Light Water Reactor Sustainability Program

Laboratory Instrument Software Controlled Spread Spectrum Time Domain Reflectometry for Electrical Cable Testing

August 2023

U.S. Department of Energy

Office of Nuclear Energy

Laboratory Instrument Software Controlled Spread Spectrum Time Domain Reflectometry for Electrical Cable Testing

S.W. Glass, J. Tedeschi, M.P. Spencer, J. Son, M. Elen, L.S. Fifield
Pacific Northwest National Laboratory

August 2023

SUMMARY

This research discusses the development of a software-controlled laboratory instrument based spread spectrum time domain reflectometry system (SSTDR). This constitutes one task within Pacific Northwest National Laboratory's (PNNL's) Light Water Sustainability Program efforts, with a mission that includes advancing nondestructive examination techniques for off-line and on-line in-situ cable condition monitoring. In 2022, PNNL evaluated SSTDR for the detection and characterization of a number of cable anomalies (Glass et al. 2022). The review included a comparison of SSTDR to Frequency Domain Reflectometry (FDR) techniques, which have enjoyed encouraging feedback and are starting to be used in nuclear power plants for periodic cable condition monitoring of cable systems as part of the plant's overall cable aging management program. The FDR test introduces a broad-band chirp onto the cable at the cable end, then listens for any reflection from a change in impedance along the cable caused by a damaged conductor or insulation, splices, contact with moisture, or other cable anomalies. The signal is captured in the frequency domain and then transformed back to the time domain using an inverse Fourier transform. Based on the velocity of propagation, the impedance response signal is plotted against distance along the cable. Peak locations along the X-axis indicate the distance along the cable where a portion of the signal has been reflected back to the instrument as a result of a cable anomaly. The FDR test is considered the gold standard of reflectometry; however, it does require the cable to be de-energized to perform the test.

The LIVEWIRE commercial SSTDR produces a similar plot to the FDR, but all processing is in the time domain. A pseudo-random noise code (PN code) is input onto the cable conductor, and the instrument listens for any reflected response from cable anomalies. The SSTDR processes the signal as an autocorrelation, comparing the input PN code to any reflected signal detected. The autocorrelation analysis for thermal aging, water and water ingress detection, ground fault and phase-to-phase fault detection at various locations along the cable and with the cable attached and detached from a motor load, and on both energized and un-energized conditions were performed. These results were contrasted with FDR measurements of the un-energized cable. Results were encouraging but indicated more work was warranted; particularly with the SSTDR, it seemed that the insulation damage would likely be better evaluated with multiple bandwidth cable tests, particularly including larger bandwidths than were possible with the current commercial instrument. The commercial instrument's bandwidth was set at 6, 12, 24, and 48 MHz but note that SSTDR and FDR definitions of bandwidth trend similarly but are not the same. The FDR response could be more broadly adjusted, and the bandwidth of 100 to 500 MHz produced the best responses. FDR responses to anomalies were clearer than SSTDR responses, and indications were that a broader bandwidth SSTDR may lead to improved SSTDR detection capability. This project used a laboratory instrument-based SSTDR (primarily using an arbitrary waveform generator and a digital oscilloscope plus Python in-house software) that allowed software adjustment of the SSTDR bandwidth, window functions applied to the exciting PN code, and other aspects of the SSTDR signal processing. Hereafter, this will be referred to as the PNNL SSTDR. Evaluating the specific performance of the PNNL SSTDR is left to a separate report. This report documents hardware and software development to produce the SSTDR cable test system.

ACKNOWLEDGEMENTS

CONTENTS

FIGURES

TABLES

ACRONYMS

ARENA	Accelerated and Real-Time Environmental Nodal Assessment
AWG	arbitrary waveform generator
BPSK	binary phase shift keying
BW	bandwidth
Fc	carrier frequency
FDR	frequency domain reflectometry
FFT	fast Fourier transform
NPP	nuclear power plant
PN code	pseudo-random noise code
PNNL	Pacific Northwest National Laboratory
SSTDR	spread spectrum time domain reflectometry
V	volt
VNA	vector network analyzer

1. INTRODUCTION

Early tests conducted by Pacific Northwest National Laboratory (PNNL) and LIVEWIRE INNOVATION Inc. to evaluate commercial spread spectrum time domain reflectometry (SSTDR) cable testing technology (Glass et al. 2022) included the following observations and conclusions:

- Both frequency domain reflectometry (FDR) and SSTDR were able to detect and locate cable anomalies, including

    - the presence of water for the undamaged non-shielded cable (but not for shielded cable),

    - thermal damage from a section of the cable passing through an oven heated to 140 °C for 62 days,

    - phase low resistance (312-Ω, 400-Ω, and 675-Ω) faults.

- The FDR data taken at 100, 300, and 500 MHz bandwidth demonstrated remarkable peaks at the positions corresponding to the water bath location and oven entry and exit for thermal aging along the cable. The SSTDR showed peaks for the cable entry to and exit from both the water bath and the thermal aging oven at 6, 12, 24, and 48 MHz bandwidths, but they were not as clear as the FDR. Moreover, the bandwidth definition of SSTDR versus FDR was not equivalent; however, the evaluation indicated that a higher bandwidth SSTDR may yield a clearer cable test. In practice, having a broader range of frequencies for either SSTDR or FDR cable tests could improve the probability of detection. Lower bandwidths propagate further along the cable; however, reflection resolution is improved with higher bandwidths. Making such a change to the commercial system was not warranted without further evaluation, and it was realized that a laboratory instrument implementation of an SSTDR could support software adjustment of test band widths, windowing, and other signal processing methods prior to making hardware and firmware changes to a commercial instrument.

This report describes the development of a software-controlled arbitrary waveform generator (AWG) and oscilloscope-based SSTDR for cable testing. The work is part of an overall effort to develop a technical basis for assessing the level and impact of cable insulation aging and degradation in nuclear power plants (NPPs). Previous related work has included:

- test-bed development (Glass et al. 2023),

- effect of thermal aging (Glass et al. 2022),

- moisture detection (Glass et al. 2021),

- environmental stressors (Glass et al. 2017), and

- bulk assessment approaches (Glass et al. 2016).

2. MOTIVATION

Nearly 20 percent of the electricity produced in the United States comes from NPPs (Joskow 2006). NPPs were originally qualified for an operational lifetime of 40 years (Subudhi 1996; Gazdzinski et al. 1996). A majority of U.S. NPPs have applied for and been granted 20-year extensions following the original license period. Within NPPs, electrical cables are susceptible to aging and degradation. These cables are commonly low- and medium-voltage critical infrastructure required for power, control, and instrumentation associated with safety and operational systems (Blocker et al. 1996). These cables may be exposed to environmental stressors, such as elevated temperatures, moisture, and gamma radiation. Such stressors can cause degradation and eventual failure of electrical cables, which may lead to the loss of safety systems if not detected and managed (IAEA 2012). Practically all utilities include some form of cable nondestructive evaluation in their aging management program, since performance testing to assure acceptable cable function is more cost effective than replacing all cables that exceed their qualified life.

Utilities are continually working to improve test quality and reduce costs associated with their cable nondestructive evaluation programs. Continued use of NPP cables beyond the initial qualification period is justified by a comprehensive cable management program that includes inspection and testing of some or all safety-related critical cables to assure that the cables can perform under normal operation and under a design-basis event.

Given the large volume of different types of electrical cables servicing NPPs, versatile prevention and detection of electrical cable failure during the operating lifetime of an NPP is a vital part of aging management. To evaluate the degradation of electrical cables and particularly the interaction of electrical cable test technologies with various damage mechanisms, PNNL developed the Accelerated and Real-time Environmental Nodal Assessment (ARENA) test bed (Glass et al. 2023) see Figure 1. The vision behind the creation of this facility is to establish a modular test facility that allows for the implementation of a broad range of test methods to detect faults and anomalies in a variety of cables and systems in a controlled environment.

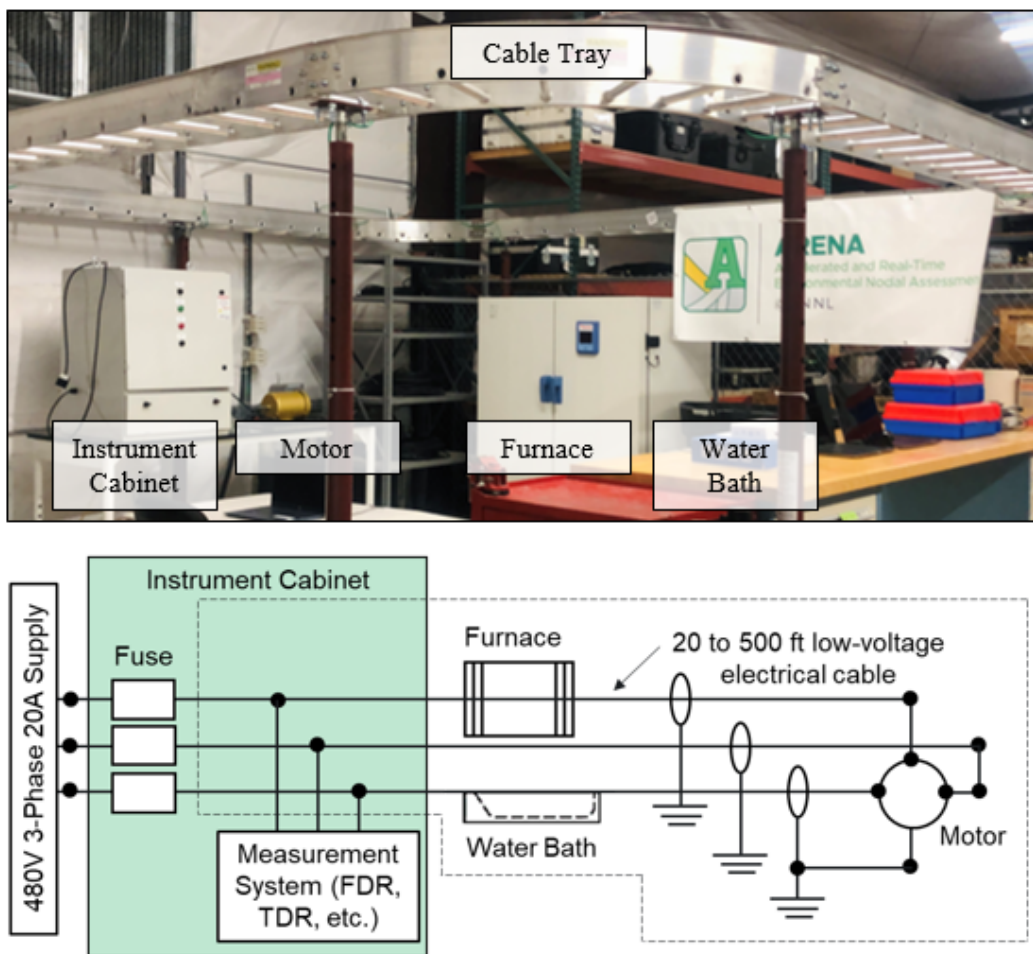Figure 1. The ARENA Test Bed (top) Digital Image and (bottom) Schematic (Glass et al. 2023)

