Introduction to The Time-adaptive Filter Repository

Xingzhi (Taylor) Zhang

February 2017

1 About the repository

Prof. Achim Kempf and his Ph.D student Yufang Hao proposed a generalized sampling theory to filter and reconstruct signals with time-varying bandwidths [1], [2], [3], [4]. Different from the existing time-varying generalizations [5], [6], [7],Kempf and Hao's method can process signals with higher accuracy and reduce the Gibbs phenomenon to a great extent [3], [4]. For the related mathematical framework and proof, see [1], [2].

The generalized sampling theory consists of signal filtering and reconstruction under time-varying bandwidth. In their papers [1], [2], Kempf and Hao present a time-adaptive filtering formula and a time-adaptive reconstruction formula to sample, reconstruct and filter signals on non-uniform sampling grids. As Prof. kempf's summer student, I implemented the two formulas with MATLAB and applied them to real signals. This Github repository stores the MATLAB implementation of the formulas and other relevant MATLAB functions. I will briefly introduce the repository content and the formulas implemented.

2 A brief introduction to the formulas

2.1 The variables included

The time-adaptive reconstruction formula is able to reconstruct signal from a set of non-uniformly sampled discrete values [1], [2]. It reads

$$\phi(t) = \sum_{n=-\infty}^{\infty} \phi(t_n) G\left(t, t_n\right) \tag{1}$$

where $G(t, t_n)$ is the time-adaptive reconstruction kernel

$$G\left(t, t_n\right) = (-1)^{z(t,t_n)} \frac{\sqrt{{t_n}'}}{|t - t_n|} \left(\sum_{m=-\infty}^{\infty} \frac{{t_m}'}{(t - t_m)^2} \right)^{-\frac{1}{2}} \tag{2}$$

.

The time-adaptive filtering formula can filter a signal along a time-varying bandwidth [1], [2]. It reads

$$\phi_{filtered}(t) = \int_{-\infty}^{\infty} \phi_{raw}(\hat{t}) P(t, \hat{t}) \nu(\hat{t}) d\hat{t} \tag{3}$$

And $P(t, t_n)$ is the time-adaptive filtering kernel

$$\begin{aligned} P(t, t_n) &= \sum_n G\left(t, t_n\right) G\left(t_n, \hat{t}\right) \\ &= (-1)^{z(t,\hat{t})} \sum_n \frac{|{t_n}'|}{\left(t - t_n\right)\left(\hat{t} - t_n\right)} \\ &\quad \times \left(\sum_m \frac{{t_m}'}{(t - t_m)^2}\right)\left(\sum_m \frac{{t_m}'}{(\hat{t} - t_m)^2}\right) \end{aligned} \tag{4}$$

The formulas derived by Kempf and Hao are in integral form since they process continuous signals. When processing discrete signals, the formula should be rewritten to meet the requirements. For discrete signals, filtering and reconstruction correspond to down-sampling and up-sampling, respectively. The time-adaptive filter can be written as

$$\phi_{filtered}\left(t_r\right) = \frac{1}{2\Omega_{raw}} \sum_{n=-\infty}^{\infty} \phi_{raw}\left(t_n\right) P\left(t_r, t_n\right) \nu\left(t_n\right) \tag{5}$$

where $P\left(t_r, t_n\right)$ is

$$P\left(t_r, t_n\right) = (-1)^{z\left(t_r, t_n\right)} \frac{\sqrt{{t_r}'}}{\left|t_r - t_n\right|} \left(\sum_m \frac{{t_m}'}{\left(t_n - t_m\right)^2}\right)^{-\frac{1}{2}} \tag{6}$$

There are two sampling grids included in the formula. By Shannon Sampling Theorem [8], a discrete signal can be down-sampled from a dense sampling grid to a sparse sampling grid when being filtered. Here, $\{t_n\}$ is the original dense sampling grid of the input signal and $\{t_r\}$ is the sparse sampling grid of the output signal. Under time-varying bandwidth, $\{t_r\}$ is no longer an equidistant sampling grid since the Nyquist rate changes with time. $\{t_m\}$ is the same sampling grid as $\{t_r\}$, and the subscript is changed to m to avoid confusion. Ω_{raw} is the bandwidth of the raw signal, which is a constant.

