Open Radar Code Architecture (ORCA): A Platform for Software-defined Coherent Chirped Radar Systems

Thomas O. Teisberg, *Student Member, IEEE,* Anna L. Broome, *Student Member, IEEE,*
Dustin M. Schroeder, *Senior Member, IEEE*

***Abstract*—Ice-penetrating radar systems are critical instruments for observing the subsurface conditions on Earth's ice sheets. Traditionally, ice-penetrating radars have not been widely accessible to the glaciological community (as in the case of resource-intensive airborne systems) and/or have been limited in their ability to be reconfigured and optimized for studying different glaciological targets (as in the case of hardware-defined radar systems). To alleviate the challenges associated with this situation, we have developed the Open Radar Code Architecture (ORCA), which is an open-source radar software codebase that allows commercially available software-defined radios (SDRs) to be used as coherent ice-penetrating radars. Here, we describe the architecture of our code, characterize coherence on SDR-based radars, and demonstrate techniques we use to improve SNR and overall performance. We also highlight the variety of SDR options available to potential users and discuss tradeoffs between different system configurations.**

***Index Terms*—Software-defined radio, radar sounding, ice-penetrating radar, radar remote sensing.**

I. Introduction

THE cryosphere is one of the most sensitive components of the Earth system to a changing climate [1]. In particular, Earth's polar ice sheets represent the largest contribution to uncertainty in mean sea level rise by 2100 according to the Intergovernmental Panel on Climate Change (IPCC) models [1]. Reducing uncertainty in predictions of future global sea level change requires a better understanding of the physical processes and conditions underlying ice-sheet motion, as well as improved models that better capture the relevant physics and processes. Ice-penetrating radars, also known as radio echo sounders, are a widely-employed class of instrument whose measurements are used to estimate numerous glaciological conditions relevant to ice sheet models [2], [3]. These glaciological conditions include ice thickness, basal material, basal roughness, englacial and basal thermal state, englacial water, englacial layering, crystal orientation fabric, and other properties. Information about these conditions may be directly derived from ice-penetrating radar observations or they may be determined using inversion-based estimation techniques [4].

Traditionally, ice-penetrating radars used in glaciology have been limited to a few airborne systems (e.g. [5]–[8]), semi-custom ground-based systems (e.g. [9]–[12]), adaptations of commercial ground-penetrating radars (e.g [13], [14]) and other one-off systems developed by individual research groups (e.g. [15], [16]). All of these systems have distinct benefits and drawbacks, as well as their own unique resource constraints. A drawback shared amongst many existing systems is that they are rigid in their system design. In other words, once the radar system has been designed, tested, and built, it is very difficult to modify its configuration, which may be necessary to study different glaciological targets. Furthermore, the existing systems that are capable of surveying moderate to extensive spatial areas are so resource intensive that their field campaigns often require significant national or multinational support (e.g. [17]). This makes targeted data collection with these systems an unrealistic option for many research groups.

Software-defined radios (SDRs) are radio communications systems that implement core functionality, such as filtering, mixing, amplification, modulation, and demodulation, primarily in the digital domain, as opposed to using dedicated analog electronic circuits. Moving this core functionality from the analog to the digital domain increases the flexibility of the system, typically widening its operating frequency range, at the expense of fine-tuned performance. Utilizing SDRs as radar transceivers increases flexibility in the implementation of radar systems, particularly in terms of waveform design, choice of center frequencies and bandwidths, rapid reconfigurability, and system architecture choices.

Recently SDRs have been used as the backbone for several cryosphere-focused remote sensing systems. In [18], an Ettus E312 SDR was mounted on a hexacopter uncrewed autonomous system (UAS) to map snow depths. A bistatic ice-penetrating radar with wireless synchronization was developed in [19] using an Ettus E312 for its receiver, and successfully observed basal reflections from 1000 m thick ice. A passive radar sounder, using the sun as a signal source, has also observed basal reflections from 1000 m thick ice and is likewise based on an Ettus E312 SDR [20], [21]. The development of a mobile, ground-based ice-penetrating radar system that uses several SDRs from National Instruments and Ettus is detailed in [22] and observed basal reflections from ice as thick as 800 meters.

To improve community access to large-scale ice-penetrating radar measurements, the Stanford Radio Glaciology Lab has developed the Open Radar Code Architecture (ORCA). ORCA is an open-source radar software codebase (available at https://github.com/radioglaciology/uhd_radar), which can be

T. O. Teisberg, A. L. Broome and D. M. Schroeder are with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305. D. M. Schroeder is also with the Department of Geophysics at Stanford University. T. O. Teisberg and A. L. Broome are co-first authors.

Manuscript received March 19, 2024.

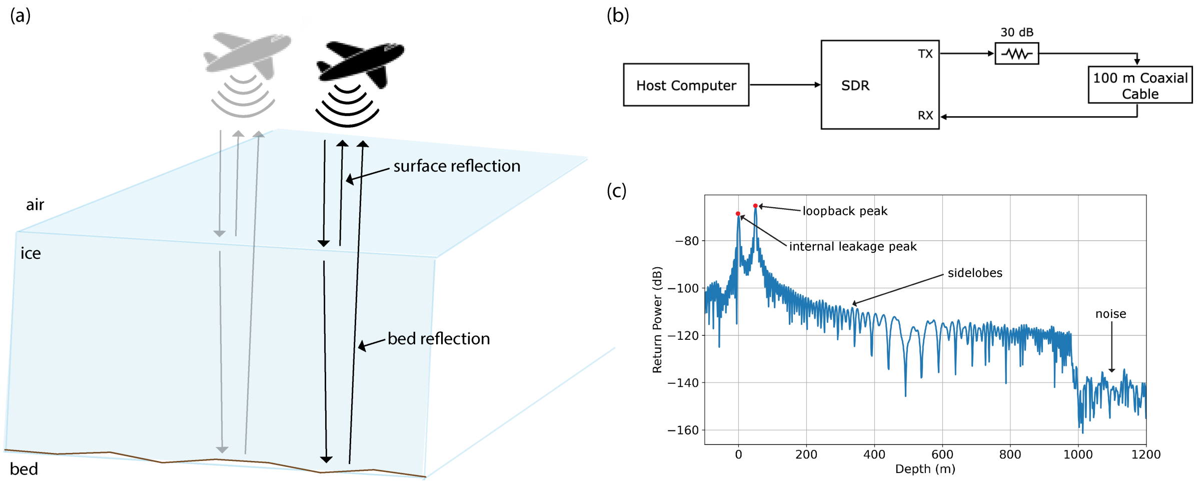

Fig. 1. (a) Cartoon of a typical ice-penetrating radar survey geometry. An electromagnetic signal is transmitted from the radar, which is depicted here on an airborne platform. These signals reflect off dielectric contrasts, such as the air-ice interface and the ice-bed interface. (b) Setup for a "loopback" test in which the SDR transmit port is directly connected to the receive port through an attenuator and coaxial cable that introduces a fixed time delay, which is dependent on the cable length. (c) Example of data produced by such a loopback test (after pulse compression), showing both the direct coupling internal to the SDR and the signal transmitted through the coaxial cable (red dots). Data collected on an Ettus B205mini SDR.

deployed on commercially available SDRs in the Ettus family [23]. This new codebase provides researchers with the ability to build reconfigurable coherent ice-penetrating radar systems from commercially available components and included open-source hardware designs at a relatively low cost. Using this codebase, SDR-based radars can mimic characteristics of the most common airborne and ground-based systems in use today, democratizing access to ice-penetrating radar technology. We have deployed ORCA on two distinct SDR-based ice-penetrating radar systems: a mobile, ground-based system using an Ettus X310 [24] and an uncrewed autonomous vehicle (UAV)-borne system using an Ettus B205mini-i [25].

