Calibrating a 4-channel Fully-Digital 60 GHz SDR

Aditya Dhananjay
New York University, Pi-Radio
Brooklyn, NY, USA

Kai Zheng
Pi-Radio
Brooklyn, NY, USA

Jaakko Haarla
Aalto University
Aalto, Finalnd

Lorenzo Iotti
UC Berkeley
Berkeley, CA, USA

Marco Mezzavilla
New York University, Pi-Radio
Brooklyn, NY, USA

Dennis Shasha
New York University, Pi-Radio
Brooklyn, NY, USA

Sundeep Rangan
New York University, Pi-Radio
Brooklyn, NY, USA

ABSTRACT

The Pi-Radio v1 software-defined radio (SDR) features a 4-channel fully-digital transceiver board operating in the 57-64 GHz band; when mated with the Xilinx RFSoC-based ZCU111 board, this forms a powerful SDR that can be used by the research community. This paper describes the calibration procedures for the SDR, with a special emphasis on not relying on expensive laboratory equipment / infrastructure like a spectrum analyzer, signal generator, or even an anechoic chamber. We hope this is an interesting read for those interested in the pipeline from hardware design schematic to fully functional SDR.

CCS CONCEPTS

• **Hardware** → *Electromagnetic interference and compatibility*; **Beamforming**; *Digital signal processing*.

KEYWORDS

mmWave, SDR, beamforming, fully-digital, calibration

ACM Reference Format:
Aditya Dhananjay, Kai Zheng, Jaakko Haarla, Lorenzo Iotti, Marco Mezzavilla, Dennis Shasha, and Sundeep Rangan. 2020. Calibrating a 4-channel Fully-Digital 60 GHz SDR. In *Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH '20), September 21, 2020, London, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3411276.3412195

1 INTRODUCTION AND MOTIVATION

Experimental research, especially in the millimeter wave bands and beyond, remains out of reach for too many members of the academic research community. Of course, theory and simulations are a critical aspect of any research being conducted; but it is not controversial to argue that more experimentation is needed. The reason for this lack of experimentation is painful: existing commercially available software defined radio (SDR) systems are prohibitively expensive, or feature technologies that are at least 10 years behind the curve. Academic groups typically do not have large budgets like the industry. Recent efforts to rectify this lack of access include the very impressive M3 effort from UC San Diego[1]. This paper describes our effort toward solving this problem.

Pi-Radio's vision[1] is to make SDR systems featuring advanced transceiver technologies available to the research community at reasonable rates. We have designed, built, and tested our v1 SDR that features a 4-channel fully-digital transceiver that operates in the 57-64 GHz band. It turned out that actually getting the hardware to work correctly was quite challenging, the main issue being *calibration*. We explicitly designed these calibration techniques to be performed without the need for expensive laboratory bench equipment like mmWave spectrum analyzers, signal generators (synthesizers), or even anechoic chambers. The various stages involve the measurement and calibration of: **a)** crystal frequency offsets; **b)** linear operating ranges; **c)** timing offset corrections; **d)** LO phase offset corrections; **e)** magnitude corrections; **f)** IQ gain imbalance corrections; and **g)** IQ quadrature LO phase imbalance corrections. We freely admit that this paper teaches no scientific breakthroughs; the math behind calibration is explained in many textbooks. However, we wanted to share with the community how we implemented these on a real SDR system, drawing attention to the ways that its behavior can deviate from ideal. To show correct calibration, we conclude this paper with a demonstration of TX/RX beamforming through geometrically determined beamforming weights.

2 HARDWARE DESCRIPTION

The SDR consists of three main parts: a) the Xilinx RFSoC-based ZCU111 baseband board; b) the Pi-Radio 4-channel fully-digital transceiver board; and c) a host computer running various software tool chains.

Xilinx ZCU111 FPGA Board [2]: This RFSoC-based FPGA board features eight 14-bit high-speed DACs and eight 12-bit high-speed ADCs. We clock the DACs and ADCs at 3932.16 MSps, leading

WiNTECH '20, September 21, 2020, London, United Kingdom
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8082-9/20/09…$15.00
https://doi.org/10.1145/3411276.3412195

[1] Pi-Radio is a spin-off from the New York State Center for Advanced Technologies in Telecommunications (CATT) located at the Tandon School of Engineering at New York University.

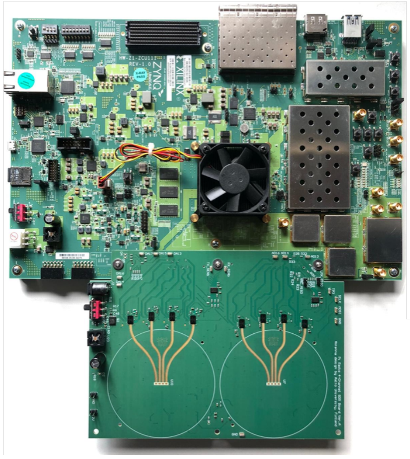

Figure 1: The bottom board is the Pi-Radio v1 transceiver board; the board at the top is the Xilinx RFSoC-based ZCU111 evaluation board. The RF shield has been removed to show the transceiver board clearly.

to a maximum theoretical baseband bandwidth of nearly 4 GHz (since the I and Q components are treated independently). The RFSoC also features four ARM Cortex-A53 cores and two ARM Cortex-R5 cores; these powerful ARM cores can be used to run Linux, various applications, and also perform real-time operations. Importantly, this FPGA contains eight soft-decision forward error correction (SD-FEC) cores in silicon; these can be used to implement Turbo, Viterbi, LDPC, or Polar decoders (highly computationally intensive operations) without using any resources on the actual FPGA.