3.   FDR AND SSTDR

Initial studies using the ARENA cable motor test bed included FDR and SSTDR tests to detect cable anomalies. FDR is beginning to be used in nuclear plants, particularly to locate areas of concern. The FDR instrument, typically a vector network analyzer or VNA, is connected to two cable conductors, one

considered the primary conductor under test and the other considered the system ground, as shown in Figure 2, or to a parallel conductor within the cable bundle (Glass et al. 2017). The instrument directs a swept-frequency chirp along the conductor and then listens for any reflection caused by an impedance change along the cable length. By listening and detecting the reflections in the frequency domain, then transforming to the time domain with an inverse Fourier transform (IFT), significant noise immunity and sensitivity to subtle impedance changes can be achieved (Glass et al. 2017). The bandwidth for the FDR is software adjustable up to 1.3 GHz, but experience shows the best responses from 100 MHz to 500 MHz. Higher bandwidth FDRs produce sharper peaks capable of spatially resolving more closely spaced impedance changes, but the higher frequencies do not propagate as far along the cable length. FDR instruments are restricted to relatively low voltages and cannot tolerate testing on energized cable systems.

The LIVEWIRE commercial SSTDR produces a similar plot to the FDR, but all processing is in the time domain. A pseudo-random noise code (PN code) is input onto the cable conductor, and the instrument listens for any reflected response from cable anomalies. The SSTDR processes the signal as an autocorrelation, comparing the input PN code to any reflected signal detected. The autocorrelation algorithm produces a robust noise-tolerant signal response; however, indications from the 2022 study (Glass et al. 2022) indicated that some higher bandwidth SSTDR responses could help with analysis and the probability of detection.
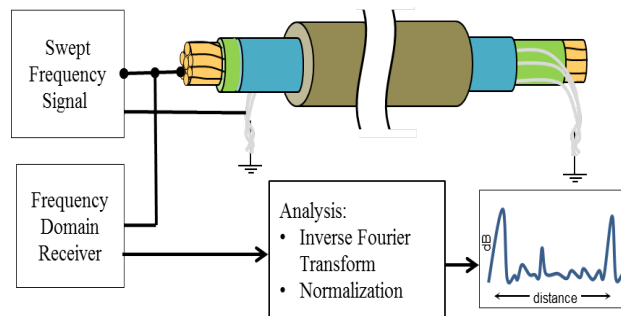
Figure 2. FDR cable test introduces a swept frequency chirp onto a conductor then listens for any reflection from any impedance change along the cable length. Listening is captured in the frequency domain then transformed to time domain using an inverse Fourier transform (Glass et al. 2017).

4. DEVELOPMENT OF LABORATORY BASED SSTDR INSTRUMENT

PNNL has developed a flexible laboratory-based SSTDR system to investigate SSTDR sensitivities to different cable impedance discontinuities as a function of bandwidth. The PNNL SSTDR system consists of the components in the function block diagram shown below in Figure 3 and the physical configuration in Figure 4. In the block diagram, the SSTDR signal generation is created by the AWG, which creates a voltage output at a specific sampling rate this instrument provides the ability to create a time series waveform representative of an SSTDR signal. The AWG provides two outputs for a single waveform as a differential pair: the (+), or 0-degree waveform, is used as the signal injected down the cable line, and the (-), or 180-degree waveform, is used as a reference to correlate against the (+) signal as it's received. This method provides a phase and time synchronous copy of the SSTDR waveform and is ideal for cross correlations.
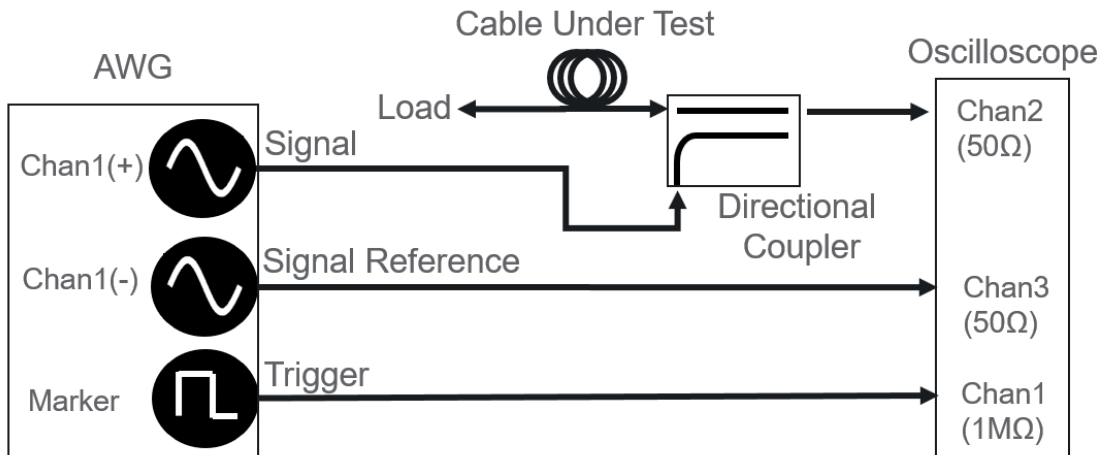
Figure 3. Block Diagram of PNNL SSTDR System
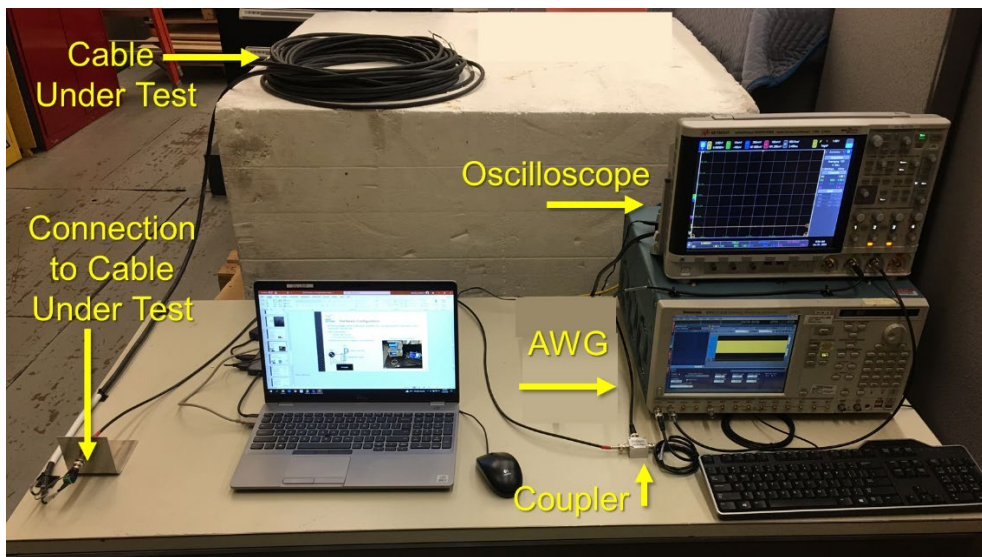
Figure 4. PNNL SSTDR Initial Test Configuration

SSTDR waveforms are created by applying a binary phase shift, 0-deg phase change or 180-deg phase change, to a carrier signal that is at a fixed frequency, termed binary phase shift keying (BPSK). The BPSK decision on 0-deg or 180-deg is randomized using a gaussian distribution (Proakis and Manolaki 1992). The example below shows the application of a BPSK modulation to a carrier frequency (Fc), first in time domain (Figure 5), and then how the time series manifests in frequency domain (Figure 6).
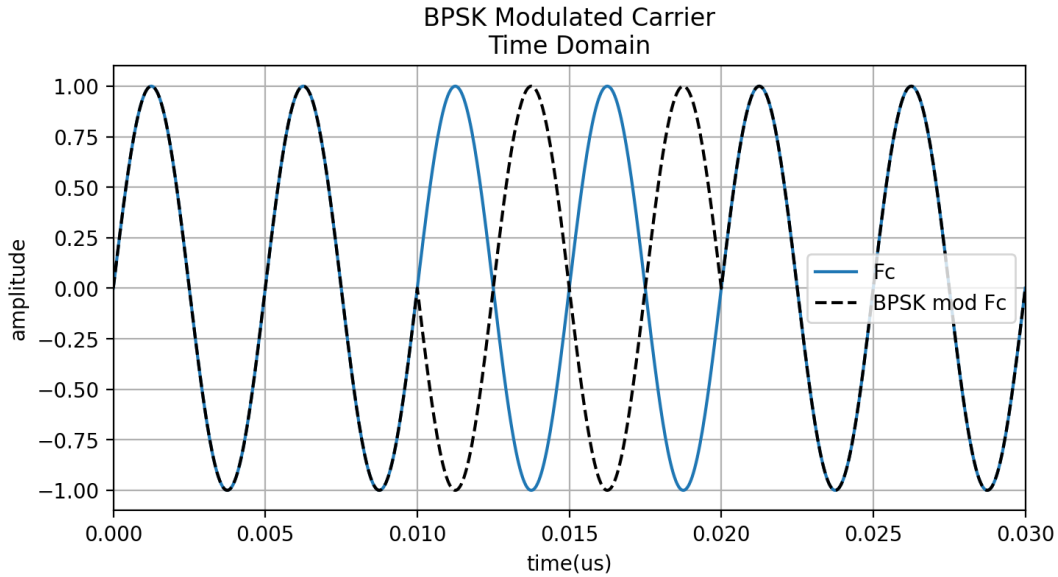
Figure 5. Time Domain Plot of 200 MHz Carrier Signal and the Carrier with a 50 MHz
BPSK Modulation Applied
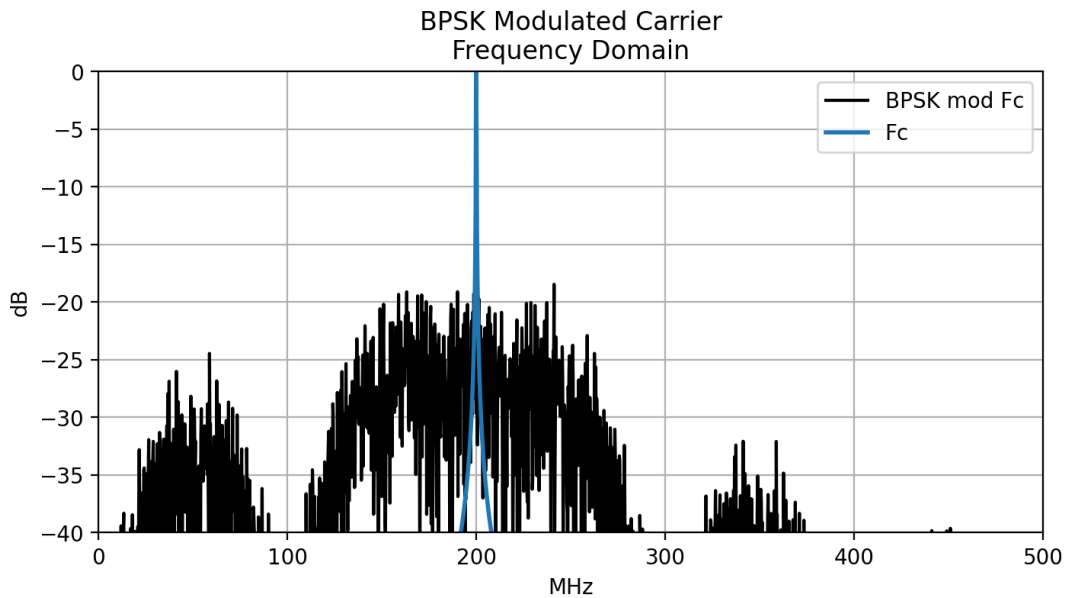
Figure 6. Frequency Domain Plot of 200 MHz Carrier Signal and the Carrier with a
50 MHz BPSK Modulate Applied