II. Prototypical Radar System

Fig. 1(a) shows a conceptual example of an ice-penetrating radar system deployed on an airborne platform. The nadir-looking radar emits an electromagnetic signal, often pulsed (e.g. [5], [7]), or in some cases continuous (e.g. [9], [10]). Some of the transmitted signal is reflected off the air-ice interface back to the receiver, while a portion of the signal is transmitted through the surface and reflects off subsurface dielectric contrasts, including internal layers and the ice-bed interface, before propagating back upwards through the ice column to the receiver. The radar system demonstrated in this paper is designed to emit a modulated waveform (for example, but not limited to, a linear frequency-modulated chirp) and to receive samples that are phase-coherent relative to each transmitted pulse. We use the terms pulses, chirps, transmissions, and waveforms interchangeably in this work. We refer to samples as the individual data points that make up a pulse, chirp, transmission, or waveform.

Because testing ice-penetrating radar equipment *in situ* is a logistically challenging endeavor, we primarily test our systems in a laboratory setting using a "loopback" configuration as shown in Fig. 1(b). In this setup the SDR transmit port is directly cabled to the receive port, with at least 30 dB of in-line attenuation added to prevent receiver damage. Varying lengths of coaxial or fiber optic cable can be used as delay lines to simulate round-trip propagation of the radar signals through ice ($v_{coax} \sim$ 1.98e8 m/s, $v_{ice} \sim$ 1.68e8 m/s).

Fig. 1(c) shows the results from a loopback test, where 100 m of coaxial cable is connected between the transmitter and receiver. The loopback peak is visible at an effective distance of 50 m. A direct path peak resulting from internal leakage of the chirp between transmitter and receiver is also visible at an effective distance of 0 m. We plot distances in terms of an effective distance, corresponding to the one-way distance to a reflector, as opposed to the round trip distance directly computed from the two-way travel time. The data in Fig. 1(c) was collected using a linear frequency-modulated chirp and processed using pulse compression, also called match filtering. The match filtering process produces sidelobes, which are visible in Fig. 1(c). The nature of the sidelobes is dependent on the amplitude of the transmitting waveform and can be modified by applying window functions to the pulse [26]. Beyond a distance corresponding to the pulse length, the sidelobes end and the noise floor is visible.

III. Radar Code Architecture

The radar code is designed designed to repeatedly transmit a pre-generated waveform (such as a chirp) and receive synchronized returns (reflections) of these waveforms. In order to achieve the highest possible duty cycle, our code splits this task between two threads: a scheduler thread responsible for enqueueing commands for the SDR and a data-writing thread responsible for pulling received data from the SDR and storing it in persistent memory. The SDR-interfacing code is written in C++ and does not utilize any custom field-programmable gate array (FPGA) code, maximizing its flexibility and portability between different SDRs. Pre- and post-processing code is written in Python and described further in Sec. VII.

The Ettus SDRs use a system of first-in-first-out (FIFO) command queues for managing requested operations. These queues are implemented on FPGAs in each Ettus device, and execution of the commands has deterministic timing relative to the system clock. Commands can be assigned to run at specific times and will wait at the top of the queue to be executed until the assigned time is reached. Time-synchronized operations can be performed by enqueueing back-to-back transmit and receive commands with the same start times, or with a fixed offset. The command queues are relatively short in length and a failed command can quickly cascade into additional errors, so the state of the queue should be actively managed by the host computer, especially for high duty cycle operation.

In addition to managing the command queues, the host computer needs to pull data from the SDR and write it to some form of persistent storage. This process is limited by the host computer's processing power, the bandwidth of the interface between the SDR and the host computer, buffer space on the host computer, and a range of other factors that are dependent on the specific system architecture (e.g. the particular Ethernet or USB host controller and write speed of storage devices). Due to other processes running on the host, the exact rate at which data can be transferred and stored is effectively non-deterministic. Because of this non-determinism, our radar code uses a separate thread solely responsible for receiving and storing data from the SDR. The scheduler thread and data-writing threads are synchronized by simple counters that track the number of transmissions enqueued, the number successfully received, and any errors that occur. The overall architecture of the code is depicted in Fig. 2.

This separation of data storage (a highly buffered process with non-deterministic timing) and command scheduling (a minimally buffered process with tight timing constraints) is necessary to achieve high duty cycle operation. The command queue maximum lengths are quite short (eight commands on most Ettus SDRs), so the host software must continuously manage the queue to ensure that the next transit and receive commands are enqueued before their scheduled times. Because the commands are small, they do not use much bandwidth over the link between the computer and SDR.

In contrast, retrieving the data from the SDR is a high bandwidth operation and the rate at which the computer can read and record this data can vary significantly as the operating system switches between the numerous processes being run on any standard computer. Data buffers, both on the SDR and on the computer, allow for this non-deterministic timing of reading data. By keeping this process separate from the command scheduling, this architecture greatly reduces the impact of delays in transferring data on keeping the command queues full.

A. Error Handling

When data is not transferred fast enough (resulting in empty or full buffers) or commands are not enqueued prior to their scheduled times, errors can occur (more details on errors can be found in Section VIII-A). When an error occurs, it usually means that the host computer has fallen behind, leading to cascading errors. To mitigate this, the host code detects errors and temporarily increases the time before the next transmission to allow the host computer to catch up. Data about errors is stored in a log file so that post-processing code can reconstruct the exact timing of each transmission.

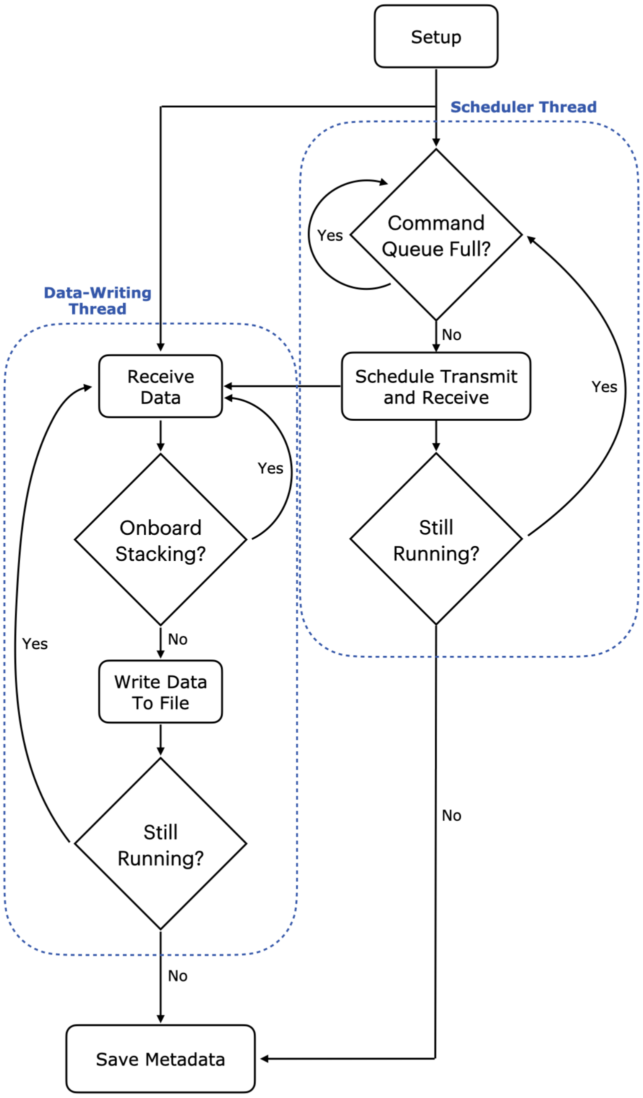

Fig. 2. The radar code runs on a host computer and manages the SDR through two threads highlighted by dashed blue boxes: a scheduling thread responsible for managing the SDR's command queue and a data-writing thread that receives and stores data.

