Modulo Folding and Signal Recovery Above Nyquist Rate

Above Nyquist Rate: Modulo Folding Does Not Hurt

We aim to recover a continuous-time bandlimited signal from its discrete-time samples, which are obtained via uniform sampling. These samples are further subjected to a modulo operation that wraps the values within a specified range. The modulo operation is defined as:

$$y[n] = \left(f(nT_s) + \frac{\Delta}{2} \right) \bmod \Delta - \frac{\Delta}{2}$$

This operation confines the observed samples to the interval $[-\Delta/2, \Delta/2]$, effectively folding any values lying outside this range back into it.

The goal is to reconstruct the original, unfolded values $f(nT_s)$ from the folded samples $y[n]$. When the sampling rate is above the Nyquist rate, and the original signal is sufficiently smooth (bandlimited), this reconstruction is possible. The idea is to exploit the continuity and predictability of bandlimited signals to identify and correct the integer multiples of Δ that were lost during folding.

In this context, even if the folded samples appear to be within a small bounded range, their actual values might lie outside, differing by one or more integer multiples of Δ. Proper unfolding restores the correct values.

Oversampling and Sample Prediction

When the oversampling factor $\zeta = \pi\varepsilon$ is positive (i.e., the signal is sampled above the Nyquist rate), perfect reconstruction of the original signal is possible.

Predicting the Next Sample from the Past

We consider the problem of predicting the next sample of a discrete-time signal from its past values. Let the sampled sequence be given by:

$$x_n = x(nT_s)$$

Assume that the sampling frequency satisfies:

$$\omega_s > 2\omega_m \quad \text{or} \quad \frac{1}{T_s} > 2W \quad \Rightarrow \quad T_s < \frac{1}{2W}$$

Suppose x_n is a real-valued signal, and it can be expressed in binary as:

$$x_n \in (-2^m, 2^m)$$

The binary expansion of x_n is:

$$x_n = -a_{n,0} \cdot 2^m + \sum_{\gamma=1}^{\infty} a_{n,\gamma} \cdot 2^{m-\gamma}$$

where $a_{n,0}, a_{n,1}, a_{n,2}, \ldots$ are the binary digits representing the signed fixed-point value. The bit $a_{n,0}$ indicates the sign.

At some point, the digits are truncated or rounded, which results in a decimated representation. Instead of representing x_n using infinite precision (all bits), we aim to represent it using only k bits — typically the most significant bits (MSBs).

Thus, we approximate the signal x_n by truncating or quantizing it:

$$x_n \to a_{n,0}, a_{n,1}, a_{n,2}, \ldots, a_{n,k-1}$$

This allows us to perform processing or prediction with finite-bit representations, enabling practical implementations in digital signal processing.

Limitations of Hardware to Accurately Represent a Signal

When the reconstruction of the signal is not perfect due to hardware limitations, we face the issue of finite-bit representation. Let:

$$x_n = -a_{n,0}2^m + \sum_{\gamma=1}^{\infty} a_{n,\gamma}2^{m-\gamma}$$

If we approximate x_n using only the R most significant bits (MSBs), we denote this approximation as $x_{n,q}$, and define the residual as:

$$x_{\text{res}} = x_n - x_{n,q}$$

This residual represents the error or loss due to quantization or approximation. The quantized value can be written as:

$$x_{n,q} = -q_{n,0}2^m + \sum_{\gamma=1}^{R-1} q_{n,\gamma}2^{m-\gamma}$$

Here, $q_{n,\gamma}$ are the bits retained in the quantized representation.

Binary Representation Strategy

In practice, we discard the lower $(\infty - R)$ bits and retain only the R MSBs. This process simplifies signal storage and processing, but introduces quantization error.

As an example:

$$x_n = 0.110011\ldots \quad \text{(binary)}$$

Truncating to $R = 3$ MSBs gives:

$$x_{n,q} = 0.110$$

and the residual becomes:

$$x_{\text{res}} = x_n - x_{n,q}$$

This residual is often small, but it can accumulate and affect signal fidelity depending on the application.

