**Twister Presentation, November 2022**

**Introduce Myself**

* Name, Research Focus.

**Topic: The TWISTER system, and some practical DSP.**

* Motivation: To share some of the system capabilities; you may want to use it.
* Motivation: To share lessons learned – they might be useful to you (and save you some time).
* Theme – little details matter.

**First: What is the TWISTER system?**

* Acquired last year on a grant from NSF.

**TWISTER system overview**

* More commonly referred to as “the MRI equipment.” It’s not an MRI machine.
* It is a collection of high-speed, high bandwidth signal generation and capture equipment.
  + LIST PARTS.

**The Majors**

* First, the major parts of the system.

**M8195A**

* 65 GSa/s, 16 GSa waveform memory. These set the maximum bandwidth and length of your signal, respectively.
* At top speed of 65 GSa/s, the memory depth (16 GSa) allows you to store 0.2462 seconds of waveform data. Slower speeds (lower bandwidth) can be longer.
* 4 Channels, but not really. If you really need all 4, give us a bucket of money and we’ll pass it on to Keysight.
* This is an impressive piece of equipment. Very useful – you can directly synthesize any waveform you want, from 0 to 32.5 GHz. Actually, you can do different waveforms at once. We’ll get to how to actually generate your own waveforms later on.

**Aside: Nyquist Limit**

* Nyquist limit – you must sample a waveform at twice the bandwidth (or faster) to preserve all the information. Same for generating a waveform. The faster you produce samples, the higher frequency features you can produce. 65 GSa/s -> 32.5 GHz bandwidth, max.

**VDI up and down converters.**

As a research group, we’re *usually* interested in frequencies slightly higher than 32 GHz – more to the tune of 320 GHz. So, we have a set of up and down converters from VDI that take an RF input signal and shift it up in frequency.

* SHOW VDI CONVERTER.
  + This is the physical version of that little “X” on system block diagrams.
* Take in a local oscillator, multiply it up using diode mixing, and finally mix the RF input signal with the upconverted LO.
* We have: 110-170, 140-220, 220-330, 330-500.
  + So, all together, we can span 0-32.5 GHz natively, then 110-500 GHz with the use of upconverters.
  + If you want to cover something in the 30-110 GHz range… sorry. Let us know and we’ll use that as an excuse to buy stuff.
* The VDI modules require a single RF input (NOT I and Q), and a high-quality local oscillator to multiply.
  + Note that this means a double-sideband signal is sent OTA. This is the first “gotcha” you might want to know about. We’ll also discuss sidebands here in a little bit.
  + We will hopefully have a sideband filter soon, as well as an amplifier for the lower end of that range.
* So how do they work? It’s magic!

**It’s not magic: Harmonics**

Just kidding; here’s a simplified version of what’s going on.

* Harmonics: Integer multiples of input, caused by nonlinearity.
* Diodes are a useful source of nonlinear behavior in electronics.

**It’s not magic: Mixing**

* Often, harmonic generation is a case of self-mixing. So what’s mixing?
* Equation in a nutshell.
* Mixing is a basic tool we have for moving things around in frequency, and for comparing one frequency to another. Very important for communications.
* Harmonic generation and mixing are the processes going on inside the VDI modules.

And now, back to the equipment rack!

**E8257D PSG (analog signal generators)**

Next on the rack is our analog signal generators.

Question: Why do we need analog generators (they only produce Sine waves, after all) when we have an *arbitrary* waveform generator right above them?

**Digitization Noise**

* Introducing another “gotcha” – digitization noise. Caused by finite precision.
  + The difference between the true value and the digital value is a source of noise.
  + The M8195A only has 8 bits of resolution. That results in digitization noise – which is already bad, but for a periodic signal (like a sine wave), those points where sampling error occurs are periodic too… meaning there will be strong spectral content there. -> Harmonics.
  + **See figures**
  + Trying to do diode mixing and up conversion with unwanted harmonics present is bad. They multiply like rabbits and give you unwanted harmonics on your LO. Then you get unwanted copies of your signal at each of those harmonics when you mix your signal onto the high frequency LO.

**Another Example**

* Another digitization noise example, for a communication waveform.
  + If you try this yourself later today (I hope you have better things to do, but still), you might get a spectrum that looks a little different. Time for a quick aside from this quick aside.