${t_r}'$ is the variable that indicates the time-varying bandwidth. According to Kempf and Hao [1], the lower the bandwidth at t_r is, the larger the value of ${t_r}'$ is. Numerically, ${t_r}'$ is the spacing between t_r and t_{r+1} (there is a more sophisticated method to calculate ${t_r}'$ [2], but the approximation using spacing is sufficient).

The function $z\left(t_r, t_n\right)$ gives the number of sample points that lie between t_n and t_r on the $\{t_r\}$ sampling grid. $\nu(t_n)$ is another function representing the bandwidth. In the implementation, its value is determined by two times the new bandwidth at t_n, which is $2\Omega_{filtered}(t_n)$ (recall that in Shannon filter, $\nu(t_n)$ is just $2\Omega_{filtered}$).

Similarly, the time-adaptive reconstruction formula for discrete signals reads

$$\phi(t_n) = \sum_{r=-\infty}^{\infty} \phi(t_r) G(t_n, t_r) \tag{7}$$

where

$$G(t_n, t_r) = (-1)^{z(t_n, t_r)} \frac{\sqrt{t_r{}'}}{|t_n - t_r|} \left(\sum_m \frac{t_m{}'}{(t_n - t_m)^2} \right)^{-\frac{1}{2}} \tag{8}$$

All the variables are the same as in the filtering formula.

2.2 Relationship with Shannon's Sampling Theory

Under constant bandwidth, Shannon Sampling Theory can be recovered as a special case [1]. If the bandwidth is set to a constant, the variable t' also becomes a constant. Then, the $t_m{}'$ and $t_r{}'$ will cancel each other. The time adaptive reconstruction kernel now becomes

$$G(t_n, t_r) = (-1)^{z(t_n, t_r)} \frac{1}{|t_n - t_r|} \left(\sum_m \frac{1}{(t_n - t_m)^2} \right)^{-\frac{1}{2}} \tag{9}$$

By the following trigonometric identity

$$\left(\frac{\pi}{sin\,(\pi z)} \right)^2 = \sum_{k=-\infty}^{\infty} \frac{1}{(z-k)^2} \tag{10}$$

The kernel can be further reduced to

$$G(t_n, t_r) = sinc\,(\Omega(t_n - t_r)) \tag{11}$$

which is the Shannon signal reconstruction kernel. The same identity can be applied to the time-adaptive filter kernel, reducing it to Shannon's sinc filter kernel.

2.3 The generation of the non-uniform sampling grid

Under time-varying bandwidth, the sampling grid is no longer uniform. Kempf and Hao have proved that the spacing of the sampling grid at an instance t is always $\frac{1}{2\Omega(t)}$ [1]. If a single t_r is known, the complete sampling grid can be found by doing $t_{r+1} = t_r + \frac{1}{2\Omega(t_r)}$ iteratively. Different from the previous section, the bandwidth information is given as a set of discrete values instead of a formula. The bandwidth at a particular instance can still be calculated using linear interpolation since the bandwidth changes slowly over the day. The grid calculation algorithm is shown in Fig. 1.

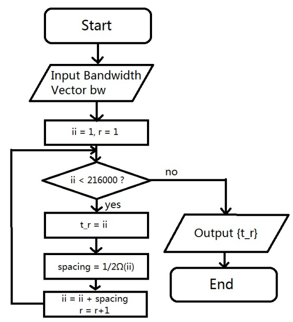

Figure 1: The Flow Chart of the Grid Calculation Algorithm

Ideally, the non-uniform sampling grid, time-adaptive filter and reconstruction kernel are all infinite. However, we are always dealing with finite signals in application. From my observation, the generalized kernel decays to 0 (approximately) at the 500th point from its center, so the kernel only needs 1000 points from the sampling grid (500 on each side).

The generated sampling grid is also finite, so there would be insufficient sample points for the kernel when the center is less than 500 points away from the ends of the sampling grid. The solution is to linearly increase the grid with a constant spacing by 1000 points (500 at each end). The spacing would be $t_2 - t_1$ at the start of the grid, and would be $t_{last} - t_{last-1}$ at the end of the grid. This is equivalent to assume the time-varying bandwidth remains constant outside the interval where the signal is sampled. Therefore, the kernel also needs an extended sampling grid to complete the filtering and reconstruction.

The MATLAB script generating the grid is included in the repository.