B. Data Storage, Metadata, and Configuration

After the desired number of pulses have been collected, the host computer directs the SDR to stop transmitting and recording, and begins the data storage process. Radar data is saved as interleaved in-phase and quadrature samples (IQ format), with bit depth dependent on the specific SDR used. Metadata is also saved at this time including the configuration file (described below), a log file, and optionally a GNSS file. The log file records errors encountered by the SDR during data reception, as well as the version of code used, which is required for post-processing. The GNSS file may be recorded when SDRs that accept GNSS input feeds, in particular those with GPS-disciplined oscillators, are used. In this case, radar data is timestamped with the GNSS time and GNSS positioning data can be recorded at a user-specified interval. All data and metadata files are timestamped with the date and time at the end of the recording.

At runtime, the user provides a configuration file in YAML format, specifying the desired radar and data storage parameters. The configuration file contains options for setting waveform related parameters (e.g. chirp type, window function, bandwidth, pulse duration, sample rate), SDR communications parameters (e.g. device IP address, clock frequency, data format), RF parameters (e.g. center frequency, receive and transmit gain, local oscillator offset, sample rate), and data collection/storage parameters (e.g. recording length, pulse repetition frequency, radar data storage location, GNSS data storage location, maximum file size). An option is also included to specify whether the system should transmit (i.e. be an active radar system) or only record data without transmitting (i.e. be a purely passive receiver). Default configuration files for example systems are included in the repository. A single configuration file fully defines the entire radar system, allowing the code to be easily deployed on many types of SDRs. Furthermore, the configuration file is automatically saved along with the radar data, ensuring all necessary parameters for post-processing are readily available and facilitating the use of common post-processing code across multiple instruments. An example configuration file is included as a supplement to this article, and more configuration files can be found in the GitHub repository.

IV. Signal Coherence Characterization and Analysis

Numerous processing methods have been developed and applied widely to ice-penetrating radar data to improve the signal-to-noise ratio (SNR) and/or resolution of targets within the ice [4]. A select number of processing methods have been developed for impulse radar systems [27], but because chirped airborne ice-penetrating radars have collected much of the existing radar data over the Greenland and Antarctic Ice Sheets, many more processing methods have been developed for and applied to coherent chirped radar data. These methods include, but are not limited to, both unfocused [28] and focused [6], [8], [29] synthetic aperture radar (SAR) processing, specularity analysis [30], layer-optimized synthetic aperture processing [31], swath imaging [32], interferometry [33], and polarimetry [34]. Crucially, all of these methods rely on phase coherence of the transmitted signal with the receiver, such that, for a given radar location and unchanging scene, the phase of the reflection, after cross-correlation, is unchanging with time.

Some confusion can result from the binary categorization of radar systems as "coherent" or "incoherent". To help clarify, we distinguish between three common uses of the term coherence:

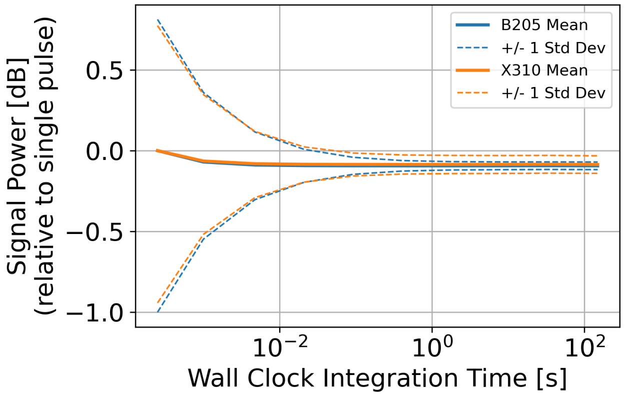

Fig. 3. Signal coherence testing was performed for both SDRs in a loopback configuration. Mean signal power dips only slightly relative to the single recording (unstacked) mean. This indicates good coherence of the system.

1) A **coherent radar system** is one in which the hardware is designed such that a transmitted signal reflecting off of a fixed set of targets is recorded with identical range and phase for each measurement.
2) **Coherent processing** is any post-processing done across measurements that relies upon the use of digitized in-phase and quadrature voltage signals (as opposed to using only the square or magnitude of the voltage signal) so that the coherence property of the radar, described in (1) can be used to improve SNR or resolution.
3) Particular imaged scenes or objects are sometimes called coherent, referring to either the predictability of phase changes as a function of time and/or space, or to constructive interference among multiple targets in a scene. These definitions are highly dependent on the imaging geometry and purpose of the data acquisition, however they are not properties of the radar system itself, so we do not consider them here.

While no hardware system is perfectly coherent, in practice, many systems, including ours, are close enough to being perfectly coherent that this subtlety can be ignored. For most applications, if the drop in mean signal power with a large amount of stacking is small compared to the targeted SNR, the system may be considered coherent. As shown in Fig. 3, the drop in signal power is about 0.1 dB. The exact degree of system coherence varies with both the choice of SDR and configuration settings (e.g. sample rate and bandwidth).

In common SDR architectures, reduced signal coherence comes from phase noise, which is a manifestation of the timing jitter caused by drift within and between onboard clocks and local oscillator(s) [35]. We quantify the coherence of our systems by transmitting repeated pulses in the loopback setup shown in Fig. 1(b) and comparing the signal peak magnitude and phase when averaging over varying numbers of transmitted pulses. Fig. 3 shows these results as a function of coherent integration time, which is the elapsed wall clock time corresponding to the duration of measurements over which we average (wall clock integration time).

Theoretically, as long as signal power remains constant,

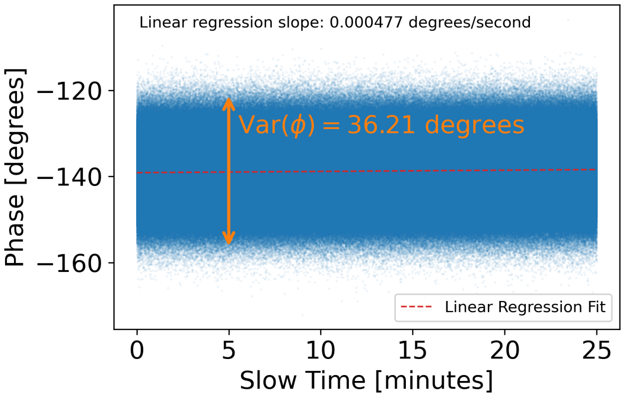

Fig. 4. Single pulse variance of the reflection phase matches well to the expected $\frac{1}{\text{SNR}}$ value of about 33 degrees. Over a long recording, the system drift may be estimated by a linear regression over the phases. The observed drift of less than 0.5 millidegrees per second is inconsequential for most applications. Data collected on a B205mini SDR in a loopback configuration with approximately 20 dB single pulse SNR.

every 10x increase in the number of coherently integrated pulses will result in a 10 dB increase in SNR (10 dB decrease in noise power) [36]. In our system, the power drops only slightly (0.1 dB) with increased stacking, as seen in Fig. 3(a), indicating good coherence. The slight drop in signal power of about 0.1 dB observed in Fig. 3 is a result of phase noise internal to the SDRs [35].