Pi-Radio v1 Transceiver Kit: This kit consists of a single board that implements a fully-digital 4-channel mmWave transceiver. On the TX side, it receives four I/Q analog baseband pairs from the eight DACs on the RFSoC, and feeds them to a bank of four Analog Devices (ADI) HMC6300 mmWave up-converters. The resulting mmWave signals are fed to a 1x4 patch antenna array using co-planar waveguide transmission lines, both designed by Aalto University (Finland). The RX side is symmetrical, with the ADI HMC6301 mmWave down-converters used to convert the four incoming RF signals to baseband, which are then fed into the eight ADCs of the ZCU111 board. While the baseband bandwidth (supported by the ZCU111) is 4 GHz, the mmWave HMC chips support 2 GHz of bandwidth. The operating frequency range is 57-64 GHz. All eight channels (four TX and four RX) are phase synchronized by a network comprising of: **1:** LO generation using the Texas Instruments LMX2595; **2:** dual-stage LO amplification using the ADI HMC962 and HMC441 parts; and **3:** LO distribution using Wilkinson dividers by Knowles Dielectric Labs. The boards are fabricated and assembled (including fully automated pick-and-place) by Sierra Circuits in Sunnyvale, CA. The boards also have two large circular *keep-out* areas around the antenna arrays, which users can use to mount their own passive dielectric lenses, if needed.

Host Computer: The ZCU111 board connects to a host computer using a gigabit ethernet interface and a simple TCP/IP control/data interface. MATLAB (or any other TCP/IP capable software like GNU Radio) can be run on the host computer to control the SDR operation. We have already implemented MATLAB-based drivers for this system. On the TX side, this allows the per-channel waveforms to be created in MATLAB, and shipped over to the RFSoC to be transmitted in a loop, until configured otherwise. On the RX side, the MATLAB code *triggers* the RFSoC to capture a set of samples from all ADCs synchronously, and then ship them over to MATLAB for further processing.

3 CALIBRATION PROCEDURES

It is well known that mmWave systems rely on *beamforming* to close the link budgets. The very process of beamforming requires the TX and RX array elements to transmit and receive with known amplitudes and phases relative to each other. Achieving this deterministic behavior requires several types of calibration, which we examine in this section. One of the goals was to perform this calibration without needing expensive lab bench equipment like mmWave spectrum analyzers and signal generators that can cost several hundreds of thousands of dollars to procure. The techniques below therefore perform *two-node calibration* that involves placing two SDR nodes in boresight (i.e., facing each other). One is the node under calibration (NUC) and the other is the reference node (REF).

3.1 Frequency Offset Calibration

Many of the calibration techniques in this paper involve performing frequency domain correlation. For this to work properly, the local oscillators (LOs) on the NUC and REF need to be very close together, without large frequency offsets. The LO on the Pi-Radio SDR board is generated by the TI LMX2595 PLL chip that takes in a reference crystal oscillator input. We use a nominal 156.25 MHz crystal on our boards, but practical crystals always show some variations from ideal. Even small variations in the crystal frequency can lead to pretty large variations in the final RF frequency. We therefore measured the frequency of the crystal on each board independently by *probing* the crystal output pads, and connecting the probe to a low-cost spectrum analyzer (Tektronix RSA306 that can measure up to 6 GHz). While calibrating two particular nodes, we observed that their crystal frequencies were 156.249725 MHz and 156.246375 MHz. This difference can lead to the mmWave RF center frequencies on the two nodes to differ by as much as 650kHz. Given that many calibration procedures utilize FFT bins as narrow as 500kHz, these frequency offsets need to be fixed.

Of course, fixing these offsets is simple. Once the crystal output frequencies are accurately measured, we simply recalculate the LMX2595 PLL registers (fractional and integer divider values) with the measured frequencies, instead of just using the nominal frequencies. To test correct performance, we modulated a single tone at the TX side of the NUC, and measured the spectrum at the RX side of the REF, and observed that the correct bin on the RX side was populated. We then recalculated the LMX2595 registers for crystal frequencies that very slightly deviated from the measured frequencies, and observed significant leakage into neighboring frequency bins. This showed that the LO frequencies were calibrated correctly, and we could proceed to the next step.

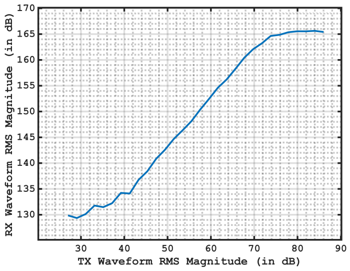

Figure 2: Measuring the Linear Operating Range of the TX: Keeping the TX waveform RMS magnitude below 165 dB ensures linear operation. Below 135dB, the system becomes noise limited.

3.2 Linearity Measurements

The HMC6300 mmWave up-converter has an output P1dB of about 12 dBm. Considering nominal patch antenna gains of 3 dB each on the TX and RX side, as well as a nominal 1m spacing between the nodes, the received power is calculated to be about −50 dBm. The input P1dB point on the HMC6301 mmWave down-converter is rated at −19 dBm. We therefore have no concerns about the linearity on the RX side, but we need to measure the linearity on the TX side to ensure correct operation and prevent saturation.

