gr:harmonia: A GNU Radio-Based Tool for Synchronization of Software-Defined Radios in a Distributed Radar Network

Cody Kieu CODYKIEU2002@GMAIL.COM
School of Electrical and Computer Engineering, Advanced Radar Research Center, University of Oklahoma

Justin Metcalf JMETCALF@OU.EDU
School of Electrical and Computer Engineering, Advanced Radar Research Center, University of Oklahoma

Russell Kenney KENNEY@OU.EDU
School of Electrical and Computer Engineering, Advanced Radar Research Center, University of Oklahoma

Abstract

Precise synchronization of time, frequency, and phase is crucial for coherent processing in distributed radar systems, especially at higher frequencies where timing errors rapidly degrade performance. To enable distributed radar operations such as coherent transmit beamforming, distributed radar systems must achieve time synchronization on the order of tens of picoseconds, requiring extraordinarily robust techniques for correcting timing errors. The `gr-harmonia` out-of-tree module implements a synchronization algorithm to achieve time, frequency, and phase alignment across distributed radar networks using UHD-compatible software-defined radio (SDR) systems. The algorithm is realized entirely in signal processing and thus requires no additional hardware, allowing it to be implemented in networks of unmodified SDRs. Because the synchronization approach is decentralized, it scales efficiently with an arbitrary number of nodes to be synchronized. The module operates exclusively through the GNU Radio message passing interface, converting pulsed data into a protocol data unit (PDU) to reduce buffering-induced latency. The module utilizes UHD burst mode with metadata flags to schedule pulses with accurate timing. It comprises blocks for waveform generation, transmission and reception of waveforms, frequency and time peak estimation, clock drift estimation, clock bias, and phase estimation. In this paper, the framework and functionality of each block are discussed, and the results of the synchronization process are shown for data collected from an open-air test with three Ettus X310s using `gr-harmonia`.

Proceedings of the 1st GNU Radio Conference, Copyright 2025 by the author(s).

1. Introduction

Distributed radar systems are a developing area of research due to the benefits they offer in comparison to traditional monostatic radars. Some of the benefits of implementing these systems include lower overall costs and the elimination of a single point of failure, while still providing results similar to those of conventional monostatic radars (Comberiate et al., 2016; Merlo & Nanzer, 2022). However, as described in numerous projects such as Shandi (2024) and Prager (2020), synchronization of software-defined radios (SDRs) in a distributed radar network is necessary to ensure coherent integration of data between each radar node. The synchronization algorithm considered in this paper draws inspiration from these projects while being implemented entirely in the signal processing domain. In doing so, this will eliminate the need for additional hardware, with the exception of antennas and software-defined radars.

One main challenge with synchronizing software-defined radars is the lack of access to blocks that aid in the synchronization process. To address this issue, SDRs must have the capability to perform radar functions, such as transmitting electromagnetic waves and processing the reflected wave to measure metrics (e.g., frequency and range). The radar signal processing and formation of the module's blocks in this paper were greatly inspired by `gr-plasma`, a radar signal processing module. Some of the blocks from this radar module were used for the testing of this project. The module `gr-harmonia` aims to provide the necessary toolkit for pulsed waveform generation, radio frequency (RF) estimation, and RF offset compensation.

`gr-harmonia` provides the necessary blocks for an all-digital synchronization of a distributed radar system through GNU Radio's message passing interface. Using

the message domain has the benefit of precisely timed burst data, which is useful for forming pulsed waveforms for radar applications (Flandermeyer et al., 2022). The received data can then be packed into protocol data units (PDUs) to help with downstream signal processing across all blocks. All blocks created for this module are implemented through the message domain for uniformity and ease of use.

The paper discusses the synchronization algorithm and its implementation in `gr-harmonia`. First, in Section 2, the assumptions and algorithmic operation are described in detail. Then Section 3 describes the signal processing chain implemented within `gr-harmonia` blocks. Section 4 outlines the experimental design, followed by Section 5, which presents the results. Lastly, Section 6 discusses the conclusions and future work for `gr-harmonia`.

2. Synchronization Algorithm

The synchronization algorithm begins by correcting for clock drift, which affects the time drift and carrier frequencies of the radars. Then, range and clock bias errors are estimated and compensated using pulse compression waveform exchanges. Finally, the carrier phases of the radars are estimated and corrected.

2.1. Radar Signal Modeling and Assumptions

The signal model used in this work follows the formulation presented in (Kenney, 2024). Several assumptions are made to simplify the signal modeling of a distributed radar network. First, it is assumed that the network consists of spatially distributed radar sensors, each equipped with an independent main oscillator (MO). It can then be assumed that each MO is operating at the same nominal frequency. Each platform's MO serves as a reference for deriving both the digital sampling clocks and the local oscillator (LO). This allows for any frequency errors in a platform's MO to result in a proportional offset in all timing- and frequency-dependent operations.