Another metric for evaluating coherence is to look at the long-term phase stability in a loopback setup. The variance of the phase of a reflection is commonly approximated as [37]:

$$\mathrm{Var}(\phi) \approx \frac{1}{\mathrm{SNR}} \tag{1}$$

Fig. 4 shows measured phases of a loopback peak over a 25 minute recording. By fitting a linear regression to the phases, the phase drift can be estimated. If desired, this drift can subsequently be removed from the recorded data by multiplying the data with a complex exponential containing a linear phase progression that is the inverse of the estimated phase drift. For many ice-penetrating radar applications, particularly those utilizing moving platforms, this level of improvement in coherence may not be necessary. We note that care should be taken to ensure that phase drift estimated in field data is intrinsic to the radar and not a property of the scene being imaged.

V. Noise Source Characterization and Analysis

Ice-penetrating radars are affected by several types of noise including thermal noise, external environmental noise, quantization noise, and leakage from internal components such as switching power converters or local oscillators (LOs) [36], [38]. Noise sources may be described and modeled as incoherent (i.e. thermal) or coherent (i.e. speckle, LO leakage). Both types affect the radar's ability to detect small signals, but each type requires different mitigation approaches. While incoherent noise can be reduced through averaging, coherent noise cannot be, since it adds in-phase with the signal. Understanding the types of noise affecting the radar system and mitigating them appropriately is thus essential to improve the radar's performance in terms of signal detection and SNR. Table I shows an overview of noise sources that may affect ice-penetrating radars and options that exist to mitigate those noise sources.

TABLE I
Potential noise sources affecting ice-penetrating radars and corresponding mitigation options.

Noise Type	Example Sources	Mitigation Options
Incoherent	thermal noise, power supply switching noise, background radiation, RFI	coherent summation (coherent averaging/stacking)*, RF front-end design, hardware filters, software filters*, shielding, antenna directivity
Coherent	local oscillators leakage, clock leakage, synchronization signals	phase dithering*, Hardware filters, software filters*, active cancellation

* Implemented as a pre- or post-processing option in the ORCA codebase.

Spectral analysis techniques may be used to determine relative contributions of different noise sources and to investigate a noise source's impact on a radar system's sensitivity. Alongside spectral analysis, a nuanced understanding of the radar system's architecture, and in our case, the architecture of the underlying SDR, is important to understand the system's noise characteristics.

A. Local Oscillator (LO) Leakage

Spectrograms are a powerful tool for understanding various noise sources that may be present in radar systems. Fig. 5 shows shows how various common noise sources may manifest themselves in spectral analysis. In Fig. 5(a) the impacts of LO leakage are clearly visible alongside a linear FM chirp. LO leakage is a form of coherent noise and limits the SNR gains achievable using pulse compression because the portion of the reference (transmitted) chirp centered around 0 Hz baseband is present in the received data at all delays, partially or wholly obfuscating reflections of the full linear FM chirp. To address the issue of LO leakage, the LO frequency can be tuned to a center frequency outside the chirp bandwidth, the digitally generated chirp signal can be moved away from baseband such that the LO contamination does not affect the chirp bandwidth of interest, or the frequencies corresponding to the LO leakage may be filtered out (using a notch filter) either during digital chirp generation or via analog filters in the front end. Our code provides support for any of these methods except the use of analog filters in the RF front end, as this would require a user-specific hardware implementation. While tuning the LO frequency outside of the chirp bandwidth either digitally or via RF/DSP tuning is an attractive option on the surface, it limits the available bandwidth that the chirped signal can occupy. This is because the sample rate of the system will be centered around the tuned LO frequency and the chirp

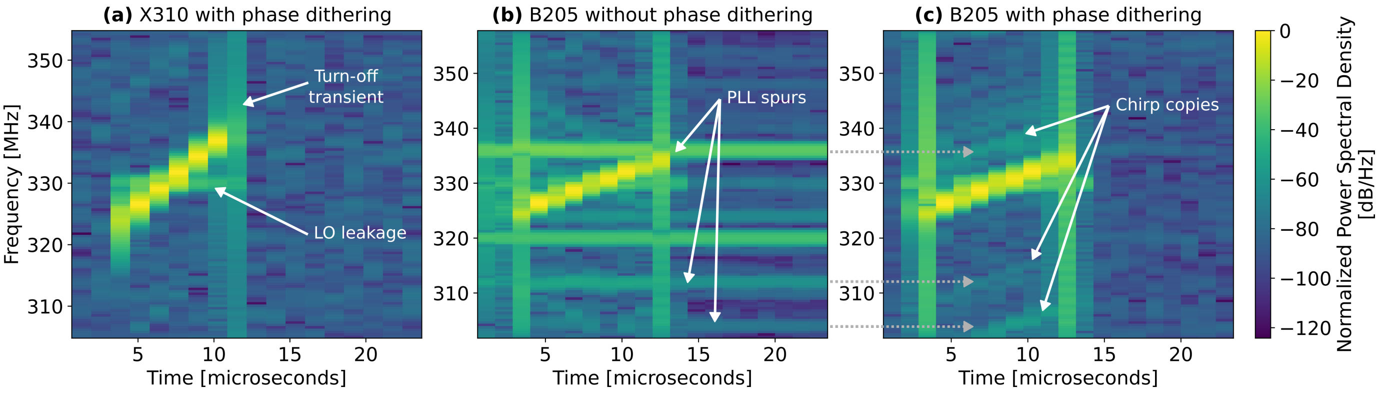

Fig. 5. Spectrograms showing various artifacts that may occur depending on the selected configuration. (a) As with all quadrature transceivers, some leakage of the local oscillator (LO) into the output signal occurs. (b) The LO fed to the mixer is generated by a phase-locked loop (PLL), which contains frequency spurs depending on the exact configuration. (c) For the same settings except with phase dithering enabled, the LO and its frequency spurs are significantly reduced, however copies of the transmitted chirp centered at the frequency spurs remain in the signal.

will inherently be one-sided around this new LO frequency, as opposed to occupying frequencies both above and below the LO as before. While this may be an acceptable tradeoff in some applications, it is often desirable to maximize the chirp bandwidth to enhance range resolution [37]. In applications necessitating maximal bandwidth, we suggest users consider processing options that include a notch filter around baseband to alleviate LO impacts.

The Ettus SDRs all use phase-locked loops (PLLs) to generate the LO signals. Although an ideal LO contains only a single frequency, the PLL output used as an LO contains other frequency components, known as spurs [39]. The location and strength of these spurs depends on the the reference oscillator frequency, the tuning frequency, and the design of the PLL. In general, we have found these spurs to be more problematic on the B205mini SDR, which always uses fractional-N PLLs. Fig. 5(b) shows examples of these spurs, with significant stacking applied to make these spurs visible. These spurs can be mitigated in stacked data by use of phase dithering (see Section V-E), however, as shown in Fig. 5(c), if the spur was present in the transmit LO signal, copies of the chirp will be maintained after phase-dithering, centered on each spur.