To do this, we ran a simple experiment. The TX NUC transmitted a single tone baseband signal from TX channel *txIndex*. We varied the transmitted power (by digitally scaling the TX waveform), and measured the receive power on one channel of the the RX REF. The results are shown in Fig. 2. The X axis is simply the RMS value of the digital TX waveform, expressed in dB. The Y axis is the received power on the RX REF, at the tone of interest. This shows that as long as we keep that RMS TX magnitude (digital) below approximately 74 dB, we will operate in the linear range. These results are consistent across different TX channels. If the TX waveform is below 40 dB, quantization and thermal noise begins to dominate. As mentioned earlier, for the link budgets under consideration, RX-side linearity is not a concern.

3.3 Timing and Phase Offset Calibration

What causes the problem? The baseband IQ transmission lines (TLines) from the RFSoC to the HMC6300/6301 chips might not be length matched. For a given channel, the IQ differential lines were designed to be length matched; however, the lengths of the TLines for different channels can be different, because length-matching TLines can lead to additional losses and routing complexity. Further, the length of the TLines between the RFICs (the HMC6300 and HMC6301) and the antenna elements are also not length matched because the additional losses in the mmWave bands caused by increasing TLine length can be very significant. These length mismatches cause *per-channel timing offsets* that need to be calibrated out. On the Pi-Radio transceiver board, the eight channels (4 TX and 4 RX) are phase synchronized using a Wilkinson tree that distributes the local oscillator (LO) signal. The TLines in this tree are also of different lengths, in order to make routing feasible and to minimize board losses. This leads to *per-channel phase offsets* that also need to be calibrated out, along with the *per-channel timing offsets*.

How do we calibrate? We now describe the method to calibrate the TX array on the NUC. The RX array on REF uses just one active channel; the other channels can either be turned off or used for redundancy. When two nodes are placed in boresight, the TX array on the NUC will be directly in front of the RX array on REF, with about two feet of spacing between them to ensure far-field operation.

The first step is to estimate the fractional timing offset. The TX NUC simultaneously transmits four orthogonal wideband sounding sequences, one per channel (randomly-chosen QPSK sequences having good autocorrelation properties). The RX REF passes the received waveform through a fractional delay filter (for several fractional timing offset hypotheses), and correlates this against each of the four sounding sequences; this process determines which fractional timing offset leads to the largest peak in the power delay profile for each of the four TX channels. This process is repeated over several iterations, and the results are *smartly* averaged (as explained later) to estimate the fractional timing offsets for each channel. The intuition is that for incorrect fractional timing offset corrections, the power in the PDP peak gets *spread out* over adjacent peaks; but for the correct fractional timing offset correction, the power is *concentrated* in just one peak.

The correctness of this method is verified by repeating the experiment, but having the TX NUC pre-compensate the sounding sequence by fractionally delaying it by the previously determined fractional timing offset. This time, we observe that the optimal fractional timing offset determined by the RX REF is very close to 0 for all channels, thereby demonstrating that the correction is indeed correct.

The next step is to estimate the *per-channel phase offsets*. Repeat the experiment above, where the transmitted waveforms are pre-compensated by the optimal fractional timing offsets for each TX channel on the NUC. On the RX REF, correlate the received waveform against each of the four sounding sequences, and observe the phase of the maximum peak in the resulting PDPs. Suppose the phases are β_n, for $n \in \{1, 2, 3, 4\}$. The phase correction factors are therefore $\gamma_n = (\beta_n - \beta_1)$ for $n \in \{1, 2, 3, 4\}$. To verify correct detection of the phase offsets, further pre-compensate the transmitted sounding sequences on the TX NUC by de-rotating each sounding sequence n by $exp(j\gamma_n)$; this time, we observe that the phases of all the received peaks are equal, thereby demonstrating correct per-channel phase offset correction.

A Note on Smart Averaging: While searching through all possible fractional timing offsets, the experiment is run over several iterations, and the results are averaged; this helps with getting clean and stable calibration factors. However, we cannot simply average the fractional timing offsets across iterations. This is because we might get −0.5 as the *best_to* (best timing offset estimate) in one iteration, but +0.5 in the next, leading to an incorrect average of 0, where in reality, the two offsets are equivalent. To overcome this issue, we use a simple and well-known trick: **a)** Multiply *best_to* by 2π; **b)** Create a complex number with the resulting phase; **c)** Add these complex numbers across all iterations; and **d)** take the phase of the summed up complex number. This leads us to another realization: the algorithm so far has been unable to distinguish between positive and negative fractional timing offsets (for example, −0.3

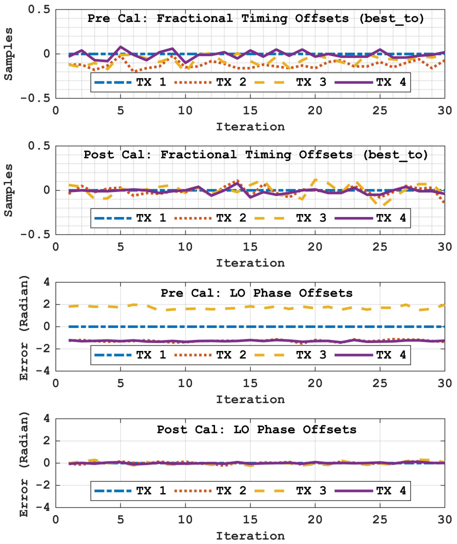

Figure 3: Calibrating the fractional timing offsets and LO phase offsets on the TX array. The integer timing offsets are 0, and have hence not been plotted. Calibrating the RX array is symmetrical.

and +0.7). So, the experiment is repeated with the TX NUC precompensating the sounding sequence by the estimated per-channel fractional timing offset and per-channel phase offsets. The RX REF captures the received waveform, and checks the location (in time samples) of all four received PDP peaks (one per TX channel). It is immediately visible as to whether some peaks are earlier than others, thereby yielding the integer timing offsets that need to be augmented to the existing fractional timing offsets.