Limitations of Hardware to Accurately Represent a Signal

When the reconstruction of the signal is not perfect due to hardware limitations, we face the issue of finite-bit representation. Let:

$$x_n = -a_{n,0}2^m + \sum_{\gamma=1}^{\infty} a_{n,\gamma}2^{m-\gamma}$$

If we approximate x_n using only the R most significant bits (MSBs), we denote this approximation as $x_{n,q}$, and define the residual as:

$$x_{\text{res}} = x_n - x_{n,q}$$

This residual represents the error or loss due to quantization or approximation. The quantized value can be written as:

$$x_{n,q} = -q_{n,0}2^m + \sum_{\gamma=1}^{R-1} q_{n,\gamma}2^{m-\gamma}$$

Here, $q_{n,\gamma}$ are the bits retained in the quantized representation.

Binary Representation Strategy

In practice, we discard the lower $(\infty - R)$ bits and retain only the R MSBs. This process simplifies signal storage and processing, but introduces quantization error.

As an example:

$$x_n = 0.110011\ldots \quad \text{(binary)}$$

Truncating to $R = 3$ MSBs gives:

$$x_{n,q} = 0.110$$

and the residual becomes:

$$x_{\text{res}} = x_n - x_{n,q}$$

This residual is often small, but it can accumulate and affect signal fidelity depending on the application.

Quantization Error and Its Bound

Let the quantization error be given by:

$$x_n - x_{n,q} = x_{\text{res}}$$

We aim to find an upper bound on this error. Consider the binary expansion of x_n, and suppose we retain only R MSBs.

Let:

$$x_n = -a_{n,0}2^m + \sum_{\gamma=1}^{\infty} a_{n,\gamma}2^{m-\gamma}$$

$$x_{n,q} = -a_{n,0}2^m + \sum_{\gamma=1}^{R-1} a_{n,\gamma}2^{m-\gamma}$$

$$x_{\text{res}} = \sum_{\gamma=R}^{\infty} a_{n,\gamma}2^{m-\gamma}$$

Since each $a_{n,\gamma} \in \{0, 1\}$, the worst-case error is when all remaining bits are 1:

$$|x_{\text{res}}| \leq \sum_{\gamma=R}^{\infty} 2^{m-\gamma}$$

This is a geometric series:

$$|x_{\text{res}}| \leq 2^{m-R+1}\left(\sum_{k=0}^{\infty}\left(\frac{1}{2}\right)^k\right) = 2^{m-R+1} \cdot \frac{1}{1-\frac{1}{2}} = 2^{m-R+2}$$

Hence, we can write:

$$|x_n - x_{n,q}| \leq \frac{2^m}{2^{R-2}} = 2^{m-R+2}$$

This shows that the quantization error decreases exponentially with the number of retained MSBs R. More bits retained implies a better approximation:

$$R \uparrow \Rightarrow x_{n,q} \approx x_n \Rightarrow x_{\text{res}} \downarrow$$

Connection Between Modulo Reconstruction and Prediction

- Recovery through prediction.
- If $\Delta \to \infty$, then no information is lost (trivial case).
- Understanding connection between problem of modulo unwrapping of discrete-time signal and the problem of predicting next sample of discrete-time signal from its past.

We use $\{x_k\}$ to reconstruct $\{f(nT_s)\}$.

$$f \in L^2, \quad \|f\|_2 < \infty$$

$$f \text{ bandlimited} \Rightarrow F(\omega) = 0 \text{ outside some } [-B, B]$$

As $\Delta \to \infty \Rightarrow \|f\|_\infty$ large.

$$F(\omega) \Rightarrow \text{Bandlimited}$$

Let $H(z) \leftarrow$ appropriate prediction function.

$$F(\omega) \to 0 \Rightarrow H(z) \to \infty$$

$$f(t) \Rightarrow \text{Bandlimited} \Rightarrow \|f\|_\infty \text{ large} \Rightarrow \text{Prediction becomes accurate}$$