The nominal frequency of each MO is given by f_{MO}. Due to the imperfect nature of crystal oscillators, the true frequency of the MO on platform i will deviate slightly from the nominal frequency, where the true frequency can be denoted by $f_{\mathrm{MO},i}$. A system's *clock drift* is defined as the ratio of the true and nominal frequencies given by

$$\alpha_i \triangleq \frac{f_{\mathrm{MO},i}}{f_{\mathrm{MO}}}. \tag{1}$$

Additionally, each platform is assumed to have a random but constant *clock bias*, denoted by ϕ_i. This bias can be attributed to multiple factors, such as the differing initialization time of each platform and transmission chain delays. Given the global time (t), clock drift, and clock bias, the local time of platform i can be expressed by

$$\tau_i = \alpha_i t + \phi_i. \tag{2}$$

Rearranging Eq. 2 to solve for global time for platform i gives

$$t = \frac{\tau_i - \phi_i}{\alpha_i}. \tag{3}$$

A few additional assumptions are made to simplify the signal modeling framework. Each platform should transmit with a nominal carrier frequency of f_c and operate with a nominal sampling frequency f_s for both the digital-to-analog (DAC) and analog-to-digital (ADC) conversions. Moreover, it is assumed that the transmitter and receiver have independent and random phase offset terms, which are given by γ_i^{tx} for the transmitter and γ_i^{rx} for the receiver.

For this project, it is also assumed that the radar platforms are stationary during the synchronization process, removing the need to compensate for platform motion. Let $s_j(t)$ be an arbitrary complex baseband waveform transmitted by platform i. Denote the distance from platform i to platform j at time t as $R_{i,j}$. Since all platforms are stationary, it can be said that $R_{i,j} = R_{j,i}$. Dividing the distance by the speed of light (c) gives the signal time-of-flight (TOF) given by

$$TOF_{i,j} = \frac{R_{i,j}}{c}. \tag{4}$$

Then, the difference between the random transmit carrier phase of platform j and the random receive carrier phase i is given by

$$\gamma_{i,j}^{\mathrm{err}} = \gamma_j^{\mathrm{tx}} - \gamma_i^{\mathrm{rx}}. \tag{5}$$

The signal received by platform i from the transmitted signal of platform j in the global time will be expressed by

$$\begin{aligned} r_{i,j}(t) =& s_j\left(\alpha_j t + \phi_j - \alpha_j \frac{R_{i,j}}{c}\right) \\ &\cdot \exp\left(j2\pi f_c \left((\alpha_j - \alpha_i)t + \phi_j - \phi_i - \alpha_j \frac{R_{i,j}}{c}\right)\right) \\ &\cdot \exp\left(j\gamma_{i,j}^{\mathrm{err}}\right) \end{aligned} \tag{6}$$

Then substituting Eq. 3 into Eq. 6, the received time of the transmitted signal of platform j observed in the local clock frame of platform i is shown by

$$\begin{aligned} r_{i,j}(\tau_i) = s_j\left(\frac{\alpha_j}{\alpha_i}\tau_i - \rho\right) \exp\left(j2\pi f_c \left(\left(\frac{\alpha_j}{\alpha_i}\right)\tau_i\right)\right) \\ \cdot \exp(j2\pi f_c \rho) \exp\left(j\gamma_{i,j}^{\mathrm{err}}\right), \end{aligned} \tag{7}$$

where ρ is a substitution variable defined by

$$\rho = \frac{\alpha_j}{\alpha_i}\phi_i - \phi_j + \alpha_j \frac{R_{i,j}}{c}. \tag{8}$$

Equation (7) illustrates the received signal impacted by clock drift, clock bias, and carrier phase. The full derivation of the signal model and the moving platform case can be found in (Kenney, 2024; Kenney et al., 2025).

2.2. Time-Division Multiple Access

Multiplexing of synchronization signals is accomplished with time-division multiple access (TDMA). It is a channel access method used in wireless and radar systems to enable multiple devices to share a common communication channel without interference between nodes. It works by allocating distinct, non-overlapping time intervals (or slots) to each transmitter, allowing them to operate in a coordinated manner. This technique is particularly effective in distributed radar systems where frequency reuse and coordination are essential.

In a TDMA system, time is divided into fixed-length slots of duration Δ_{TDMA}, with each slot assigned to a different node in the network (Prager et al., 2020; Merlo et al., 2024a). Suppose there are N total nodes in the system. Each node is assigned a unique integer index 1 to N, which determines its designated transmit time slot. If the nodes were to communicate in ascending order according to their assigned integer values, then the transmit time for node j, relative to its own clock time, is given by

$$\tau_j^{\text{tx}} = (j-1)\Delta_{\text{TDMA}}, \quad j = 1, 2, \ldots, N, \tag{9}$$

The TDMA schedule is known throughout all sensors in the network, allowing each sensor to infer the source of any signal received within the scheduled time.