As a final test, the experiment is repeated with the TX NUC precompensating the per-channel sounding sequence by the aggregate timing offsets (fractional and integer) and the per-channel phase offsets. At the RX REF, we can observe that all peaks appear in the same time sample and have the same phase. This is verified over several iterations, and we are done. We have just described the procedure to calibrate the *per-channel timing offset* and *per-channel phase offset* for the TX array. Calibration of the RX array is symmetrical, and has therefore been left to the imagination of the reader. Alternatively, the reader can pull the code from GitHub [3] and take a look at the gory details.

3.4 Magnitude Correction

What causes the problem? Multiple copies of the same module (like the HMC6300) will have manufacturing and packaging variations that lead to the conversion gain showing variances. There are further variances caused by temperature gradients on the transceiver board. Warmer modules will show lower gain than the cooler modules during steady state operation. Correct beamforming operation relies on deterministic gains or signal powers across these modules; these therefore have to be measured and calibrated out.

How do we calibrate? We describe the magnitude calibration of the TX array on NUC; the RX calibration is symmetrical. This procedure is rather simple. Transmit a wideband signal from each TX channel *txIndex*, one channel at a time (i.e., not simultaneously). Measure the receive power on REF, and divide it by the received power when $txIndex = 1$ was transmitting. Take the average of these correction factors across all RX channels on REF. The result is the magnitude correction factor for each TX channel on NUC. Run this experiment over several iterations, and take the average of the results for stability and correctness. The results from magnitude calibration (in an experiment where TX and RX magnitude calibration are jointly performed) is shown in Fig. 4; observe that within a few iterations, the TX and RX magnitude correction factors have stabilized.

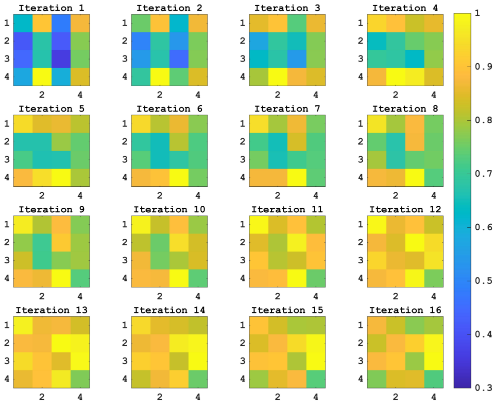

Figure 4: Calibrating the TX and RX magnitudes (digital gains). The rows and columns correspond to TX and RX indices. Magnitudes are normalized by the (TX, RX) pair with maximum power within that iteration.

3.5 IQ Calibration on the Receiver

What causes the problem? Consider a receiver that down-converts from RF to baseband. Without loss of generality, assume that the input RF signal is $cos(2\pi(f_c + f_m)t$, where f_c is the carrier frequency and f_m is the transmitted baseband signal frequency. An ideal receiver produces the complex baseband signal by mixing the received signal with a complex carrier at frequency f_c, and passing through a low pass filter (LPF). This can be written as:

$$i(t) = LPF(cos(2\pi f_c t)cos(2\pi(fc + f_m)t)) \tag{1}$$

$$q(t) = LPF(sin(2\pi f_c t)cos(2\pi(fc + f_m)t)) \tag{2}$$

This yields $i(t) = cos(2\pi f_m t)$ and $q(t) = sin(2\pi f_m t)$, as desired. However, a practical receiver will have the following imperfections: **a)** the conversion gain on the I channel and Q channel can differ by a factor of $\alpha \neq 1$; and **b)** the quadrature carrier might not be perfectly $\pi/2$ radian out of phase with the in-phase carrier, the phase error being $v > 0$ radian. Using standard trigonometric identities, the practically demodulated signals reduce to:

$$i'(t) = \alpha cos(2\pi f_m t) \tag{3}$$

$$q'(t) = sin(2\pi f_m t + v) \tag{4}$$

Note that the magnitude factor α and phase factor v were arbitrarily assigned to the i and q channels respectively, without any loss of generality. These imbalances lead to poor suppression of undesired sidebands, thereby driving up the error vector magnitudes (EVMs) in the receiver. Expanding equation (4) and writing in matrix notation, we get:

$$\begin{bmatrix} i'(t) \\ q'(t) \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ sin(v) & cos(v) \end{bmatrix} \begin{bmatrix} i(t) \\ q(t) \end{bmatrix} \tag{5}$$

Call this middle matrix M; the values of α and v determine how imbalanced the receiver is, and therefore the magnitude of the undesired sidebands. Overcoming these imbalances involves estimating and inverting this matrix M to recover the desired baseband signal.

Generating clean sinusoids: Some IQ calibration procedures rely on being able to generate clean sinusoids (at the mmWave frequencies) without sidebands or spurs; but how do we generate such signals? Lab bench synthesizers that can generate mmWave signals cost several hundreds of thousands of dollars, thereby making it out of reach for most. Another possibility is to configuring REF (the SDR other than the node under calibration, or NUC) at center frequency f_c and transmitting a modulated tone at f_m to generate a sinusoid at $f_c + f_m$; however, will lead to the transmitter-side IQ imbalances messing up the signal, and the calibration technique will be unable to distinguish between TX-side and RX-side imbalances. Our work-around is to use *Offset LO*.