B. Switching Noise

Noise from switching DC-DC power converters can also couple into the receiver system and is often difficult to detect in a spectrogram. Fig. 6(a) shows an example of this noise in the time domain when a commercial switching power supply is used in close proximity to the receive antenna. The roughly 2 MHz repetition frequency of the spikes corresponds to the switching frequency of the power supply. Note that there is no electrical connection between the power supply and the SDR in this test setup. The recorded noise is the result of radiated emissions. Switching noise is typically not coherent, but it still presents a nuisance for recovering and interpreting weak and/or distant reflections. As shown in Fig. 6(b), the switching power supply noise can be reduced by coherent summation, however the noise level for a given coherent integration length always remains elevated compared to the noise-free case. Switching noise from power converters is often best mitigated by improving the design of the power supply, making use of non-switching power converters, or including more effective shielding [38].

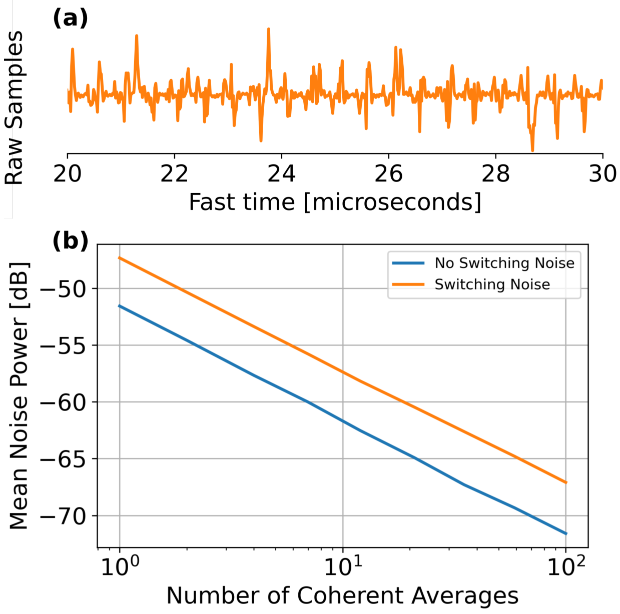

Fig. 6. (a) Noise from a switching power supply presents in the raw time-domain samples as spikes at a rate of 2 MHz (the power supply switching frequency). (b) The effect of this is to increase the initial single-pulse noise floor. While the noise floor can still be reduced by coherent integration, starting at a lower noise floor is desirable to increase SNR for a given coherent integration time. Data collected on a B205mini.

C. Radio-frequency Interference

External environmental (e.g. man-made) noise or radio-frequency interference (RFI) typically presents less of a problem for active radar systems as compared to passive ones (e.g. [21], [40]), but nonetheless can be a nuisance. The inherent flexibility of SDR-based radar systems provides several options for mitigating RFI. One option involves conducting a spectral scan at the field site and using statistical methods to determine which frequency bands have RFI contamination and which do not [21]. Then, only the clean frequency bands can be selected for use in the radar survey. Another option is to perform post-processing RFI removal to filter out in-band contamination prior to additional processing [21], [40].

D. Thermal Noise

Thermal noise (also called Nyquist noise or Johnson noise) is the noise generated by thermally-induced movement of bound charges, such as electrons [41]. The random fluctuations of thermal noise can be modeled as an additive white Gaussian noise source [42]. Because thermal noise is additive and incoherent relative to the radar signal, its impact can be reduced through averaging [36]. For complex-valued data, coherent averaging of N samples preserves the signal magnitude and phase, while reducing the mean noise power by a factor of N [19], [36]. Operating in a thermal noise-dominated regime is thus advantageous, because coherent averaging of data collected from a perfectly coherent radar system can in theory produce infinite decreases in mean noise power, and thus infinite increases in SNR. In reality, these performance improvements are finite, even for a perfectly coherent radar system, because eventually multiplicative noise such as clutter or sidelobes will become dominant over the remaining thermal noise.

Understanding the dominating noise regime at any given stage of the radar system and data processing pipeline is critical for systems without significant performance margins, as is the case in most SDR-based radar systems. Different noise regimes cause different limits on system performance and overcoming these limits requires distinct techniques depending on the primary noise source. Below we explore the impacts of coherent noise and how it is addressed in our radar platform.

E. Phase Dithering to Overcome Coherent Noise

Most radar systems we envision being built upon this architecture will have the noise for each recorded pulse dominated by an additive noise source, such as thermal noise at the antenna. As long as the target(s) of interest remain phase coherent (i.e. are not moving in the scene), coherent averaging across multiple pulses can then be used to reduce the noise power level while keeping the signal power constant (coherent summation, as opposed to averaging, would result in increased signal power and constant noise power) [36]. As the noise floor decreases, however, the system eventually will become dominated by a noise source that is coherent with the transmitted and received signals. Clocks and oscillators internal to the SDR that leak directly into the transmitted or received signal may be such a noise source. Eventually, clutter or sidelobes may also appear to be a limiting coherent noise source. Because these noise sources are coherent with the radar system, the mean noise power will level off as additional stacking is performed. This is the case shown by the orange lines in Fig. 7. In this case, LO leakage within the radar system quickly moves the system out of an incoherent noise dominated regime into a coherent noise dominated one.

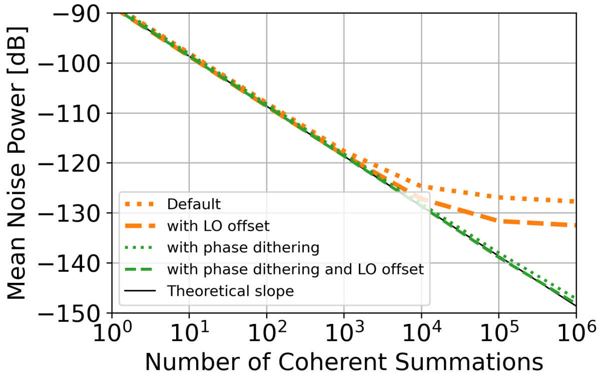

Fig. 7. Mean noise power as a function of number of coherent stacks (averages). The thin dotted orange line shows a coherent noise dominated system, where mean noise power levels off as the stacking amount increases. Shifting the LO away from the signal (thick dashed orange line) allows for a lower mean noise power but the system still reaches a coherent noise limited state. The green lines (dashed with LO offset and dotted without LO offset) show the results of applying phase dithering to make the coherent noise sources incoherent. This keeps the system in an additive noise dominated regime and allows for additional coherent processing gain to be achieved. This data is collected on an Ettus X310. Exact results will change depending on the system and initial single-pulse SNR.

One approach for mitigating coherent noise used by some radar systems is phase dithering [36]. Phase dithering consists of applying a variable phase offset to each transmitted waveform and inverting that phase offset upon reception. This process effectively decorrelates coherent noise sources while maintaining coherency of the transmitted waveform. After inverting for the applied phase shift, received data can be coherently averaged to realize a factor of N increase in SNR (assuming thermal noise is now the dominant noise source).

Phase dithering can be implemented in a number of ways depending on how the radar is designed. For an SDR-based radar, no additional hardware is needed. Each transmitted chirp is simply multiplied by a complex exponential and the received data is multiplied by the inverse. Our code implements phase dithering by applying a pseudo-random phase shift in this manner. Phase shifts are generated using a seeded pseudo-random number generator and are undone prior to writing received data to storage. Apart from minor computational overhead, there is no downside to phase dithering on an SDR-based system. Phase dithering only helps however, once a coherent noise source becomes dominant. The point at which this happens can vary significantly between systems and is dependent on the initial single-pulse SNR.