2.3. Estimation of Clock Drift

To begin the synchronization algorithm derivation from (Kenney, 2024), each platform is designated the same single-tone waveform at baseband. Platform j transmits a single tone at a baseband frequency f_j^α with a pulse duration of T_p, and its transmit time is determined by (9). The single-tone baseband transmit signal, expressed in its local time τ_j, is defined as

$$s_j^\alpha(\tau_j) = \exp\left(j2\pi f_j^\alpha \tau_j\right) \cdot \text{rect}\left(\frac{\tau_j - \frac{T_p}{2}}{T_p}\right), \tag{10}$$

where the rectangular function is defined by

$$\text{rect}(t) \triangleq \begin{cases} 1, & |t| < \frac{1}{2} \\ 0, & \text{else}\,. \end{cases} \tag{11}$$

Then the received waveform at platform i can be shown by

$$\begin{aligned} r_{i,j}^\alpha(\tau_i) = \exp&\left(j2\pi\left[\left(f^c\frac{\alpha_j}{\alpha_i} - 1\right) + \frac{\alpha_j}{\alpha_i}f^{\alpha_j}\right]\tau_i\right) \\ &\cdot \exp\left(-j2\pi\left(f^c + f^{\alpha_j}\right)\rho\right) \\ &\cdot \text{rect}\left(\frac{\frac{\alpha_j}{\alpha_i}\tau_i - \rho - \frac{T_p}{2}}{T_p}\right) \\ &\cdot \exp\left(j\gamma_{i,j}^{\text{err}}\right) \end{aligned} \tag{12}$$

Once receiver i captures the single-tone waveform, the observed frequency can be estimated by computing the Fourier transform of the received signal, as modeled by

$$\begin{aligned} R_{i,j}^\alpha(f) = \frac{\alpha_i}{\alpha_j}T_p\,\text{sinc}&\left(\frac{\alpha_i}{\alpha_j}T_p(f - f_{i,j}^\alpha)\right) \\ &\cdot \exp\left[-j2\pi(f - f_{i,j}^\alpha)\left(\frac{\alpha_i}{\alpha_j}\rho + \frac{\alpha_i}{\alpha_j}\frac{T_p}{2}\right)\right] \\ &\cdot \exp\left[-j2\pi(f_c + f_j^\alpha)\rho\right] \\ &\cdot \exp\left(j\gamma_{i,j}^{\text{err}}\right). \end{aligned} \tag{13}$$

where the normalized sinc function is defined as

$$\text{sinc}(x) \triangleq \frac{\sin(\pi x)}{\pi x}, \tag{14}$$

and the observed frequency of the single tone sent from platform j to i is given by

$$f_{i,j}^\alpha = \left(\frac{\alpha_j}{\alpha_i} - 1\right) f_c + \frac{\alpha_j}{\alpha_i} f_j^\alpha. \tag{15}$$

The peak of the magnitude of the Fourier transform $|R_{i,j}^\alpha(f)|$ occurs at $f = f_{i,j}^\alpha$, which allows platform i to estimate $\hat{f}_{i,j}^\alpha$ by simply finding the global maximum of the received signal.

Once all platforms transmit their single-tone signals, platform i has all frequency estimates, $\hat{f}_{i,j}^\alpha$, for each platform such that $j \neq i$. Rearranging (15) yields a linear equation of the unknown clock drifts:

$$\left(\hat{f}_{i,j}^\alpha + f_c\right)\alpha_i - \left(f_j^\alpha + f_c\right)\alpha_j = 0. \tag{16}$$

Since the clock drift values cannot be resolved without an external reference, one platform's drift (e.g., α_1) is conventionally set to unity

$$\alpha_1 = 1. \tag{17}$$

This gives $N_p - 1$ independent equations with N_p unknowns, resulting in a fully determined system that can be solved. This method is called the *single-tone without exchange technique*. To improve the robustness of the system, platforms can also exchange their frequency estimates

$\hat{f}_{i,j}^{\alpha}$ with one another. With this additional information, a system of $N_p^2 - N_p + 1$ equations is formed, which allows for a more accurate solution using a least-squares approach. This method is referred to as *single-tone with exchange* and is beneficial in noisy environments. The algorithm in this module will be implemented using this method.

Additionally, to estimate frequency offsets with high precision, the sinc-based nonlinear least squares (NL-LS) algorithm is employed to improve the accuracy of the frequency estimates beyond the resolution of the discrete Fourier transform (DFT). Sinc NL-LS implements a sinc-based kernel to best-fit the received data using nonlinear least-squares (Prager et al., 2020; Kenney et al., 2024).

After using sinc NL-LS to estimate the frequency of each received signal, clock drift can be estimated by solving the set of linear equations. The general system of linear equations can be represented as

$$\mathbf{Ax} = \mathbf{b}, \tag{18}$$

where A is an $M_b \times M_x$ matrix of coefficients of the system of equations given from Eq. 16 and that relates vector **x** to vector **b**. The vector **x** has M_x values to be solved for, and **b** is an $M_b \times 1$ solution vector containing the values for the solution of each equation. Assuming $M_b > M_x$, the solution to the system of equations is ordinary least-squares (OLS), which uses the Moore-Penrose pseudo-inverse matrix. The solution is computed by

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{19}$$

It is assumed $\mathbf{b}$ is corrupted by additive noise with covariance $\mathbf{C_b}$. If the variance of each component of $\mathbf{b}$ is equivalent, then OLS can be used. Otherwise, if the variance of each component is different, then weighted least-squares (WLS) can be used to improve estimation accuracy; however, for simplicity, OLS estimation is used for the implementation of this algorithm.

