Arya Keni, RCT DSP

# Analyzing Complex Frequency-Time (FT) Domain Multi-transmitter Signal Generation and Decomposition

# Problem Statement

- The goal is to produce, and then analyze an optimal method to separate FT (frequency-time) domain unique signals generated from “N” transmitters in an open space, recorded by a single SDR (Software Defined Radio). Without knowing the amplitudes of each transmitter, or their ″active times″ (and hence “inactive times”), but only the sampling and central frequency (CF) that the SDR receives transmitter communications with, is it possible to predict the slight offsets from each transmitter in the CF, that are present in a sampling sequence?

- The first part is the develop a complex signal overlay that satisfies such conditions, and then to test it with a set of procedural algorithms that have been proven to work well on single frequency counterparts.

# Devised Procedure ~ Overview

- To focus on the signal generation: each transmitter can fire a strictly real signal (real waveform at a certain power level that is randomly generated, with a pre-quadrature sampling (90 degrees in an out of phase sampling) to determine the aliased frequencies relative to the receivers heterodyned (intermediate) frequency given the sets of offsets to determine), and after firing, sleep for a set amount of time. Since all transmitters are independent of each other, these values randomly vary, since they are unknown but are recorded. Hence, any/all/none transmitters may overlap partially/wholly at any given interval of time, and since the sleep times and firing times have no correlation, some interesting patterns that may/may not repeat depending on the complexity of the combinations of these timings may arise. The general theory is that with enough of these timesteps, a pattern can be deciphered, but usually, there is a limited timestep recording due to data limitations.

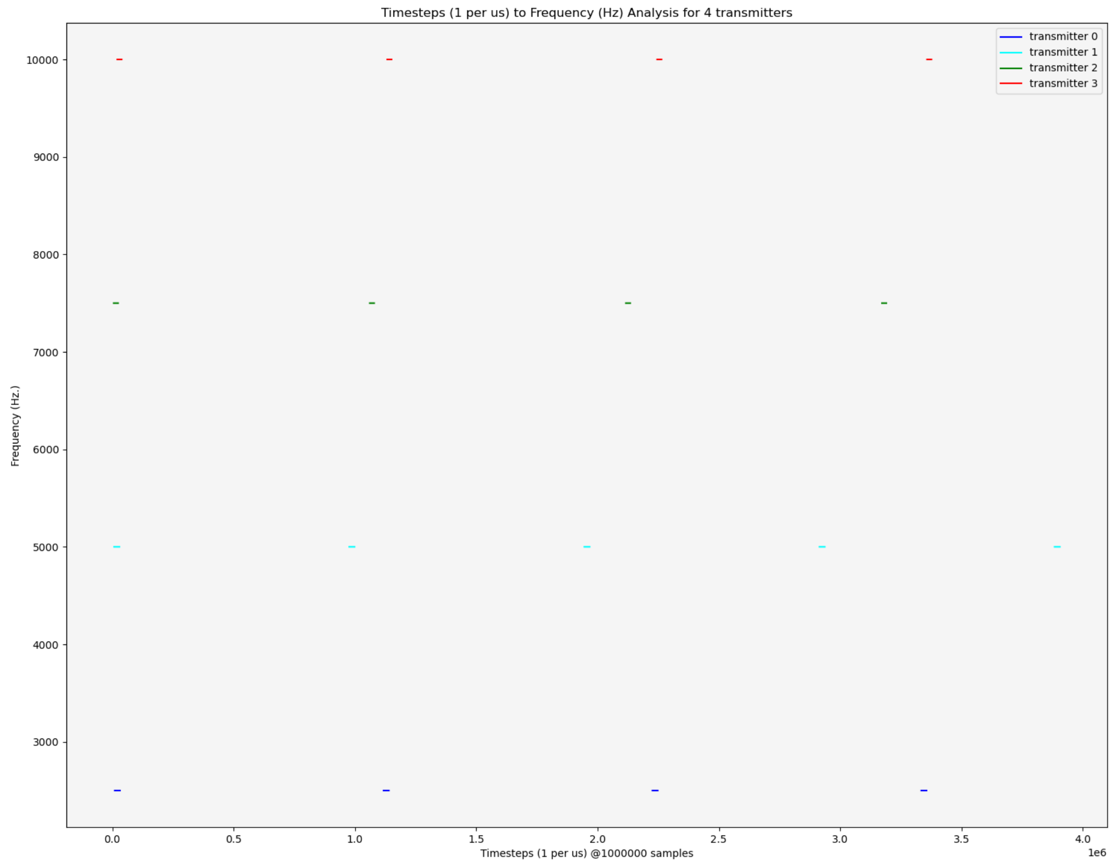
- This signal is then fed into a multi-BLUE (Best Linear Unbiased Estimator) algorithm tweaked to approximate for certain delay ranges, given a CF and sampling rate, for a calculable amount of minimum period. Of course, since this period is hard to determine as previously mentioned: running multi-BLUE with tweaks to run independent of power values and estimate a single offset each time with near perfect accuracy, and minimal error, the issue of separating this behavior for a given set across the offsets arises. Especially if the time-domain is considered, then there are more issues with coverage of accurate Fourier decompositions for a given interval range, which also may rapidly vary.