Note that the Fc is much higher in amplitude before the BPSK modulation is applied. This is due to all the energy within the time series concentrating in the single frequency Fc. However, for the Fc-modulated sequence, the phase flipping manifests by spreading out the frequency response over many frequency bins, hence the spread spectrum. In general, the 3 dB bandwidth of the spread spectrum response is double the BPSK modulation frequency since both positive and negative frequencies are used. If no Fc is used, the 3 dB bandwidth is limited to just the BPSK modulation frequency. An advantage of the spread spectrum signal relative to a single frequency tone, such as the unmodulated Fc, is that the spread spectrum signal appears as noise and will likely not interfere with other active systems.

Additionally, the spread spectrum sequence can be used as a matched filter for cross correlations by saving a copy of the waveform to compare against spread spectrum measurements, distinguishing them from other signals and even other spread spectrum waveforms in the same bandwidth.

PNNL used open-source software (Python) to create the time series waveforms that are uploaded to the high-speed AWG. Using an external directional coupler (Figure 5, Figure 6) the spread spectrum waveforms are 1) sent to an oscilloscope as a reference signal, 2) transmitted down the cable line under test, and 3) received back at the oscilloscope for a measurement waveform of the cable under test. The control, acquisition, and signal processing have all been automated in a streamlined series of Python scripts to allow for efficient and continuous measurements over time and to facilitate the data acquisition of the approximate 400 test conditions planned for this work.

5. SELECTION OF SOFTWARE AND HARDWARE PARAMETERS FOR INSPECTION DATA

For the PNNL SSTDR, software algorithm parameters influence key characteristics of the waveform to improve performance and maximize the probability of anomaly detection. When evaluating the performance of an SSTDR system, the metrics to review are signal-to-thermal noise, signal-to-clutter, and downrange resolution (Figure 7). Signal=to-thermal noise is defined as the ratio between target anomaly signal amplitude versus. the randomized thermal noise from resistive components in the system. Signal-to-clutter compares the target anomaly signal response to undesired signal components, such as harmonics, correlation artifacts, environmental noise, and multipath signal reflections. Downrange resolution is defined as the ability to resolve two peaks in distance by their 3 dB point, or half-power.
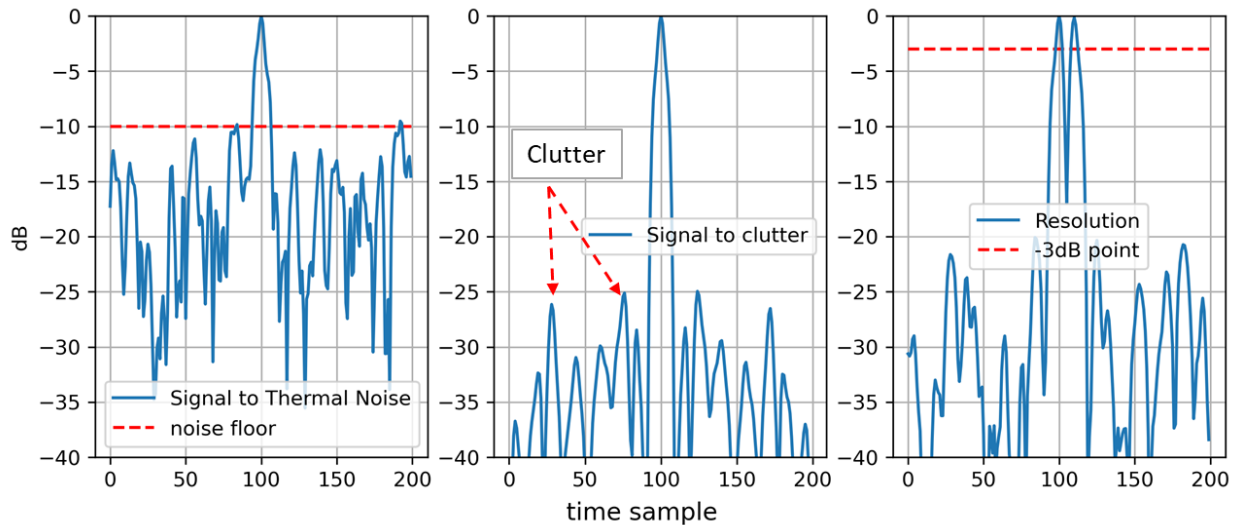
Figure 7. Examples of Signal-to-Thermal Noise Ratio (left), Signal-to-Clutter Noise Ratio (middle), and Resolution (right)

Observe that for the signal-to-thermal noise ratio, the target anomaly signal is centered at time sample 100, and there is randomized thermal noise appearing at -10 to -20 dB relative to the desired peak. For the signal-to-clutter case, the target anomaly peak is again at time sample 100; the randomized noise is not observable, but symmetrical sidebands at +/- 65-time samples are present due to the cross correlation. In the plot for resolution, there are two distinct peaks, and the dip between them is at -15 dB from the maximum. Resolution is defined as the -3 dB point.

It's important to note which aspects of an SSTDR system impact these parameters specifically and how to optimize the SSTDR waveform appropriately for a given use case. Table 1 below shows different parameters and, as they increase or decrease, how they impact the performance of the SSTDR system.

Table 1. Stoplight Chart of SSTDR Waveform Characteristics and Their Impact on SSTDR Performance

	Cable Length	Fc	Bandwidth	PN Code Length	Signal Power
Signal to Thermal Noise Ratio	degrades as length increases	degrades as Fc increases	degrades as bandwidth increases	improves as PN code increases	Improves as power increases
Signal to Clutter Ratio	No effect	With carrier frequencies close to 0 Hz, SSTDR ensues clutter	No effect	improves as PN code increases	No effect
Resolution	No effect	Degrades with Fc < ½ BPSK	Improves as bandwidth increases	No effect	No effect

From the stoplight table shown above, improvements can be made by tuning the bandwidth specifically for the cable scenario and increasing the signal power, but the key feature for improving signal quality is the PN code length. PN code length comes at a cost in system design, with ultra-long PN codes that require high memory depth and increase the amount of time it takes to collect a single scan. PNNL's SSTDR waveform parameters have been selected based on the instrumentation available for this study. As noted previously, signal quality could be further improved with instrumentation that would allow for longer PN codes. Hardware parameters for the SSTDR tests are shown in Table 2.

Table 2. Measurement Hardware Parameters for PNNL SSTDR Implementation

Bandwidth (MHz)	Carrier Freq (MHz)	PN Code Length (s)	Output Power(dBm)	AWG Clock Rate (SPS)	Scope Sample Rate (SPS)	Samples
50	25	1.00E-05	-16	2.00E+08	5.00E+09	2.85E+04
100	50	1.00E-05	-16	4.00E+08	5.00E+09	2.85E+05
200	100	1.00E-05	-16	8.00E+08	5.00E+09	2.85E+05
300	150	1.00E-05	-16	1.20E+09	5.00E+09	2.85E+05
400	200	1.00E-05	-16	1.60E+09	5.00E+09	2.85E+05
500	250	1.00E-05	-16	2.00E+09	5.00E+09	2.85E+05

The software parameters for processing the measured waveform from the cable under test involve:

1. cropping the time series to the start of the AWG trigger
2. applying a custom-modified Hamming window in the frequency domain of the received data,
3. performing a cross-correlation of the processed and received data with the measured reference,
4. computing the imaginary component of the crosscorrelated result using a Hilbert Transform, and
5. converting from time to distance using the propagation velocity of the cable.