After the clock drift estimation procedure is completed, each platform i has an estimate of its clock drift, denoted by $\hat{\alpha}_i$. With this information, each platform can compensate for its clock drift by modifying the time scaling of its transmitted and received signals to align them with a common reference frame, achieving frequency synchronization. It is important to note that Doppler shifts resulting from platform motion can impact frequency synchronization (Merlo et al., 2024b); however, it is ignored in this paper since all radar platforms are stationary.

Starting with the local time-domain transmit signal from platform j, the time dilation due to clock drift α_j and corresponding carrier shift can be compensated by first reconstructing the baseband signal using the corrected sampling rate. The corrected baseband waveform $s'_j(\tau_j)$ is given by

$$s'_j(\tau_j) = s_j\left(\frac{\tau_j}{\hat{\alpha}_j}\right) \cdot \exp\left[j2\pi f_c\left(\frac{1}{\hat{\alpha}_j} - 1\right)\tau_j\right], \tag{20}$$

for which the corrected receive time axis can be defined as

$$\tau'_i = \frac{\tau_i}{\hat{\alpha}_i}. \tag{21}$$

To correct for the carrier frequency offset, signal $b_i(\tau'_i)$ must be multiplied to the received signal. The carrier compensation signal is defined as

$$b_i(\tau'_i) = \exp\left[j2\pi f_c(\hat{\alpha}_i - 1)\tau'_i\right]. \tag{22}$$

This results in waveforms exchanged between nodes being free of time dilation and carrier frequency offset, enabling coherent operation among the distributed platforms.

2.4. Estimation of Clock Bias and Carrier Phase

To achieve full synchronization, the unknown time and phase shifts caused by clock biases and RF transmitter and receiver phase shifts must be accounted for. The procedure for estimating clock bias and carrier phase is given in (Prager et al., 2020; Kenney, 2024). It is assumed that the radar systems are all synchronized in frequency. Additionally, since the platforms are stationary for testing, the ranges between platforms are symmetric $R_{i,j} = R_{j,i}$.

The synchronization of clock bias and carrier phase begins with each platform broadcasting a pulse-compression waveform, which can be used for time-delay estimation. In this paper, the pulse-compression waveform is a linear frequency modulation (LFM) pulse defined by

$$x_p(t) = a(t)\exp\left[j\pi\left(-\beta t + \frac{\beta}{T_p}t^2\right)\right], \tag{23}$$

where $a(t)$ is the time-varying amplitude of the signal defined by (11) and β is the signal bandwidth.

Pulse compression, also known as match filtering, provides the benefit of maximizing the signal-to-noise ratio (SNR) of the input signal. The matched filter response is built by taking the time-reversed conjugate of the transmitted signal, given by

$$h(t) = x_p^*(-t). \tag{24}$$

The pulse-compressed signal $y(t)$ may then be obtained by the convolution of the received signal $x(t)$ with $h(t)$.

Now that the LFM waveform and matched filtering are laid out, the process of estimating clock bias and carrier phase can be discussed. After each platform j transmits an LFM, all other platforms compute the pulse compression of

the received signal $r^b_{i,j}(\tau'_i)$ with the known baseband LFM waveform $s_j(\tau'_i)$. This then gives each platform the cross-correlation signal $d_{i,j}(\tau'_i)$, from which each platform can compute the peak value of the cross-correlation $m_{i,j}$. However, due to the resolution of the DFT, the time-domain sinc NL-LS is used to further improve the time estimation of the peak value of the $d_{i,j}$.

The delay values $m_{i,j}$ determined by platform i for all platforms j where $j \neq i$ can be expressed by

$$m_{i,j} = \frac{\rho}{\hat{\alpha}_j} = \frac{1}{\hat{\alpha}_i}\phi_i - \frac{1}{\hat{\alpha}_j}\phi_j + \frac{\alpha_j}{\hat{\alpha}_j}\frac{R_{i,j}}{c}. \tag{25}$$