**Really quick aside: Spectral leakage**

* Served fresh to you from Wikipedia!
* I wasn’t sure where to put this, but I wanted to include it because this was driving me crazy for a long time and maybe I can save someone a bit of trouble later on.
  + It’s another signal effect caused by digitization, so it fits here.
* Discuss spectral leakage :
  + Windowing a sinusoid produces spectral leakage. (Scalloping)
    - You must window because no signal lasts forever.
  + When you sample that windowed sinusoid, your window duration sets where on the spectrum your sample points fall.
  + If you have an integer number of cycles per window, your samples fall so that it gives the illusion of no spectral leakage.
  + Note: scalloping can result in “loss” of the signal peak.
* Having an integer number of carrier cycles per waveform window ends up being important for sending communication waveforms with the AWG, but for a not-entirely-identical reason. We’ll discuss that a bit later.
* Back to the rack!

**The DSOV254A**

The last piece of major equipment on the rack is our 4-channel digital oscilloscope. It can sample up to 80 GSa/s, which gives it a Nyquist bandwidth of 40 GHz.

* 80 GSa/s => 3.8mm at lightspeed.
* Immediate gotcha: The Nyquist bandwidth may be 40 GHz, but the front-end amplifier is less. Practically, you have 25 GHz of bandwidth.
* But then again, as Russ can tell you, you can still *capture* signals at higher frequencies. In general, though, you’re limited to 25 GHz of bandwidth at the scope.
* You can set the sample rate lower – divide 80 GSa/s by any power of two.
  + Like the signal generator, this saves you waveform memory, making things faster (display updates, waveform processing, file transfers, etc) and allows you to capture longer time durations before running out of memory (205 MPts).
* This scope is a powerhouse – it’s incredibly powerful and versatile, and it’s also got quite the learning curve!

**“Minor” research instrumentation**

Like any good ensemble, it’s really the thoughtful accessories that make-or-break the whole setup. So… here’s some of the “minor” research instrumentation to be aware of.

**Practical Considerations**

Introducing some of the unsung heroes of the setup:

<SHOW some of these>

* Big ‘ole box fan. Make sure it’s on before using the system.
  + Seems silly, except without that cooling the rack will overheat and threaten the calibration on the equipment! Practical details matter! (that’s a theme)
* **THz absorbers (!! To be discussed !!)**. You may need these to prevent multipath and standing waves. They *do* make a difference, and not having them messed up our antenna characterizations for a long time.
* Antennas and lenses. Increase directivity = range / SNR, potentially at the cost of side lobes, phase aberrations and / or dispersion.
* 2.9 mm cables, adapters, and connectors. Safe handling procedures and documentation.
  + Gotcha: 2.9, 3.6, and SMA are all *mechanically* compatible. But they’re different tolerance and specs. In general, don’t mix-and-match. In some cases it’s okay, but READ THE ONLINE DOCUMENTATION before trying anything clever. Cables are $$.
* **The LNA (!! To be discussed !!)**

**Multipath Propagation**

* Discuss signal reflection, scattering, multipath, and how to make an absorber.

**The LNA - PROBLEM**

* Noise spurs. Digitization, sampling, dynamic range.
  + This is with no input. Just noise. 10 and 20 GHz are nasty.
* Frequency resolution versus noise floor.

**The LNA - SOLUTION**

* Boosts SNR two different ways.
  + 1) By overcoming cable losses between the downconverter and the ‘scope.
  + 2) More importantly, it boosts the received signal above the noise floor of the scope.
    - Important: The difference between the noise floor of the *signal* and the noise floor of the *scope*.
    - The LNA boosts the signal power *and* the signal noise floor by the same amount, then adds a little extra noise of its own. But as long the signal noise remains *below* the noise floor of the scope, the LNA is increasing our SNR.
    - In this case, our little LNA is giving us a boost of over 20 dB. Yes please!

**Antenna Systems**

One more set of practical issues relate to antenna systems.

* Goal: Increase directivity (more power, fewer problems)
* Use lenses, dishes, and horns.
  + Lenses: easy to align, dispersion could be problematic. Reflection coatings. Magnifying glass.
  + Horns: Very nice, but not high enough gain.
  + Dishes: great. Anybody want to design one for us?
  + GOTCHA: How big can you make a dish antenna? (DRAW THIS)

**Introducing an example case**

So let’s set up an example case – say you want to make a sine wave at 305 GHz, shoot it down a line-of-sight path between the receiver and transmitter (CCU and CCD), then capture it on the scope and save it for analysis. Simple, right?

**First stop: making the waveform.**

You can make lots of standard waveforms with the AWG front-panel software (BTW, both the DSO and the AWG run a full Windows OS), which is great – but let’s embrace the “arbitrary” part of the name and define our own waveform file that we’ll load into memory.