Figure 8 through Figure 11 below show the progression through the five signal processing steps for a 200 MHz bandwidth SSTDR acquired data set on a 110 ft unshielded cable with an open circuit at 110 ft.
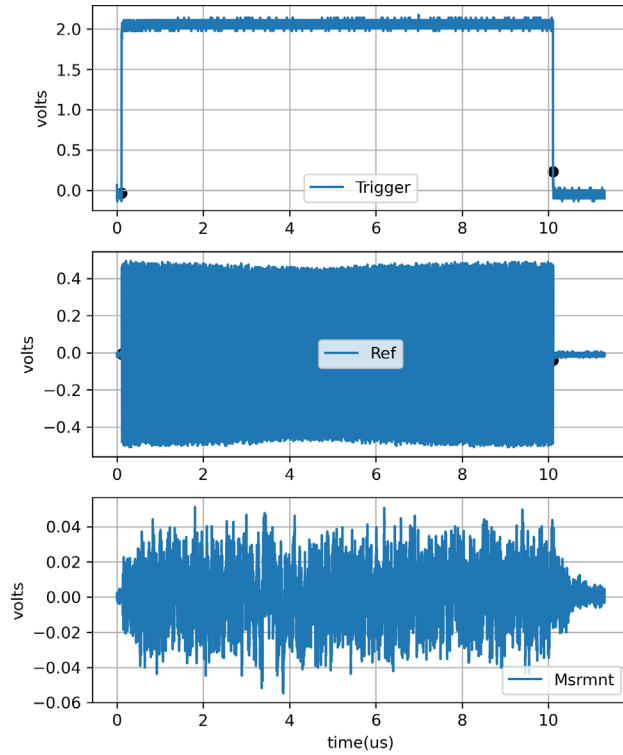
Figure 8. Time Series Data Collected for a 100 ft Cable and SSTDR Waveform
with 200 MHz Bandwidth
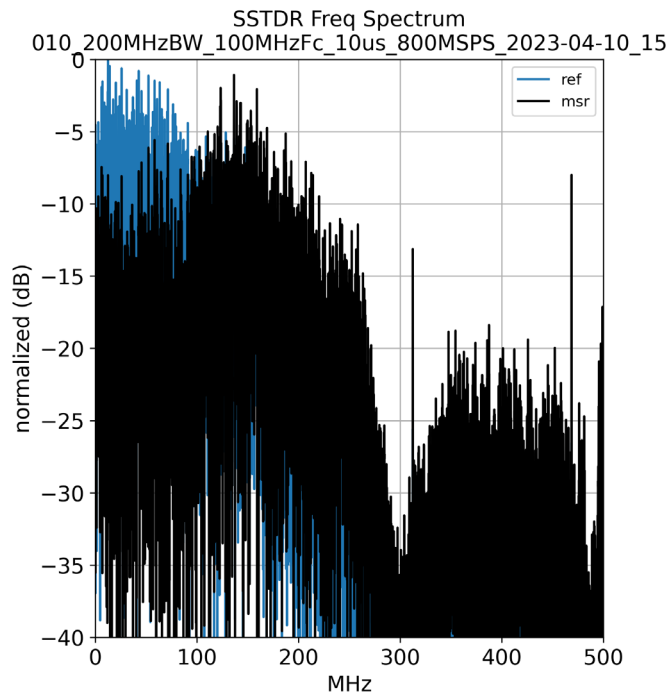
Figure 9. Frequency Domain Content of Reference and Measured Waveform
Post Application of Windowing Function
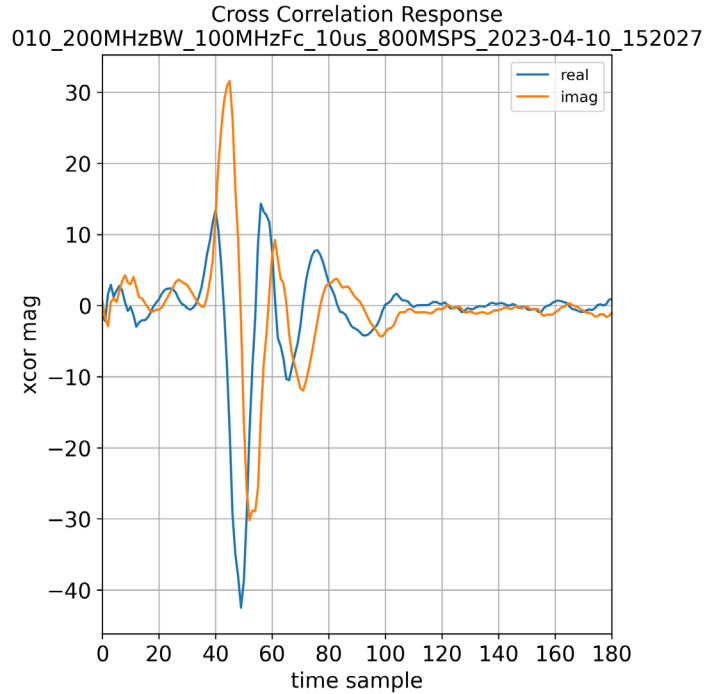
Figure 10. Resulting Real and Imaginary Component of Correlated Data from Hilbert Transform
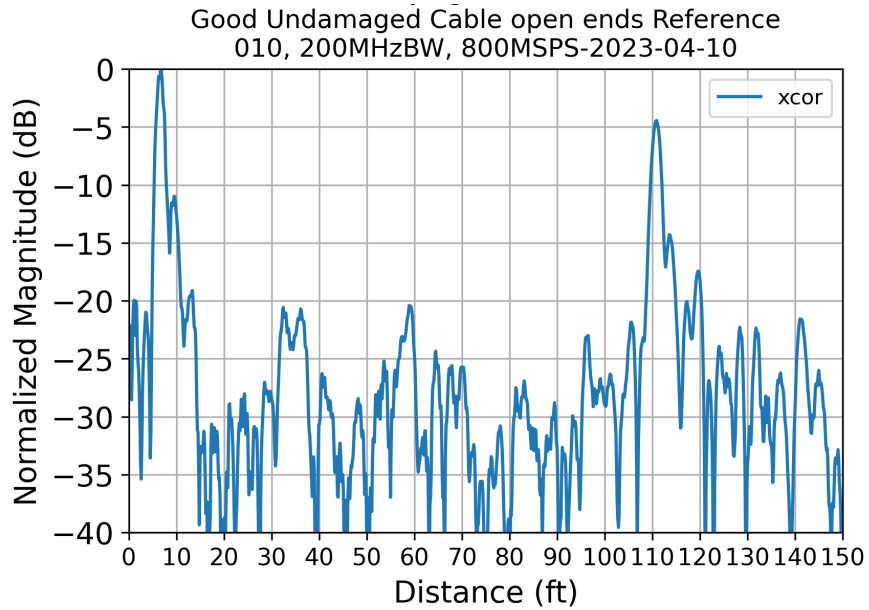
Figure 11. PNNL SSTDR Result Measuring 100 ft Cable with Open Circuit at Cable End (110 ft)

The results of the computation produce good correlation results that track with the expected performance versus SSTDR measurement parameters from Table 1 stoplight chart. It is important to note that there is a test-lead cable between the PNNL SSTDR system and the cable under test that is used to inject signals into the cable under test. This test-lead length shifts the range domain response of the cable under test by several feet; in this case, the test-lead length is why the peak lands at 110 ft rather than 100 ft.

5.1 SSTDR Data Collection and Processing Flow

Figure 12 below shows the flow of data collection and the signal processing applied to the data to compute the SSTDR results. The raw data acquisition is accomplished using an oscilloscope triggered by the arbitrary waveform generator that creates the spread-spectrum signals.

Raw Data Acquisition	→	Trigger & WFM Alignment	→	Decimation	→	Frequency Domain Windowing

Cross Correlation	→	Hilbert Transform and Zero Padding	→	Range Domain Conversion
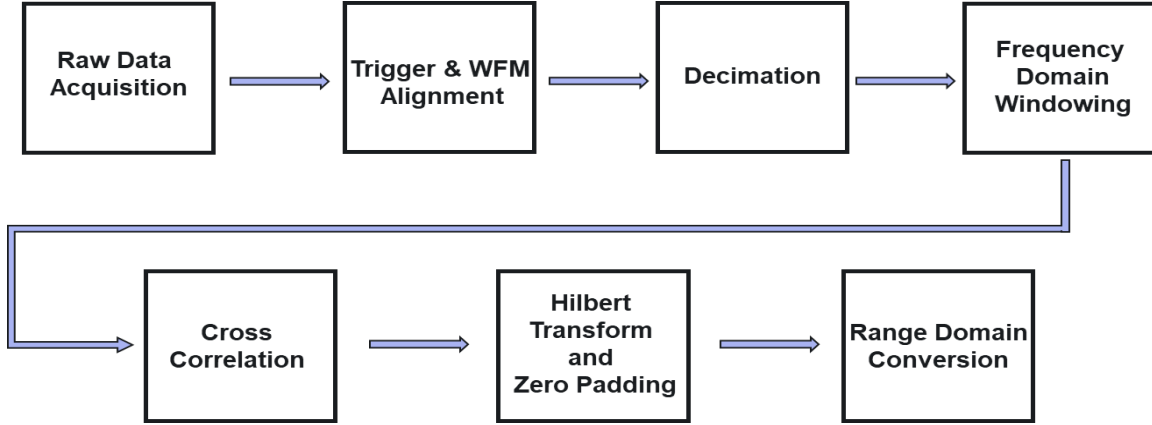
Figure 12. SSTDR Data Collection and Processing Flow Diagram

The trigger and waveform alignment occurs in software, which uses the start of the acquired trigger signal to determine the starting sample of both the reference and measurement waveforms. The end of the trigger waveform is used to find the end of the reference waveform. This is important because only using time samples where the reference waveform exists ensures that the reference signal used for cross-correlation against the measurement waveform does not have any extraneous information that will decorrelate. Additionally, this provides the time shift zero location translating to 0 ft in range.

Decimation applied to the time series information is used to optimize the frequency domain content to fit a customized Hamming window. If the time series information is over sampled, this will attenuate frequencies on the outer edges of the frequency span too much and produce negative results. Therefore, the highly sampled time series information is decimated to be sampled only at 4x the SSTDR bandwidth; this is double the Nyquist sampling theory requirement and provides a radio frequency spectrum with aliased frequencies of 4x the SSTDR bandwidth.

After the decimation is applied, a frequency domain window targets attenuation in the 0-5 MHz frequency range to remove undesired DC offset effects and low-frequency modulations. This then provides a clean, measured frequency spectrum to compare against the reference waveform. The cross correlation is defined as:

$$(f_{msrd} * f_{ref})(\tau) \triangleq \int \overline{f_{msrd}(t)} f_{ref}(t + \tau) dt$$

This cross correlation slides the f_{ref} function across the f_{msrd} function and computes the integral of their product at each position. As the two waveforms align, the result is positively correlated and produces a large amplitude. And as the waveforms are oppositely aligned in amplitude, this creates a large null from being negatively correlated. The cross-correlation result is then time shifted by the length of the reference waveform to properly align the time 0 index, producing results similar to the real data set previously shown in Figure 10. The results from the cross-correlation are real-only data sets; therefore, the cross-correlation result is translated into the frequency domain, the negative (or aliased) frequencies are multiplied by zero, and the waveform is padded with zeros. This accomplishes two things: 1) removing the negative frequencies creates a complex data set when the data are transferred back into the time domain, and 2) the zero padding improves the time domain sampling, creating smoother results. The

computation of the imaginary component of the complex waveform in this accomplishment results from implementing a standard Hilbert Transform. This aids in computing the complex magnitude of the SSTDR result which is normally reported in dB or logarithmic scaling.