As an example, say we are calibrating the RX IQ imbalances on the NUC at center frequency 58 GHz, and suppose we want f_m = -1 GHz. To generate this 57 GHz signal, we will configure REF at f_c = 56 GHz, and modulate a baseband signal at frequency +1 GHz. The transmitted signal will therefore have a desired sideband at 57 GHz, and a smaller undesired sideband at 55 GHz. At the receiver NUC, f_c = 58 GHz. The undesired sideband is 3 GHz down from center, and is therefore filtered away in the IF stage of down-conversion. Using this simple Offset LO technique, we can generate the required clean tones as seen by the receiver NUC.

How do we calibrate? Our technique is inspired by Ellingson's IQ calibration procedure [4]. Let us start with the α calibration on the RX channels of the NUC. The NUC is configured at center frequency 58 GHz, and is expecting to receive a clean sinusoid at 57 GHz, which is generated by the REF TX as explained in the previous paragraph. Capture the I and Q baseband waveforms on the NUC, and measure the power on them. Quite simply, α is the integrated energy on the I channel divided by the integrated energy on the Q channel. This measurement is repeated over several iterations and averaged. There is a further improvement, wherein the time-domain signals are passed through an FFT, and only the energy corresponding to the frequency bin of interest (-1 GHz) is used in averaging. The two methods provide results that are virtually indistinguishable. Once the α term is known, calibration simply involves dividing the real component of the received time-domain waveform by α, and leaving the imaginary component untouched.

Once the α imbalance is corrected, equation (5) reduces to:

$$\begin{bmatrix} i'(t) \\ q'(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ sin(v) & cos(v) \end{bmatrix} \begin{bmatrix} i(t) \\ q(t) \end{bmatrix} \tag{6}$$

Next, calibrate out the quadrature phase imbalances v. The phase imbalance matrix in the equation above needs to be inverted to recover I and Q from I' and Q' as shown:

$$\begin{bmatrix} i(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -tan(v) & sec(v) \end{bmatrix} \begin{bmatrix} i'(t) \\ q'(t) \end{bmatrix} \tag{7}$$

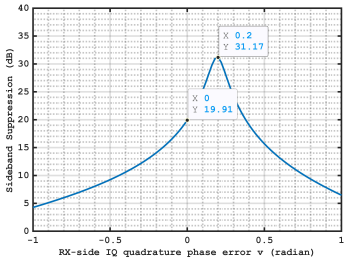

Figure 5: Estimating the quadrature phase offset on one RX channel on the NUC (one iteration). Observe that after v correction, the sideband suppression is improved by over 11dB.

This task boils down to estimating the value of v, applying equation (7), and we are done; but how do we estimate v? The idea is to take the received waveform, and apply corrections for a large number of v hypotheses in the range of [-1 1] radian. For each of the hypotheses, measure the sideband suppression (i.e., the received baseband power at −1 GHz divided by the received baseband power at +1 GHz). The v hypothesis that leads to the greatest suppression is the correct estimate. Average over a number of iterations for good performance and stability. The results from one such iteration is shown in Fig. 5. This graph plots the resulting sideband suppression as a function of different phase correction factors that have been applied. It is plain to see that the I and Q LO paths in the receiver are not $\pi/2$ out of phase, but instead have an additional phase error of 0.2 radian. Once this phase correction is applied (equation 7), the sideband suppression improves from 19.91 dB to 31.17 dB.

We have so just described how to estimate and correct the RX-side IQ imbalances: the magnitude correction factor α and the phase correction factor v. To finally show its overall performance, we run an experiment that does the following: **a)** capture samples at the RX, plot the spectrum, and show the uncalibrated sideband suppression; **b)** apply the α correction to the same signals; and **c)** apply the v correction to the same signals. Fig. 6 shows the sideband suppression getting better as the α and v corrections are applied. The columns correspond to RX channel 1, 2, and 3 (4 is not shown for space reasons). Despite the HMC6301 (mmWave receiver) being a heterodyne down-converter (as opposed to a direct converter), we observed that the typical v correction factors were rather high, in the range of -0.3 to +0.3 radian. Fig. 6 shows that correctly performing RX-side IQ calibration results in additional sideband suppression in the 10-13dB range.

3.6 IQ Calibration at the Transmitter

What causes the problem? An up-converting transmitter takes the complex baseband signal (say it is a single tone $e^{j2\pi f_m t}$) and mixes it with a complex carrier $e^{j2\pi f_c t}$. The real component of this result is then amplified and transmitted. However, the quadrature component of the carrier might not be phased at perfectly $\pi/2$ away from the in-phase component of the carrier; this might have a phase error of v radian. Further, the conversion gains on the I and Q channels might be differ by a factor of α. If the baseband signal is given by $i(t) = cos(2\pi f_m t)$ and $q(t) = sin(2\pi f_m t)$, the baseband equivalent of the transmitted signal in a practical transmitter is given by: $i'(t) = \alpha cos(2\pi f_m t)$ and $q'(t) = sin(2\pi f_m t + v)$;