Additionally, to help compute carrier phase offsets, the phase of the cross-correlational signal at the peak value $\angle d_{i,j}(m_{i,j})$ should be calculated for each platform. The phase of the peak value is then given by

$$\angle d_{i,j}(m_{i,j}) = \gamma^{err}_{i,j} - 2\pi f_c m_{i,j}. \tag{26}$$

Then all platforms exchange their values of $m_{i,j}$ and $\angle d_{i,j}(m_{i,j})$ for all (i,j) pairs where $j \neq i$. Once all values are received, every platform should have the same knowledge of $m_{i,j}$ and $\angle d_{i,j}(m_{i,j})$. Range estimates $\hat{R}_{i,j}$ can then be calculated by

$$\hat{R}_{i,j} = \frac{\alpha_j}{\hat{\alpha}_j}R_{i,j} = \frac{c(m_{i,j} + m_{j,i})}{2}, \tag{27}$$

and an estimate of the clock bias differences can be computed by

$$\hat{\phi}_{i,j} = \frac{1}{\hat{\alpha}_i}\phi_i - \frac{1}{\hat{\alpha}_j}\phi_j = \frac{m_{i,j} - m_{j,i}}{2}. \tag{28}$$

2.5. Clock Bias and Carrier Phase Compensation

After estimating clock bias differences, the clock bias of each platform can be adjusted to align with the network mean. Using the clock bias differences, platform i computes a clock bias correction as

$$\tilde{\phi}_i = \frac{1}{N_p}\sum_j \hat{\phi}_{i,j}. \tag{29}$$

Using the estimated clock bias, platform j can then correct its transmit signal by applying the fractional delay $\tilde{\phi}_j$ and a phase correction of $\exp(-j2\pi f^c \tilde{\phi}_j)$. Then, on the receive for platform i, the signal can be corrected by applying a fractional delay of $-\tilde{\phi}_i$ and a phase correction of $\exp(j2\pi f^c \tilde{\phi}_i)$. This results in clock bias being compensated for.

For the compensation of the carrier phase, the carrier phase estimates for all pairs of i and j where $i \neq j$ may be calculated by

$$\hat{\gamma}^{err}_{i,j} = \angle d_{i,j}(m_{i,j}) + 2\pi f^c m_{i,j}. \tag{30}$$

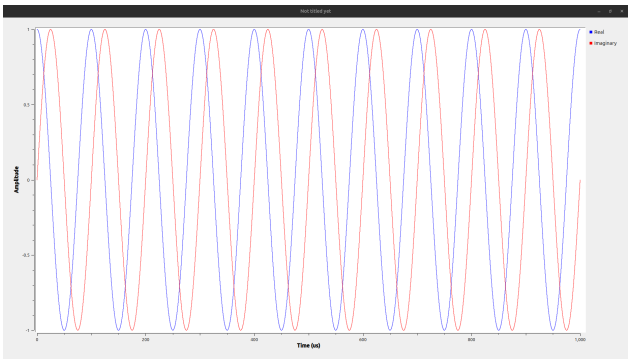

Figure 1. A single-tone waveform.

Then, using the carrier phase error estimates, the transmit carrier phase γ^{tx} and receive carrier phase γ^{rx} can be solved in a $N_p^2 - N_p + 1$ system where one equation is $\gamma^{\text{tx}}_1 = 0$ and the rest of the linear equations are described by (5). Using the carrier phase estimates, each platform i can compensate for carrier phase offsets by applying a phase shift of $-\gamma^{\text{tx}}_i$ to all waveforms transmitted by itself. Afterwards, on all received signals on platform i, a phase shift of γ^{rx}_j can be applied.

3. Blocks and Features

The `gr-harmonia` module is organized into a sequence of blocks that follow the synchronization process from transmission to compensation. This section describes the signal processing operations and metadata handled by each block. The blocks are separated into five categories: waveform generation, USRP interfacing and control, clock drift estimation, clock bias and carrier phase estimation, and waveform compensation.

3.1. Waveform Generation

The two radar waveforms implemented in `gr-harmonia` are the single-tone waveform and linear frequency-modulated waveform (LFM). The single-tone block in `gr-harmonia` implements this waveform using (20). The time-domain output of the single-tone block is shown in Figure 1 where the waveform has $f^{\alpha} = 10\ kHz$ and $T_p = 1\ ms$.

The LFM block generates an LFM at complex baseband given by (23). The time-domain representation of the waveform is shown in Figure 2 where the waveform has $\beta = 25$ MHz and $T_p = 1$ ms. Additionally, the LFM block includes three input ports to take in estimates of clock drift, clock bias, and carrier phase, and compensate for the transmit offsets.

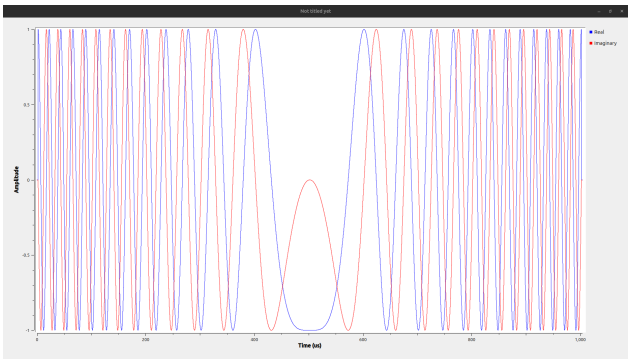

Figure 2. An LFM waveform.

3.2. USRP Interfacing and Control

Since this module primarily serves as a test bench for evaluating the synchronization algorithm, the data from each SDR was processed on a single host computer. In a practical setup, each SDR would typically be connected to a separate host. As a result, `gr-harmonia` uses a single custom block to handle all the signal exchanges between each radar node, eliminating the need for any communication waveforms.

The UHD block operates by processing waveforms in the form of protocol data units (PDUs). Each input waveform is accompanied by metadata indicating which estimates were received (e.g. clock drift, clock bias). These corrections are then applied to both the transmit and receive times, and then the block computes the necessary fractional time delay needed to adjust each transmit waveform accordingly. Based on this metadata, the block automatically advances to the next synchronization step.

Furthermore, the block provides TDMA control, enabling the user to configure the time offset between transmissions of the SDRs. Depending on the receive time and capture time, the block captures the received data and packages it into a PDU. It then calculates the fractional delay required for proper alignment in the received waveform and tags this value as metadata along with the corresponding SDR that transmitted the signal.

3.3. Clock Drift Estimation

Clock drift estimation is divided into two components: the Peak Frequency Estimation block and the Clock Drift Estimation block. The Peak Frequency Estimation block accepts input waveforms in the form of PDUs and computes their FFT. Then it calculates the spectral peak, which the block can then refine the frequency estimates by using sinc NL-LS. The resulting frequency estimate is then packaged into a PDU, along with metadata identifying the SDR that received the waveform. The Clock Drift Estimation block takes these PDUs as input and, once all have been received, computes the clock drifts using OLS. The final clock drift estimates are then encapsulated in a PDU output with the associated SDR metadata.

3.4. Clock Bias and Carrier Phase Estimation

The clock bias and carrier phase estimation are divided into two blocks: the Peak Time Estimation block and the Clock Bias and Carrier Phase Estimation block. The Peak Time Estimation block accepts input waveforms (e.g. received waveforms and matched filters) as PDUs. Each waveform is pulsed-compressed to provide a peak estimate that is further refined through sinc-based NL-LS. The resulting time estimate is packaged into a PDU and output with the corresponding SDR metadata. The Clock Bias and Carrier Phase Estimation block accepts the time estimates contained in PDUs and waits until all the values from each SDR have been received. It then computes the range estimates between nodes, determines the mean clock bias, and solves for the transmit and receive carrier phases using OLS. The final estimates are then packed into PDUs with metadata specifying which SDR they should be applied to.

3.5. Waveform Compensation

The Waveform Compensation block applies fractional delay techniques and phase adjustments that result from prior estimates. It accepts received waveforms in the form of PDUs and, if a fractional time delay is required, it applies the corresponding delay to the waveform. The block also accepts the estimated clock drift, clock bias, and carrier phase to perform the necessary phase corrections before outputting the compensated waveform as a PDU.

4. Experiment Design

The module was verified through an open-air experiment in an anechoic chamber using low-cost commercial-off-the-shelf (COTS) Universal Software Radio Peripheral (USRP) software-defined radios (SDRs). In this experiment, three USRP X310 with corresponding UBX 160 daughterboards were used. Additionally, monopole antennas were attached to the transmitter and receiver ports of each SDR. The host was used to interface with all three SDRs.

The experiment setup is demonstrated in the block diagram shown in Figure 3. Additionally, the parameter specifications used in the experiment are as shown in Table 1. To simulate coarse synchronization, UHD's `set_time_now()` command was used to set the internal timers of each SDR to $t = 0$, allowing each SDR's timer to be within a few milliseconds of one another. Then, to mitigate the impact of multipath, the SDRs were placed in

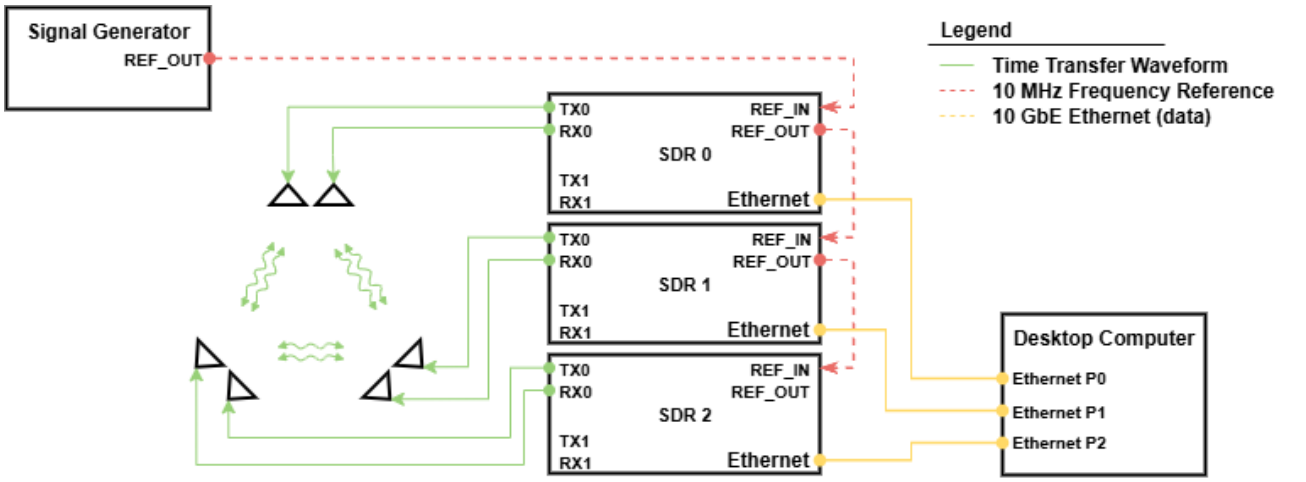

Figure 3. Block diagram synchronization setup.