* After all, whatever your research is, you probably have some specific waveform you want to send, and it might be highly specialized. You might also want to have a copy of the original waveform you loaded into the scope as a reference when you process your capture later.
* Good news: I’ve done most of the work for you. (If you’ve been wondering what part of this presentation gives results of my work over the last couple semesters… we’re starting to get there).
  + Make your signal in matlab, then feed the signal and time vectors to the “makeAWGfiles.m” script. It turns your signal into a binary file the AWG needs.
    - Don’t like matlab? No problem. It can read in .csv or .mat files (made any way you’d like) and perform the conversion that way, too. You just need both *time* and a *signal*.
  + There are a few “gotchas” here. I’m going to tell you about them because they were great learning experiences, and the real point of this presentation is to mention all the little details that you don’t hear about when you learn the theory in class.
    - (1) Digitization noise. Remember me?
    - (2) Dynamic range. Scale the waveform to take up as much dynamic range as possible.
    - (3) Make sure there’s no DC component!!! You’ll damage the upconverters.
    - (4) Most interesting: If you were to just read the Keysight manual (yikes) turn your waveform into a bunch of 8-bit samples, then try to load it into the AWG, you’re *probably* going to get a helpful error message: “Waveform could not be imported with current settings.” But not always. Helpful, I know. So what gives?

Here’s the root of the problem: your waveform has to be a multiple of 256 samples. This is a requirement of the AWG. It loads waveform samples out of memory and into the output buffer in blocks of 256, and if you don’t have an exact multiple of 256 samples, it won’t allow you to output the signal. The error messages don’t tell you any of that, but they really should. Here’s how you fix it:

* Waveform adjustments tab. You can copy the waveform in memory (may take up to 256 copies, impossible for long waveforms since you run out of room), you can zero pad, you and chop samples off, or you can resample when you load the waveform.
  + **DISCUSS EACH OF THESE.**
* Resampling is the best, but there’s a catch – if you want to *remotely* load a waveform, you can’t resample.
* Best solution: make sure your *original* waveform is sampled at a rate that results in a total length that’s a multiple of 256.
  + This was a huge pain to get figured out. Fortunately, the function I wrote does all that for you, and resamples (if necessary) to ensure it only generates “friendly” files.
  + This was a major “gotcha.”

**Setting the LO frequencies**

Okay then. At this point, you’ll want to follow the system start-up documentation to get everything connected safely and powered up correctly. I won’t go into detail on that. Instead, I’ll deal with your next decision: How to place your signal where you want it.

* Okay. If we want 305 GHz, that means we’ll need the 220-330 GHz converters. N= 12,
* We want a 305 GHz signal over-the-air, so the TX\_LO should be 305/12 = 25.4167 GHz as the PSG output, right? Not so fast – it’s time to talk about sidebands!
  + Assume fc is 5 GHz.
  + Mixing IF with a 305 GHz carrier, you’re going to get 2 sidebands.
  + **NOTE:** One is shifted **UP by 305 GHz**, the other shifted **DOWN by 305 GHz** (which mirrors at 0 Hz and wraps back up). These are sidebands, and they’re identical… but neither one is at 305 GHz.
* Pick upper sideband for 305 GHz.
  + If the upper sideband is at 305, then we must subtract the IF of the signal (5 GHz) to get the carrier – 300 GHz. That’s where we want our multiplied LO to end up at, so we set the PSG to 300/12 = 25 GHz.
* This results in a sine wave at 305 GHz, with a duplicate copy at 295 GHz.
* Now we need to set the RX.

**Setting the receiver – Heterodyne versus Homodyne.**

Great. Now we need to set the receive LO.

* Option 1: Homodyne mixing. The most obvious thing to do is to set the receiver LO to be 25 GHz as well. If you do, then you’re performing homodyne mixing. So what happens?
  + You’re mixing a 300 GHz oscillator with a double-sideband signal, also at 300 GHz. Thus, you’ll get two double sideband signals, one shifted UP to 300 + 300 = 600 GHz, which is filtered out, and the other shifted DOWN to 0 Hz.
  + Note: This puts the sidebands at 0 +/- 5 GHz. The negative component, again, gets mirrored back up to + 5 GHz – right on top of the other sideband!
  + Stacking both sidebands in the sample place is the “homodyne advantage” – saving 3dB (double the power) can be a huge deal!
  + Gotcha: If the two oscillators aren’t *exactly* locked together, the sidebands won’t stack perfectly – they’ll be slightly off. This causes them to beat one against the other, alternating in phase and out of phase. This means you go between double the power (yay!) and destructive interference (0 power – very bad).
    - Connect both reference cables on the PSGs, wait 45 minutes, then adjust the phase of the oscillators to get maximum power.
    - Matthew has automated that process, but we haven’t done anything with a variable distance link yet.
    - We’re expecting a new addition to the family – a pair of Rubidium clocks that should let us lock the oscillators together over a long distance (when sync cables are impractical).
  + Gotcha: You can’t separate the sidebands. If you only want to study the 305 GHz sideband, you don’t want the 295 GHz band in your received signal too. If you need to separate the two bands, then you need heterodyne mixing.
* Option 2: Heterodyne mixing. Let’s set the receiver’s LO to be something other than 25 GHz. This will split the sidebands, at the cost of half your power wasted.
  + This is the most general case. Homodyne is just a special case of mixing, in which the two sidebands end up in the same place. In heterodyne (general) case, they will be different.
  + Let’s set our receiver LO to be 25.8333 GHz. Then our upconverted LO is 310 GHz, so our dual sidebands are shifted down to 300 – 310 = -10 GHz, with the upper sideband at -5 GHz and the lower sideband at -15 GHz. On the scope, those negative frequencies mirror, so the lower sideband will appear at 15 GHz, and the upper sideband at 5 GHz.
  + Throw away the 15 GHz signal (DSP, filtering, scope bandwidth, etc). The 5 GHz signal is the one you want – the 305 GHz sine wave.
  + Yes, placing the sidebands can be a huge pain. The takeaway: study up on sidebands and mixing and work out what your setup should be, so you know which signal is the one you want!

**Setting up the scope.**

Okay – you have the signal loaded into memory on the AWG. You have the PSGs outputting the right frequencies. Everything is powered up and safely connected according to the documentation online. Now what?

The last thing is to make sure the scope is set up correctly. Let’s talk about aliasing again.

* Is the bandwidth right? You need to sample at 2x the highest frequency you don’t want to alias.
* In our example, we want to capture a 5 GHz sine wave, so a sample rate of 10 GSa/s is sufficient, correct?
  + PROBABLY NOT!!! Remember, that lower sideband is floating around, too, up at 15 GHz. If you sample at 10 GSa/s, it will alias the 15 GHz lower sideband, and guess where it ends up!?! Right at 5 GHz.
    - **SEE DIAGRAM**
  + Pro tip: You can adjust the bandwidth of the DSO’s front-end amplifier. Set it to something between 5 and 15 GHz – that way, even though the 15 GHz lower sideband would technically alias, it will be attenuated below the noise floor before aliasing!
  + Which leads to…
* Be sure the front-end filter is set correctly.
  + Set too low: Even if you’re sampling fast enough to observe the Nyquist limit, your filter bandwidth could be set too low, and attenuate the signal before it is sampled.
  + Set too high: Not *theoretically* a problem, but practically it can let in unwanted signal (noise). It’s best to set the filter bandwidth just higher than the maximum frequency you want to measure.
    - Even though you can filter it out later in processing, noise spikes can eat up your dynamic range, so it’s best to reduce them via filtering if you can.
* Finally, you save the waveform on the ‘scope, then transfer it to your computer for whatever analysis you want to do.

**Communication Waveform Requirements**

So, that’s great, but my research isn’t just about sinusoids. Well, I guess by Fourier theory it actually is, but I’m interested in things with high bandwidth – communication waveforms, to be precise. I’m especially interested in being able to measure the BER.

* I need a communication waveform, as well as a software-simulated receiver.
* Won’t go into the receiver, but I do want to talk about some constraints that arise from **needing to repeat the waveform sequences back-to-back**.
* You need an integer number of carrier cycles per waveform (for repeated copies).
* You need the record length to be a multiple of 256.
* You need the sample rate to be between 54 and 65 GSa/s for the AWG (without dividers).
* You need the sample rate to have an integer number of samples per waveform duration.
  + Obviously you won’t have a non-integer sample number, but you can have a waveform duration that spans 256 and ¾ sample periods. Problem: What happens to the little bit of extra time at the end? It gets dropped – and this means there’s a discontinuity when frames are looped back-to-back. You lose time, which totally messes up your receiver.
* You must satisfy all of these at once.
  + Not possible for arbitrary combinations of symbol rate (sets duration) and carrier frequency. Solution: adjust the carrier center frequency by a small amount.
    - For long-ish waveforms, this is less than a tenth of a percent.

**Illustrate**

<Draw on this slide: Why are each of the requirements problematic if violated?>

**Questions?**

I hope you enjoyed the talk, and learned something interesting.

If not that, then I hope I saved you some time late on!