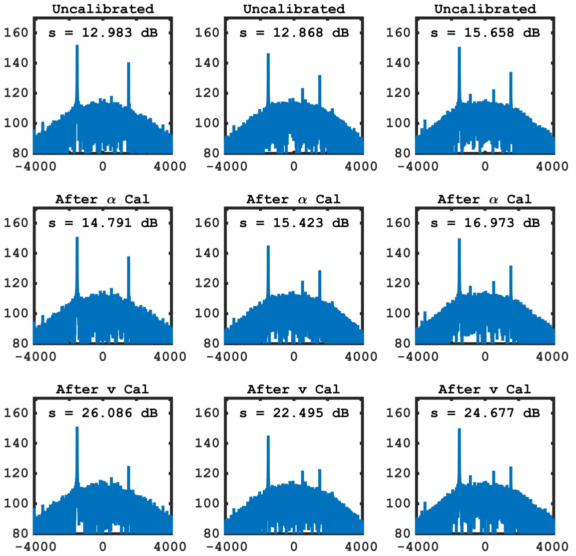

Figure 6: Applying the RX IQ calibration factors results in *additional* suppression of the unwanted sideband by 10-13dB (the total suppression is denoted by s). The columns represent RX channel 1, 2, and 3. The X-axis is the subcarrier index (frequency) and the Y-axis is the power in dB.

we have arbitrarily assigned the α and v factors to the I and Q channels respectively, without any loss of generality. Using standard trigonometric identities, this can be written as:

$$\begin{bmatrix} i'(t) \\ q'(t) \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ sin(v) & cos(v) \end{bmatrix} \begin{bmatrix} i(t) \\ q(t) \end{bmatrix} \tag{8}$$

How do we calibrate? The first step is to measure and calibrate out the α factor. Activate one TX channel at a time on the NUC, and transmit only the real component of a single tone baseband signal; measure the received power on the RX REF. Next, transmit only the imaginary component of the baseband signal, and measure the received power on the RX REF. The ratio of these two received power values, when averaged over all RX channels on REF, is the estimate of α. Average over multiple iterations to get stable calibration factors. Once the α calibration factors are known, applying them simply involves dividing the real component of the transmitted baseband waveform by α and leaving the imaginary component untouched. The next step is measuring and calibrating the v phase offsets in the TX chains.

After the α correction factors have been applied, equation (8) reduces to:

$$\begin{bmatrix} i'(t) \\ q'(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ sin(v) & cos(v) \end{bmatrix} \begin{bmatrix} i(t) \\ q(t) \end{bmatrix} \tag{9}$$

Overcoming the v imbalances involves *precoding* the transmitted waveform by the inverse of the central matrix in equation (9). This inverse is given by:

$$M^{-1} = \begin{bmatrix} 1 & 0 \\ -tan(v) & sec(v) \end{bmatrix} \tag{10}$$

What we have left to do is to estimate v for each of the four channels on the TX NUC. Let's say the TX NUC is operating at f_c = 58 GHz. The intuition behind the technique is the following: **a)** transmit a single tone (at f_m = +1 GHz) from the TX NUC, leading to the desired (upper) sideband at 59 GHz, and the undesired (lower) sideband at 57 GHz; **b)** the RX REF is operating at an offset LO of f_c = 56 GHz, and is capable of measuring the power in only the undesired lower sideband at 57 GHz; **c)** sequentially cycle through and apply the precoding matrices for every v hypothesis on the TX NUC as per equation (10); **d)** use the RX REF to determine which v precoding led to the lowest undesired sideband; and **e)** since v is the only variable, this must correspond to the *best* v correction on the TX NUC.

The *Offset LO* method ensures that the RX REF sees only the transmitted lower sideband at 57 GHz, but cannot see the transmitted upper sideband at 59 GHz (this gets filtered out in the IF stage). The procedure above is repeated for every $txIndex \in [1, 2, 3, 4]$. In each run, the power in the undesired lower sideband is measured by REF across all v hypotheses and all RX channels on REF. The optimal v value is estimated using an MMSE error metric across all v hypotheses and all 4 RX channels on REF. These v estimates should be averaged over a few iterations for stability.

To examine the TX-side IQ v calibration process, we ran a simplified experiment where only one RX channel on REF was used (otherwise the data is too much to plot in this paper). Each TX channel sequentially cycles through multiple v hypotheses in the range of [-1, 1] radian; in each case, the RX REF measures the power in the undesired lower sideband at 57 GHz. This has been plotted in Fig. 7; the four graphs correspond to different TX channels $txIndex$ on the NUC. Consider $txIndex$ = 1; the optimal v correction is 0.46 radian, showing that the hardware imperfections can be *really large*. If this correction had not been applied, this undesired sideband would be larger by a 8.567 dB. Channel $txIndex$ = 3 has a small IQ v error of 0.12 radian, and the additional sideband suppression offered by v calibration is small (0.8663 dB). Channel $txIndex$ = 4 has a moderate IQ v correction of 0.26 radian, but the benefit of applying the calibration is quite large (11.59 dB).

Why are the graphs in Fig. 7 noisy? The goal of the *Offset LO* method is to ensure that the RX REF sees only the transmitted lower sideband, with the transmitted upper sideband being filtered away. We therefore want the f_c at the TX NUC and the RX REF to be far apart. The tone corresponding to the transmitted lower sideband (57 GHz) is near the left edge of the transition band of the transmitter IF filter; similarly, this tone is also near the right edge of the transition band of the receiver IF filter. This observation, when combined with the fact that the lower sideband is not the desired sideband at the TX NUC, makes the signal very weak at the RX REF. These noise limited signals make the measurements in Fig. 7 appear noisy.

Summary of IQ Calibration: Despite the HMC6300 mmWave up-converter and the HMC6301 mmWave down-converter having heterodyne architectures (as opposed to direct conversion), the IQ imbalances are quite significant. This can lead to large unwanted sidebands, thereby lowering the EVM of the signals. We have observed that applying the IQ corrections can lead to additional suppression of up to 15 dB, thereby demonstrating the importance of performing these calibrations carefully.

