	Real actual UDP data stream received By laptop 2	Lost UDP datagrams in communication channel
90kbps	91kbps	44.6kbps	35.0%
1000kbps	1000kbps	42.7kbps	94.0%

Figure 6 – Overview of *iperf3* measurement results regarding a bidirectional data stream.

Figure 8 – Overview of the components that were used for the experiment.

tioned. Since a default gateway was not configured, it is essential that both laptops are within the same IP subnet. A /24 subnet with subnet mask 255.255.255.0 was chosen.

Configuration of the *RemAud* Software

The *RemAud* software allows remote operation of the speaker, microphone, and PTT button of a transceiver, referred to as, Remote Access in this article. This means that a transceiver located in the shack can be operated remotely from another location without requiring physical access to that transceiver. See **Figures 12** and **13**.

Remote Access via *RemAud* can be achieved within a Local Area Network (LAN) or, if desired, via the internet. In this experiment, the *RemAud* software was used with the two IC-9700 transceivers, which wirelessly communicate with each other via the 23 cm band using the DD connection providing a bandwidth of 45-80 kbps at OSI layer 4, depending on the amount of bidirectional transmissions via the DD connection.

In the experiment, all components are set up on one physical table, and an imaginary Remote Access boundary is drawn between the two IC-9700 transceivers, which, like the IC-7300, are connected to dummy loads to ensure that RF signals on the 23 cm and 10/11 m bands do not disturb other users of those bands.

The *RemAud* software consists of a Client and a Server. The Client part must be installed on the computer that is remote from the transceiver to be operated, which in this experiment is laptop 1. The Server part must be installed on the computer

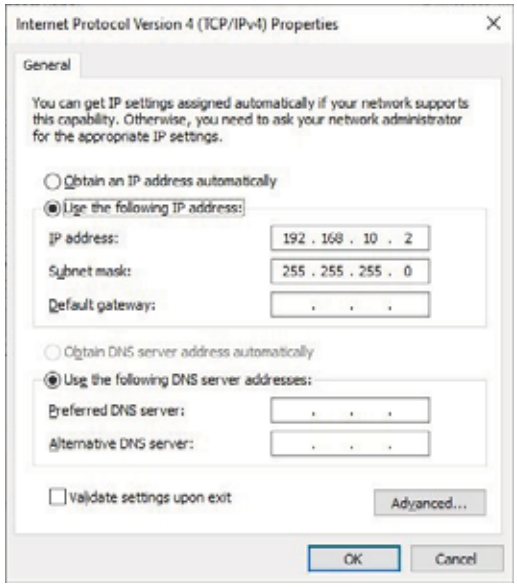


Figure 10 – IPv4 settings on laptop 2.

directly connected to the transceiver, which in this experiment is laptop 2, controlling the IC-7300.

The following settings are required for the *RemAud Server* software:

1. An IP port for Client and Server to use, which in the experiment is 4003.
2. A username/password for authentication by the *RemAud Client* software.
3. Audio devices: in the experiment, the (virtual) speaker and microphone of the IC-7300.
4. Codec: GSM 6.10 was chosen, with a sample rate of 8 kHz and a transfer rate of 1.8 kB/sec.
5. PTT: DTR via the (virtual) COM port of the IC-7300 was used for controlling PTT.

For the *RemAud Client* software, the following settings are required:

1. In the General tab, the Use PTT on Transmit option needs to be enabled.
2. As the remote connection, the IP address and IP port of the *RemAud Server* are to be used.

Figure 9 – Schematic view of the components that were used for the experiment.

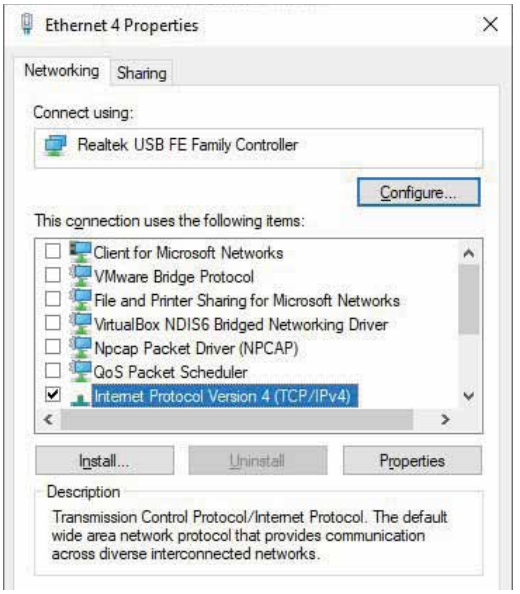


Figure 11 – List of active services on network adapter of laptop 1 and laptop 2.

3. Audio devices: the speakers and microphone of the relevant laptop are to be used.
4. Codec settings must be identical to those in the *RemAud* Server software.

In the experiment, the buffer settings in both the Server and Client software were left unchanged. See **Figures 14** and **15** for screenshots of *RemAud Client* software. See **Figures 16** and **17** for screenshots of *RemAud Server* software.

Configuration of the IC-9700 Transceivers

First, both IC-9700 transceivers need to have a working D-STAR configuration, of which the setup is not discussed in detail in this article. Assuming the D-STAR configurations are functional, both IC-9700 transceivers need to be set to DD mode on identical frequencies. The TX mode needs to be activated using the Transmit button. The TX Inhibit must be disabled. This ensures that any data sent to the LAN port is transmitted over the 23 cm band, so that it can be received by the other IC-9700. See **Figure 18** for screenshot of the IC-9700 in DD mode.

The settings in the DV/DD SET menu must be identical for both IC-9700 transceivers. The DD Packet Output option was not set to All, but to Normal. This option is not documented in the Advanced Manual, but it seems that the All setting forwards data frames, including multicast and broadcast packets, over the 128 kbps connection. The Normal setting does not, leaving more bandwidth available for transferring *RemAud* audio and PTT signals during the experiment.

Figures 12, 13 – *RemAud Server* and *RemAud Client* software.

Configuration of the IC-7300 Transceiver

In the Connectors menu, specific settings need to be applied for correctly transmitting both audio signals and PTT control signals from the USB-connected laptop. The settings from the Connectors menu that were used are listed below. Other settings can be applied for different applications.

The IC-7300 should be set to FM mode without enabling DATA mode. A frequency was chosen on which both the President Randy III and the IC-7300 (with dummy load) can transmit and receive. See **Figure 19**, screenshot of IC-7300.

The ACC/USB AF SQL setting should be turned to ON to prevent continuous noise through the *RemAud* audio connection.

The DATA OFF MOD should be set to USB and not to MIC, ACC, or MIC,ACC, so the speaker and microphone signals are transmitted via the USB cable, not via the microphone input on the front of the IC-7300.

In the USB SEND/Keying submenu, the USB SEND option should be set to DTR so that the *RemAud Server* software can use the DTR pin of the (virtual) COM port of the IC-7300 to control PTT.

With the above settings, it is possible to remotely control the IC-7300 through the *RemAud* software.

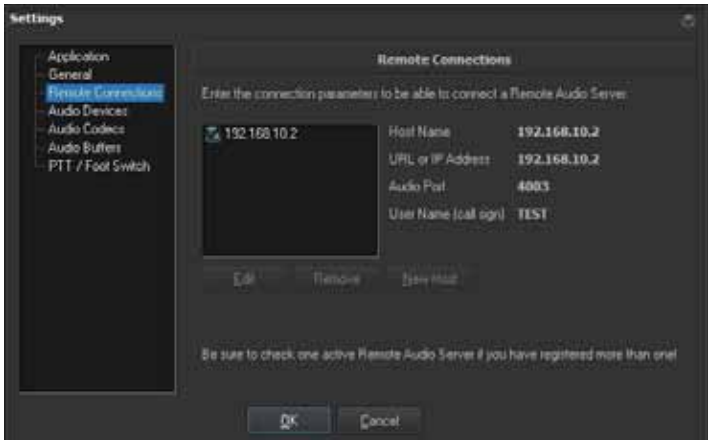
Experiment considered from the perspective of the OSI Model

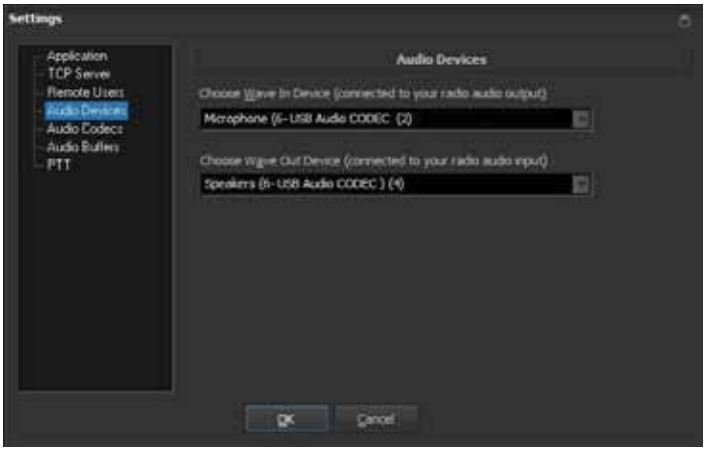
Figure 20 shows the OSI model, this time with an indicative application for each layer in the experiment. To keep the illustration simple and legible, specific protocols and equipment were not always displayed in multiple layers.