Using Prediction for Modulo Reconstruction

We use this *unfolded* sample for prediction of the prediction filter:

$$x_n = \begin{cases} x_n, & |n| \leq N \\ 0, & \text{otherwise} \end{cases}$$

$$x_n = \sum_{m=-N}^{N} x_m \cdot h_{n-m}$$

Here, h is the prediction filter.
Let the estimate be $\hat{x}_n$. Let the error be $e_n = x_n - \hat{x}_n$
From earlier,

$$\hat{x}_n = \sum_{k=1}^{\infty} a_k x_{n-k} \Rightarrow e_n = x_n - \sum_{k=1}^{\infty} a_k x_{n-k}$$

What do we *have*? $\to x_n$ modulo folded.
What do we *estimate*? $\to x_n$
So, we reconstruct via:

$$x_n - \hat{x}_n = e_n \Rightarrow x_n = \hat{x}_n + e_n$$

We attempt to estimate x_n from previous samples $x_{n-1}, x_{n-2}, \ldots$, and reduce the prediction error e_n.
Goal: Estimate x_n using prediction

$$\hat{x}_n = \sum_{k=1}^{\infty} a_k x_{n-k} \Rightarrow e_n = x_n - \hat{x}_n$$

We estimate x_n as:

$$x_n = \hat{x}_n + e_n$$

For $n \leq N$, we already know $x_n \Rightarrow$ We can compute $\hat{x}_n \Rightarrow$ We can find $e_n = x_n - \hat{x}_n$

But for $n > N$, we do not know $x_n \Rightarrow$ We do not know $e_n \Rightarrow$ We **estimate** e_n from a filter (say, use the previous values of e_k)

Let:

$$\hat{e}_n \approx e_n \Rightarrow \hat{x}_n + \hat{e}_n \approx x_n$$

$$\hat{e}_n = (x_n - \hat{x}_n) \approx \text{prediction error} \Rightarrow x_n \approx \hat{x}_n + \hat{e}_n$$

Modulo constraint:

$$|e_n| < \frac{\Delta}{2}$$

So, ensure error in estimate is below this value $\Rightarrow$ stability of filter is important $\Rightarrow$ need to always satisfy this condition **Key idea:** Use unfolded x_n using:

$$x_n = \hat{x}_n + e_n \Rightarrow e_n = x_n - \hat{x}_n$$

Condition for correct recovery:

$$|e_n| < \frac{\Delta}{2}$$

So recovery of x_n is possible if:

$$|x_n - \hat{x}_n| < \frac{\Delta}{2}$$

Strategy: Use a predictor:

$$\hat{x}_n = \sum_{k=1}^{\infty} a_k x_{n-k}$$

Let $e_n = x_n - \hat{x}_n$

Then,

$$|e_n|^2 = \sum_{n \in \mathbb{Z}} |x_n - \sum_{k=1}^{\infty} a_k x_{n-k}|^2 = \sum_{n \in \mathbb{Z}} |e_n|^2$$

Minimize: Total squared error

Let x_n be generated by a bandlimited function:

$$x_n = f(nT), \quad \text{with } f(\omega) = 0 \text{ for } |\omega| > \omega_m$$

So x_n is in the span of $\text{sinc}(n-k)$ basis for $|k| \leq \omega_m$

(You have a rough diagram near $\omega_m = 5$; that may mean frequency cutoff is 5 rad/sec.)

Use optimal predictor in this bandlimited setting to ensure:

$$|e_n| < \frac{\Delta}{2} \Rightarrow \text{Successful unfolding}$$

Monotonic Filter (Causal): Assume $c[0] = 1$

$$c[n] = \delta[n] - h[n]$$

If $h[n] = \delta[n] - c[n]$ (from above), then:

$$c[n] = \delta[n] - h[n] \Rightarrow C(z) = 1 - H(z)$$

rest of the understood part is in my notebook...couldn't get it over here...will explain from notebook