Two distinct methods for synthesizing a signal from its short-time Fourier transform have previously been proposed. We called these methods the filter-bank summation (FBS) method and the over-lap add (OLA) method .

In the ***weighted overlap add*** **(WOLA)** method, we apply a *second window* after the inverse DFT and prior to the final overlap-add to create the output signal. Such a window can be called a synthesis window, postwindow, or simply output window.''

VMD

It is an entirely non-recursive variational mode decomposition model, where the modes are extracted concurrently. The model looks for an ensemble of modes and their respective center frequencies, such that the modes collectively reproduce the input signal, while each being smooth after demodulation into baseband. In Fourier domain, this corresponds to a narrow-band prior. We show important relations to Wiener filter denoising. Indeed, the proposed method is a generalization of the classic Wiener filter into multiple, adaptive bands. Our model provides a solution to the decomposition problem that is theoretically well founded and still easy to understand. The variational model is efficiently optimized using an alternating direction method of multipliers approach. Preliminary results show attractive performance with respect to existing mode decomposition models. In particular, our proposed model is much more robust to sampling and noise. Finally, we show promising practical decomposition results on a series of artificial and real data.[1]**https://ieeexplore.ieee.org/document/6655981**

The goal of VMD is to decompose a real valued input signal f into a discrete number of sub-signals (modes), u_k, that have specific sparsity properties while reproducing the input. Here,the sparsity prior of each mode is chosen to be its bandwidth in spectral domain. In other words, we assume each mode k to be mostly compact around a center pulsation ω_k, which is to be determined along with the decomposition .

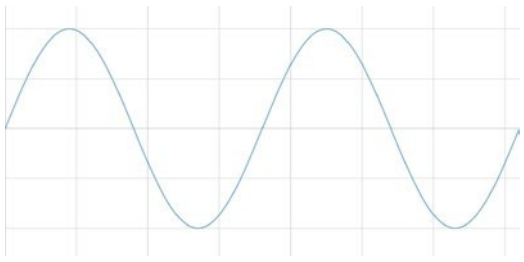
Real-world signals are usually viewed as a voltage changing over time. This is referred to as the time domain. Fourier's theorem states that any waveform in the time domain can be represented by the weighted sum of sines and cosines. The Fourier transform deconstructs a time domain representation of a signal into the frequency domain representation. The frequency domain shows the voltages present at varying frequencies. It is a different way to look at the same signal.

A digitizer samples a waveform and transforms it into discrete values. Because of this transformation, the Fourier transform will not work on this data. Instead, the discrete Fourier transform (DFT) is used, which produces as its result the frequency domain components in discrete values, or bins. The fast Fourier (FFT) is an optimized implementation of a DFT that takes less computation to perform but essentially just deconstructs a signal. The frequency domain is great at showing if a clean signal in the time domain actually contains cross talk, noise, or jitter.

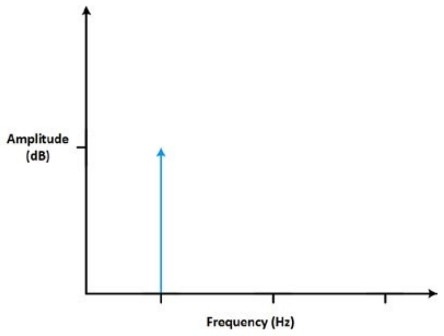
Windowing Although performing an FFT on a signal can provide great insight, it is important to know the limitations of the FFT and how to improve the signal clarity using windowing.

What Is Windowing

When you use the FFT to measure the frequency component of a signal, you are basing the analysis on a finite set of data. The actual FFT transform assumes that it is a finite data set, a continuous spectrum that is one period of a periodic signal. For the FFT, both the time domain and the frequency domain are circular topologies, so the two endpoints of the time waveform are interpreted as though they were connected together. When the measured signal is periodic and an integer number of periods fill the acquisition time interval, the FFT turns out fine as it matches this assumption.



(A)

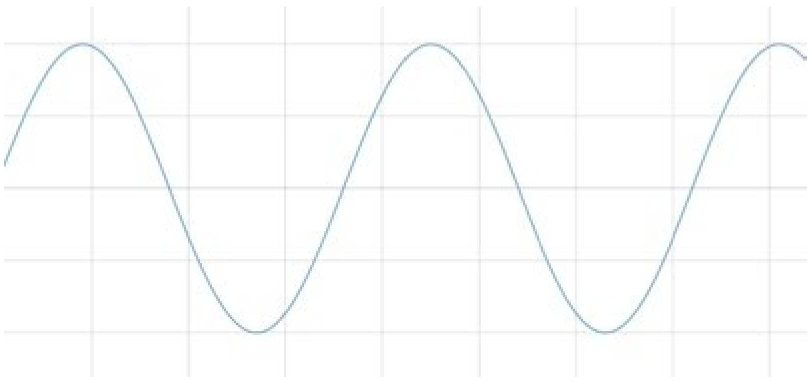


(B)

Figure : Measuring an integer number of periods (A) gives an ideal FFT (B).