Finally, the complex SSTDR resulting time axis is transferred into distance using the velocity of propagation or VoP (normally reported as a fraction of the speed of light). VoP varies as a function of cable design but once calibrated, reflections can be correlated to specific lengths along the cable.

6. TEST PLAN AND PRELIMINARY RESULTS

To evaluate the PNNL SSTDR instrument's performance for detection and locating anomalies, it is desirable to have a significant number of observations. A test matrix was prepared that included approximately 30 test conditions with the target of taking both FDR and SSTDR data using a range of software-controlled parameters (> 500 total tests). Detailed analysis of these tests will include manual review as well as neural network, artificial intelligence, and machine learning analysis, but that will be the subject of a subsequent report. Select preliminary results are shown below.

6.1 New/Good Unshielded Cable – No Anomalies Routed beside Thermally Aged Cable in Oven

Figure 13 shows a new undamaged cable routed similarly to the thermally aged cable. The cable is approximately 100-ft. long, with a 20-ft. lead between the cable start and the test instruments. Cable ends are clearly visible, with the noise floor more than 10 dB below the cable end responses.
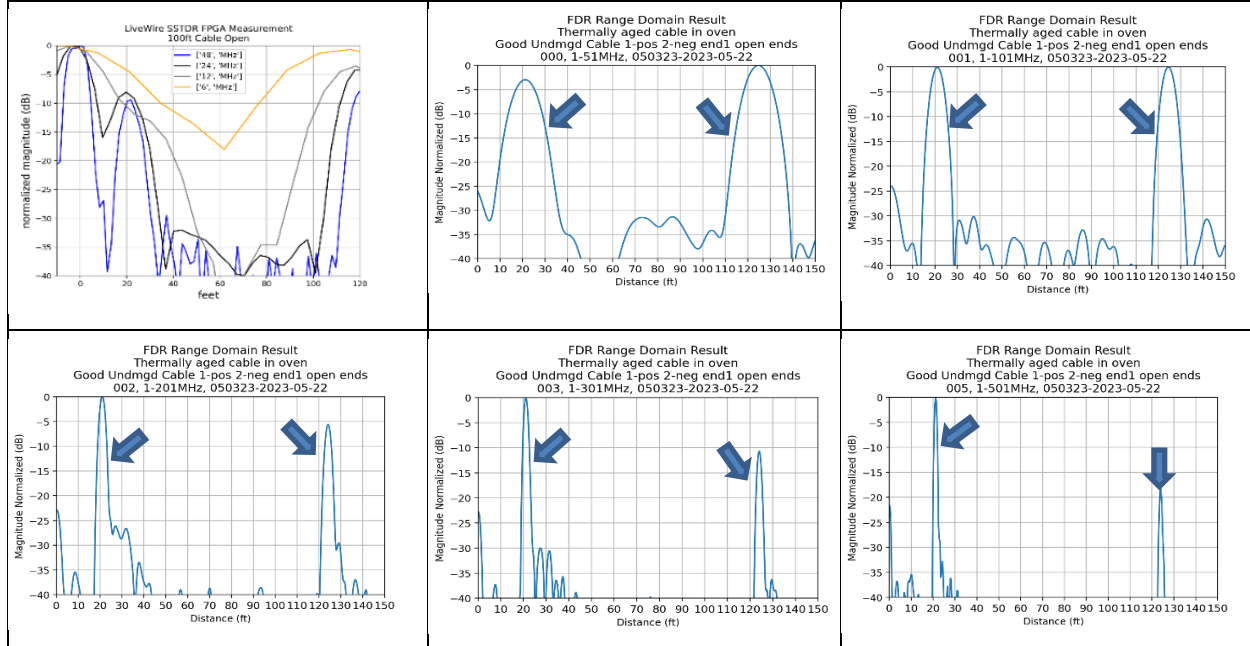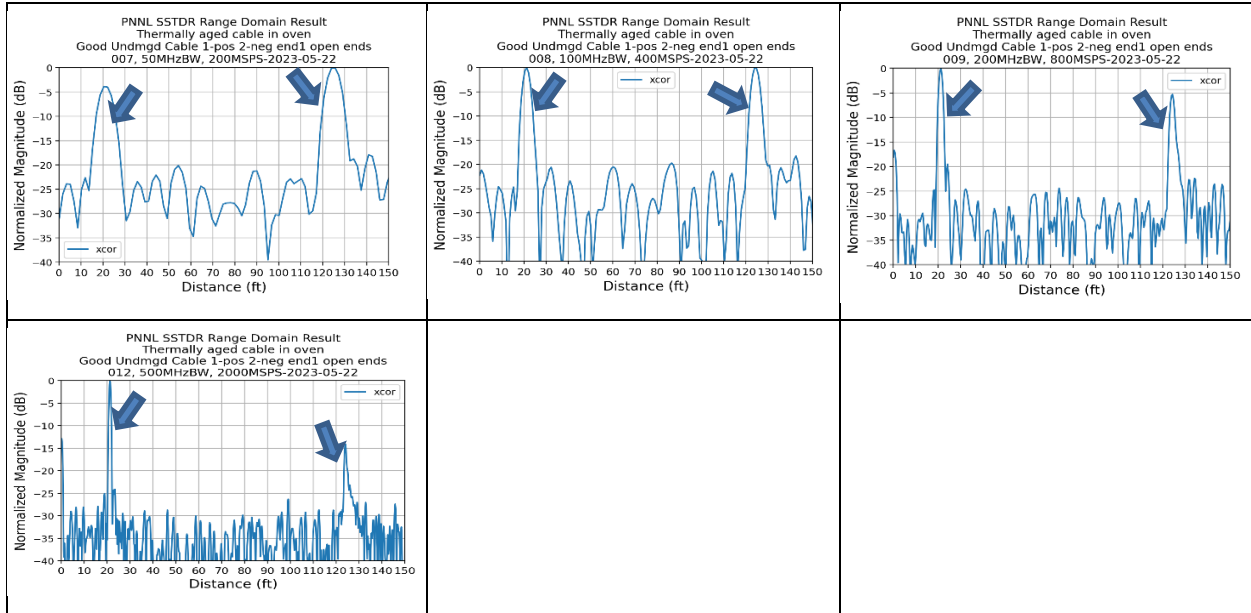
Figure 13. New/Good Unshielded Cable – No anomalies. The cable is routed beside thermally aged cable in cable trays and oven. LIVEWIRE, VNA, and PNNL SSTDR with multiple bandwidth data are shown. Cable-end peaks are indicated by arrows.

6.2   Thermally Aged Unshielded Cable

Figure 14 shows a 60-day thermally aged cable with data from LIVEWIRE, VNA, and PNNL SSTDR. The edges of the thermally damaged region are most clearly visible in the lower-bandwidth VNA plots but peaks near the noise floor are also indicated with some frequency bandwidths of the SSTDR plots.
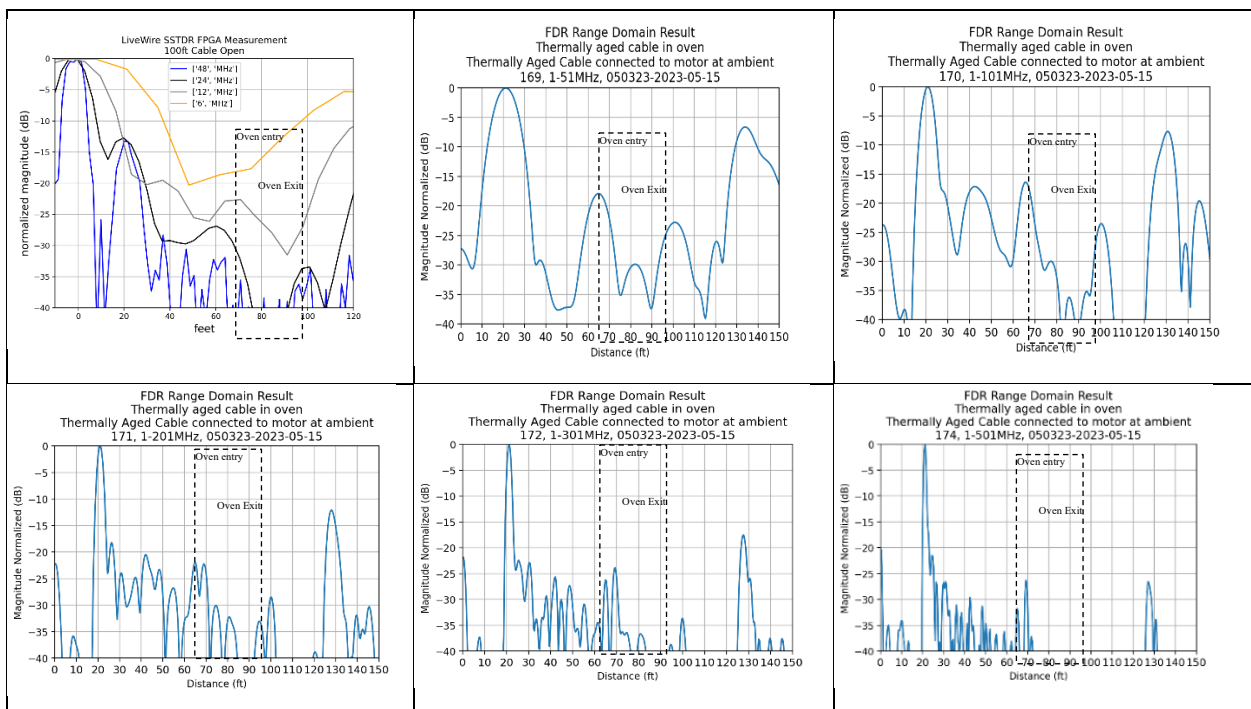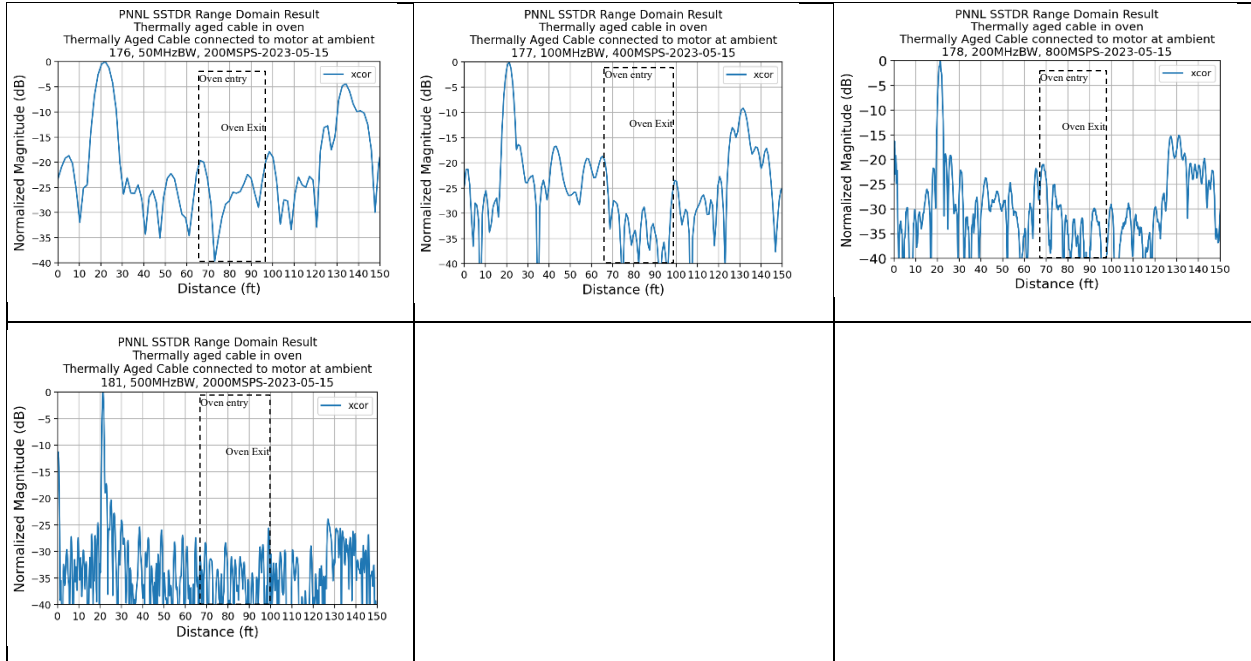
Figure 14. A 100-ft thermally aged cable (aged for 60 days) between 65 and 90-ft is shown with LIVEWIRE SSTDR, FDR, and PNNL SSTDR data taken at multiple bandwidths. Locations where oven entry and exit are only visible at lower bandwidth FDRs.