Fig. 7 highlights the impacts of different noise regimes and how they can be mitigated. Without mitigation, the noise power eventually levels out as the LO's coherent noise dominates (orange lines in Fig. 7). Shifting the chirp away from the LO allows for reaching a lower noise floor (dashed orange line), however other clocks and spurs of the LO can still result in a coherent noise dominated regime. By employing phase dithering (green lines in Fig. 7), these coherent noise sources are effectively made incoherent, allowing the mean noise power to continue declining at the theoretically expected

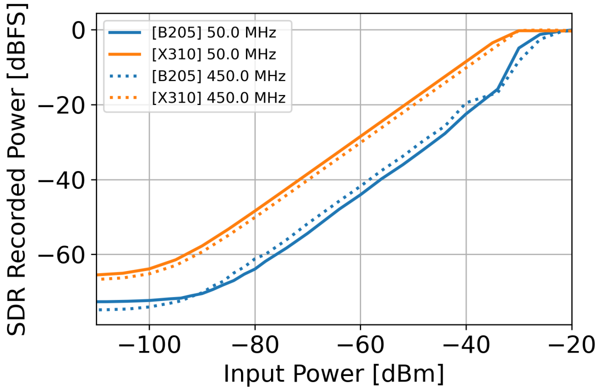

Fig. 8. Mapping between received power in physical units (dBm) and SDR units (dBFS) for the X310 (receive gain 30 dB) and the B205mini (receive gain 36 dB). The X310 has better linearity and a wider dynamic range, making it more suitable to applications requiring radiometric precision.

TABLE II
Select Ettus SDR Options as of Spring 2024 [23]

SDR	Frequency Range (GHz)	Bandwidth (MHz)	Channels	Mass (kg)	Cost
X310 with UBX 160	0.01-6	200	2 TX 2 RX	1.7	$14,000
X410	0.001-7.2	400	4 TX 4 RX	2.7	$28,000
B205mini	0.07-6	56	1 TX 1 RX	0.024	$1,500
B210	0.07-6	56	2 TX 2 RX	0.35	$2,200
N310	0.01-6	100	4 TX 4 RX	3.13	$18,000

10 dB per 10x increase in stacking, resulting in the ideal coherent processing gain of 60 dB for 1 million coherent averages.

VI. SDR Calibration

Radiometric calibration of the SDR may be useful for some studies, including those that seek to quantify reflectivity of the ice-bed interface [16] or surface properties [43]. Microwave radiometers, which are deployed in polar regions to observe englacial temperature [44], are effectively non-transmitting radars and also require radiometric calibration of the SDR. In its most basic form, this radiometric calibration is a knowledge of the actual power (in physical units) transmitted out of the SDR, plus a mapping of the received power in SDR units to physical units. Both of these calibrations are primarily a function of the user-specified SDR transmit and receive gain, and to a lesser degree, are a function of frequency.

The relation between SDR transmit gain and physically transmitted power is provided by Ettus for most of the SDR systems they produce (e.g. [45]). We have found this data to be largely accurate and do not reproduce it here. This relation can be confirmed by transmitting from the SDR into a calibrated receiver. The mapping of measured to physical power on the receive side is done by transmitting from a calibrated source into the SDR. Fig. 8 depicts the resulting calibration curve for an X310 system at an SDR receive gain of 30 dB (orange) and a B205mini system with receive gain of 36 dB (blue). Different values of SDR receive gain shift this curve left or right and different SDR platforms have different dynamic ranges and linearity.

VII. Processing Code

While the main focus of this paper is on the code that controls the SDRs and their behavior as radars, we briefly discuss the processing methods included within the ORCA repository for completeness.

The primary processing methods are written in Python, and example processing scripts are written in Jupyter Notebooks. Radar data is saved from the SDR into a binary file with samples in IQ format. We provide a method to read in the radar data, as well as associated metadata (i.e. the YAML file that catalogs the user-specified operating parameters), and convert the saved binary file into Zarr format, which is an open-source compressed file format meant for storing chunked, multi-dimensional arrays [46]. Converting the data to Zarr format slightly reduces storage requirements and is useful both for faster local and cloud-based processing. The radar data is loaded from the Zarr file into an Xarray Dataset, arranged by fast time sample index and slow time pulse index. The metadata from the associated YAML configuration file and reported errors that occurred during recording are loaded as attributes into the dataset.

Basic processing methods we include are stacking, which coherently averages together N pulses, and pulse compression, which uses a copy of the analytic transmitted signal to implement a match filter on the received data. Example scripts we provide include a basic field processing notebook (Field Processing.ipynb) that loads data, views a single pulse of raw data (useful to check for clipping), performs stacking and pulse compression, and displays 1D and 2D radargrams. This script also displays the power spectrum and fast time spectrogram of the data, which are both useful for debugging RFI and other issues. We also provide an example script, similar to the one used to produce Fig. 7, to compute SNR-related statistics such as mean noise power and noise power variance, as a function of the amount of stacking. This script can be useful for evaluating the coherence of the system. Notebooks to reproduce each of the figures in this paper are also included in the repository.

VIII. SDR & Host Hardware Options

Our code can be deployed on any SDR in the Ettus family that utilizes a host computer. The E3xx series SDRs, which contain their own embedded computers, are currently not supported though, to the best of our knowledge, there is nothing preventing adaptation of our code to these platforms. Table II lists examples of several Ettus SDRs and some of their capabilities most relevant for ice-penetrating radar systems. For some ice-penetrating radar applications, the bandwidth may be particularly important. Higher bandwidth SDRs are well suited to resolving fine details in the near surface ("snow" radars) while the lower cost, lighter weight systems with less bandwidth may be more appealing if the primary goal is detecting the ice-bed interface. Our code may be adapted to use intrinsically lower bandwidth SDRs in a stepped-frequency architecture to synthesize larger overall effective bandwidths. For an end user with a specific target, platform, or radar requirement in mind, multiple SDRs may be sufficient to complete the task or only one may meet the user's needs. We show this data, current as of early 2024, to demonstrate the range of choices one has in SDR selection.

Beyond SDR selection, the choice of host computer can have a significant impact on the performance of the radar system. Each SDR has a specific set of communications protocols it supports for getting commands from and transferring data to the host. For example, the Ettus X310 can utilize either 1 Gb or 10 Gb Ethernet connected via an optical transceiver for host communication, while the B205mini is restricted to USB 3. Selection of the communications interface hardware is also an important consideration.

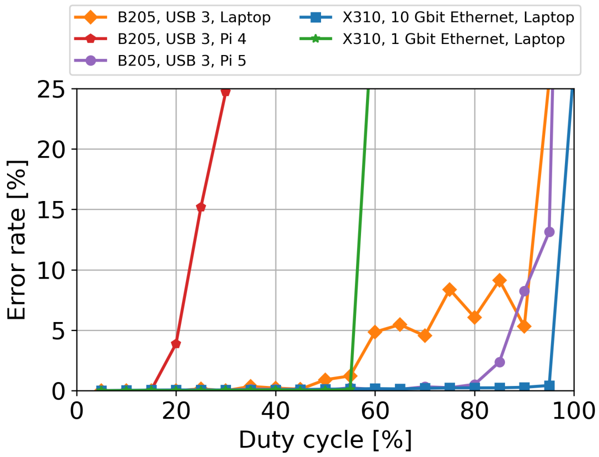

Fig. 9. Error rate as a function of duty cycle for various combinations of SDRs and host computers: B205mini with laptop via USB3 (orange diamonds), B205mini with Raspberry Pi 4 via USB3 (red pentagons), B205mini with Raspberry Pi 5 via USB3 (purple circles), X310 with laptop via 1 Gb Ethernet (green stars), and X310 with laptop via 10 Gb Ethernet (blue squares). All data is collected in a loopback configuration, with a sample rate of 50 MHz for the X310 data and 56 MHz for the B205mini data.