However, many times, the measured signal isn't an integer number of periods. Therefore, the finiteness of the measured signal may result in a truncated waveform with different characteristics from the original continuous-time signal, and the finiteness can introduce sharp transition changes into the measured signal. The sharp transitions are discontinuities.

When the number of periods in the acquisition is not an integer, the endpoints are discontinuous. These artificial discontinuities show up in the FFT as high-frequency components not present in the original signal. These frequencies can be much higher than the Nyquist frequency and are aliased between 0 and half of your sampling rate. The spectrum you get by using a FFT, therefore, is not the actual spectrum of the original signal, but a smeared version. It appears as if energy at one frequency leaks into other frequencies. This phenomenon is known as spectral leakage, which causes the fine spectral lines to spread into wider signals.



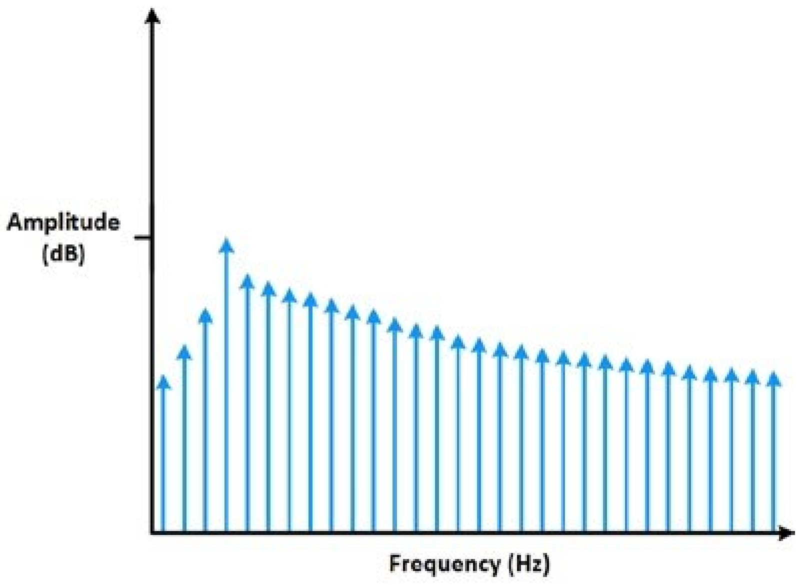


Figure : Measuring a noninteger number of periods (A) adds spectral leakage to the FFT (B).

You can minimize the effects of performing an FFT over a noninteger number of cycles by using a technique called windowing. Windowing reduces the amplitude of the discontinuities at the boundaries of each finite sequence acquired by the digitizer. Windowing consists of multiplying the time record by a finite-length window with an amplitude that varies smoothly and gradually toward zero at the edges. This makes the endpoints of the waveform meet and, therefore, results in a continuous waveform without sharp transitions. This technique is also referred to as applying a window.