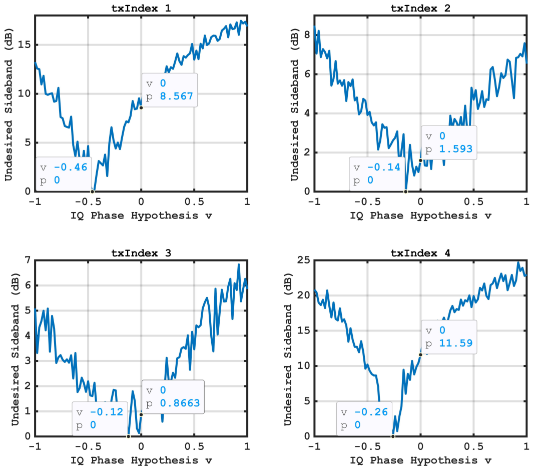

Figure 7: TX IQ calibration: Each TX cycles through and precodes using multiple v hypotheses; in each case, the power in the undesired (lower) sideband is measured by the RX REF. Observe that the v corrections can be quite large (as high as 0.46 radian), and correcting the v imbalance can yield additional sideband suppression of up to 11dB. The data marker v is the IQ phase hypothesis, and p refers to the power in the undesired sideband.

3.7 What can go wrong with these Calibration techniques?

First, suppose the NUC placement was mis-aligned, say by 3^o. The measured array phase factors will be erroneous, and all resulting beams will be offset by 3^o. In our system, the beams are fairly fat (the HPBW is about 30^o); mis-alignment therefore does not hurt us much. Still, we have built a simple rectangular harness (using laser-cut acrylic) that ensures that the nodes are perfectly in boresight for calibration. We note that the timing offset, magnitude, and IQ calibration factors remain unaffected by any mis-alignment.

Second, power-cycling the system can lead to DAC/ADC alignment and LO starting phase uncertainty, thereby messing up the calibration. We have designed and implemented self-calibration techniques that can be run upon power-up to detect and correct these variations. Running these self-cal techniques does not require the nodes to be placed in boresight; they can be run wherever they are deployed without any dependencies on other nodes. We do not discuss these techniques further because of space constraints.

Third, a question can be asked: *how close to ideal (i.e., by using expensive lab-bench equipment) do our calibration techniques get?* The short answer is that we do not know yet, and this will be investigated in future work.

Fourth, a question can be asked: *will these techniques scale to when there are 128 channels?* The RX-side calibration techniques will work using the same running-time as a 4-channel system because there is no SINR penalty when the number of RX channels is increased. The TX-side techniques might need to be divided into sets of 4 channels at a time to prevent SINR degradation; the time needed to calibrate will therefore scale linearly. However, by implementing these directly on the FPGA (as opposed to the current method that uses MATLAB), we will get orders of magnitude improvement in execution speed.

Summary of Calibration: The suite of TX and RX array calibrations (fractional + integer timing offsets, LO phase offsets, and magnitude calibration) takes about 30 minutes to run. This process calibrates both nodes, since they take turns behaving as the NUC and the REF. These calibrations take a long time to run because of the discontinuous nature of the MATLAB drivers; shipping a new waveform to the FPGA, or triggering and receiving the waveform in MATLAB takes about $1s$ to run. Given that the calibration process makes several hundreds of such iterations, the process takes time. Of course, we can massively speed it up by running the calibration directly on the FPGA; this is planned in future work. The IQ calibrations takes about an hour to run, the time being dominated by the sheer number of waveform writes and reads needed by the TX IQ v calibration. Once all the procedures are run, the calibration factors are written to a file, and these can be loaded upon startup. However, we realized that there are random starting LO phase variations across power cycles; re-running the calibration routines upon every power cycle is infeasible. We therefore designed and implemented a self-calibration routine that can be run every power cycle; for space reasons, this is not described further.

4 BEAMFORMING DEMONSTRATION

The ultimate test of whether an array has been calibrated properly is if we can correctly form beams in the required directions. To do this, we first apply all the required calibration factors to the signal to/from each TX/RX channel. To transmit a beam in direction θ (where $\theta = 0$ is boresight), the baseband signal for TX channel $txIndex$ should be multiplied by $e^{j\cdot\pi\cdot txIndex\cdot sin(\theta)}$; this is classical geometric beamforming in the case when the antenna elements are spaced $\lambda/2$ apart. Conversely on the receiver side, to receive a signal in direction θ (where $\theta = 0$ is boresight), the baseband signal from each RX channel $rxIndex$ needs to be multiplied by $e^{j\cdot\pi\cdot rxIndex\cdot sin(\theta)}$ prior to being combined. To demonstrate beamforming on two nodes (A,B), the experiment involves:

- Transmit from a single channel on A. Receive on all RX channels of B. Apply multiple beamforming vectors to *look* in all directions;
- Transmit from all channels on A. Sequentially, apply beamforing vectors to scan the transmit beams in different directions; in each case, use a single RX channel on B, and measure the power;
- Transmit from a single channel on B. Receive on all RX channels of A. Apply multiple beamforming vectors to *look* in all directions;
- Transmit from all channels on B. Sequentially, apply beamforing vectors to scan the transmit beams in different directions; in each case, use a single RX channel on A, and measure the power;

Essentially, this demonstrates RX and TX beamforming on both nodes in the link, along with the estimation of the angle of arrival (AoA) and angle of departure (AoD). We first place both nodes (named `trx-0002` and `trx-0003`) directly facing each other and ran the experiment. Fig. 8 shows the TX and RX beam patterns, showing that the AoA and AoD are estimated correctly. Next, we

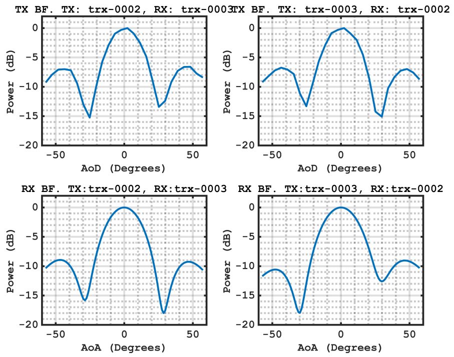

Figure 8: When the two nodes are placed in boresight, the AoA and AoD values are correctly estimated to be 0 on both ends.