6.3   Phase to Phase Fault Cable

For phase-to-phase low-resistance faults, only the 100-ohm fault was visible. As indicated in Figure 15, the fault was most easily detected with the higher bandwidths of both FDRs and both LIVEWIRE's and PNNL's SSTDRs.
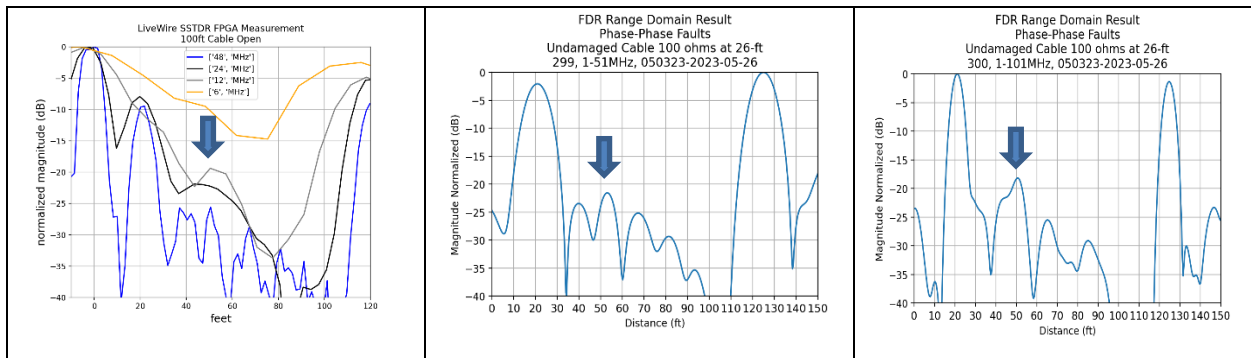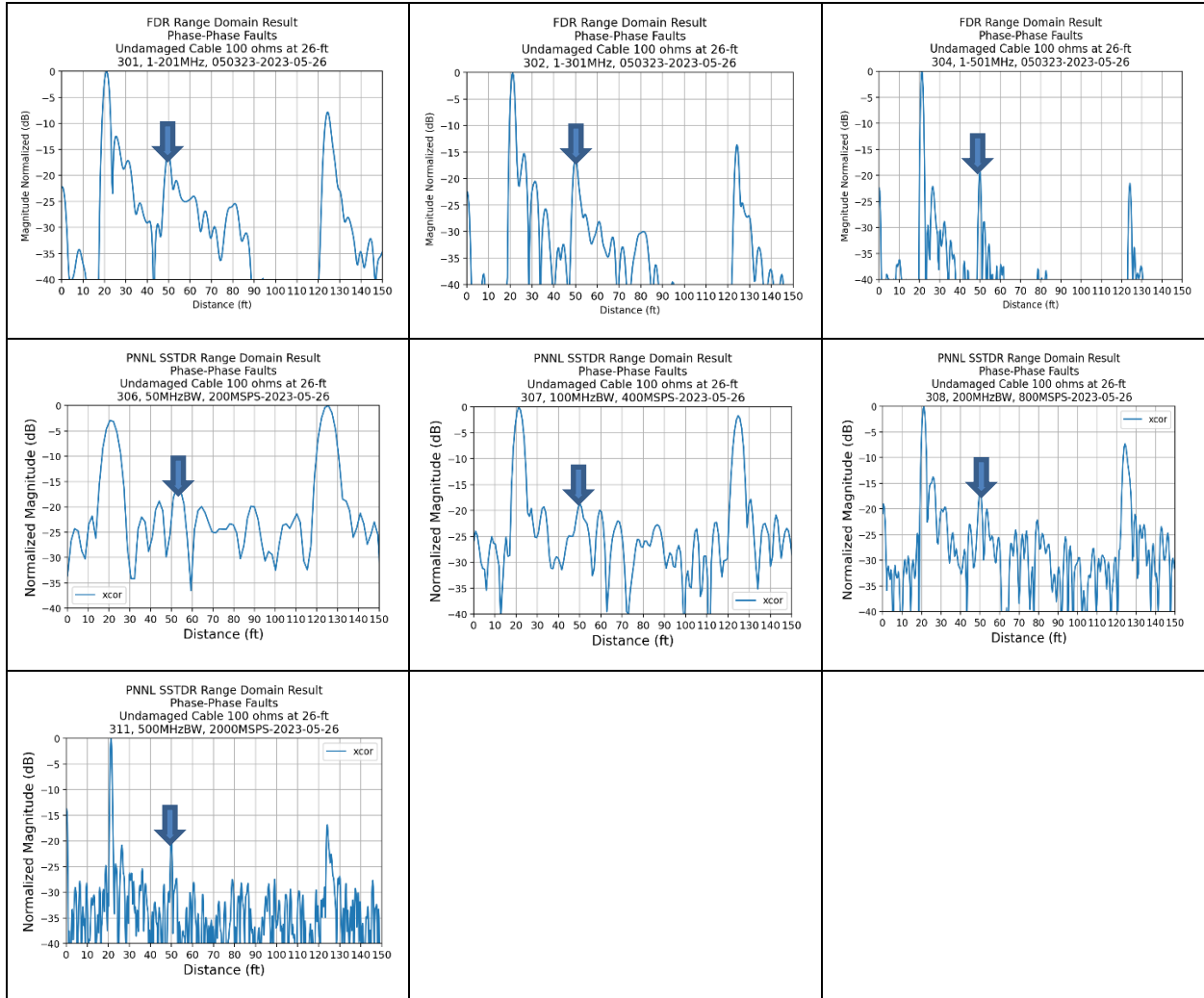
Figure 15. LIVEWIRE, FDR, and PNNL SSTDR data at various bandwidths for phase-phase 100-ohm fault at 26 ft. Note that there is a 20-ft. measurement cable between the cable start and the instrument, so the fault appears at about 50 ft.

7. OBSERVATIONS AND CONCLUSIONS

This project demonstrated that an SSTDR can be implemented with a laboratory ARB for signal generation, an oscilloscope for data capture, and some PYTHON signal processing code (see appendix). The frequency bandwidth of the commercial SSTDR system spans from 6 MHz to 48 MHz. The PNNL SSTDR extends the analysis bandwidth to 500 MHz, providing additional cable test information for analysts to consider. The PNNL SSTDR was tested and compared to FDR tests using a Copper Mountain TR1300 vector network analyzer and a LIVEWIRE commercial instrument. Observations and conclusions based on tests using the ARENA cable motor test bed with new good cables and cables with anomalies were as follows:

1. Having multiple frequency bandwidths extending above the 48 MHz of the commercial instrument provides additional data that can aid in interpreting the SSTDR test.

2. Generally, the FDR data were clearer than either the LIVEWIRE instrument or the PNNL SSTDR.

3. All instruments showed sharper peaks with higher bandwidths and broader peaks with lower bandwidths.

4. Baseline noise levels generally increase with higher bandwidths. We also noted reduced amplitude responses from the distal cable end as the cable length was extended. This could also be noted from the multiple reflections of the 100-ft. cables that were the primary focus of the comparative tests.

5. Thermal aging was difficult to detect with all instruments. The best indications were with the lower-bandwidth FDR plots.

6. Phase-to-phase faults with a 100-ohm resistor were easier to detect with higher frequency tests, both based on FDR responses and PNNL SSTDR responses.

8. REFERENCES

Blocker, E., S. Smith, L. Philpot, and J. Conley. 1996. *Aging Management Guideline for Commercial Nuclear Power Plants - Electrical Cable and Terminations.* Sandia National Labs SAND96-0344, Albuquerque, NM. http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=204243.

Fabiani, D., S. V. Suraci, and S. Bulzaga. 2018. "Aging Investigation of Low-Voltage Cable Insulation Used in Nuclear Power Plants." In *2018 IEEE Electrical Insulation Conference (EIC)*, 516–19. IEEE. https://doi.org/10.1109/EIC.2018.8481139.

Fifield, L. S., and R. Duckworth. 2015. "Progress in the Investigation Fo Nuclear Power Plant Cable Aging." In *17th International Conference on Environmental Degradation of Materials in Nuclear Power Systems -- Water Reactors* August 9-13, 2015, Ottowa, Ontario, Canada.

Furse, C., Y. C. Chung, R. Dangol, M. Nielsen, G. Mabey, and R. Woodward. 2003. "Frequency-Domain Reflectometry for on-Board Testing of Aging Aircraft Wiring." *IEEE Transactions on Electromagnetic Compatibility* 45 (2): 306–15. https://doi.org/10.1109/TEMC.2003.811305.