# Taking a step back …

- Since there are many concepts that are relevant to this topic, here is a good overview on the majority of it, logically speaking, to the use of this development:
- A: **Why BLUE?** Well, BLUE works best for single offsets to CF (99.8%~100%, slightly underestimated values) while using only at most 1% of the sample data recording (even in conditions as "bad" as -96dB SNR of noise, and -40dB SNR (signal to noise ratio) of ping strength). While this is only for single offsets and was rigorously tested in predicting a variety of offsets within the possible range, this is promising enough given BLUE's linearity, to adjust it in modicum to the current constraints.
- B: **Okay, so what is the good range of offsets and transmitting frequency BLUE works best in**? Typically, BLUE works best for any Mid-GHz frequency of transmission, or lower. Any higher, and more complicated estimators, such as Angle of mean, Mean of Angle, reuse, non-reuse, MLE (Maximum likelihood estimator) will need to be used, as depending on the trend of the "data" (rather ping of frequency offset in time), the use of these 6 algorithms depend. Given the offsets that are possible as any real number from - 0.5 GHz to 0.5 GHz, for a sampling rate of 1GHz, the best specifications arise to be the BLUE Algorithm.

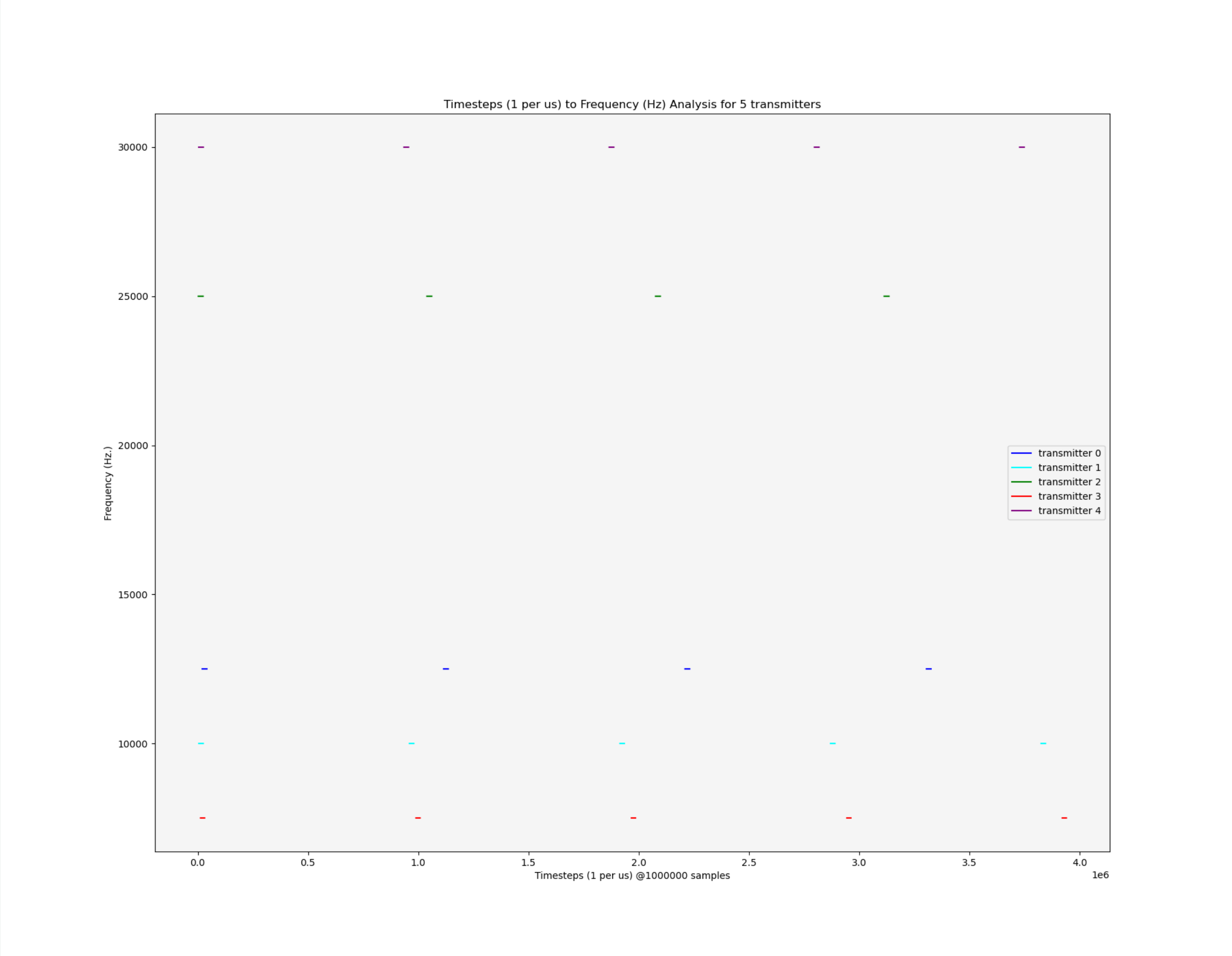
# Current Solution – Non-Analytical Solvability