A. *Performance Benchmarking*

For many ice-penetrating radar applications, particularly those using relatively low-power transmitters, maximizing coherent processing gain is crucial for improving SNR. This means it is desirable to maximize the number of pulses transmitted (and received) in a given processing aperture. From a radar system design point of view, it is therefore important to design a radar capable of performing at a high pulse repetition frequency (PRF).

If the combination of SDR, communication interface, and host computer, is unable to perform at a given PRF, the most common behavior is that the command queue described in Sec. III becomes full of out-of-date commands and scheduling of data recording corresponding to each transmitted pulse results in a timing error (late command error). When this error occurs, data is not recorded for the error pulse. We note that we do not observe a loss of phase coherence between pulses before and after these errors.

Another error behavior sometimes observed (more commonly in scenarios with long pulse lengths) is the occurrence of overflow and underflow errors. These errors happen when the host either does not consume or does not produce data samples fast enough to keep up with the specified rate. These errors are addressed through careful choice of sample rates and performance tuning on the host side (e.g. by elevating thread priorities or writing to a RAM drive). For some applications and unavoidable host computer limitations, it may be necessary to utilize RF Network on Chip (RFNoC) blocks, such as the Replay block which utilizes DRAM on the SDR to buffer incoming and outgoing data, reducing the demands on the host computer [47].

Characterizing a realistically achievable maximum PRF is important for understanding the operating limitations of the radar system. In Fig. 9 we show the error rate achieved as a function of duty cycle and PRF for several SDR/interface/host computer combinations. Each error results in a small gap in data recording, so high error rates may pose issues for some post-processing algorithms. There is no hard threshold for what error rate is tolerable, but we treat an error rate of approximately 10% as an upper limit of what might be tolerable. In general, systems should be designed for much lower error rates if possible.

Fig. 9 shows that the limiting bandwidth is different for each SDR/interface/host combination. The X310 SDR connects to the host computer via a single 1 Gb, single 10 Gb, or dual 10 Gb Ethernet connection. When the host computer has sufficient processing power (as is the case for our host computers) it is the bandwidth of the Ethernet interface that limits maximum sample rates on the X310 system. When connecting to the SDR via 1 Gb Ethernet, Ettus states the expected maximum sample rate for a single channel with 16 bit I/Q samples should be 25 Msps, while when using a single 10 Gb Ethernet connection the maximum sample rate for a single channel should be 200 Msps [48]. To achieve the full 200 Msps rate on multiple channels requires dual 10 Gb Ethernet [48]. We do not currently have a use case for and do not test full 200 Msps streaming from all channels using dual 10 Gb Ethernet.

Fig. 9 demonstrates that our code is capable of running at the respective data rate limits on the X310 with either 1 Gb or 10 Gb Ethernet connections. X310 loopback data in Fig. 9 were collected using a sample rate of 50 MHz, meaning that a 1 Gb Ethernet connection should only support duty cycles of approximately 50%, while a 10 Gb Ethernet connection should be capable of supporting 100% duty cycles. Operation at these limits is demonstrated by the green (1 GbE) and blue lines (10 GbE) in Fig. 9. Beneath the Ethernet data rate limit, error rates of less than 2% are reliably achieved.

For systems based on the B205mini SDR, which uses a USB 3.0 interface, achievable data rates are heavily dependent on the host computer's specifications, especially the USB 3 controller. Use of a Raspberry Pi 4, for example, is possible but significantly limits the achievable duty cycle, as compared to use with a high performance laptop or even a Raspberry Pi 5, shown by the red, orange, and purple lines in Fig. 9, respectively. Users with applications that demand high PRFs should exercise caution when using SDRs with USB interfaces, as well as if using host computers with limited processing power.

IX. Conclusion

The Open Radar Code Architecture (ORCA) is a powerful platform for developing ice-penetrating radar systems that we hope will enable more glaciologists to collect their own radar data. ORCA is an open-source codebase for running coherent, chirped ice-penetrating radars on software-defined radios (SDRs) within the Ettus family. ORCA includes options to overcome thermal noise impacts (via coherent stacking) and coherent noise impacts (via phase dithering and filtering). ORCA has been successfully deployed on two distinct systems developed by the Stanford Radio Glaciology Lab: Peregrine, a fixed-wing UAV ice-penetrating radar [25] and MAPPERR, a towed ground-based ice-penetrating radar system [24]. ORCA code, as well as open-source hardware designs and building instructions for Peregrine and MAPPERR are available at https://github.com/radioglaciology/uhd_radar.

Acknowledgment

The authors would like to thank NASA, the NSF, NDSEG, the Heising-Simons foundation, and Stanford University for financial and logistical support that made the development and testing of the described system possible.

T. O. Teisberg was supported by a NASA FINESST Grant (80NSSC23K0271). A. L. Broome was supported by the NDSEG Fellowship Program.

This material is based upon work supported by the National Science Foundation under Grant No. 284791. Field testing was supported by NASA under Grant 80NSSC22K0782.

Additional support was provided by the Heising-Simons Foundation, the TomKat Center for Sustainable Energy, Stanford Data Science, and the Stanford Institute for Human-Centered Artificial Intelligence.

References

[1] B. Fox-Kemper *et al.*, *Ocean, Cryosphere and Sea Level Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2021, p. 1211–1362.

[2] E. Larour, H. Seroussi, M. Morlighem, and E. Rignot, "Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM)," *J. Geophys. Research: Earth Surface*, vol. 117, no. F1, pp. 1–20, 2012.

[3] H. Seroussi *et al.*, "ISMIP6 Antarctica: a multi-model ensemble of the Antarctic ice sheet evolution over the 21st century," *The Cryosphere*, vol. 14, no. 9, pp. 3033–3070, 2020.

[4] D. Schroeder *et al.*, "Five decades of radioglaciology," *Annals of Glaciology*, vol. 61, no. 81, pp. 1–13, 2020.

[5] F. Rodriguez-Morales *et al.*, "Advanced Multifrequency Radar Instrumentation for Polar Research," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2824–2842, 2014.

[6] M. Peters *et al.*, "Along-track focusing of airborne radar sounding data from West Antarctica for improving basal reflection analysis and layer detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 9, pp. 2725–2736, 2007.

[7] D. A. Young, D. M. Schroeder, D. D. Blankenship, S. D. Kempf, and E. Quartini, "The distribution of basal water between Antarctic subglacial lakes from radar sounding," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2059, p. 20140297, 2016.

[8] F. Hélière, C.-C. Lin, H. Corr, and D. Vaughan, "Radio Echo Sounding of Pine Island Glacier, West Antarctica: Aperture Synthesis Processing and Analysis of Feasibility from Space," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2573–2582, 2007.

[9] P. Brennan *et al.*, "Phase-sensitive FMCW radar system for high-precision Antarctic ice shelf profile monitoring," *IET Radar, Sonar & Navigation*, vol. 8, no. 7, pp. 776–786, 2014.

[10] K. W. Nicholls, H. F. Corr, C. L. Stewart, L. B. Lok, P. V. Brennan, and D. G. Vaughan, "A ground-based radar for measuring vertical strain rates and time-varying basal melt rates in ice sheets and shelves," *Journal of Glaciology*, vol. 61, no. 230, pp. 1079–1087, 2015.