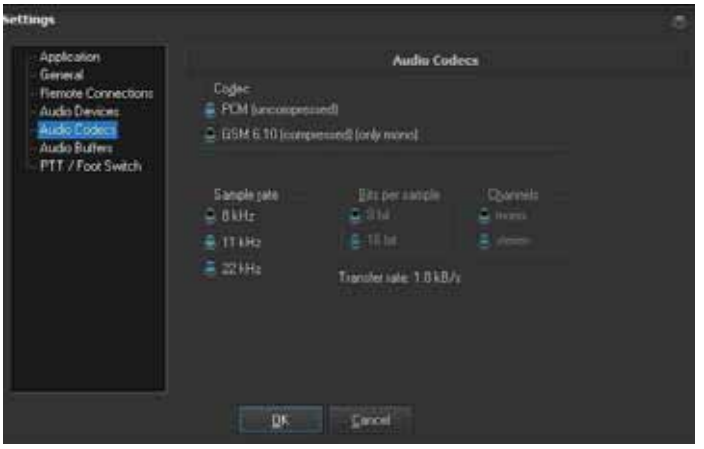
Part 3: Conclusions and Recommendations

Based on the tests and the experiment conducted, several conclusions can be drawn and recommendations can be made.

The DD function of the IC-9700 is technically usable, but it is challenging to make modern applications work well over a DD connection in practice. The effective bandwidth of the connection is quite low. Low bandwidth easily results in a congested connection, resulting in UDP datagram loss and an







Figures 14, 15 – Screenshots of settings in *RemAud Client* software.

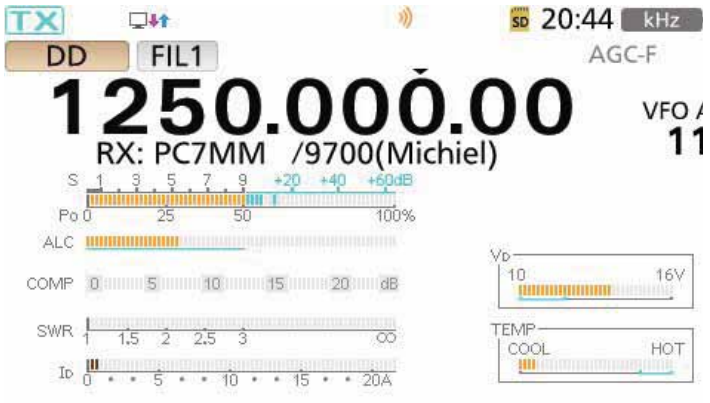
Figures 16, 17 – Screenshots of settings in *RemAud Server* software.

excessive number of TCP retransmits, causing application software to malfunction.

The effective throughput and latency of the DD connection is largely dependent on the amount of simultaneous bidirectional traffic. If there is data traffic in only one direction, the available bandwidth on OSI layer 4 is about 80 kbps. If there is continuous and simultaneous data traffic in both directions, the available bandwidth in one direction is about 45 kbps, with the total data traffic in both directions being around 80-90 kbps.

The IP address settings on the IC-9700 appear to be irrelevant for the proper operation of the DD function. However, the IP address set on the IC-9700 is relevant for the proper operation of other functions, such as the DV gateway function and the Icom RS-MS3W and Icom ST-4001W software.

Although not showing in the video, during the experimentation process, the Internet Connection Sharing (ICS) function of Microsoft Windows was used on laptop 1 to provide laptop 2 with internet connectivity via laptop 1 through the DD connection. After properly configuring ICS on laptop 1 and setting DHCP on the network card of laptop 2, internet connectivity was available on laptop 2 via the DD connection. It was



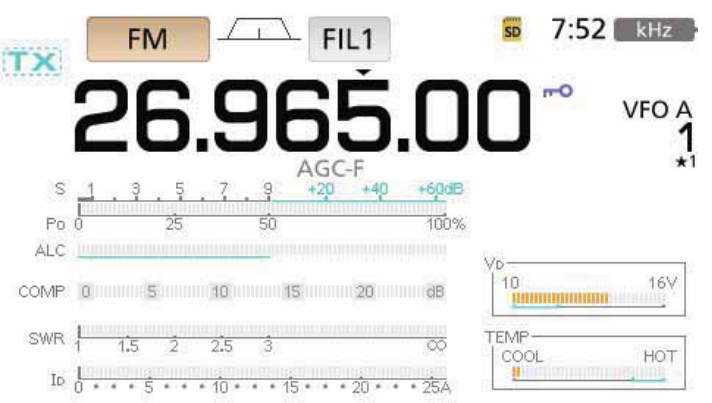


Figure 18 – Screenshot of the IC-9700 in DD mode, operating on 1250 MHz.

Figure 19 – Screenshot of the IC-7300 in FM mode, operating on 26.965 MHz.

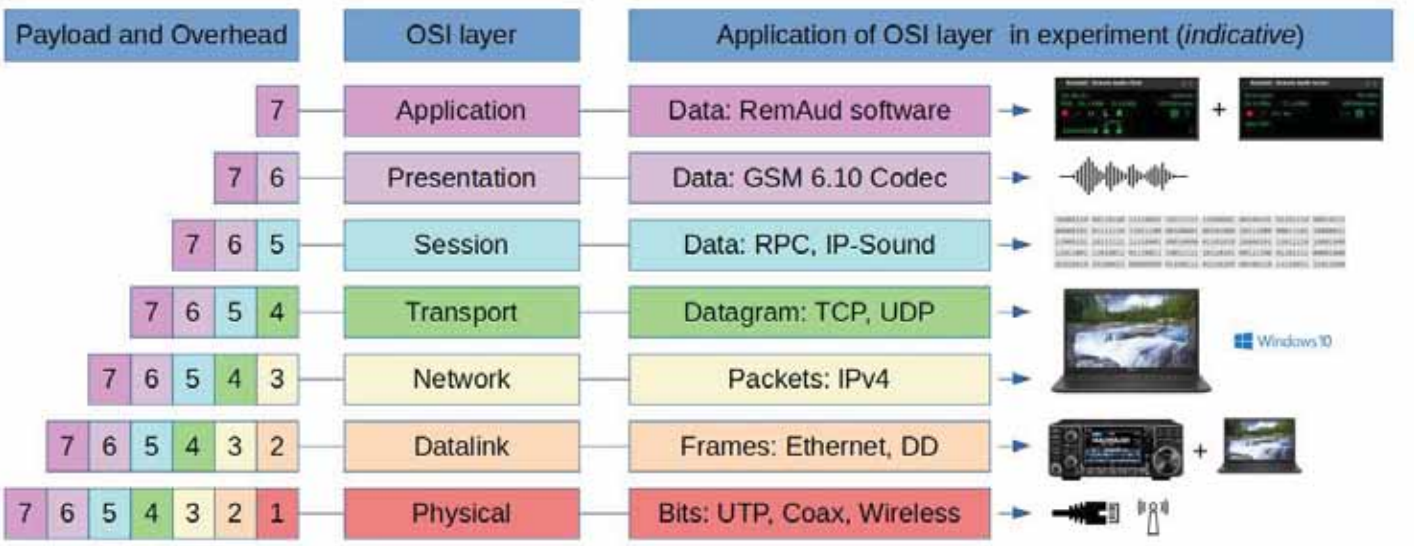


Figure 20 – Schematic overview of OSI layers with indication of application in the experiment.

then possible to open a website on laptop 2, but it took a very long time for the page to open. If Windows and/or installed Windows software searched for updates simultaneously via the DD connection, there was not enough bandwidth left for browsing websites or for other applications. The connection became congested, and only timeouts were seen at ICMP level. **Figure 21** is an example of the results of an ICMP ping to google.com without any other Windows software intentionally generating network traffic.

In the experiment, only direct communication was established between two IC-9700 transceivers with dummy loads. No DD repeater was used, partly because a DD repeater is not available in the local region. It would be interesting to repeat this experiment while using a DD repeater. A separate article might be published in the future on a similar experiment that uses a DD repeater.

```
C:\Windows\system32\cmd.exe

C:\iperf>ping google.com -n 15

Pinging google.com [172.217.23.206] with 32 bytes of data:
Reply from 172.217.23.206: bytes=32 time=141ms TTL=57
Reply from 172.217.23.206: bytes=32 time=77ms TTL=57
Reply from 172.217.23.206: bytes=32 time=80ms TTL=57
Reply from 172.217.23.206: bytes=32 time=75ms TTL=57
Request timed out.
Reply from 172.217.23.206: bytes=32 time=86ms TTL=57
Reply from 172.217.23.206: bytes=32 time=78ms TTL=57
Reply from 172.217.23.206: bytes=32 time=76ms TTL=57
Reply from 172.217.23.206: bytes=32 time=83ms TTL=57
Reply from 172.217.23.206: bytes=32 time=83ms TTL=57
Reply from 172.217.23.206: bytes=32 time=74ms TTL=57
Reply from 172.217.23.206: bytes=32 time=116ms TTL=57
Reply from 172.217.23.206: bytes=32 time=78ms TTL=57
Reply from 172.217.23.206: bytes=32 time=76ms TTL=57
Reply from 172.217.23.206: bytes=32 time=85ms TTL=57

Ping statistics for 172.217.23.206:
    Packets: Sent = 15, Received = 14, Lost = 1 (6% loss),
Approximate round trip times in milli-seconds:
    Minimum = 74ms, Maximum = 141ms, Average = 86ms

C:\iperf>_
```

Figure 21 – Screenshot with results of ICMP ping to google.com via ICS via DD connection with a timeout as a result of other unintended simultaneous network traffic through the DD connection.

The effective bandwidth available for software applications is not very high. 45-80 kbps is about 5-10 kB/s (kilobytes per second), which proved to be insufficient for working properly with the PCM Codec of *RemAud*, which requires 8 kB/s. Therefore, the GSM 6.10 Codec in *RemAud* was used, which requires 1.8 kB/s. This also provides a de facto answer to the question about the feasibility of transporting an AAC audio stream over a DD connection.

Initially, the idea was to remotely control the IC-7300 by using the Icom RS-BA1 software instead of *RemAud* software. However, it turned out that even the CI-V data stream requires more than the 45-80 kbps bandwidth, causing the connection to become completely congested and the RS-BA1 software to crash, even without adding a TX or RX audio stream to the CI-V data stream. Actual bandwidth requirements of the CI-V data stream were not measured.

As shown in the video, the TX/RX relay of the IC-9700 switches about 10 times per second when the DD function is enabled and data is transfered. That equals 36,000 times per hour and 864,000 times per day. After having the DD function enabled for some days, the TX/RX relay might have reached the end of its lifespan.

Given the relatively low effective bandwidth available, it is important to avoid unnecessary network traffic through the DD connection. During the experiment this was achieved by directly connecting the laptops to the IC-9700 transceivers, not setting a default gateway and DNS server, and disabling all network services except IPv4 on the network cards. If the IC-9700 were to be integrated into a network, it is recommended to place the transceivers in a separate VLAN where only unicast traffic is routed through, and if possible, to apply traffic shaping. During the experiment, the IC-9700 was connected to a busy network. Just the ARP and DHCP-related packets were enough to overwhelm the DD connection.

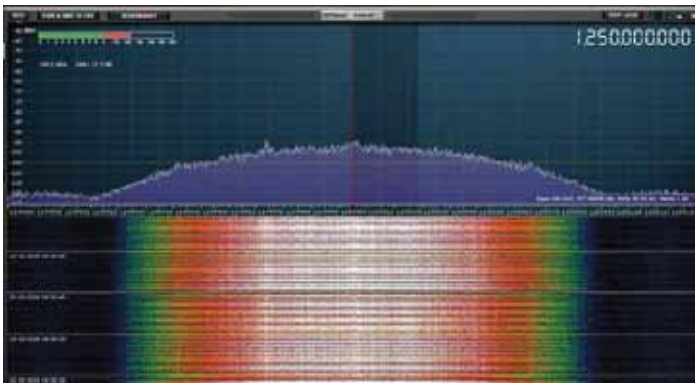


Figure 22 – The 170 kHz-wide GMSK modulated DD signal visualized in the SDRUno software.

The DD connection appeared to be too slow to comfortably use modern internet/cloud applications in practice. If it is desirable to use the DD connection for internet/cloud applications, besides applying traffic shaping, it is recommended to setup a proxy server on the server side and configure this proxy server only in the software application that needs internet connectivity. This prevents other software applications and/or the Windows Update mechanism from using the DD connection.

During the experimentation process, the DD signal was analyzed, among other tools with the *SDRUno* software. The signal, **Figure 22**, appeared to be about 170 kHz wide. It was discovered that the DD signal is a digitally modulated GMSK signal and that the IC-9700 uses a software-selective filter of 300 kHz width for DD.

Afterword

Through this article, we liked sharing the knowledge and experiences we have gained and we hope to inspire the ham radio community to start further experimenting with the DD function of the IC-9700.

This article is not intended to be a comprehensive manual for installing and/or configuring the components mentioned. Only the most essential parameters necessary to create a technically functioning system are mentioned in the article.

If you have any questions following this article or the video, please feel free to contact us!

Michael, PC7MM, has worked as a self-employed IT project manager and IT consultant for over 15 years and is mostly engaged in large and complex transition projects. He specializes in cyber security and has been a licensed ham radio amateur since 2023. He is a proud member of the Dutch VERON A08 chapter and is always eager to broaden and deepen his knowledge.

Richard, PD3RFR, works as a senior technical telecom project coordinator at a company that specializes in rail infrastructure projects. He is a licensed ham radio amateur since 2013 and is an active board member of the Dutch VERON A08 chapter. He actively contributes to organizational and technical aspects of several regional radio and television broadcasters.

Notes

[1] Download link for IC-9700 product sheet and data sheet**: https://www.icomfrance.com/uploads/files/produit/doc-IC-9700-en.pdf**

[2] Download link for the IC-9700 Advanced Manual: **https://www.icomjapan.com/support/manual/2166/**

[3] Video- Experiment with the Digital Data mode of the Icom IC-9700" by PD3RFR and PC7MM: **https://www.youtube.com/watch?v=Q0x0RkCm94k**

[4] Download link for Wireshark network protocol analyzer software: **https://www.wireshark.org**

[5] Download link for iperf3 network bandwidth measurement tool: **https://github.com/esnet/iperf**

[6] Download link for *RemAud* software, developed by DF3CB: **https://df3cb.com/remaud/download/**

Michelle Thompson, W5NYV
5379 Carmel Knolls Dr., San Diego, CA 92130 | **mountain.michelle@gmail.com**

Real and Complex Signal Basics

The magic of radio is rooted in mathematics. Some of that math can be complicated or scary looking. We are going to break things down bit by bit, so that we can better understand what it means when we say that we are going to transmit a complex baseband signal.

Real Signals

Everything in this article is based on a single-carrier real signal, even when we get to complex modulation. A single-carrier real signal is where our data is a single-dimensional value that we want to communicate, and we multiply it by a carrier wave (a cosine wave) at a carrier frequency (fc). Let's call the value we want to communicate *alpha*. Alpha is short for *alphabet*. An alphabet is a set that contains all the possible values that we might want to send.

Because we are dealing with digital signals, the value that we are transmitting is held for a period of time, called *T*. During the next period of time, we send another constant value, and so on. We send discrete values for a period of time *T*, one after another until we are all done sending data. This is a different situation than if we were sending continuous values over time, which is the case with analog signals.

Let's say we are using four different signal amplitudes to represent four different values of alpha. During each time period, *T*, we select one of these four amplitude values. We hold that value for the entire time period. These values can be thought of as single-dimensional values. One amplitude value uniquely identifies the value of alpha we want to send. Sending one of four values at a time means we are sending two bits of data at a time, represented as 00, 01, 10, and 11.

In order to send our amplitude value over the air from transmitter to receiver, we multiply alpha by our carrier frequency. The result is alpha*cos(2*pi*fc*t). Cosine is a function of time t. The 2*pi term converts radians per second to cycles per second, which is something that most of us find easier to deal with.

When we multiply in the time domain, we cause a different mathematical process to happen in the frequency domain. This process is called *convolution.* This mathematical process creates images, or copies, of our baseband signal in the frequency domain. After multiplication in time (or convolution in frequency) one of our baseband values will show up at fc and the other will appear at -fc.

Real signals have a special characteristic: They are symmetric. So, we have these two images, -fc and fc. Let's filter out the lower image, and transmit fc, which is our modulated signal at the carrier frequency. At the receiver, we multiply what we receive by the same cosine wave that we used to transmit, cos(2*pi*fc*t). We multiply the received fc by fc in the time domain (convolution in the frequency domain). This results in images at 2*fc and 0 Hz. We use a low-pass filter to get rid of the unwanted images at 2*fc and by integrating over the time period *T*, we get a scaled version of the original value (alpha) that was sent.

Why do we integrate over the time period *T*? Because we are dealing with digital signals, and we need to collect the signal power over the time that a specific value is being sent. Integrating over time period *T* maximizes the received signal power and lets us filter out noise and interference.

Amazing! We transmitted a signal, then we reversed the process, and got our original sent value back.

Complex Signals

So, what's all this complex signal stuff all about? Why mess with success? We have our single carrier signal and our four amplitude values. We send out a modulated signal, we demodulate at the receiver. With a little filtering we get back what we sent. What more could we want? Well, we want to be able to send more than a "shave and a haircut" number of bits!

If we want to send more bits in the same time period, then we must use a bigger alphabet. Let's say that we want to double our throughput. We now pick from 16 different amplitudes, sending the value we picked out for a

Table 1 – Alpha Value Representation in Bits

alpha	*bits*
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

period of time *T* as a single-carrier real signal. Now, each alpha value stands for four bits as shown in **Table 1**.

We have a minor problem. Sending 16 different voltage levels on a single carrier means that we have to be able to differentiate with finer and finer resolution at our receiver. Before, we only had to distinguish between four different levels. Now we have 16. This means we had better have a really clear channel and a lot of transmit power to tell the difference between two neighboring signals. But we don't always have plenty of power and clear channels. It's expensive and a bit unreasonable to demand, and there is a better way.

We know we want to send out one of 16 values, not just one of four. We know it will be expensive and fussy if we insist upon remaining with our single-dimensional thinking. However, we can turn our one-dimensional problem into a two-dimensional problem, and assign a real single-carrier signal to, say, the vertical dimension, and then a second single-carrier real signal to the horizontal dimension. The vertical dimension handles four levels. The horizontal handles four levels. We still have the same time period *T*. We are now performing digital signal processing!

Transmitting the Complex Signal

Figure 1 shows that we just now have a two-dimensional coordinate system to represent our data, instead of a one-dimensional coordinate system. This is referred to as 16QAM, meaning *quadrature amplitude modulation* with 16 values. "Quadrature" refers to the two signal dimensions being at right angles.

But how can we send two real signals over the air, at the same time? We can't just add them together, can we? They will step on each other and we'll get a noisy mess at the receiver.

Math saves us! We actually can add these two signals together, send them as a sum, and then extract each dimension back out, but only if we prepare them properly. Here is how that is done.

Look at the two-dimensional diagram of 16QAM in Figure 1. The vertical axis is labeled Q (for quadrature), and the horizontal axis is labeled I (for in-phase). When we want to indicate the vertical dimension of our value (pick any one of them), then we use that value's vertical location. For example, in Figure 1, the vertical axis position of -1 is used for the values represented by 0011, 0111, 1111, and 1011. And, then we multiply that -1 by sin(2*pi*fc*t). We now have our Q signal.

If we want to transmit the value 1111, we also need its horizontal location. That is +1 on the I axis. We multiply this value of +1, which gives the horizontal dimension location of 1111 bits, by cos(2*pi*fc*t). We now have our I signal. The alpha value 1111 can then be represented by the combination of a Q signal value of -1 and an I signal value of +1.

The Q axis value is multiplied by a sine wave. The I axis value is multiplied by a cosine. The resulting Q and I signals are out of phase by 90 degrees, also referred to as quadrature. Two signals in quadrature but at the same frequency creates what is called a complex signal because each combination of Q and I signal values can be treated as a complex number.

The Q and I signals are added together and transmitted for the duration of the sample period. Both of them are transmitted at the same time and the pair of signals represents a coordinate pair for a particular alpha shown as the circles in Figure 1. Adding the I and Q signals together and transmitting them sends (I axis value) * cos(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t).

Receiving the Complex Signal

At the receiver, we have to somehow split the combined signals back into vertical (Q) and horizontal (I) coordinates, so that we can figure out that we sent 1111. We receive the signal, and create two copies of what we received. We take one copy and we multiply it by cos(2*pi*fc*t). We take the other copy and we multiply it by sin(2*pi*fc*t). We integrate over our time period *T*. This integration over *T* is important not just because it helps mitigate noise and interference, but also because it lets us take advantage of several trigonometric identities.

First, let's look at the [received signal] * cos(2*pi*fc*t) multiplication. Here is the result of this multiplication with the received signal in square brackets [ ]:

[(I axis value) * cos(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t)] * cos(2*pi*fc*t)

Rewrite it to distribute (multiply) our cos(2*pi*fc*t) through.

(I axis value) * cos(2*pi*fc*t) * cos(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t) * cos(2*pi*fc*t)

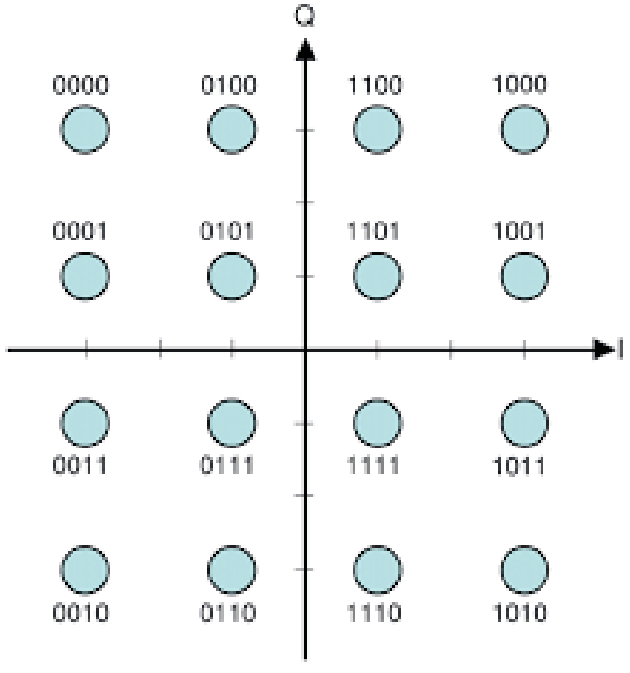


Figure 1 – The graph illustrates quadrature modulation (QAM) by which 16 different values can be transmitted. Each circle represents a unique value of 0000 to 1111 that is transmitted as a single combination of the Q and I signals, each with a value of 00 to 11.

Let's focus on the term on the left-hand side first, with the I axis value. We'll set aside the Q axis value term for now.

(I axis value) * cos(2*pi*fc*t) * cos(2*pi*fc*t)

Aha! We can convert that cos() * cos() term to something we can use. Using the half angle identity, square each side, and double all the angle measurements (easy, right?). After this bit of cleverness, here is the result.

cos(2*pi*fc*t) * cos(2*pi*fc*t) = 1/2 * [1 + cos(2*pi*2fc*t)]

So now we can replace cos(2*pi*fc*t) * cos(2*pi*fc*t) with something that will be useful in the next few steps.

(I axis value) * 1/2 * [1 + cos(2*pi*2fc*t)]

See that 2fc term in there? We had a 2fc term in our earlier demodulation. Let's rearrange things by distributing the I axis value through.

(I axis value/2) + [(I axis value/2) * cos(2*pi*2fc*t)]

Remember we are integrating over the time period *T*, at the receiver. We now have one of the two terms rewritten in a useful way. What happens when we integrate a cosine signal from 0 to *T*? That value happens to be zero! This leaves just the integration of (I axis value/2)!

The result at the receiver for the multiplication and integration of the first copy of the received signal is (I axis value)*(*T*/2). We know *T* and can divide it by 2, so we know the I axis dimension value.

But wait! We only did the first part. Remember we had

(I axis value) * cos(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t) * cos(2*pi*fc*t)

We recovered the I axis value from the term before the plus sign. But what about the term after the plus sign? We set it aside, and now we need to take a look at it.

(Q axis value) * sin(2*pi*fc*t) * cos(2*pi*fc*t)

What do we do here? Trigonometry saves us here as well. When we integrate sin(2*pi*fc*t)*cos(2*pi*fc*t) from 0 to period *T*, it just happens to be zero, so the entire Q axis value term drops out. Does the same technique work for the copy of the received signal that we multiply by sin(2*pi*fc*t)?

You bet it does! First, let's multiply and distribute our sin(2*pi*fc*t) across the second copy of the summed I and Q signals that we made. We multiply and get:

[(I axis value) * cos(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t)] * sin(2*pi*fc*t)

And rewrite it to distribute our cos(2*pi*fc*t).

(I axis value) * cos(2*pi*fc*t) * sin(2*pi*fc*t) + (Q axis value) * sin(2*pi*fc*t) * sin(2*pi*fc*t)

Now that we know that integrating cos(2*pi*fc*t) * sin(2*pi*fc*t) from 0 to *T* is zero, we can drop out the I axis value term right away. That's good because we already have the I axis value from multiplying our received summed signal by cos(2*pi*fc*t) and using trigonometry cleverness.

We are left with

(Q axis value) * sin(2*pi*fc*t) * sin(2*pi*fc*t)

And we once again use the half angle trig identity, square each side, and then double all angle measurements.

sin(2*pi*fc*t) * sin(2*pi*fc*t) = 1/2 * [1 – cos(2*pi*2fc*t)]

We can replace sin(2*pi*fc*t) * sin(2*pi*fc*t) with 1/2 * [1 – cos(2*pi*2fc*t)] which gives us

(Q axis value) * 1/2 * [1 – cos(2*pi*2fc*t)]

And when we multiply through by the Q axis value, we get

(Q axis value/2) – [(Q axis value/2) * cos(2*pi*2fc*t)]

Hey, guess what goes to zero again? That's right, cosine integrated from 0 to *T* is zero. We are left with the constant term that integrates to (Q axis value) * (*T*/2).

The following is key: When we multiply the summed signal that we received by cosine, we get the I axis value. When we multiply the summed signal that we received by sine, we get the Q axis value. This technique is called *quadrature mixing*.

Advantages of Complex Modulation

I and Q give us the coordinates on the 16QAM chart in Figure 1. As long as the receiver frequency is synchronized with the transmitter (a whole other story) and as long as our map of which point stands for which value at the receiver is the same as at the transmitter, then we have successfully received what was sent.

Moving from a single-carrier real signal to a complex signal, where two real signals are sent at the same time using math to separate them at the receiver, gives us advantages. We can send more bits without having to send more amplitude levels. Our two signals are each handling four levels, but using the results in a two-dimensional grid gives us a lot more bits per unit time without having to change our performance expectations. Sending 16 different amplitude levels is harder than sending four. So, we send four levels twice—once using each carrier. We use some mathematical cleverness to unlock the grid coordinates from the transmitted waveform.

However, using this complex modulation scheme gives us yet another advantage. Because of the math we just did, we eliminate an entire image when compared to a single carrier real signal. Remember when we multiplied the amplitude by our carrier and we got an image at fc and -fc? Quadrature mixing produces just one image. Filtering becomes less difficult because we no longer create a second image.

A third advantage of I and Q modulation is that it isn't restricted just to sending 16QAM. Using an I and a Q signal, and a fast enough sample period *T*, means you can send any type of modulation or waveform that you can dream up. Now that's some radio power!

[There are practical considerations beyond the scope of this article regarding the amount of information that can be transmitted in a fixed bandwidth (Shannon's limit), the waveform of the modulating signals, and the ratio of signal energy for each bit compared to received noise (Eb/NO). These would have obscured the basic mathematical techniques of complex modulation and have been omitted for clarity. – Ed.]

Complex modulation does require some signal processing at the receiver, but this type of signal handling is at the heart of every software defined radio. And now you know how complex signals work!

Upcoming Conferences

GNU Radio Conference (GRCon25)

September 8 – 12, 2025
Everett, Washington

https://events.gnuradio.org/event/26/

GRCon25, the 15th Annual GNU Radio Conference, will be held Monday, September 8 through Friday September 12, 2025, at the Edward D. Hansen Conference Center in Everett, Washington.

Zero Retries Digital Conference 2025 (ZRDC2525)

September 13, 2025
Everett, Washington

https://www.zeroretries.org/p/conference

The Zero Retries Digital Conference 2025 will be held Saturday, September 13, at the Edward D. Hansen Conference Center in Everett, Washington. *While GRCon2025 and ZRDC 2025 are being held consecutively at the same venue, the two events are independent of each other.*

Pacific Northwest VHF Society Conference

October 10 – 11, 2025
Wenatchee, Washington

www.pnwvhfs.org

The Pacific Northwest VHF Society Annual Conference will be held Friday and Saturday, October 10 – 11, 2025 in Wenatchee, Washington.

The RSGB 2025 Convention

October 10 – 12, 2025
Milton Keynes, United Kingdom

www.rsbg.org

The Radio Society of Great Britain 2025 Convention will be held October 10 – 11, 2025 at the Kents Hill Conference Centre, Kents Hill, Milton Keynes, United Kingdom.

43rd Annual AMSAT Space Symposium

October 16 – 19, 2025
Phoenix, Arizona

www.amsat.org

The 43rd Annual AMSAT Space Symposium and Annual General Meeting will be held Thursday through Sunday, October 16 – 19, 2025, at the Holiday Inn & Suites Phoenix Airport North in Phoenix, Arizona.

Microwave Update 2025

October 17 – 18, 2025
Tucson, Arizona

https://microwaveupdate.org

Microwave Update 2025 will be held Friday and Saturday, October 17 – 18, 2025, at the Casino Del Sol Conference Center in Tucson, Arizona. The event is being hosted by the Arizona Microwave Group.

Jim Willis, KG6TW
P.O. Box 1232, Young Harris, GA 30582 | **jimwill172@gmail.com**

A Visual Exploration of HF Propagation Using WSPR Data

A review of factors that influence HF propagation using WSPR data.

Introduction

There are several reasons hams get on the air: Regularly scheduled nets typically operate in a defined geographical area and at set times. DXers are looking for contacts outside of their country. EmComm operators may want to use NVIS. The rest of us just get on the air and call CQ! Each of these scenarios requires us to pick a frequency that is suitable for radio communications depending on our location, the season, time of day, distance, and band conditions. The question is: "How do we pick the right frequency?"

The goal of this article is to illustrate the factors which influence HF propagation using WSPR data. (WSPR is the beacon mode that is part of the *WSJT-X* software suite.) We will consider the following factors:

- The regions of the ionosphere
- The secant law
- Geometry
- Absorption in the D-region
- Critical frequencies for the F2-, F1- and E-layers

Note that in this article we consider only *single-hop* HF propagation. Other factors that affect HF propagation include space weather, atmospheric and man-made noise.

The Layers of the Ionosphere

The ionosphere is characterized by the presence of ionized regions. We will be focusing on the D, E and F regions. The F2-layer **[1]** is present throughout the day, while the F1-layer and E-layer are present only during the daytime. The degree of ionization and its ability to *refract* HF propagation is characterized by the layer's critical frequency, fo, and height. **Table 1** illustrates the variation of fo and height on a typical June day. In the table, foF2 **[3]**, foF1 and foE are the critical frequencies for the F2-, F1- and E-layers. hmF2, hmF1, hmE are the maximum heights for the F2-, F1- and E-layers.

The Secant Law

The secant law describes whether signals of a given frequency will be reflected **[4]** back to Earth when given the transmitting frequency, the critical frequency of the governing layer, and the incident angle, φ.**[5]** The secant law states:

MUF = fo * secant φ

The secant law describes the minimum propagation distance. For example, for Near Vertical Incidence Skywave (NVIS) propagation to work, the transmitted frequency must be less than or equal to the governing critical frequency.**[6]** This is an important consideration for ARES planners and other emergency responders.

Geometry

The secant law explains the minimum propagation distance but what is responsible for the maximum propagation distance? The answer is "geometry." The maximum propagation distance is determined solely by the height of the reflecting layer. For example, if radio signals are reflected by the F2 layer at 350 km then the maximum propagation distance will be about 3500 km. Conversely, if the signal is reflected by the E-layer at a height of 110 km then the maximum propagation distance is about 1500 km.

Absorption in the D-region

The D-region absorbs HF signals. The amount of absorption is approximately **[7]** proportional to the wavelength squared and is characterized by the frequency at which there is 1 dB of absorption. In the example shown in **Figure 1**, the frequency

Table 1 – Typical Ionosonde data from Eglin AFB [2]

Date/time UTC	*Critical Frequency (MHz)*				*Layer Height km*	
	foF2	**foF1**	**foE**	**hmF2**	**hmF1**	**hmE**
6/15/2024 6:15	8.725	-	-	385.7	-	-
6/15/2024 21:00	9.663	5.32	3.58	304.1	186.2	99.5

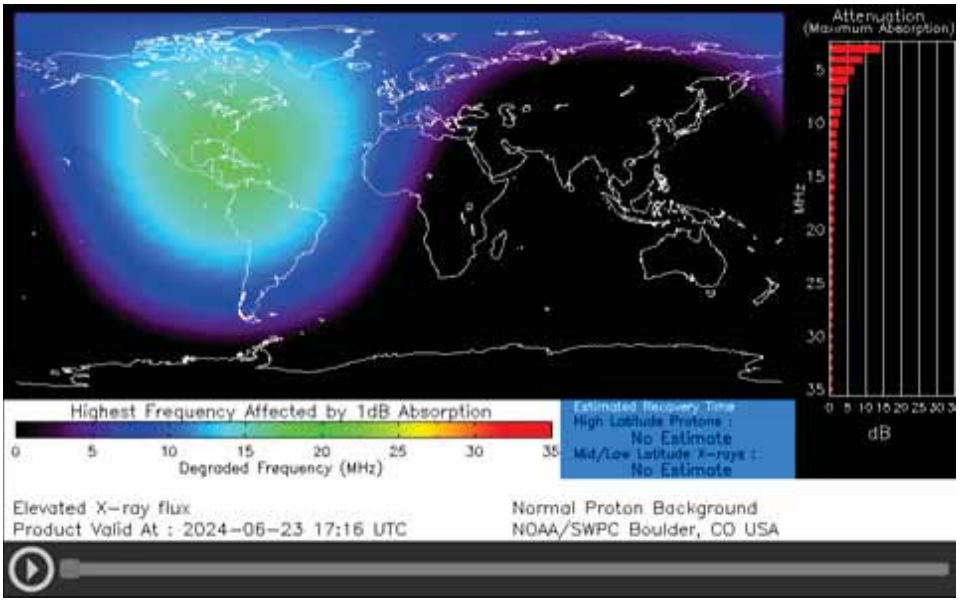


Figure 1 – "Heat map" of the highest frequency for 1 dB attenuation in the D-region.**[8]**

shown for North America at 1716 UTC on 23 June 2024 is about 18 MHz.

D-region absorption is only significant during daylight hours, as can be seen in Figure 1. This fact has important consequences for daytime operation on the 80- and 40-meter bands.

Critical Frequencies for the F2-, F1- and E-regions

As mentioned earlier, the critical frequency of the governing layer determines if the HF signal will be reflected and if so, the incident angles that will be reflected. The critical frequency of the governing layer and the layer height determines the minimum propagation distance for that band, location and time of day. Referring to **Figure 2**, we see the critical frequency for the F2 layer increases during the day, until about noon and then slowly decreases during the afternoon and nighttime. We would expect to see the minimum propagation distance for HF signals being reflected by the F2 layer to increase during the nighttime due to the secant law. **Figure 3** shows a world-wide map of estimated critical frequencies on 8 January 2025 at 2045 UTC.

The height of the governing layer determines the maximum propagation distance, independent of frequency. This means the maximum propagation distance is the same for all the HF bands. The calculated and observed maximum propagation distance is about 3500 km for a layer height of 350 km.**[9]** 40-meters is the exception to this rule because during the daytime, E is the governing layer.

The F1- and E-layers are only present during the daytime. The critical frequencies, for both layers, are proportional to the amount of sunlight present given the location, season and time of day. Based on the data presented in this article, it appears the E-layer is more important for HF propagation than the F1-layer.

80-meters

Referring to **Figure 4**, the number of spots per hour increases from about 0000 UTC until about 0800 UTC then decreases to zero by about 1400 UTC. 80-meters is a nighttime band due to D-region absorption present during the daytime. The observed maximum propagation distance is about 3500 km in agreement with the maximum propagation distance mentioned above.

40-meters

Referring to **Figure 5**, the number of spots per hour increases from about 1900 UTC until about 0300 UTC, remains

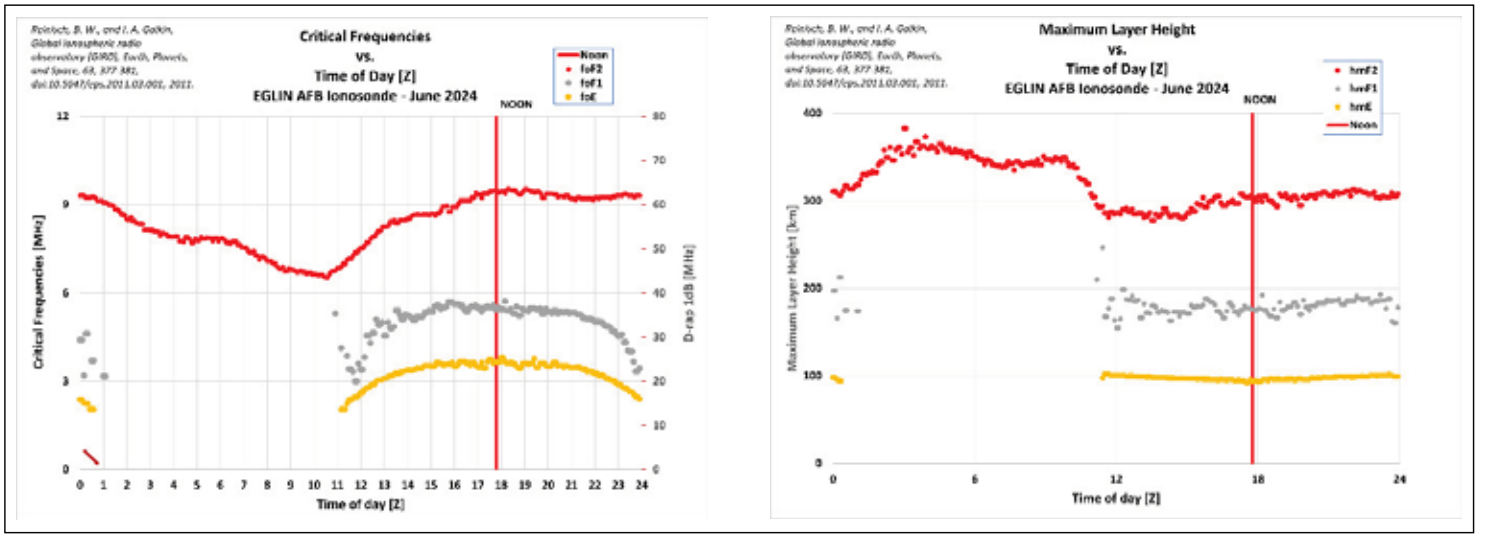


Figure 2 – Critical frequencies and maximum layer height averaged over the month of June 2024, EGLIN AFB.

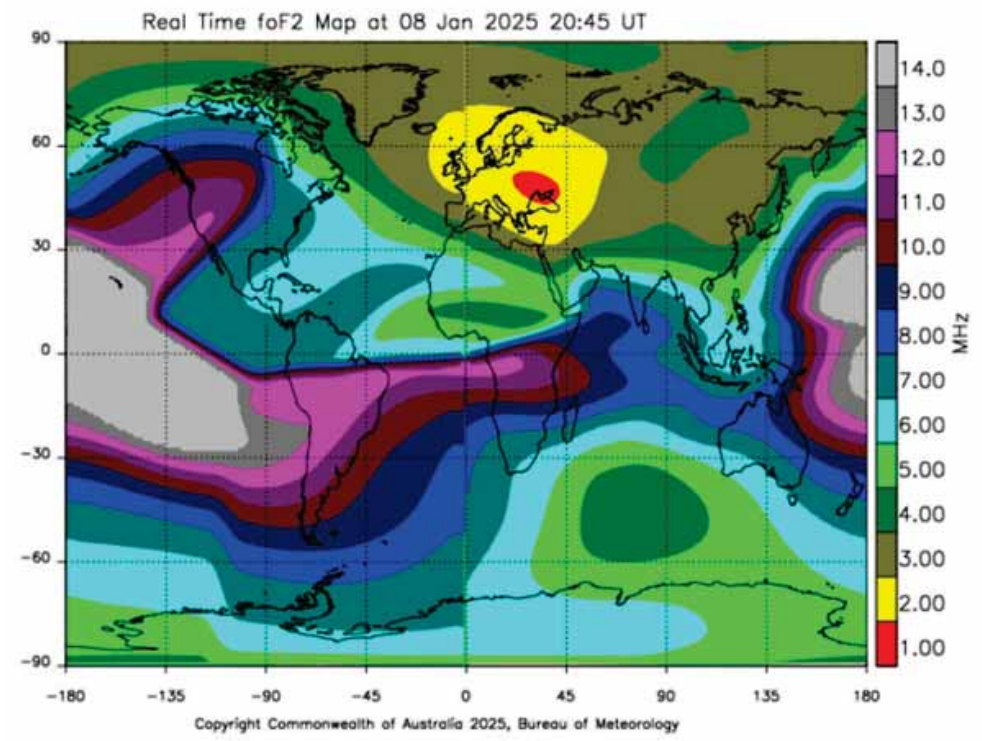


Figure 3 – Estimated critical frequencies map, foF2, for 8 Jan, 2025 at 2045 UTC.[**10**]

more or less constant until about 1200 UTC, decreases until about 1600 UTC, then remains constant until about 1900 UTC. The observed maximum nighttime propagation distance is about 3500 km indicating reflection from the F2 layer at a height of about 350 km. The observed maximum daytime propagation distance varies from about 750 km to about 1000 km, indicating reflection from the E-layer at a height of about 100 km coupled with significant absorption from the D-region. Previous studies show the maximum propagation distance from the E-layer should be about 1500 km. This difference can be attributed to absorption in the D-region.

20-meters

Referring to **Figure 6**, the number of spots per hour decreases from about 0000 UTC until about 0900 UTC, remains more of less constant until about 1000 UTC then increases until about 1300 UTC, remains constant until about 2100 UTC, then increases until about 0000 UTC. The observed maximum propagation distance, during both daytime and nighttime, is about 3500 km, indicating reflection from the F2-layer at a height of about 350 km. The observed minimum propagation distance varies throughout the day due to changes in the critical frequency foF2. For example, the "notch" in the minimum propagation distance from about 0600 UTC to about 1200 UTC is due to the decrease in foF2.

15-meters

Referring to **Figure 7**, the number of spots per hour decreases from about 0000 UTC until about 0600 UTC, remains more or less constant at about zero until about 1200 UTC, then increases until about 1500 UTC, remains constant until about 2200 UTC, then increases until about 0000 UTC. The observed maximum propagation distance, during both daytime and nighttime, is about 3500 km, indicating reflection from the F2 layer at a height of about 350 km. The observed minimum propagation distance varies throughout the day due to changes in the critical frequency foF2. For example, the "notch" in the minimum propagation distance from about 0600 UTC to about 1200 UTC is due to the decrease in foF2. The minimum propagation distance of about 750 km, observed between 1400 UTC and 0000 UTC, is consistent with the F2-layer height of about 300 km and a critical frequency of about 12.5 MHz.

10-meters

Referring to **Figure 8**, the number of spots per hour decreases from about 0100 UTC until about 0400 UTC, remains

Table 2 – Daytime and nighttime, minimum and maximum propagation distances and activity levels for the 80-, 40-, 20-, 15-, and 10-meter bands.

	Propagation Distance km				*Activity*	
Band (meters)	*Daytime minimum*	*Nighttime maximum*	*Daytime minimum*	*Nighttime maximum*	*Daytime*	*Nighttime*
80	N/A	N/A	0	3500	N/A	Good
40	0	1500	0	3500	Varies	Good
20	750	3500	750-1500	3500	Good	Varies
15	750	3500	2000	3500	Good	Varies
10	750	3500	3000	3500	Good	Varies

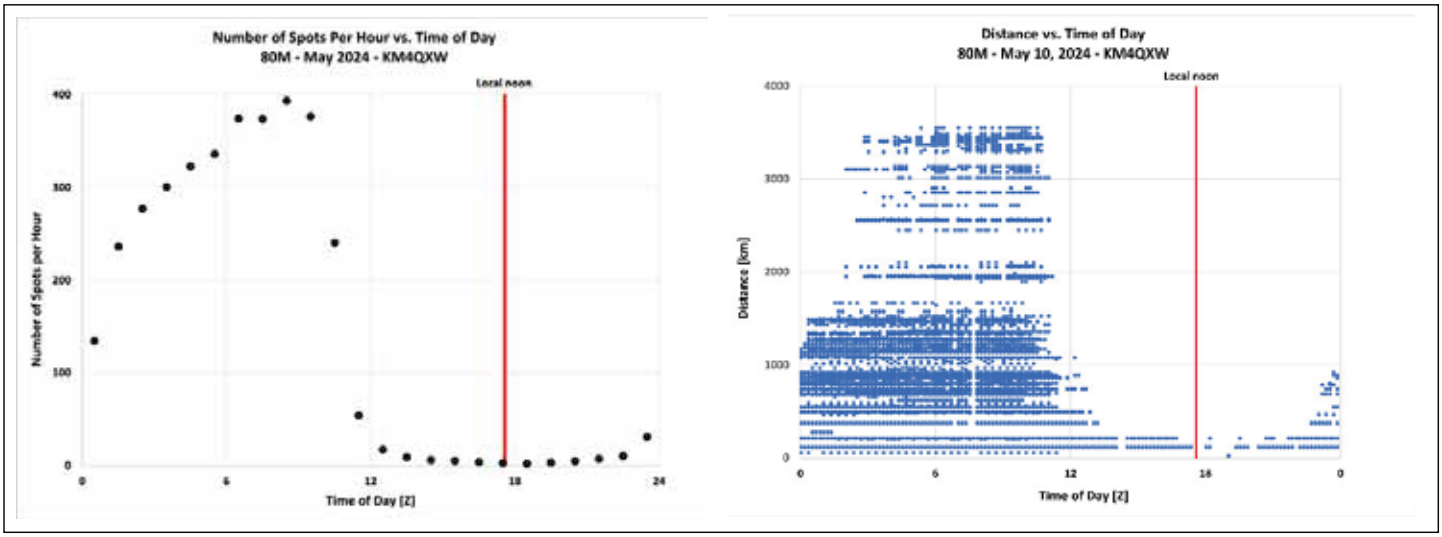


Figure 4 – Number of spots per hour and distance versus time of day for 80-meters.

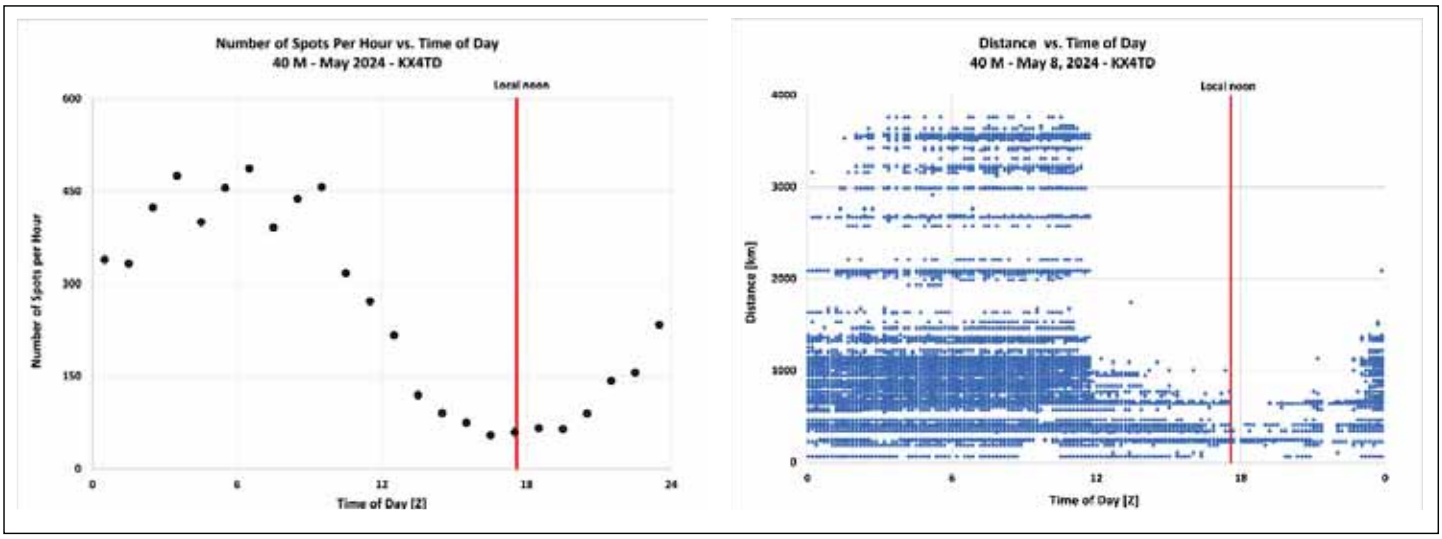


Figure 5 – Number of spots per hour and distance versus time of day for 40-meters.

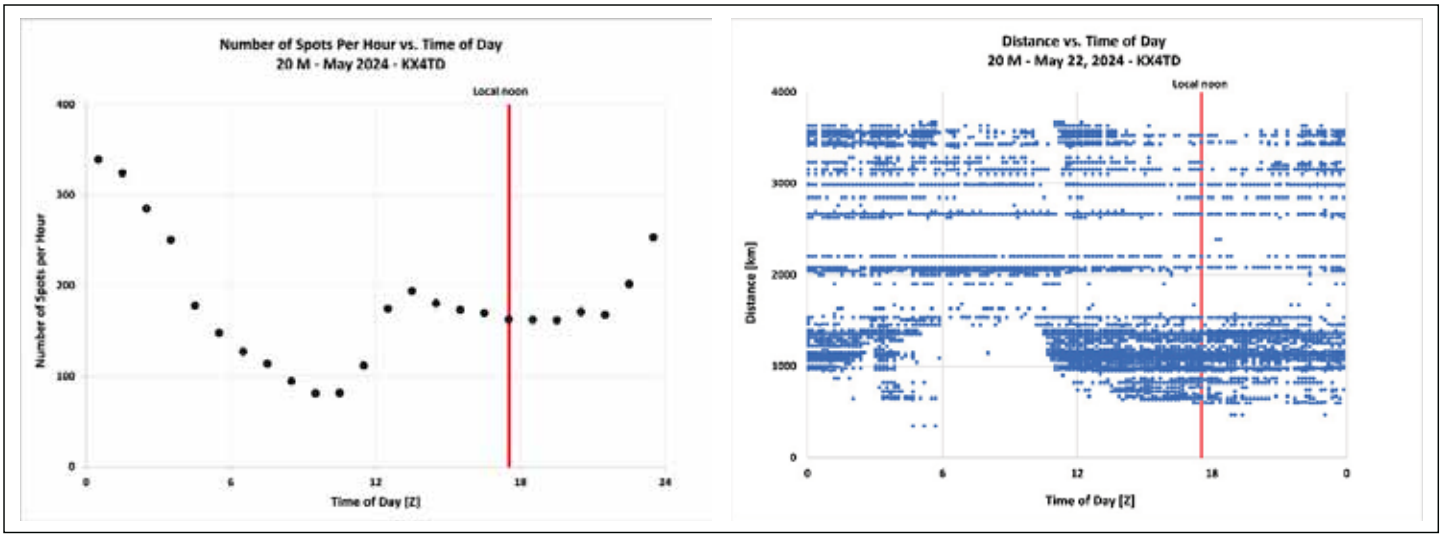


Figure 6 – Number of spots per hour and distance versus time of day for 20-meters.

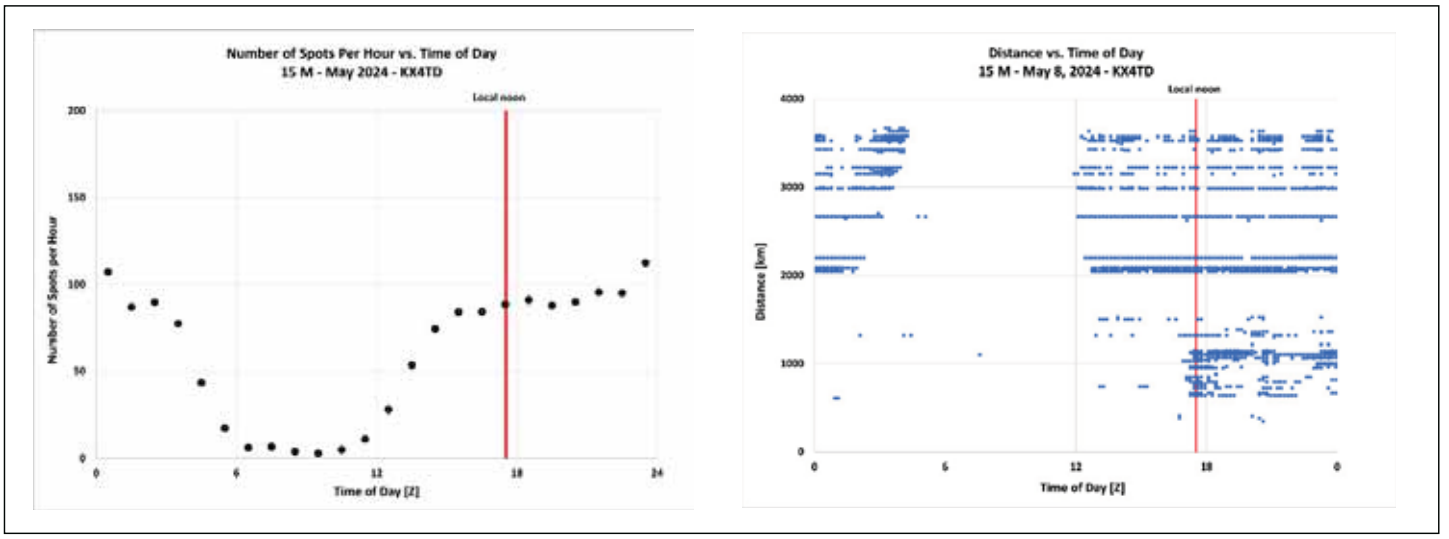


Figure 7 – Number of spots per hour and distance versus time of day for 15-meters.

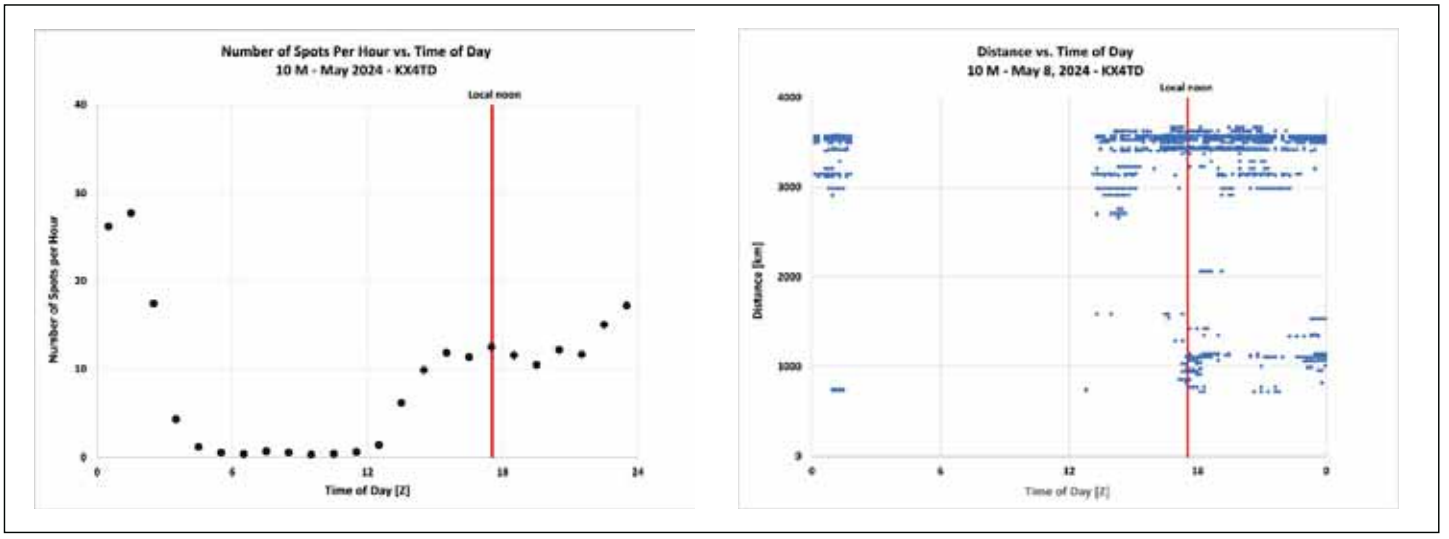


Figure 8 – Number of spots per hour and distance versus time of day for 10-meters.

more or less constant at about zero until about 1200 UTC, then increases until about 1500 UTC, remains constant until about 2100 UTC, then increases until about 0100 UTC. The observed maximum propagation distance, during both daytime and nighttime, is about 3500 km, indicating reflection from the F2-layer at a height of about 350 km. The observed minimum propagation distance varies throughout the day due to changes in the critical frequency foF2.

For example, the "notch" in the minimum propagation distance from about 1300 UTC to about 1700 UTC is due to the variation in foF2. The minimum propagation distance of about 750 km, observed between 1800 UTC and 0000 UTC, is consistent with the F2-layer height of about 300 km and a critical frequency of about 12.5 MHz.

Discussion

As with all amateur radio, to make a contact (or a spot), we need to select an operating frequency based on the location of the transmitter and receiver, the time of day, the season and the sunspot cycle. As stated earlier, the calculated maximum propagation distance depends only on the layer height and not on the frequency (provided the secant law is satisfied).

The observed maximum daytime propagation distance on 40-meters varied between about 500 and 750 km. However, the same transmitter recorded a maximum propagation daytime distance on 40-meters of 1500 km during the previous winter. The amount of D-region absorption varies depending on the season, with winter having the least amount of absorption and summer having the greatest amount of absorption.**[11]** From this, we conclude that the difference between summer and winter maximum propagation distance on 40-meters is due to D-region absorption.

Picking the "best" NVIS frequency is a balance between staying below the governing critical frequency and the daytime absorption on the lower frequency bands.

Jim Willis, KG6TW, was first licensed as WN4CCA in 1974 and spent most of his career in high tech engineering. His interest in amateur radio includes ARES, supporting public service

events, and HF propagation. He holds an Amateur Extra-class license and is retired and living in the north Georgia mountains.

Notes

[1] I use "layer" to denote a particular instance, for example "the critical frequency of the F2-layer" and "region" to denote the concept, for example, "the D-region."

[2] Critical frequency data provided by: B.W. Reinisch and I.A. Galkin, Global ionospheric radio observatory (GIRO), *Earth, Planets, and Space*, 63, 377-381, doi:10.5047/eps.2011.03.001, 2011.

[3] foF is the critical frequency for the F-region. The critical frequency is the highest frequency where the transmitted signal is reflected from the ionosphere vertically. The "o" stands for "ordinary" as opposed to "extraordinary." The difference in the two is beyond the scope of this article.

[4] Radio waves are refracted or "bent" when passing through regions of varying index of refraction. When the radio waves are bent enough to return to Earth they appear to be reflected. The reflection appears to take place at the "virtual" height which is higher than the actual radio wave path. See: Letter Circular LC 575, The Ionosphere and Radio Transmission Conditions, with Special Reference to the Observing and Reporting Service of the National Bureau of Standards, (1939), pp. 5, Fig. 2.

[5] φ is the angle between the incident radio wave and the normal to the ionosphere layer. MUF is the maximum usable frequency. The Secant Law was proposed by Breit and Tuve, published *Phys. Rev*. 28, 554, (1926).

[6] Although this article refers to reflection of "ordinary" radio waves, reflection of "extraordinary" radio waves occurs at a slightly higher frequency than ordinary waves and may be useful for NVIS operators.

[7] K.A. Zawdie, D.P. Drob, D.E. Siskind, and C. Coker (2017), Calculating the absorption of HF radio waves in the ionosphere, *Radio Sci*., 52, 767–783, doi:10.1002/2017RS006256, Eq. 1 and 2.

[8] From the NOAA website, **https://www.swpc.noaa.gov/products/d-region-absorption-predictions-d-rap**.

[9] *K9YA Telegraph*, Volume 20, Issue 3, March 2023.

[10] https://www.sws.bom.gov.au/HF_Systems/6/5

[11] In the Northern Hemisphere.

DX Engineering Makes

Contest Season

Fun!

We make it easy to get what you need for higher scores: DX Engineering's RF-PRO-1B® Active Magnetic Loop Antenna and other receiving devices; HamPlus Antenna Switches; RigSelect's Pro Transceiver Switch and SO2R Controller; VA6AM, RF Meca, and 4O3A band pass filters; rigs like the Icom IC-7610; and more.

Contact us for the best products, fastest shipping, and most responsive service from experienced operators—a winning combination every time!

Make the Change to a More Satisfying Ham Radio Purchasing Experience

- Easy ordering by phone or web
- Products from over 180 leading manufacturers
- Friendly customer service from hams with a combined 1,000+ years in amateur radio
- Fastest shipping in the industry
- Responsive and ongoing technical support
- Not 100% happy? We make it right!

Order by Phone

800-777-0703 Country Code: +1
9 am to midnight ET, Monday-Friday
9 am to 5 pm ET, Weekends

Order Online

www.DXEngineering.com
Most orders over $99 ship free
Request a Free Catalog

Tech Support

330-572-3200
DXEngineering@DXEngineering.com
9 am to 7 pm ET, Monday-Friday
9 am to 5 pm ET, Saturday

DX 25™ ENGINEERING

bhi · HEIL HAM RADIO · INRAD international radio · PHONEMA · vezdx ELECTRONICS DESIGN INC.

ON ALL BANDS

OnAllBands.com is dedicated to educating and informing the Amateur Radio community.

HANDBOOK 101

The Next Generation of Amateur Radio!

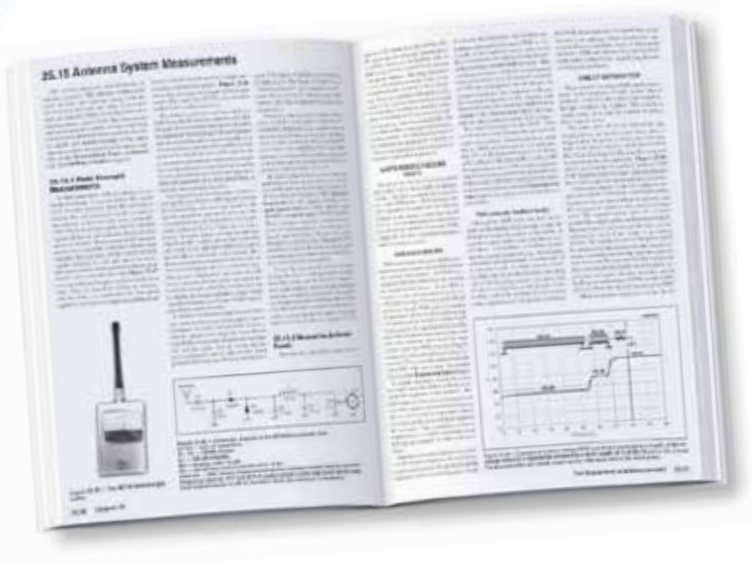
Softcover Book
Includes e-book.
Retail $69.95

Six-Volume Set
Includes e-book.
Retail $69.95

e-Book
Retail $69.95

Major Updates:

- Electromagnetic analysis and inexpensive tools for modeling circuits, antennas, and propagation.
- Higher-level modeling of transmitters and receivers.
- Preparing your station for emergency operations.
- Radio astronomy receiver and antenna design.
- Batteries and battery safety.
- NEC4 and antenna modeling software.
- SWR meters and related tests.
- RF safety and compliance with FCC exposure regulations.



The ***ARRL Handbook*** is your complete guide to wireless technology experimentation, practice, and development. Since 1926, the ***Handbook*** has captured the state of radio science and technology in one authoritative work. Use it to delve into radio electronics, circuit design, digital modulation techniques, and equipment construction.

BONUS e-Book Download!

Your purchase includes the fully searchable digital edition of the printed book (PDF format), plus expanded content, software, PC board templates, and other support files.

Order online at www.arrl.org/shop | Call toll-free US 1-888-277-5289