Gazdzinski, R.F., W.M. Denny, G.J. Toman, and R.T. Butwin. 1996. *Aging Managment Guideline for Commercial Nuclear Power Plants - Electrical Cable and Terminations*. Sandia National Labs SAND96-0344. Albuquerque, NM. https://www.nrc.gov/docs/ML0311/ML031140264.pdf.

Glass, S. W., A. M. Jones, L. S. Fifield, and T. S. Hartman. 2017. "Frequency Domain Reflectometry NDE for Aging Cables in Nuclear Power Plants." In *AIP Conference Proceedings* 1806, 080015. https://doi.org/10.1063/1.4974640.

Glass, S.W., Leonard S. Fifield, D. Gerges, T. Jonathan, A.M. Jones, and T.S. Hartman. 2015. *State of the Art Assessment of NDE Techniques for Aging Cable Management in Nuclear Power Plants*. Pacific Northwest National Laboratory PNNL-24649. Richland, WA. https://doi.org/10.2172/1242348.

Glass, S.W., A.E. Holmes, Leonard S. Fifield, and K.K. Anderson. 2020. *Assessment of EPRI's Tan Delta Approach to Manage Cables in Submerged Environments: Statistical Review of EPRI Data*. Pacific Northwest National Laboratory PNNL-28542-1. Richland, WA. https://doi.org/10.2172/1713064.

Glass, S.W., A.M. Jones, Leonard S. Fifield, and T.S. Hartman. 2016. *Bulk and Distributed Electrical Cable Non-Destructive Examination Methods for Nuclear Power Plant Cable Aging Management Programs*. Pacific Northwest National Laboratory PNNL-25634. Richland, WA. https://doi.org/10.2172/1328060

Glass, S.W., A.M. Jones, Leonard S. Fifield, T.S. Hartman, and Nicola Bowler. 2017. *Physics-Based Modeling of Cable Insulation Conditions for Frequency Domain Reflectometry (FDR)*. Pacific Northwest National Laboratory PNNL-26493. Richland, WA. http://www.pnnl.gov/main/publications/external/technical_reports/PNNL-26493.pdf

Glass, S.W., Mychal P. Spencer, M. Prowant, A. Sriraman, J. Son, and Leonard S. Fifield. 2023. "The ARENA Test Bed - A Versatile Resource for I&C Development and Validation." In *13th Nuclear Plant Instrumentation, Control & Human-Machine Interface Technologies*. Knoxville, TN, USA.

Glass, S.W., Mychal P. Spencer, A. Sriraman, Leonard S. Fifield, and M. Prowant. 2021. *Nondestructive Evaluation (NDE) of Cable Moisture Exposure Using Frequency Domain Reflectometry (FDR)*. Pacific Northwest National Laboratory PNNL-31934. Richland, WA. https://doi.org/10.2172/1820614

Glass, S.W., A. Sriraman, M. Prowant, Mychal P. Spencer, Leonard S. Fifield, and S. Kingston. 2022. *Nondestructive Evaluation (NDE) of Cable Anomalies Using Frequency Domain Reflectometry (FDR) and Spread Spectrum Time Domain Reflectometry (SSTDR)*. Pacific Northwest National Laboratory PNNL-33334.

IAEA. 2012. *Assessing and Managing Cable Ageing in Nuclear Power Plants.* International Atomic Energy Agency Nuclear Energy Series No. NP-T-3.6. Vienna, Austria. https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1554_web.pdf

Joskow, P. L. 2006. *The Future of Nuclear Power in the United States: Economic and Regulatory Challenges.* Massachusetts Institute of Technology MIT-CEEPR 06-019WP. Cambridge, MA https://ceepr.mit.edu/wp-content/uploads/2023/02/2006-019.pdf

Proakis, J.G., and D.G. Manolaki. 1992. *Digital Signal Processing, Principles, Algorithms, and Applications*. 4th ed: Prentice Hall.

APPENDIX: SPREAD SPECTRUM TIME DOMAIN REFLECTOMETRY SOFTWARE

The following three primary scripts perform the functions of signal generation, acquisition, and computation. Note that there are supporting Python modules and specific instrumentation control scripts that are not shown. The purpose of these primary scripts is to educate the reader on how to generate their own spread spectrum time domain reflectometry (SSTDR) waveforms, details on acquisition that matter as far as sampling rates, and basic SSTDR computation.

A1: SSTDR Signal Generation

```
"""
Binary Phase Shift Keying PN Code Generation

v0
Jonathan Tedeschi 10/26/22

"""
import numpy as np
import matplotlib.pyplot as plt
import FFT


path='\\\\pnl\\projects\\CABLE_NDE_PROJ\\People\\Tedeschi\\FY23\\sstdr_implimentation\\test_matrix_SSTDR_wfms\\'

# WFM parameters
duration=[1e-6,10e-6,100e-6] # list of PN code lengths
f_bpsk=200e6# BPSK modulation frequency (Hz)
f_carrier=f_bpsk


# ARB Parameters
Tek_AWG_sample_rate=8*f_bpsk#

for max_wfm_duration in duration:
    #%%
    half_cycles=int(max_wfm_duration*f_bpsk*2) # digital clock 1 or a 0 at the clock rate
    states=np.random.randint(0,2,half_cycles)*2-1 # randomized BPSK PN code
    state_holding_pts=(Tek_AWG_sample_rate/f_bpsk)/2 # determines how many pts a 1 or 0 holds for
    t=np.linspace(0,max_wfm_duration,int(len(states)*state_holding_pts))
    bpsk_wfm=np.repeat(states, state_holding_pts)*np.sin(2*np.pi*f_carrier*t) # repeats the BPSK PN code appropriately for the modulation frequency f_mod

    #%%
    # =========================================================================
    # Plotting Results
    # =========================================================================
    plt.figure(figsize=(6,7))
    plt.subplot(211)
    plt.plot(t*1e6,bpsk_wfm)
    plt.grid()
    plt.xlabel('us')
    plt.ylabel('Volts')
    plt.title('Modulated Carrier\n SSTDR BW: '+str(int(f_bpsk/1e6))+'MHz, Fc: '+str(int(f_carrier/1e6))+'MHz')

    BPSK_WFM=FFT.fft0(bpsk_wfm, t)

    plt.subplot(212)
    plt.plot(BPSK_WFM[0]/1e6,FFT.dB(BPSK_WFM[1],True))
    plt.grid()
    plt.xlabel('MHz')
    plt.ylabel('Volts')
    # plt.title('BPSK Sequence')
    plt.xlim(0,1000)
    plt.ylim(-40,0)
    plt.savefig(path+str(int(2*f_bpsk/1e6))+'MHz_BPSK_'+str(int(f_carrier/1e6))+'MHz_Fc_'+str(int(max_wfm_duration*1e9))+'ns.png',dpi=200)

#%%

# =========================================================================
# Saving waveforms and markers for triggering
```

```
# ==============================================================================
  marker=np.ones(len(bpsk_wfm))
  marker[0]=0
  tek_wfms=np.flipud(np.rot90(np.array([bpsk_wfm,marker,marker]),k=1))


np.savetxt(path+str(int(f_bpsk/1e6))+'MHz_BPSK_'+str(int(f_carrier/1e6))+'MHz_Fc_'+str(int(max_wfm_duration*1e9))+'ns_'+str(int(Tek_AWG_sample_rate/1e6))
+'MSaps.txt',tek_wfms,delimiter=',')
```

A2: SSTDR Acquisition

```
'''

PNNL SSTDR DATA Collection and Computation

Author: Jonathan Tedeschi
5/25/23

'''
import numpy as np
import matplotlib.pyplot as plt
import Keysight_DSOX_4104A_measurement_automation_r3 as scope
import FFT
import msrd_data_xcor_function_r4 as xcor
import os
import time
import Tekawg_wrapper as TekAWG
#%%
def SSTDR_capture_and_compute(SSTDR_BW,
  SSTDR_fc,
  SSTDR_wfm_len,
  AWG_clock,
  cable_len,
  path,
  fname):

  # ==========================================================================
  # Oscilloscope Presets
  # ==========================================================================
  # Expected Oscillscope Channel Configuration
  trig_chan=1 # marker from AWG
  data_chan=2 # measurement of cable
  ref_chan=3 # reference channel

  # Oscilloscope Acquisition Buffers on either end of WFMs
  start_buffer= 100e-9 # seconds
  end_time_buffer= 1e-6 # seconds

  scope_acq_time=start_buffer+SSTDR_wfm_len + (cable_len/.7)*(1e-9) + end_time_buffer

  # rounding scope acquisition time up to the nearest 100ns
  scope_acq_time=int(np.ceil(scope_acq_time*1e7))/1e7

  # ts=1/(SSTDR_fc*4) # unused, saving maximum # of pts in time span

  num_pts=0#int(scope_acq_time/ts)


  # ==========================================================================
  # Data Acquisition
  # ==========================================================================

  # setting the oscilloscope parameters
  scope.set_scale_and_reference(15e-3,0,scope_acq_time,scope_acq_time/2-start_buffer,2)

  time.sleep(.5)

  # acquring data
  scope.save_all_chans_3(num_pts,['Trig','wfm','ref','NA'],path,fname,20,0)

  #%%
  data=np.loadtxt(path+fname+'.csv',skiprows=1,delimiter=',')

  # ==========================================================================
  # Calling Computation Script
  # ==========================================================================
  scope_channels=[trig_chan,ref_chan,data_chan]
  # xcor.compute_xcor(data,fname,path,scope_channels,True)
  xcor.compute_xcor(data,fname,path,scope_channels,False,SSTDR_BW*(3/2),False)
#%%
```

```
if __name__ == '__main__':

    SSTDR_BW=200e6
    SSTDR_fc=100e6
    SSTDR_wfm_len=10e-6
    AWG_clock=800e6
    cable_len=100#feet
    path='\\\\pnl\\projects\\CABLE_NDE_PROJ\\People\\Tedeschi\\FY23\\sstdr_implimentation\\data\\stability_testing\\'
    for f in range(100):


        SSTDR_capture_and_compute(SSTDR_BW,SSTDR_fc,SSTDR_wfm_len,AWG_clock,cable_len,path,"{:03d}k".format(f))
```