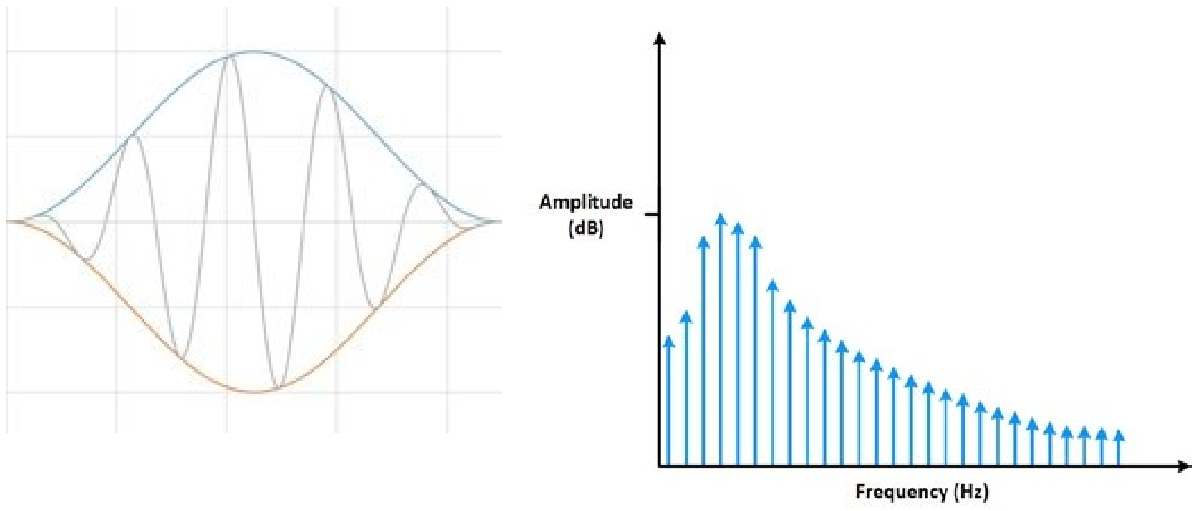


Figure: Applying a window minimizes the effect of spectral leakage.

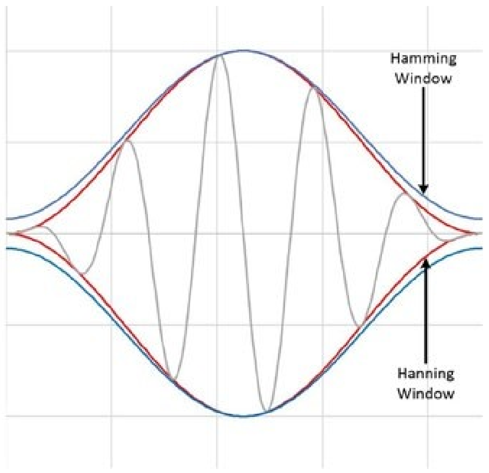
Windowing Functions

There are several different types of window functions that you can apply depending on the signal. To understand how a given window affects the frequency spectrum, you need to understand more about the frequency characteristics of windows.
An actual plot of a window shows that the frequency characteristic of a window is a continuous spectrum with a main lobe and several side lobes. The main lobe is centered at each frequency component of the time-domain signal, and the side lobes approach zero. The height of the side lobes indicates the affect the windowing function has on frequencies around main lobes. The side lobe response of a strong sinusoidal signal can overpower the main lobe response of a nearby weak sinusoidal signal. Typically, lower side lobes reduce leakage in the measured FFT but increase the bandwidth of the major lobe. The side lobe roll-off rate is the asymptotic decay rate of the side lobe peaks. By increasing the side lobe roll-off rate, you can reduce spectral leakage. Selecting a window function is not a simple task. Each window function has its own characteristics and suitability for different applications. To choose a window function, you must estimate the frequency content of the signal.

- If the signal contains strong interfering frequency components distant from the frequency of interest, choose a smoothing window with a high side lobe roll-off rate.
- If the signal contains strong interfering signals near the frequency of interest, choose a window function with a low maximum side lobe level.
- If the frequency of interest contains two or more signals very near to each other, spectral resolution is important. In this case, it is best to choose a smoothing window with a very narrow main lobe.
- If the amplitude accuracy of a single frequency component is more important than the exact location of the component in a given frequency bin, choose a window with a wide main lobe.
- If the signal spectrum is rather flat or broadband in frequency content, use the uniform window, or no window.
- In general, the Hanning (Hann) window is satisfactory in 95 percent of cases. It has good frequency resolution and reduced spectral leakage. If you do not know the nature of the signal but you want to apply a smoothing window, start with the Hann window.

Even if you use no window, the signal is convolved with a rectangular-shaped window of uniform height, by the nature of taking a snapshot in time of the input signal and working with a discrete signal. This convolution has a sine function characteristic spectrum. For this reason, no window is often called the uniform or rectangular window because there is still a windowing effect.

The **Hamming and Hann window** functions both have a sinusoidal shape. Both windows result in a wide peak but low side lobes. However, the Hann window touches zero at both ends eliminating all discontinuity. The Hamming window doesn’t quite reach zero and thus still has a slight discontinuity in the signal. Because of this difference, the Hamming window does a better job of cancelling the nearest side lobe but a poorer job of canceling any others. These window functions are useful for noise measurements where better frequency resolution than some of the other windows is wanted but moderate side lobes do not present a problem.