3 The contents in the repository

There are six MATLAB files in the repository, including four functions and two scripts. Please notice that **the unit of frequency is Hertz [Hz] and the unit of sampling grid is second [s].**

3.1 calculate_grid.m

This is the script that generates the non-uniform sampling grid from a given time-varying bandwidth. The script needs two vectors: the equidistant dense sampling grid and the vector indicating the bandwidth value at each point in the dense sampling grid. The algorithm The algorithm is introduced in the previous section. The output of this script will be the sparse non-uniform sampling grid named "filtered_grid" and the linearly extended "longer_filtered_grid".

3.2 time_adaptive_filter

This function implements the time-adaptive filter. It takes five input arguments: sample, raw_grid, filtered_grid, longer_filtered_grid, and V_t. "sample" is a set of samples of the raw signal, and "raw_grid" is the dense sampling grid on which the samples are taken. "filtered_grid" and "longer_filtered_grid" are the two non-uniform sampling grids generated according to the time-varying bandwidth. "V_t" is a vector of scaling factors. "filtered_grid", "longer_filtered_grid", and "V_t" can be generated by the "calculate_grid" script. This function filters a signal along a time-varying bandwidth and down-sample it from "raw_grid" to "filtered_grid". The output of the function is a new set of samples taken on "filtered_grid".

3.3 time_adaptive_reconstruction

This function implements the time-adaptive reconstruction formula. Similar to the filter, it takes four inputs: sample, sample_grid, longer_sample_grid, and grid_points. Here, "sample" is the set of samples taken on the non-uniform sampling grid "sample_grid". "longer_sample_grid" is the linearly extended "sample_grid", and "grid_points" is the uniform dense sampling grid we want to up-sample the signal to. This function reconstructs/up-samples a signal from a set of non-uniformly taken samples along. The output is a new set of samples taken on the dense uniform sampling grid "grid_points".

3.4 time_varying_signal_generator

This function generates a sample signal that possesses a time-varying bandwidth. The sample signal is generated by adding time-adaptive-reconstruction kernels with different centers and amplitudes together. It takes the non-uniform sampling grid "grid" as input and outputs the signal as samples taken on "grid". The output still needs to be up-sampled by the time-adaptive reconstruction function.

3.5 white_noise_filtering

This is a script that applies the time-adaptive filtering and reconstruction MATLAB functions. The script needs two vectors: a uniform sampling grid and the bandwidth value at each sample point. The script first generates the non-uniform sampling grid according to the provided bandwidth and a piece of white noise. Then, the script time-dependently filter the white noise using the time-adaptive-filter along the provided bandwidth. The final output is the vector named "filtered_white_noise_long". One can plot the output and observe the result of time-adaptive filtering.

3.6 tafc

This function is the combination of the filtering and reconstruction kernels. In the previous implementations, the user has to run a separate script to generate sampling grids before the filtering. The tafc function takes the discrete raw signal, the uniform grid on which the raw signal is defined, and time-varying bandwidth as inputs, and directly outputs the signal after filtering.

References

[1] Y. Hao and A. Kempf. On a non-Fourier generalization of Shannon sampling theory. *Proc. of the 10th Canadian Workshop on Information Theory CWIT'07*, pages 193–196, 2007.

[2] Y. Hao and A. Kempf. Significant reduction of Gibbs overshoot with generalized sampling method. *Proc. 8th International Conference of Sampling Theory and Applications SampTA'09*, 2009.

[3] Y. Hao and A. Kempf. Filtering, sampling, and reconstruction with time-varying bandwidths. *IEEE Signal Processing Letters*, 14(3):241, 2010.

[4] Y. Hao and A. Kempf. Generalized Shannon sampling method reduces the Gibbs overshoot in the approximation of a step function. *Journal of Concrete and Applicable Mathematics*, 8(3):540–554, 2010.

[5] H.J. Landau. Necessary density conditions for sampling and interpolation of certain entire functions. *Acta Math*, 117:37–52, 2009.

[6] R. Paley and N. Wiener. *Fourier Transforms in the complex domain*, volume 19. American Math Soc. Colloq., 1934.

[7] J.L. Yen. On non-uniform sampling of band-limited signals. *IRE Trans. on Circuit Theory*, pages 251–257, 1956.

[8] A. Zayed. *Advances in Shannon's Sampling Theory.* CRC Press, 1993.