[11] L. Mingo and G. Flowers, "An integrated lightweight ice-penetrating radar system," *J. Glaciology*, vol. 56, no. 198, pp. 709–714, 2010.

[12] L. Mingo, G. E. Flowers, A. J. Crawford, D. R. Mueller, and D. G. Bigelow, "A stationary impulse-radar system for autonomous deployment in cold and temperate environments," *Annals of Glaciology*, vol. 61, no. 81, pp. 99–107, 2020.

[13] S. Campbell, Z. Courville, S. Sinclair, and J. Wilner, "Brine, englacial structure and basal properties near the terminus of McMurdo Ice Shelf, Antarctica," *Annals of Glaciology*, vol. 58, no. 74, pp. 1–11, 2017.

[14] M. G. P. Cavitte *et al.*, "From ice core to ground-penetrating radar: representativeness of SMB at three ice rises along the Princess Ragnhild Coast, East Antarctica," *Journal of Glaciology*, pp. 1–13, 2022.

[15] B. C. Welch and R. W. Jacobel, "Analysis of deep-penetrating radar surveys of West Antarctica, US-ITASE 2001," *Geophysical Research Letters*, vol. 30, no. 8, 2003.

[16] K. Christianson *et al.*, "Basal conditions at the grounding zone of Whillans Ice Stream, West Antarctica, from ice-penetrating radar," *J. Geophys. Research: Earth Surface*, vol. 121, no. 11, pp. 1954–1983, 2016.

[17] J. A. MacGregor *et al.*, "The scientific legacy of NASA's Operation IceBridge," *Reviews of Geophysics*, vol. 59, no. 2, 2021.

[18] S. Prager, G. Sexstone, D. McGrath, J. Fulton, and M. Moghaddam, "Snow Depth Retrieval With an Autonomous UAV-Mounted Software-Defined Radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[19] N. Bienert *et al.*, "Post-processing synchronized bistatic radar for long offset glacier sounding," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.

[20] S. T. Peters, D. M. Schroeder, M. S. Haynes, D. Castelletti, and A. Romero-Wolf, "Passive Synthetic Aperture Radar Imaging Using Radio-Astronomical Sources," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2021.

[21] S. Peters, D. Schroeder, D. Castelletti, M. Haynes, and A. Romero-Wolf, "In situ demonstration of a passive radio sounding approach using the sun for echo detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7338–7349, 2018.

[22] P. Liu *et al.*, "Software-Defined Radar Systems for Polar Ice-Sheet Research," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 3, pp. 803–820, 2019.

[23] Ettus Products. [Online]. Available: https://www.ettus.com/products/

[24] A. L. Broome, D. M. Schroeder, and J. T. Johnson, "First Results from Mapperr: The Multi-Frequency Active Passive Polar Exploration Radar-Radiometer," *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, vol. 00, pp. 36–39, 2023.

[25] T. O. Teisberg, D. M. Schroeder, A. L. Broome, F. Lurie, and D. Woo, "Development of a UAV-Borne Pulsed Ice-Penetrating Radar System," *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, vol. 00, pp. 7405–7408, 2022.

[26] G. Heinzel, A. Rudiger, and R. Schilling, "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows." 2002. [Online]. Available: https://holometer.fnal.gov/GH_FFT.pdf

[27] D. A. Lilien, B. H. Hills, J. Driscol, R. Jacobel, and K. Christianson, "ImpDAR: an open-source impulse radar processor," *Annals of Glaciology*, pp. 1–10, 2020.

[28] M. Peters, D. Blankenship, and D. Morse, "Analysis techniques for coherent airborne radar sounding: Application to West Antarctic ice streams," *J. Geophys. Research: Solid Earth*, vol. 110, no. B6, 2005.

[29] J. J. Legarsky, S. P. Gogineni, and T. L. Akins, "Focused Synthetic Aperture Radar Processing of Ice-Sounder Data Collected Over the Greenland Ice Sheet," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 10, p. 2109, 2001.

[30] D. Schroeder *et al.*, "Estimating subglacial water geometry using radar bed echo specularity: Application to Thwaites Glacier, West Antarctica," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 443–447, 2015.

[31] D. Castelletti, D. M. Schroeder, E. Mantelli, and A. Hilger, "Layer optimized SAR processing and slope estimation in radar sounder data," *Journal of Glaciology*, vol. 65, no. 254, pp. 983–988, 2019.

[32] J. Paden, T. Akins, D. Dunson, C. Allen, and P. Gogineni, "Ice-sheet bed 3-D tomography," *Journal of Glaciology*, vol. 56, no. 195, pp. 3–11, 2010.

[33] A. Jenkins, H. F. Corr, K. W. Nicholls, C. L. Stewart, and C. S. Doake, "Interactions between ice and ocean observed with phase-sensitive radar near an Antarctic ice-shelf grounding line," *Journal of Glaciology*, vol. 52, no. 178, pp. 325–346, 2006.

[34] T. M. Jordan, D. M. Schroeder, D. Castelletti, J. Li, and J. Dall, "A polarimetric coherence method to determine ice crystal orientation fabric from radar sounding: Application to the NEEM Ice Core region," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8641–8657, 2019.

[35] T. O. Teisberg, D. M. Schroeder, A. L. Broome, and R. Culberg, "Coherence and phase noise in software-defined radio-based ice-penetrating radar instruments," *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, 2024, in review.

[36] C. T. Allen, S. N. Mozaffar, and T. L. Akins, "Suppressing Coherent Noise in Radar Applications With Long Dwell Times," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 3, pp. 284–286, 2005.

[37] F. Ulaby and D. Long, *Microwave Radar and Radiometric Remote Sensing*. Artech House, 2015.

[38] K. Player, T. Stumpf, J.-B. Yan, F. Rodriguez-Morales, J. Paden, and S. Gogineni, "Characterization and Mitigation of RFI Signals in Radar Depth Sounder Data of Greenland Ice Sheet," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 6, pp. 1060–1067, 2013.

[39] C. Barrett, "Fractional/Integer-N PLL Basics," 1999, TI Publication SWRA029.

[40] M. Andrews, J. Johnson, M. Brogioni, G. Macelloni, and K. Jezek, "Properties of the 500-2000-MHz RFI environment observed in high-latitude airborne radiometer measurements," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2021.

[41] D. M. Pozar, *Microwave engineering*. John wiley & sons, 2011.

[42] J. C. Curlander and R. N. McDonough, *Synthetic aperture radar systems and signal processing*. Wiley, New York, 1991, vol. 11.

[43] C. Grima *et al.*, "Surface and basal boundary conditions at the Southern McMurdo and Ross Ice Shelves, Antarctica," *J. Glaciology*, vol. 65, no. 252, pp. 675–688, 2019.

[44] C. Yardim *et al.*, "Greenland ice sheet subsurface temperature estimation using ultrawideband microwave radiometry," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2021.

[45] (2021) UBX - RF Performance Data. [Online]. Available: https://kb.ettus.com/UBX#RF_Performance_Data

[46] "Zarr-python documentation." [Online]. Available: https://zarr.readthedocs.io/en/stable/index.html

[47] (2022) Using the RFNoC Replay Block in UHD 4. [Online]. Available: https://kb.ettus.com/Using_the_RFNoC_Replay_Block_in_UHD_4

[48] (2016) About usrp bandwidths and sampling rates. [Online]. Available: https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates