Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

EEE 310

Communication Systems I Laboratory

Project Report

Project Title:

Signal Sampling and Reconstruction at Sub-Nyquist Sampling Frequencies using TPSL-ADC Sampling Method

Submitted To:

Dr. Md. Forkan Uddin
Professor, Department of EEE, BUET
Tashfiq Ahmed
Lecturer, Department of EEE, BUET

Submitted by:

Sariha Noor Azad (1706051)

Saleh Ahmed Khan (1706053)

Tiasa Mondal (1706054)

Sadat Tahmeed Azad (1706064)

Section: A2

Date of Submission: 27/July/2021

Keywords:

Compressive Sensing, Sub-Nyquist Sampling, L1 norm, Sparse signal, L1-magic algorithm, Fourier Basis Matrix, Sensing Matrix, Measurement Matrix

Introduction:

In the field of communication and signal processing, a signal must be sampled at Nyquist frequency or higher, in order to successfully reconstruct it from the discrete time signal using any interpolation formula, such as Whittaker-Shannon Interpolation formula which can reconstruct the original signal just from the Nyquist Rate signal.

According to sampling theorem, if a signal is sampled below the Nyquist Rate, then it is impossible to reconstruct it using any conventional methods due to Aliasing.

Overview of Compressive Sensing and TPSL-ADC method

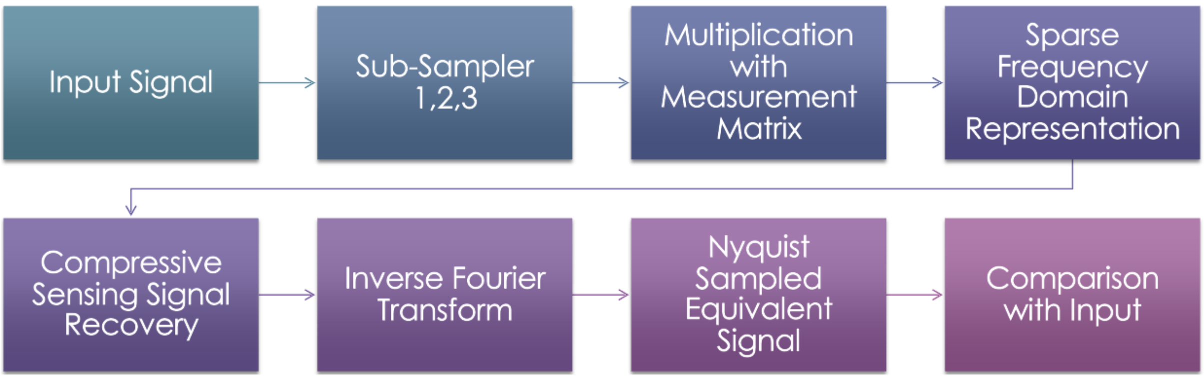
Compressive Sensing is the central idea of our project. Usually, the Nyquist sampling theorem assumes that the signal is broadband. But often, in the Fourier domain, majority of the frequency components are zero, which means that the frequency domain representation is sparse, and the spectral information of these sparse signal is much less than its bandwidth. Compressive Sensing is the idea that using this property we can recover the original signal by obtaining the sparsest solution of its frequency domain representation.

The reconstruction requires the pre-modulation of the original signal with a pseudo-random sequence. This is usually done by sampling the signal at random positions (Random Equivalent Sampling or RES) by varying the sampling interval, which increases the complexity of the system.

In the paper[1] that we have tried to follow, the alternative proposed was to use three parallel sub- Nyquist low-rate Analog-to-Digital converters (TPSL-ADC). In this method, three samplers, each with a sampling frequency much lower than the Nyquist Frequency of the sampled signal is used and the information from all three samplers are combined while reconstructing the original signal. They are set at different sub-Nyquist rates, which are coprime to each other, so that samples do not overlap or become uniformly distributed. The starting time of the samplers are also set different. As a result, the resulted parallelly sampled sequence is pseudo-random, and the full potential of the CS method can be achieved. The recovery algorithm is discussed in the appropriate section. The advantage is that, unlike RES, we don’t need complex circuitry to vary sampling interval.

Sub-Nyquist Sampling can have a multitude of uses, from less memory consumption, low cost of equipment to less transmission bandwidth in communications.

Methodology of TPSL-ADC Method:



In our project, Sub-Nyquist sampling and Signal Reconstruction will be demonstrated in two ways:

(i) Sampling and Reconstruction of some common Signals
(ii) Sampling and Reconstruction of a speech Signal

Part – I: Sampling using 3 Parallel Samplers

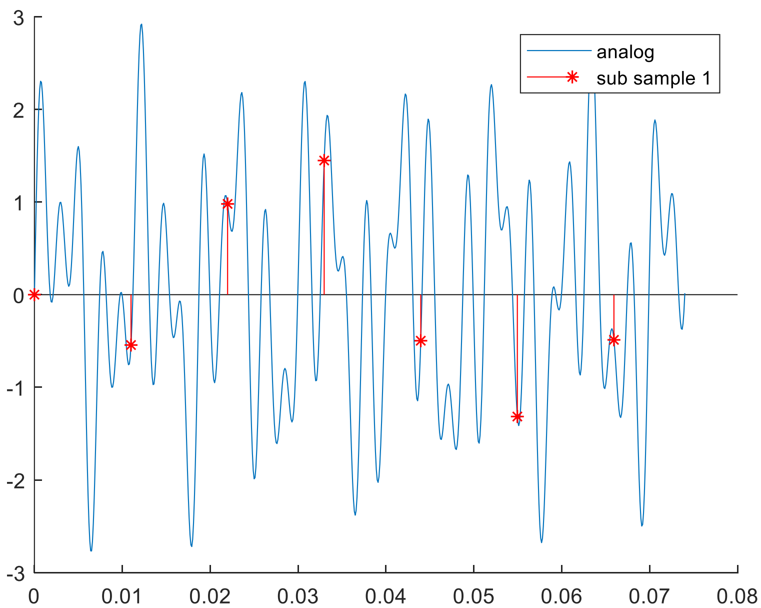
Input Analog Signal:

$$\sin(2*pi*70*t) + \sin(2*pi*275*t) + \sin(2*pi*400*t)$$

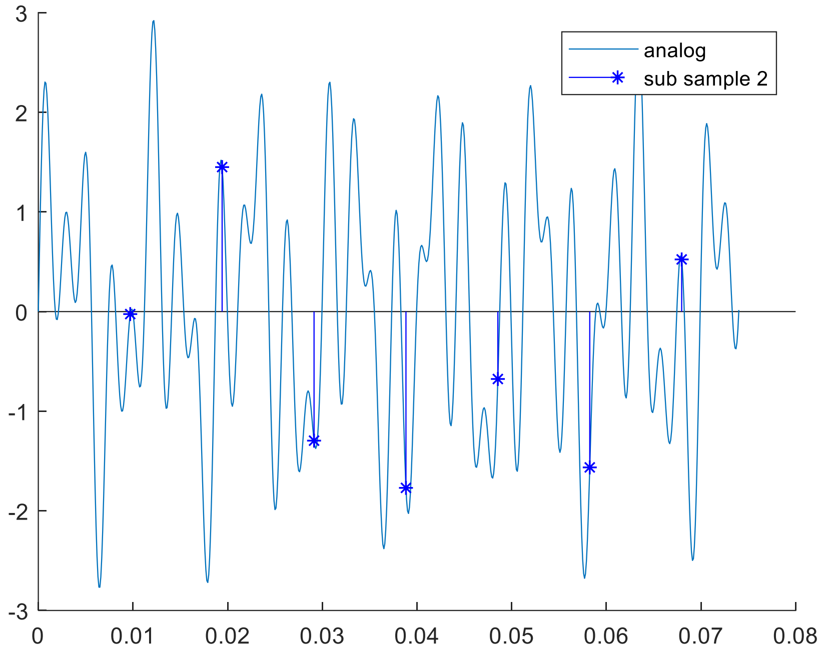
From inspection of the signal, it can be easily understood that the Nyquist Frequency is Fs = 800 Hz and normal sampling frequency is taken as 1000 Hz

However, the samplers are set to take samples at frequencies,

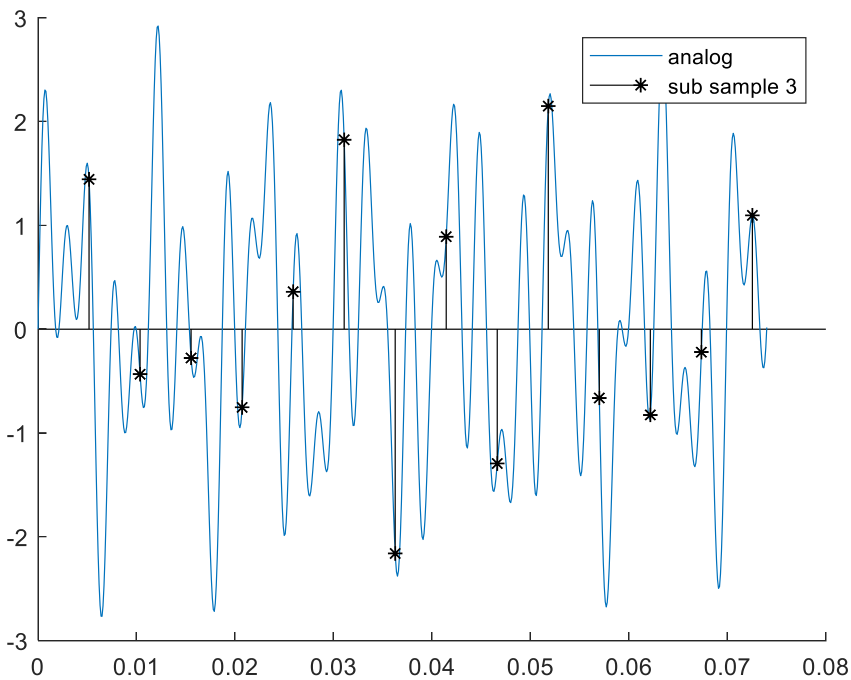
Sub Sampler 1: Frequency, F1 = 91 Hz



Sub Sampler 2: Frequency, F2 = 103 Hz, with starting delay from first sampler $\tau_{21} = 1/F2$

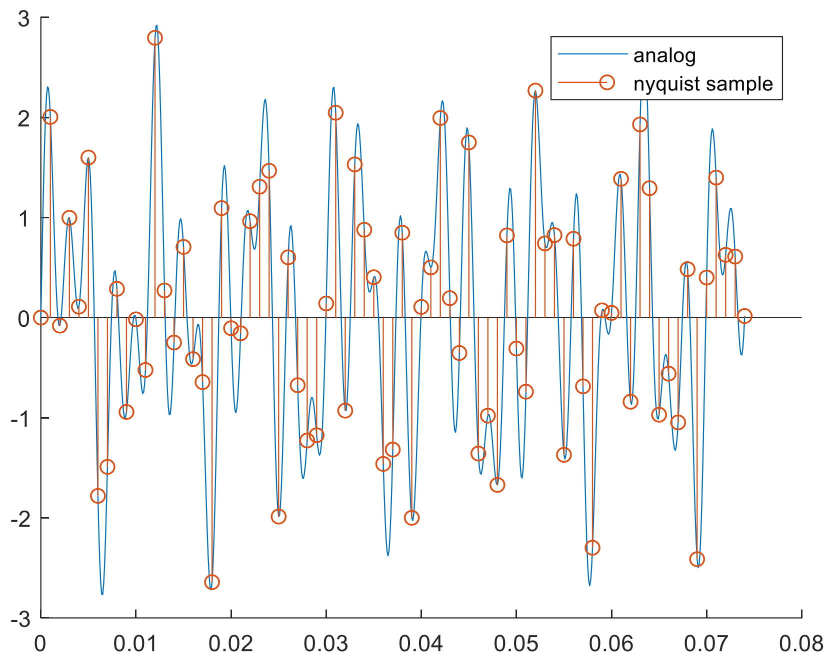


Sub Sampler 3: Frequency, F3 = 193 Hz, with delay $\tau_{31} = 1/F3$



Plot of Samples taken from the original signal using Sub-Samplers

Parallelly, these emulate a pseudo-random sequence.



An equivalent sampled frequency sampled signal has also been taken here for reference.

Part – II: Creation of the Measurement Matrix

In order to reconstruct the Nyquist frequency signal from its subsampled counterparts, we need to establish a relationship between the two. This comes in the form of a measurement matrix. By multiplying the Nyquist/Eqivalent frequency sampled Discrete time signal with its corresponding measurement matrix, we can find the sub-sampled discrete time signal.

$$E_{\text{out}}[m] = E_{\text{out}}(mT_e), \quad 1 \leq m \leq N.$$

E_{out} is the original continuous time signal, which has been sampled at sampling frequency greater than Nyquist rate. Te is sampling interval.

The sub-Nyquist sampled signals can be represented as

$$\begin{cases} E_1[m] = E_{\text{out}}(mN_1T_e),\ 1 \leq m \leq M_1 \\ E_2[m] = E_{\text{out}}(mN_2T_e + \tau_{21}),\ 1 \leq m \leq M_2\ , \\ E_3[m] = E_{\text{out}}(mN_3T_e + \tau_{31}),\ 1 \leq m \leq M_3 \end{cases}$$

Where τ is the respective starting time delay from the first sampler.

Using Whittaker–Shannon interpolation theory, we can write -

$$E_1[m] \approx \sum_{n=1}^{N} E_{\text{out}}(nT_e) \cdot \text{sinc}\left(\frac{mN_1T_e}{T_e} - n\right), 1 \le m \le M_1$$

$$E_2[m] \approx \sum_{n=1}^{N} E_{\text{out}}(nT_e) \cdot \text{sinc}\left(\frac{mN_2T_e + \tau_{21}}{T_e} - n\right), 1 \le m \le M_2.$$

$$E_3[m] \approx \sum_{n=1}^{N} E_{\text{out}}(nT_e) \cdot \text{sinc}\left(\frac{mN_3T_e + \tau_{31}}{T_e} - n\right), 1 \le m \le M_3$$

Where M is the respective number of sub-sampled points. The interpretation of these equations is – the N point uniformly sampling signals (sampled above Nyquist rate) is interpolated, and then subsampled with the respective sub-Nyquist frequency.

In our code, we have started the sampling indices from 0, and modifications were done appropriately.

Each sinc part can be taken as an M_i x N matrix (i= 1,2,3), and then concatenated to form the measurement matrix. If the three matrices are denoted as Φ, φ and γ, then the resulted measurement matrix is-

$$\mathbf{\Phi} = [\phi^{\mathrm{T}}, \varphi^{\mathrm{T}}, \gamma^{\mathrm{T}}]^{\mathrm{T}}.$$

Whose dimension is (M1 + M2 + M3) x N, or M x N. Multiplying this matrix with the Nx1 Nyquist-sampled signal will yield E = $[E1\ E2\ E3]^T$, an Mx1 matrix which emulates a pseudo random signal.

Also, these matrices satisfy restricted isometry property, which is required in signal reconstruction.

Part – III: Relation Between the Signals sampled at Sub-Nyquist and Nyquist rate

As explained previously, the measurement matrix allows us to establish the relation-

Sub-Nyquist Sampled Sequence =

(Measurement Matrix) x (Nyquist Sampled Sequence)

Mathematically, $E = \Phi . E_{out}$

Now, the Nyquist sampled signal can be broken down into the $E_{out} = \Psi \cdot X$, where Ψ is the Fourier Basis Matrix, and X is the corresponding Fourier Transform of the Nyquist sampled signal. Finally, we can write -

$E = \Phi . E_{out} = \Phi \cdot \Psi \cdot X = A \cdot X$

Here, A is called the sensitivity matrix, whose dimension is M x N, assuming N is the N-point Fourier transform of the signal. In the paper followed, FT was taken of the length 5N instead, for reducing leakage. However, this does the opposite sometimes, in cases where the frequency components are integer multiples of the frequency interval, and the sparsest spectra is achieved at FT length N, not 5N. But increasing the number of points do make the frequency spectrum sparser, and so there is a trade-off.

In can be noted that this results in an underdetermined system E=AX (i.e., infinite solutions), where we have N unknowns in X but have M equations at our disposal (M < N). This is where the idea of compressive sensing comes in. We use the sparsity property of practical frequency domain signal to our advantage and use an algorithm to find the sparsest solution of X (where most of the components are zero) that satisfies our underdetermined system. This wouldn't be possible if we used $E_{out}[n]$ instead.

Sparse Representation of the Input Signal

For example, a simple sinusoid in time domain is not a sparse signal but in frequency domain, it has only two frequency components and thus can be used a sparse matrix in compressive sensing signal recovery. Same holds true for most signals.

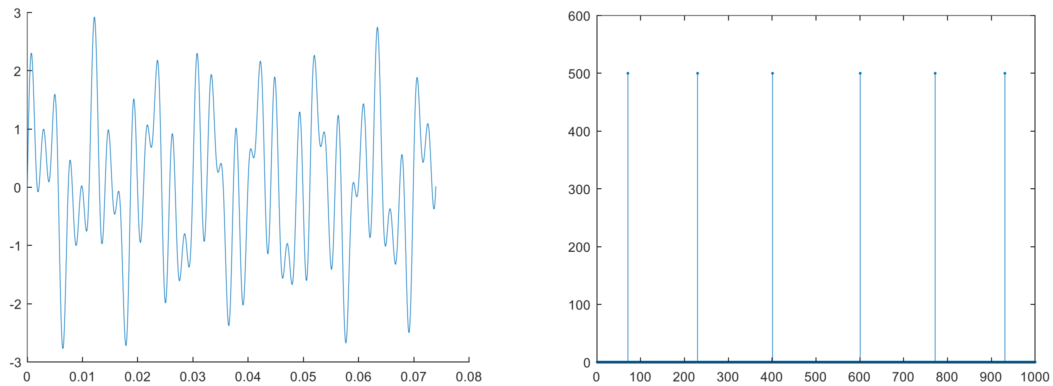


Fig: time domain vs sparse signal representation (FT)

Part IV: Simplified Reconstruction Algorithm

Our underdetermined system is E=AX, where E is the subsampled signal emulating pseudo-random sampling, A is sensitivity matrix, and X is the unknown Fourier Spectrum.

The simplified algorithm proposed in paper followed is this- each column of A (N in total, or 5N in the paper) represents the weight of each of the N frequencies in the Fourier spectra. However, as most of the components are zero, we don't need all the **N** columns. So, we only choose the columns that have the greatest weight on **E**, and put them in another MxN matrix **B** (initialized as an MxN zero matrix), in the corresponding columns.

In the algorithm proposed in the paper, this was achieved by correlating the error Mx1 matrix **r** (initialized as r=E)**,** which is the deviation from E at the end of each iteration – with each column of A. Thus, we get N correlation coefficients, the greatest of which contributes the most to the error. Thus, we put the corresponding column of **A** in the matrix **B**, and solve for **X#** using least square regression for E=BX, using the formula-

$$\mathbf{X}^{\#} = (\mathbf{B}_i^{\mathrm{T}}\mathbf{B}_i)^{-1}\mathbf{B}_i^{\mathrm{T}}\mathbf{E}$$

Then, we find the corresponding error by

$$\mathbf{r} = \mathbf{E} - \mathbf{B}_i\mathbf{X}^{\#}.$$

Which then is used in the next iteration.

However, we were able to implement this algorithm for real sparse signal, but for complex signals, the size of the matrix increased the number of complex multiplications, further complicating the calculation, and thus giving wrong results. Additionally, the least square method (which uses L2 norm of error) is not good when it comes to sparse signal reconstruction. So, we used a function that solves any given underdetermined system for the sparsest solution using L1 norm instead, obtained from the toolbox referred to below.[2]

Used function: l1eq_ (x0, A, [], y)

In our case, A is sensitivity matrix, y=E, and x0 = $A^T y$, which is initial correlation coefficient matrix.

From the obtained Fourier domain signal, we can get the time domain reconstruction using inverse Fourier transformation.

From there, we can acquire the time domain signal.

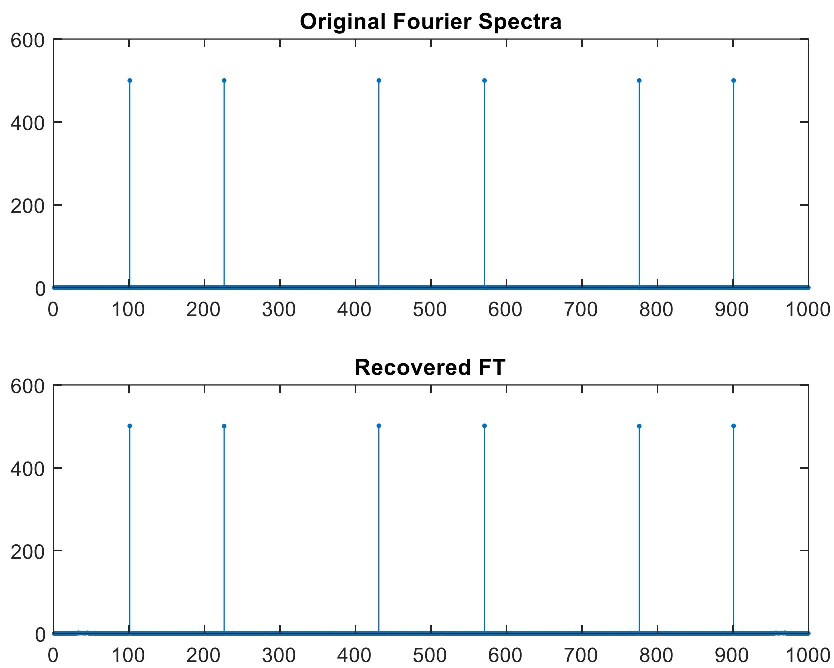
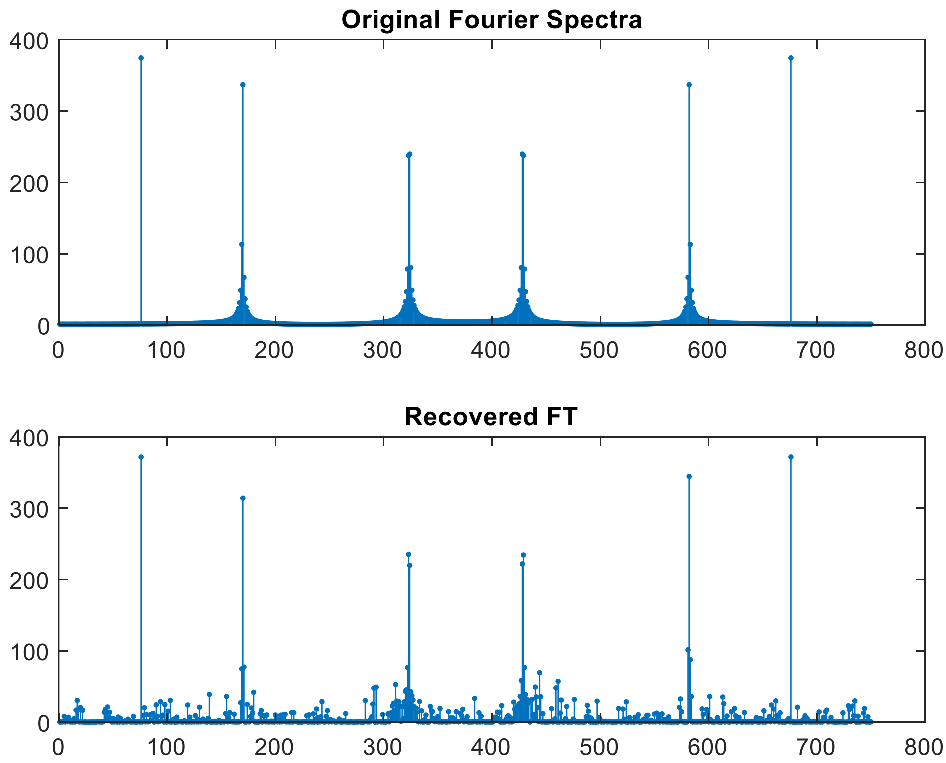


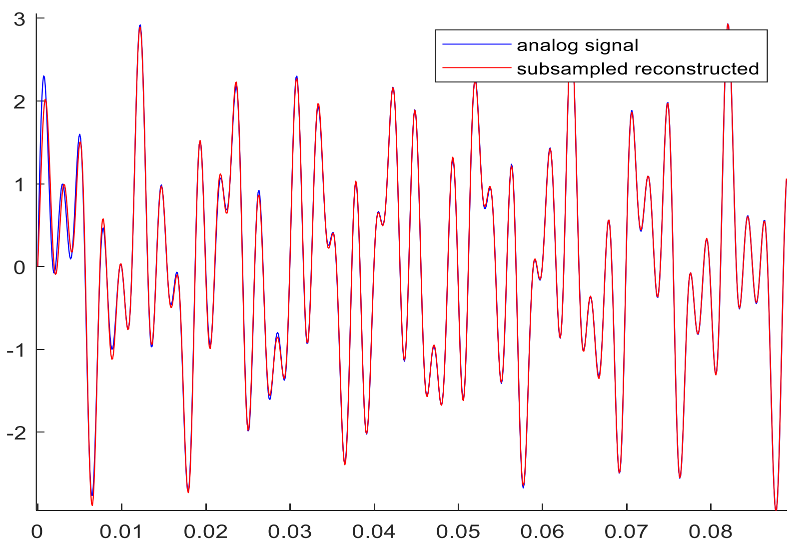
Figure: Reconstructed Fourier Spectra

As we can see, the sparse signal reconstruction is performed well. In this example, we have chosen the sampling frequency Fc and signal length N such that the frequency components are integer multiples of Fc/N. However, this is not always practical, as we can't know the frequencies in a signal. So, in those cases, there will be leakage, and reconstruction won't be so perfect, as can be seen in the following example (The number of points N was varied, and thus leakage was introduced).



Then, time domain signal is recovered using inverse Fourier transform.

Results



The Input Signal has been reconstructed satisfactorily. Although, there are a few distortions in the two edges of the signal, because we are using a Whittaker-Shannon Interpolation for a finite duration signal.

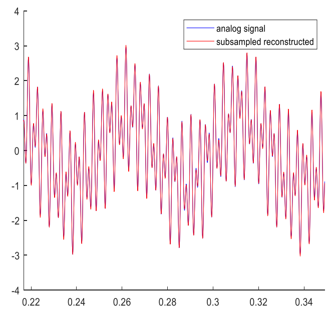
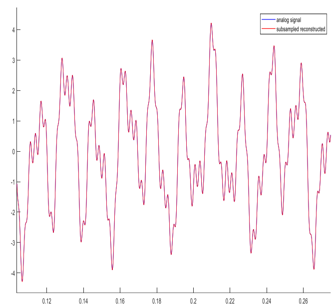
The Signal-to-Noise Ratio of the Signal = 40.167 dB (For the no leakage case).

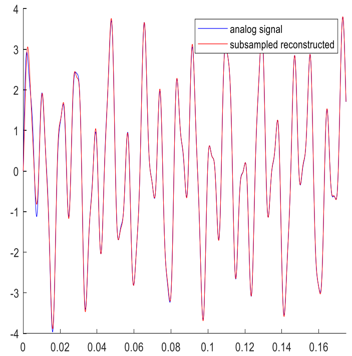
Performance with other signals and frequencies

We also tested our reconstruction scheme with various input signals such as sinusoids of various frequencies as well as rectangular and sinc functions to see test the limitations of the program.

(I) Different periodic Signals

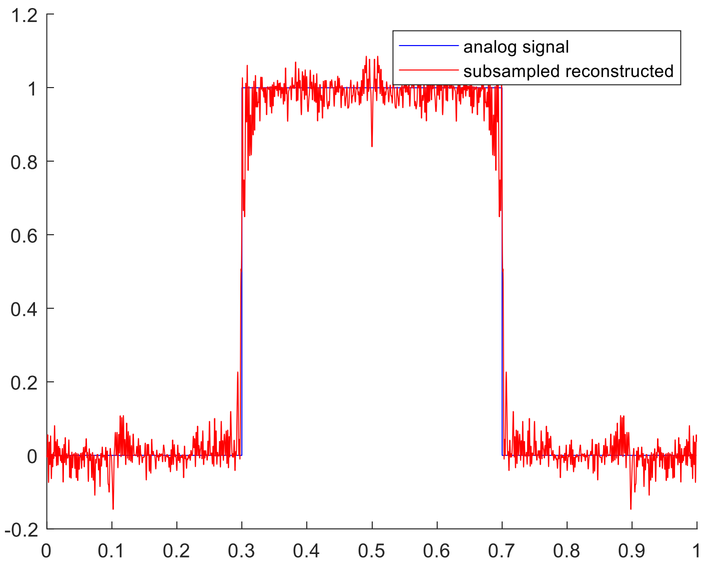
An array to different tests is performed with sinusoids of varying frequency. The results are given below.

Input Signal	The Frequencies (F1, F2, F3, FC)	M, N and N/M	SNR (dB)	Input Vs. Output Graph
`sin(2*pi*20*t) + sin(2*pi*280*t) + sin(2*pi*490*t)`	91, 103, 193, 1000 Hz	385, 1000, 2.597	29	
`sin(2*pi*25*t) + 0.5*sin(2*pi*345*t) + sin(2*pi*120*t) + 2*sin(2*pi*63*t)`	83, 97, 117, 1000 Hz	295, 1000 3.389	53	

`1.5*sin(2*pi*48*t) + 0.5*sin(2*pi*233*t) + 2*sin(2*pi*111*t);`	83, 97, 117, 500 Hz	295, 500 1.695	38	

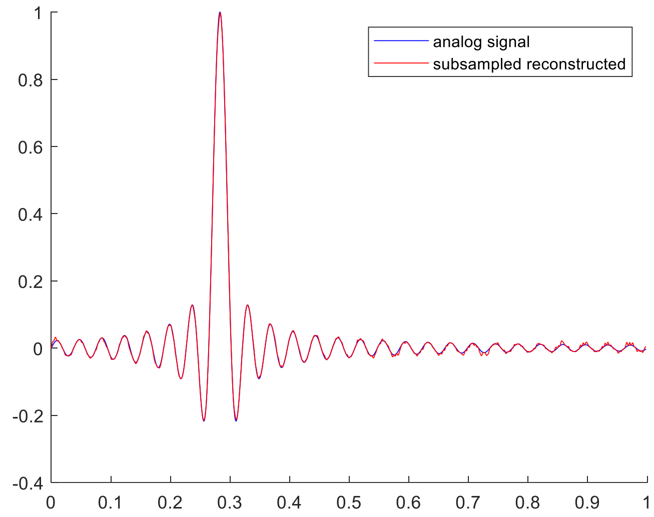
(II) *Rectangular Function:*

We know that, rectangular functions ideally have an infinite bandwidth, so we tried to reconstruct a rectangular pulse using our method. The resulting figure has a lot of high frequency ripples. SNR = 19.92 dB



(III) Sinc Function:

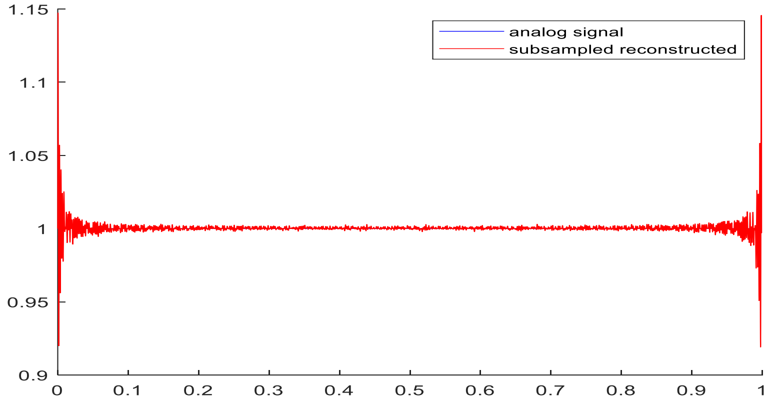
The frequency domain representation of a Sinc function is a rectangular pulse. So, it is difficult to represent it sparsely like a sinusoid, so Sinc function representation is subject to a noisy reconstruction if its FT domain signal is too wide.



(IV) Pure DC value:

Reconstruction a pure DC value also results in high frequency ripples at the edges.

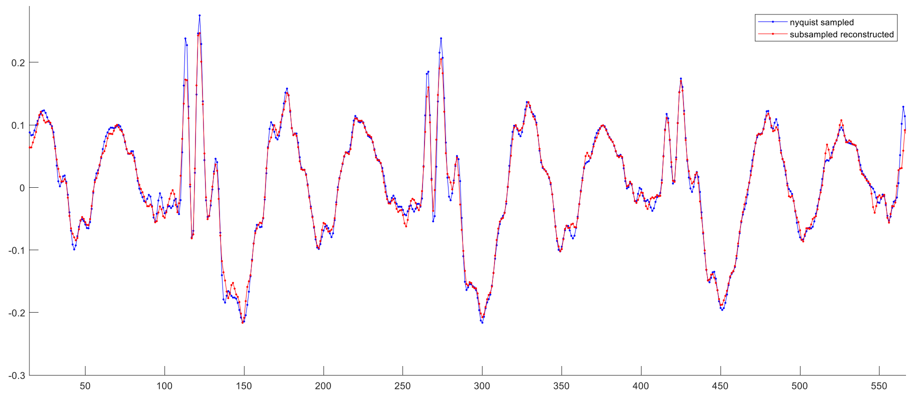
Here, the value of the input signal is 1.

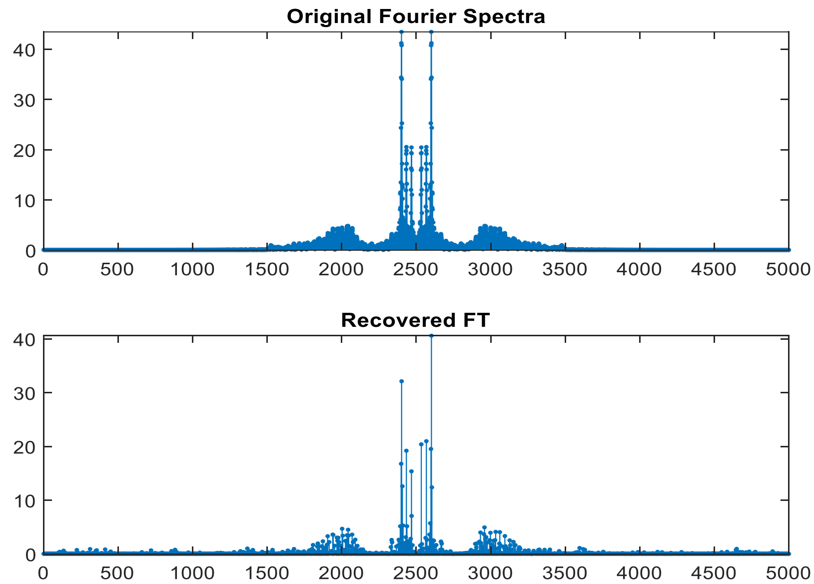


Reconstruction of a Voice Signal

To test a practical signal, we have recorded our voice and applied our reconstruction scheme on it. The results are shown below,

In case of voice reconstruction, the delay needs to be set to $\tau_{21} = 1/F3$, $\tau_{31} = 1/f2$ to emulate pseudo-randomness.





F1 = 3300, F2 = 4650, F3 = 7530, Fc = 22050 Hz

Sub-sampled Signal Length = 696

Input Signal Length = 1000

SNR = 15.81 dB

Full Signal Reconstruction

Full Sub-sampled Signal Length = 98414

Full Input Signal Length = 141120

Full Signal SNR = 16.23 dB

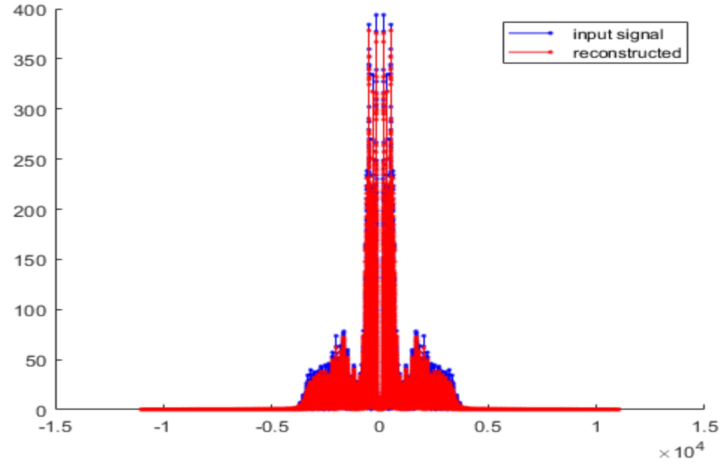


Figure: Frequency Spectrum of Input and Reconstructed Signals

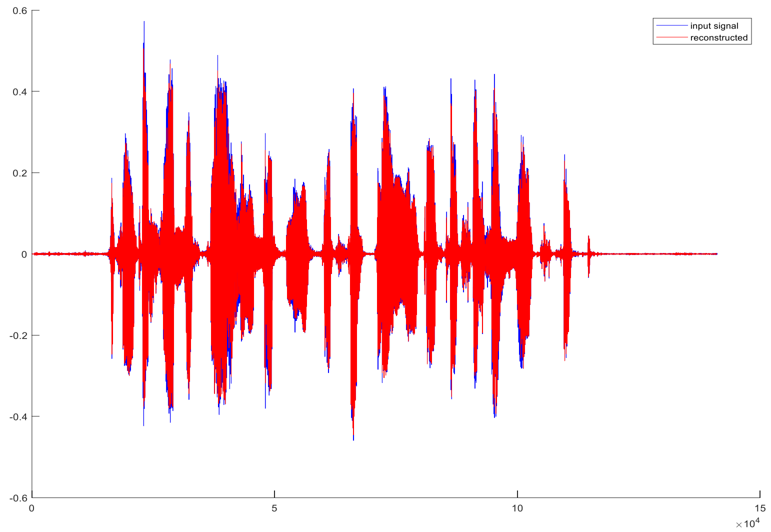


Figure: Time Domain graph of Input and Reconstructed Signals

The Practical Signal is prone to some noise, in order to implement this signal in a practical scenario a post process noise removal filter or program will be able to suppress additional noise in the signal. Apart from that, we can conclude the signal has been reconstructed satisfactorily.

Limitations and Adjustments

As the system is based on sub-Nyquist sampling, as well as a Compressive Sensing algorithm working on complex values, the capabilities of this type of signal recovery subject to some limitations. Such as,

1. ***Spectral Leakage:***

   We are using the Fourier Transform of the Input Signal for Signal Recovery, which is subject to spectral leakage if the number of points in the Fourier transform doesn't match with the frequencies in the signal, which results in noise in the output signal.

2. ***Signals that cannot be Sparsely Represented:***

   The algorithm has some difficulty in reconstructing signals whose Fourier transform is not sparse enough, such as the Sinc function and in some instance, real voice signals are also reconstructed with a low SNR and moderate amount of noise that must be filtered after reconstruction.

3. ***Issues with Compressive Sensing Algorithm:***

   Compressive Sensing Algorithm is the central concept of our program, but in some cases, the Sensing Matrix and/or the subsampled matrix are either poorly conditioned, does not emulate pseudo-randomness or does not satisfy the *restricted isometry property* which is essential in order to solve the underdetermined system.

Some adjustments are needed to overcome the following limitations, if they ever arise in operation:

1. ***Adjusting the size of the Sensing Matrix:***

   The size of the Sensing Matrix, or the Fourier Basis Matrix creates the sparse input signal, however, if too many points are taken, the system is prone to Spectral Leakage just like any discrete frequency domain signal. Therefore, it must be adjusted to have the least leakage for best results of signal recovery.

2. ***Increasing the Target Sampler Frequency:***

   The frequency at which a signal is sampled plays a key role in its bandwidth as well as frequency spectrum, if a signal cannot be represented as a sparse signal, then by increasing the target frequency (Fc) in our program, we can squeeze the frequency spectrum of that signal to add more zeros and make the signal sparsely represented.

3. ***Adjusting the Sub Samplers Frequency or Delay:***

   Adjusting the three sub-samplers' delay and/or their sampling frequencies will change the matrices used in CS calculation, and thus if any previous values had been poorly conditioned, doing so will create new data and allow the algorithm to run.

Practical Uses and Merits of TPSL-ADC Method

(i) Lower Bandwidth Requirement:

The sampling frequency of the 3 Sub-samplers combined is much lower than the Nyquist frequency. Therefore, if the signal is transmitted and sent using FDM/QM or any similar method, it will consume significantly less bandwidth, compared to the broadband Nyquist Signal. If utilized by the communication companies, the system has the potential to make the transmission and bandwidth usage more efficient.

Potential Application Field: Cognitive Radio, Ultra-wideband Communications, Wideband RADAR

(ii) ***Low Memory Consumption while storing Sub sampled digital signal:***

As demonstrated earlier, not only does can the system sample at lower frequencies but also store more data in less space. As apparent by the size of Sub-sampled matrix compared to the Nyquist sampled matrix (M<N). Therefore, TPSL-ADC sampled signal requires less memory storage space.

(iii) ***Overcome Hardware limitations to sample Ultra-High Frequency Signals:***

With the advancement of communication technologies, the demand of higher sampling rate is increasing day by day. But it's difficult to construct Analog to Digital Converter with higher bits per sample at high sampling rate. So sub-Nyquist rate creates an alternative approach in this way.
Potential Application Field: High frequency Electric Field Measurement.

(iv) Lower Equipment Cost:

The cost of Samplers and associative Equipment get higher with the sampling frequency, using TPSL-ADC method will allow the user to process and transmit signal to use less bandwidth and lower the cost of overall operation. Additionally, system complexity is increased in high frequency applications. In such cases, lowering sampling frequency means greater sampling interval and thus more feasible method.

Drawbacks of TPSL-ADC Method

(i) Practical Signals and Voices are prone Noise:

As demonstrated in the report, to increase the robustness of this system in a practical scenario, a filter/hardware to remove noise must be added to the output.

(ii) Increased Hardware and Sophistication:

While TPSL-ADC is a good at sampling at high frequencies using cheap and slow samplers, the system needs more hardware (3 samplers and other necessary accessories) as well as a calculation medium to do that same task that a single high-speed sampler and receiver can manage by itself. This increases the hardware requirement and sophistication of the entire system.

(iii) Computationally Expensive:

The Compressive Sensing Algorithm is based on matrix calculations. Therefore, it requires some time to reconstruct the signal. And this calculation time increases with the number of samples and the associative frequencies. While reconstructing a practical signal, this process can take up-to a few seconds.

Conclusion:

TPSL-ADC signal sampling and reconstruction is a relatively new and unconventional method in the field of communication and technology. If its limitations, such as the randomness aspect of Compressive Sensing and issues of Noise can be resolve can be resolved, then this system has huge potential in optimizing the problems of the memory limitations, bandwidth restrictions and might be able to tackle the constant demand of high speed and sophisticated sampling and signal processing devices that are beginning to emerge with the recent evolution and developments in the communication sector.

References:

[1] https://www.researchgate.net/publication/337546953_Sub-Nyquist_sampling_method_and_its_application_in_high-frequency_electric-field_measurement

[2] https://statweb.stanford.edu/~candes/software/l1magic/

Appendix:

List of all Included Codes:

1. Time domain sampling and reconstruction of some basic signals

“”

2. Time domain sampling and reconstruction of some basic signals with noise

“”

3. Time domain sampling and reconstruction of some basic signals with noise (graph plotting)

“”

4. Sampling and Reconstruction of truncated voice signal

“”

5. Sampling and Reconstruction of full voice signal

“”

6. Whittaker-Shannon Interpolation Function

“ws_interp”

7. L1-magic CS Function from the link

“l1eq_pd.m”

8. L1-magic CS Function implemented by our group

“”