Parameters	Value
Sample rate	100 MS/s
Center frequency	1 GHz
Baseband frequency	10 MHz
Bandwidth	25 MHz
Pulse width	1 ms
Capture length	9 ms

Table 1. Radar parameters.

SDR	Standard Deviation
2	7.17 ppb
3	6.77 ppb

Table 2. Clock drift estimation results.

an anechoic chamber, as shown in Figure 4.

5. Experiment Results

To characterize the performance of the clock drift and clock bias, 100 consecutive trials were performed in the anechoic chamber. For the estimation of clock drift, all SDRs were given an external reference clock of 10 MHz. Running the synchronization algorithm produced a relatively high standard deviation, with SDR 2 having an error of 7.17 parts per billion (ppb) and SDR 3 showing 6.77 ppb, as shown in Table 2. Consequently, an accurate estimation of clock bias and phase is unlikely.

For the estimation of clock bias, two separate experiments were conducted: one using the external reference clock and assuming clock drift estimates were accurate, and the other using the internal clock of the SDRs. Table 3 shows that using the internal clock and clock drift estimates results in

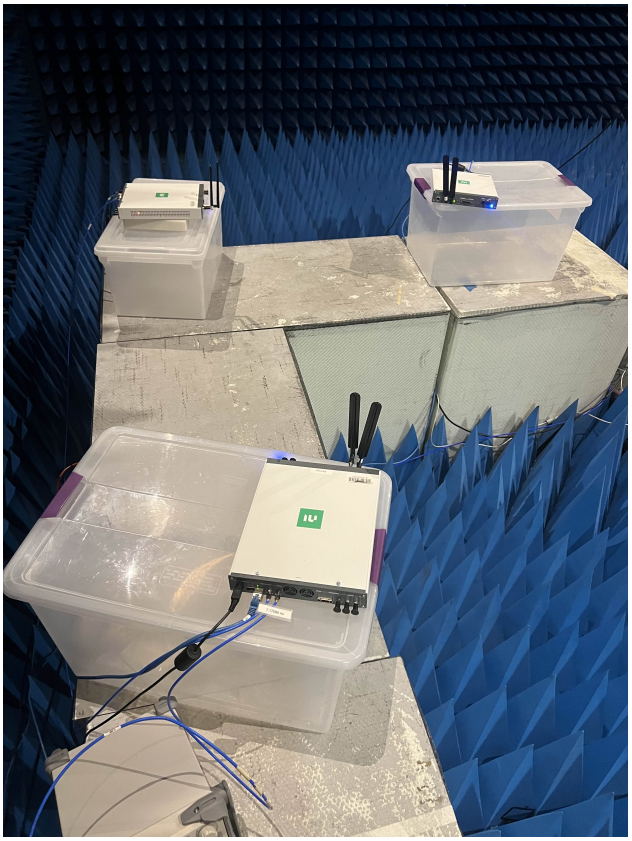

Figure 4. Wireless configuration experimental setup.

deviations in the range estimates. On the contrary, using the external reference clock, the standard deviation is low, ranging from 0.01 ns to 0.15 ns of error.

Test	Dev 1 → Dev 2	Dev 1 → Dev 3	Dev 2 → Dev 3
	Standard Deviation		
Internal Clock	26.18 cm (0.87 ns)	45.73 cm (1.52 ns)	61.57 cm (2.05 ns)
External Clock	3.22 cm (0.10 ns)	3.92 cm (0.13 ns)	4.48 cm (0.15 ns)

Table 3. Clock bias estimation results.

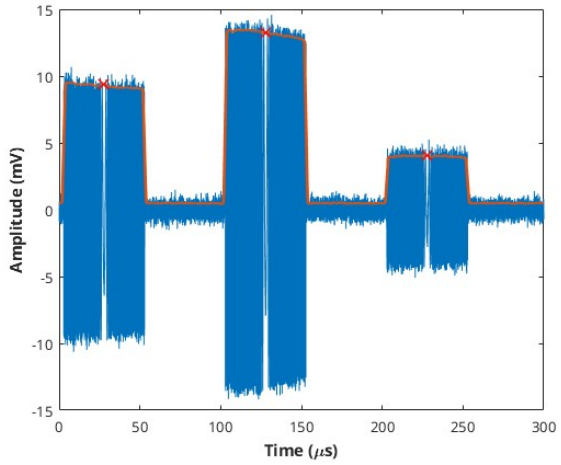

Figure 5. Coherent summation of pulsed LFMs from nodes 2 and 3 to node 1. From left to right, the waveforms are transmitted by node 2, node 2 and 3, and node 3. The orange line is the averaging window across the waveform, and the red "X" marks are the sampled points used for the coherent gain estimation.