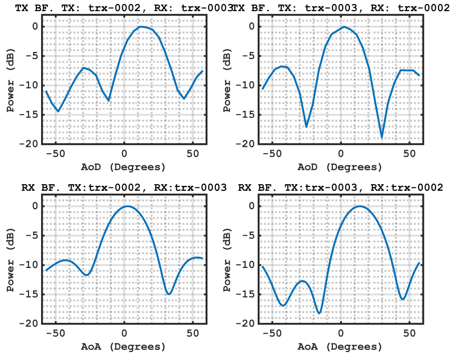

Figure 9: Beamforming demonstration when node `trx-0002` is rotated counter-clockwise by about 15 degrees.

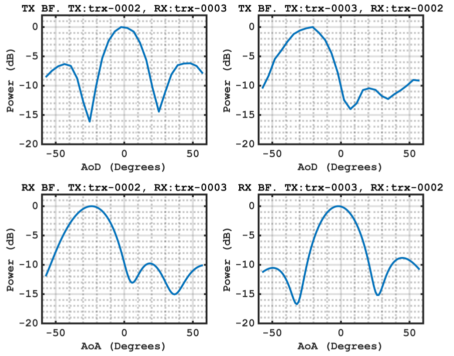

Figure 10: Beamforming demonstration when node `trx-0003` is rotated clockwise by about 25 degrees.

rotated node `trx-0002` counter clockwise by about 15 degrees, and repeated the experiment; the results are shown in Fig. 9, showing that the correct AoA and AoD have been detected. Finally, we returned `trx-0002` to the original position, but rotated `trx-0003` clockwise by about 25 degrees; the results are shown in Fig. 10; again, the correct AoA and AoD were detected. These experiments demonstrate that the node calibration has been performed correctly, as evidenced by successful beamforming and estimation of AoA and AoD values.

5 CONCLUSION

Getting the Pi-Radio v1 SDR to work correctly needed careful calibration. There are many ways in which the behavior of practical devices deviates from ideal, and these need to be calibrated out in order to get beams: **a)** crystal frequency offset correction; **b)** identification of linear operating ranges; **c)** timing offset corrections; **d)** LO phase offset corrections; **e)** magnitude corrections; **f)** IQ gain imbalance corrections; and **g)** IQ quadrature LO phase imbalance corrections. The goal of this paper is to be a tutorial in how to carefully calibrate an SDR node. Performing calibration has hitherto relied on expensive lab bench equipment like spectrum analyzers and signal synthesizers; in the mmWave frequency range, these are painfully expensive. This paper aims to demonstrate one set of techniques that can be used to calibrate SDRs *in an affordable manner*. We do not claim scientific novelty; the techniques described are well-known or simple enough to be obvious. All calibration code (as well as Pi-Radio's v1 SDR hardware design schematics) have been released on GitHub [3] using the free and highly permissive MIT license; anybody can use it in any way that they choose. It is our vision to democratize access to experimental wireless research not only through affordable SDRs that feature advanced transceiver technologies (like fully-digital), but also through open sourcing calibration code that can be used by the community either on the Pi-Radio or any other SDR platform.

ACKNOWLEDGEMENTS

The Pi-Radio v1 SDR hardware was designed, built, and tested (along with early calibration efforts) during the performance period of NSF STTR Phase-I Award #1821150, *A Fully-Digital Transceiver Design for mmWave Communications* (Phase-II pending). More advanced calibration techniques were developed and implemented during the performance period of the ARMY STTR Phase-I Award, *Millimeter Waveforms For Tactical Networking* #W911NF20P0038. An early version of the SDR (based on a different architecture) was developed at New York University, partly funded by a NIST grant, *An End-to-End Research Platform for Public Safety Communications above 6 GHz* (Award # 70NANB17H166).

We would like to thank Prof. Arjuna Madanayake from Florida International University, Prof. Jim Buckwalter, Prof. Mark Rodwell, and Prof. Upamanyu Madhow from the University of California - Santa Barbara for enjoyable discussions about calibration. We also would like to thank Dr. Vasilii Semkin (UC Louvain, Belgium) for his help with antenna design. Finally, we would like to thank the NSF RCN community for their energy and vocal demands for increased access to affordable real-world mmWave experimentation.

REFERENCES

[1] R. Zhao, T. Woodford, T. Wei, K. Qian, and X. Zhang, "M-Cube: A Millimeter-Wave Massive MIMO Software Radio." Mobicom 2020.
[2] "Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit." [Online]. Available: https://www.xilinx.com/products/boards-and-kits/zcu111.html
[3] "Fully-digital SDR Repo." [Online]. Available: https://github.com/adityadhananjay/fully-digital
[4] S. W. Ellingson, "Correcting I-Q Imlabance in Direct Conversion Receivers." [Online]. Available: https://www.faculty.ece.vt.edu/swe/argus/iqbal.pdf