A3: SSTDR Signal Processing and Documentation

```
# =============================================================================
# Author: Jonathan Tedeschi
# rev 3 incorperates optional frequency domain windowing 3/27/23
# =============================================================================
import numpy as np
import matplotlib.pyplot as plt
import FFT
from scipy import signal
import glob
import hilbert_transform
import scipy as sc
import Q_and_BW_functions_r3 as BW
import digital_filtering_r0 as filt

#%% Functions

# =============================================================================
#
# =============================================================================
def ref_start_stop_index(array,trigger_channel):

    for start_index in range(len(array[:,trigger_channel])):
    if array[start_index,trigger_channel]>.5:
    break

    if start_index<3:
    start_index=3

    for stop_index in range(len(array[:,trigger_channel])):
    if array[stop_index+start_index,trigger_channel]<.5:
    stop_index=stop_index+start_index
    break


    return(start_index-3,stop_index+1)



# =============================================================================
#
# =============================================================================
def compute_xcor(oscope_data,fname,save_path,chan_index_list,peak_BW,freq_domain_window_peak=0,test_plots=False):
    '''
    compares the digitized ref vs. digitized measured data based off the trigger wfm

    oscope_data: 5 column data set from a single measurement

    fname: string to be used in plots


    chan_index_list: list of positions for which is the trigger, ref and msrmt channel
    [trig_chan,ref_chan,msrmt_chan]


    peak_BW: False, or the dB relative to the peak to calculate the downrange performance


    freq_domain_window_peak: recommended to be at 1.5 x BPSK freq

    '''

    trig_chan=chan_index_list[0]
```

```python
ref_chan=chan_index_list[1]
data_chan=chan_index_list[2]

# determining where the reference starts and stops in the time series based off of the AWG trigger
ref_indices=ref_start_stop_index(oscope_data,trig_chan)

# computing the cross correlation

input1=oscope_data[ref_indices[0]:,data_chan]
input2=oscope_data[ref_indices[0]:ref_indices[1],ref_chan]

# rev 3 update including frequency domain windowing on received xcor data
if freq_domain_window_peak!=0:
print('Applying Frequency Domain Window')
INPUT1=FFT.fft0(input1, oscope_data[ref_indices[0]:,0])

window_peak_index=int(np.argmin(np.abs(INPUT1[0]-freq_domain_window_peak)))

window_shape=np.hamming(window_peak_index*2)

window=np.ones(len(INPUT1[1]))

window[0:window_peak_index]=window_shape[0:window_peak_index]

window[-1*window_peak_index:]=window_shape[window_peak_index:]

WFM_windowed=INPUT1[1]*window

bpsk_wfm_windowed=FFT.ifft(INPUT1[0],WFM_windowed)[1][0:len(input1)]

input1=bpsk_wfm_windowed

xcor=signal.correlate(input1,input2,mode='full')# computing xcor

# determining the shift due to length of reference and measurement wfm
shifted_xcor=np.roll(xcor,len(oscope_data[ref_indices[0]:,data_chan]))

# calculating imaginary component using hilbert transform
xcor_cmplx=hilbert_transform.hilbert(shifted_xcor)

# xcor_cmplx=np.abs(xcor_cmplx_2x)*np.exp(1j*np.unwrap(np.angle(xcor_cmplx_2x))/2)


# standard cable velocity factor
cable_velocity_factor=0.69

# computing time based on sample rate from oscilloscope data
ts=oscope_data[1,0]-oscope_data[0,0]
t=np.linspace(0,(len(xcor)-1)*ts,len(xcor))/2# /2 due to round trip

# computing distance
# r=t*(3e8*cable_velocity_factor) # meters per second
r=t*(9.836e8*cable_velocity_factor) # feet per second


if bool(peak_BW):
down_range_resolution=BW.calc_BW(r,20*np.log10(np.abs(xcor_cmplx)),3,'peak',True)


np.savetxt(save_path+fname+'_xcor_result.csv',np.flipud(np.rot90(np.array([t,r,xcor_cmplx.real,xcor_cmplx.imag,20*np.log10(np.abs(xcor_cmplx))]))),delimiter=',',
header='time(s),range(f),xcor (real), xcor (imag), xcor (dB)')

plt.figure()
plt.plot(r,20*np.log10(np.abs(xcor_cmplx))-max(20*np.log10(np.abs(xcor_cmplx))),label='xcor')#-np.max(10*np.log10(s)))
plt.grid()
plt.xticks(np.linspace(0,150,16),fontsize=11)
plt.xlim(0,150)
plt.yticks(fontsize=14)
plt.xlabel('Distance (ft)',fontsize=14)
plt.ylabel('Normalized Magnitude (dB)',fontsize=14)
plt.legend(loc='best')
if bool(peak_BW):
plt.title('Measured Data Complex Magnitude of XCOR\nOne Way Distance, 3dB Downrange Resolution: '+str(int(down_range_resolution*100))+'cm\n'+fname)
else:
# plt.title('Measured Data Complex Magnitude of XCOR\nOne Way Distance\n'+fname)
if '_' in fname:
plt.title('PNNL SSTDR Range Domain Result\n'+save_path.split('\\')[-4]+'\n'+save_path.split('\\')[-3]+'\n'+fname.split('_')[0]+', '+fname.split('_')[1]+',
'+fname.split('_')[4]+'-'+fname.split('_')[5])
else:
```

```python
plt.title('PNNL SSTDR Range Domain Result\n'+fname)

plt.ylim(-40,0)
plt.savefig(save_path+fname+'_xcor_range_domain_normalized.png',bbox_inches='tight',dpi=200)


'''
supporting test plots
'''
if bool(test_plots):
    # ===============================================================================
    # Time Series Plots
    # ===============================================================================
    plt.figure(figsize=(6,8))
    plt.subplot(311)
    plt.title('Time Series Data\n'+fname+' Case')
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,trig_chan],label='Trigger')#-np.max(10*np.log10(s)))
    plt.scatter(oscope_data[ref_indices,0]*1e6,oscope_data[ref_indices,1],color='k')
    plt.grid()
    # plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')
    plt.subplot(312)
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,ref_chan],label='Ref')#-np.max(10*np.log10(s)))
    plt.scatter(oscope_data[ref_indices,0]*1e6,oscope_data[ref_indices,ref_chan],color='k')
    plt.grid()
    # plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')
    plt.subplot(313)
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,data_chan],label='Msrmnt')#-np.max(10*np.log10(s)))
    plt.grid()
    plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')


    xlimit=.2
    plt.figure(figsize=(6,8))
    plt.subplot(311)
    plt.title('Time Series Data\n'+fname+' Case')
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,trig_chan],label='Trigger')#-np.max(10*np.log10(s)))
    plt.scatter(oscope_data[ref_indices,0]*1e6,oscope_data[ref_indices,trig_chan],color='k')
    plt.grid()
    plt.xlim(0,xlimit)
    # plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')
    plt.subplot(312)
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,ref_chan],label='Ref')#-np.max(10*np.log10(s)))
    plt.scatter(oscope_data[ref_indices,0]*1e6,oscope_data[ref_indices,ref_chan],color='k')
    plt.xlim(0,xlimit)
    plt.grid()
    # plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')
    plt.subplot(313)
    plt.plot(oscope_data[:,0]*1e6,oscope_data[:,data_chan],label='Msrmnt')#-np.max(10*np.log10(s)))
    plt.xlim(0,xlimit)
    plt.grid()
    plt.xlabel('time(us)')
    plt.ylabel('volts')
    plt.legend(loc='best')


    # ===============================================================================
    # Frequency Domain Plots of Measured Data
    # ===============================================================================
    REF_WFM=FFT.fft0(oscope_data[:,ref_chan],oscope_data[:,0],True)
    MSR_WFM=FFT.fft0(input1,oscope_data[ref_indices[0]:,0],True)

    plt.figure(figsize=(6,7))
    plt.plot(REF_WFM[0]/1e6,20*np.log10(np.abs(REF_WFM[1]))-max(20*np.log10(np.abs(REF_WFM[1]))),label='ref')
    plt.plot(MSR_WFM[0]/1e6,20*np.log10(np.abs(MSR_WFM[1]))-max(20*np.log10(np.abs(MSR_WFM[1]))),color='k',label='msr')
    plt.grid()
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.ylim(-40,0)
    plt.xlim(0,500)
    plt.xlabel('MHz',fontsize=14)
```

```python
    plt.ylabel('normalized (dB)',fontsize=14)
    plt.legend(loc='best')
    plt.title('SSTDR Freq Spectrum\n'+fname[:-4],fontsize=14)


    # ========================================================================
    # XCOR Plots
    # ========================================================================
    plt.figure(figsize=(6,7))
    plt.plot(xcor_cmplx.real,label='real')
    plt.plot(xcor_cmplx.imag,label='imag')
    plt.grid()
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.xlabel('time sample',fontsize=14)
    plt.ylabel('xcor mag',fontsize=14)
    plt.legend(loc='best')
    plt.title('Cross Correlation Response\n'+fname,fontsize=14)

    plt.figure(figsize=(6,7))
    plt.plot(xcor_cmplx.real,label='real')
    plt.plot(xcor_cmplx.imag,label='imag')
    plt.grid()
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.xlabel('time sample',fontsize=14)
    plt.ylabel('xcor mag',fontsize=14)
    plt.legend(loc='best')
    plt.xlim(0,180)
    plt.title('Cross Correlation Response\n'+fname,fontsize=14)

    return(xcor_cmplx,t,r)


def filter_data(data,chans_2_filter,fc,BW):
    data1=np.copy(data)
    ts=data[1,0]-data[0,0]
    print(1/ts)
    for f in chans_2_filter:
    filt_wfm=filt.butter_filter(data[:,f],np.array([fc-BW/2,fc+BW/2]), 1/ts,5,'bandpass')


    data1[:,f]=filt_wfm
    #
    return(data1)

#%%

if __name__ == '__main__':


    path='\\\\pnl\\projects\cable_nde_proj\\FY23\\SSTDR Testing\\Jan2023_Data\\Thermally aged cable in oven\\Good Undamaged Cable open ends
Reference\\PNNL_SSTDR\\'


    fname='010_200MHzBW_100MHzFc_10us_800MSPS_2023-04-10_152027.csv'

    raw_data=np.loadtxt(path+fname,delimiter=',',skiprows=1)


    trig_chan=1
    data_chan=2
    ref_chan=3


    oscope_chan_indices=[trig_chan,ref_chan,data_chan]

    compute_xcor(raw_data,fname[:-4],path,oscope_chan_indices,False,200e6*(3/2),True)
    # compute_xcor(data,fname,path,scope_channels,False,SSTDR_BW*(3/2),False)
```

22