To demonstrate beamforming performance and phase coherency with an external reference clock, nodes 2 and 3 were configured to transmit toward node 1. Both nodes were assigned a 50 μs LFM that incorporated compensation for clock bias and range offsets. Given three TDMA time slots, node 2 was set to transmit during the first and second slots, while node 3 was assigned the second and third. This configuration resulted in the signal shown in Figure 5. An averaging filter with a 200-sample window was then applied to the received waveform to compute the power. Afterwards, the power of each pulse was sampled at the midpoint of each LFM. Using these sampled values, the ratio of the combined transmit power and the sum of the single-transmit powers yielded a coherent gain estimate of 98.62% of the maximum possible coherent gain.

6. Conclusion and Future Work

The `gr-harmonia` module has been presented as a fully digitized synchronization process for software-defined radios in a distributed network using GNU Radio. The module functions entirely in the message domain, where all signal processing blocks exchange information in the form of PDUs. At its current stage, the module assumes stationary platforms and demonstrates the ability to estimate key synchronization parameters, including clock drift, clock bias, and carrier phase. The experiments demonstrate sub-nanosecond accuracy in clock bias estimation using an external reference clock and achieve a coherent gain of 98.62% in beamforming, confirming both time and phase alignment.

Future developments of `gr-harmonia` will focus on revising blocks to include SNR estimation, WLS estimation, and Kalman filtering, improving estimation performance during synchronization. In particular, Kalman filtering can improve the estimation of clock drift, enabling reliable integration for future experiments. Additionally, the module is planned to be extended to compensate for velocity in scenarios involving moving platforms.

References

Comberiate, Thomas, Zilevu, Kojo, Hodkin, Jason, and Nanzer, Jeffrey. Distributed transmit beamforming on mobile platforms using high-accuracy microwave wireless positioning. pp. 98291S, 05 2016. doi: 10.1117/12.2231793.

Flandermeyer, Shane, Mattingly, Rylee, and Metcalf, Justin. gr-plasma: A new gnu radio-based tool for software-defined radar. *Proceedings of the GNU Radio Conference*, 7(1), 2022. URL https://pubs.gnuradio.org/index.php/grcon/article/view/121.

Kenney, Russell. *An Approach to Simultaneous Wireless Synchronization and Navigation for Mobile Distributed Networks of Radar Systems*. Ph.d. dissertation, University of Oklahoma, Norman, Oklahoma, 2024.

Kenney, Russell H., Metcalf, Justin G., and McDaniel, Jay W. Wireless distributed frequency and phase synchronization for mobile platforms in cooperative digital radar networks. *IEEE Transactions on Radar Systems*, 2:268–287, 2024. doi: 10.1109/TRS.2024.3369043.

Kenney, Russell H, Metcalf, Justin G, and McDaniel, Jay W. Concept and theoretical performance analysis for decentralised digital synchronisation in distributed radar sensor networks. *IET radar, sonar navigation.*, 19(1), 2025. ISSN 1751-8784.

Merlo, Jason M. and Nanzer, Jeffrey A. High accuracy wireless time synchronization for distributed antenna arrays. In *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 1966–1967, 2022. doi: 10.1109/AP-S/USNC-URSI47032.2022.9887217.

Merlo, Jason M., Shandi, Naim, Dula, Matthew, Bhattacharyya, Ahona, and Nanzer, Jeffrey A. Fully wireless collaborative beamforming using a three-element coherent distributed phased array. In *2024 IEEE International Symposium on Phased Array Systems and Technology (ARRAY)*, pp. 1–8, 2024a. doi: 10.1109/ARRAY58370.2024.10880359.

Merlo, Jason M., Wagner, Samuel, Lancaster, John, and Nanzer, Jeffrey A. Fully wireless coherent distributed phased array system for networked radar applications. *IEEE Microwave and Wireless Technology Letters*, 34(6):837–840, 2024b. doi: 10.1109/LMWT.2024.3375085.

Prager, Samuel, Haynes, Mark S., and Moghaddam, Mahta. Wireless subnanosecond rf synchronization for distributed ultrawideband software-defined radar networks. *IEEE Transactions on Microwave Theory and Techniques*, 68(11):4787–4804, 2020. doi: 10.1109/TMTT.2020.3014876.

Shandi, Naim, Merlo, Jason M., and Nanzer, Jeffrey A. Decentralized picosecond synchronization for distributed wireless systems. *IEEE Transactions on Communications*, 73(6):4425–4438, 2024. doi: 10.1109/TCOMM.2024.3480993.