- Example 1: Let's assume 4 transmitters, all firing at non-equal rates, and sleeping at non-equal rates. For the sake of knowledge, and verification, here are the sets of offset frequencies (keep in mind they can be negative as well):
- `{2500 5000 7500  10000}` in Hz
- *(Old Version of Python Spectrogram: Issues with current waterfall library for some reason. Removed gridlines for ease of review)*



# Current Solution – Non-Analytical Solvability

- Example 2: For a second, more slightly convoluted scenario, consider 5 transmitters, but with a larger range of offsets, and in a non-deterministic order:
- `{12500,10000,25000,7500,30000}` in Hz
- To re-iterate: Each signal looks like a pulse here, as it is a discrete capture of the active (at a certain frequency), and then 0 at inactive as there is no data from the transmitter. Each transmitter is a separated signal by color, exactly as recorded.
- **(ZOOM IN TO SEE THE DETAILS: Labelled, reduced "sleep" time per transmitter → same minimum channel separation in frequency domain of 2.5kHz kept: this condition is verified for each transmitter to be valid)**



# Mathematically defining the Signal

- The signal as "seen" by the SDR can be defined as follows (knowing the starting time (represented as a real phase $\phi_i$ (offset from the beginning of samples within[0, 2π)), the real sleeping time (about 1s on average as $c_i$), the real active time (about 20ms on average as $b_i$), the real amplitude (about -60dB with 10% variance, as $a_i$), the real period (the central frequency, here 172MHz or $f_c$), and the variance in each of the factors (about 10% in each)) for "n" transmitted signals and an integer "i" for the $i^{th}$ transmitter at a statistically independent (0 covariance value) channel (from 1 to n inclusive), hence for a given transmitter at "i":

- $G_m(t)=\Sigma_i \Sigma_k (a_i e^{j(2\pi f\_c+f\_i+\phi\_i)} +\varepsilon_i)(\xi(i,k,t))$ **(where $G_m(t)$ is the full received mixed signal as a form from signal processing papers (Chen et. Al.) describing transmit symbols)**

- Where k is a natural positive number from 1 to floor($t_{end}/(c_i+b_i)$), $\varepsilon_i$ is AWGN (additive white gaussian noise with 0 mean) from surroundings (to be filtered out).
- Where the goal is to ascertain the set $f_i$ for all "i" (sets of frequencies of offset). All of this assumes a recoding time from 0 to $t_{end}$ in any temporal scale described by a sampling frequency ($f_s$) at 1MHz.
- The function ($\xi(i,k,t)$) is a permittance function that separates each "i" in the FT domain: for each k: $\xi(i,k,t)=1$ where t is in between an including $k(c_i+b_i)$ and $k(c_i+b_i)+b_i$ and is $\xi(i,k,t)=0$ otherwise. Note that $f_i$ typically takes values from $-f_c$ to $f_c$ inclusive.

# Aiming to Solve this problem

- The approach to *possibly* alleviate the multiplicity of variables that are not known can be solved by modifying BLUE to run parallelly, not only in search of frequency domain irregularities from a set offset (singly evaluated and references), but also during a chunk of timesteps to classify existing offsets in said chunk, which solves the issue of recording offsets in general.
- Given this, since the complexity of the recorded signal from n transmitters containing n offsets all at any combinatorically probable set of combinations, and the ping strength being unknown, the aim is for BLUE to denoise the signal, bypass the requirement of amplitude domain (as with single frequency), and try and classify time-dependent data for n offsets, where n is not known to the algorithm either. Hence a unique modification to BLUE to run parallelly in multiple steps must be devised.
- *An additional aspect of interest, so to say, could also be that once such an algorithm has been developed, to minimize the sampling rate, and hence by extension, amount of data, required to be an input for multi-BLUE*
- To re-iterate, a key challenge in this process is to smartly apply a version of multi-BLUE in enough active time intervals such that the complete set of frequencies of offsets are deciphered, given the number to decipher being known. Chunking time-steps in intervals, and after a FFT process, attempting to decode n frequencies parallelly seems to be the best idea, in summary

# Future Implications

- We can utilize this signal generation code effectively to test out under many conditions, automated for the modified BLUE algorithm, to train it and adjust it to fit the specifications of solving the problem (i.e. finding out a set of CF offsets, along with possible active times per timestep of sample).

- In addition to the proposed solution: the challenge here lies in adjusting the specifications, as BLUE benefits greatly from amplitude/signal power related data for its training purposes, but there may be a way to utilize Fourier decomposition (Transform + Separation) and some preliminary spectral analysis to bypass this issue, and produce meaningful results from a very complicated signal capture (as read by the receiver).