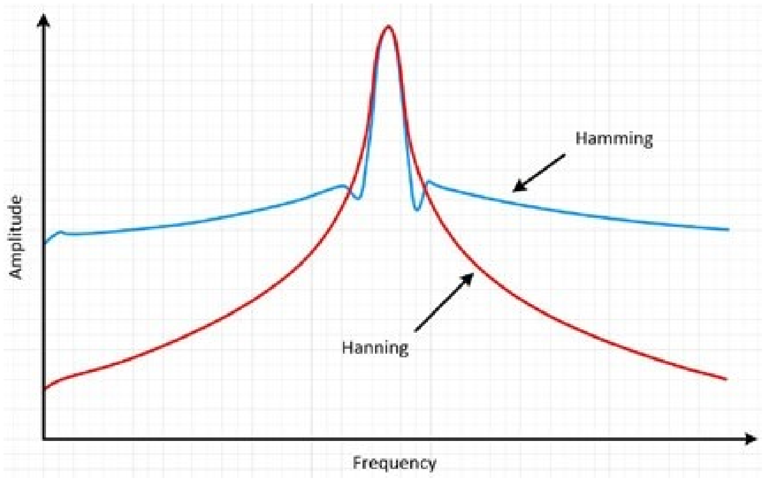
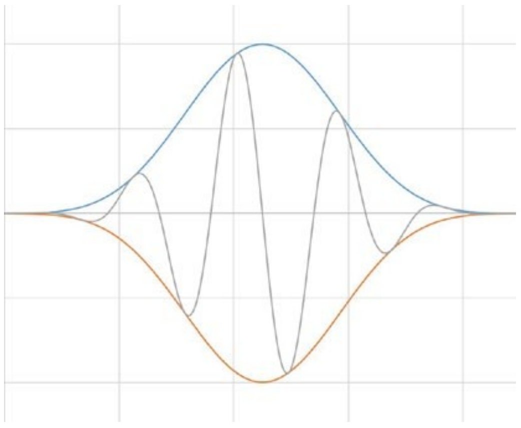


Figure : and Hann windowing result in a wide peak but nice low side lobes

The Blackman-Harris window is similar to Hamming and Hann windows. The resulting spectrum has a wide peak, but good side lobe compression. There are two main types of this window. The 4-term Blackman-Harris is a good general-purpose window, having side lobe rejection in the high 90s dB and a moderately wide main lobe. The 7-term Blackman-Harris window function has all the dynamic range you should ever need, but it comes with a wide main lobe.



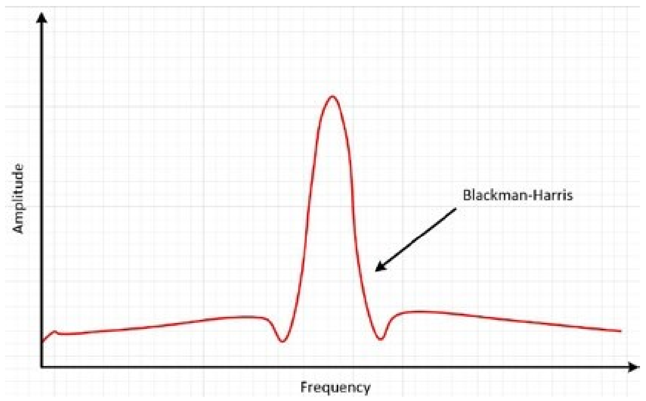


Figure : The Blackman-Harris results in a wide peak, but good side lobe compression.

A Kaiser-Bessel window strikes a balance among the various conflicting goals of amplitude accuracy, side lobe distance, and side lobe height. It compares roughly to the Blackman-Harris window functions, but for the same main lobe width, the near side lobes tend to be higher, but the further out side lobes are lower. Choosing this window often reveals signals close to the noise floor. The flat top window is sinusoidal as well, but it actually crosses the zero line. This causes a much broader peak in the frequency domain, which is closer to the true amplitude of the signal than with other windows. These are just a few of the possible window functions. There is no universal approach for selecting a window function. However, the table below can help you in your initial choice. Always compare the performance of different window functions to find the best one for the application.

In order to assess the bandwidth of a mode, we propose the following scheme: 1) for each mode u_k, compute the associated analytic signal by means of the Hilbert transform in order to obtain a unilateralfrequency spectrum. 2) for each mode, shift the mode's frequency spectrum to "baseband", by mixing with an exponential tuned to the respective estimated center frequency. 3) The bandwidth is now estimated through the H1 Gaussian smoothness of the demodulated signal, i.e. the squared L2-norm of the gradient. The resulting constrained variational problem is the following:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}$$
$$\text{s.t.} \quad \sum_k u_k = f \qquad (14)$$

where $\{u_k\} := \{u_1, \ldots, u_K\}$ and $\{\omega_k\} := \{\omega_1, \ldots, \omega_K\}$ are shorthand notations for the set of all modes and their center frequencies, respectively. Equally, $\sum_k := \sum_{k=1}^K$ is understood as the summation over all modes.

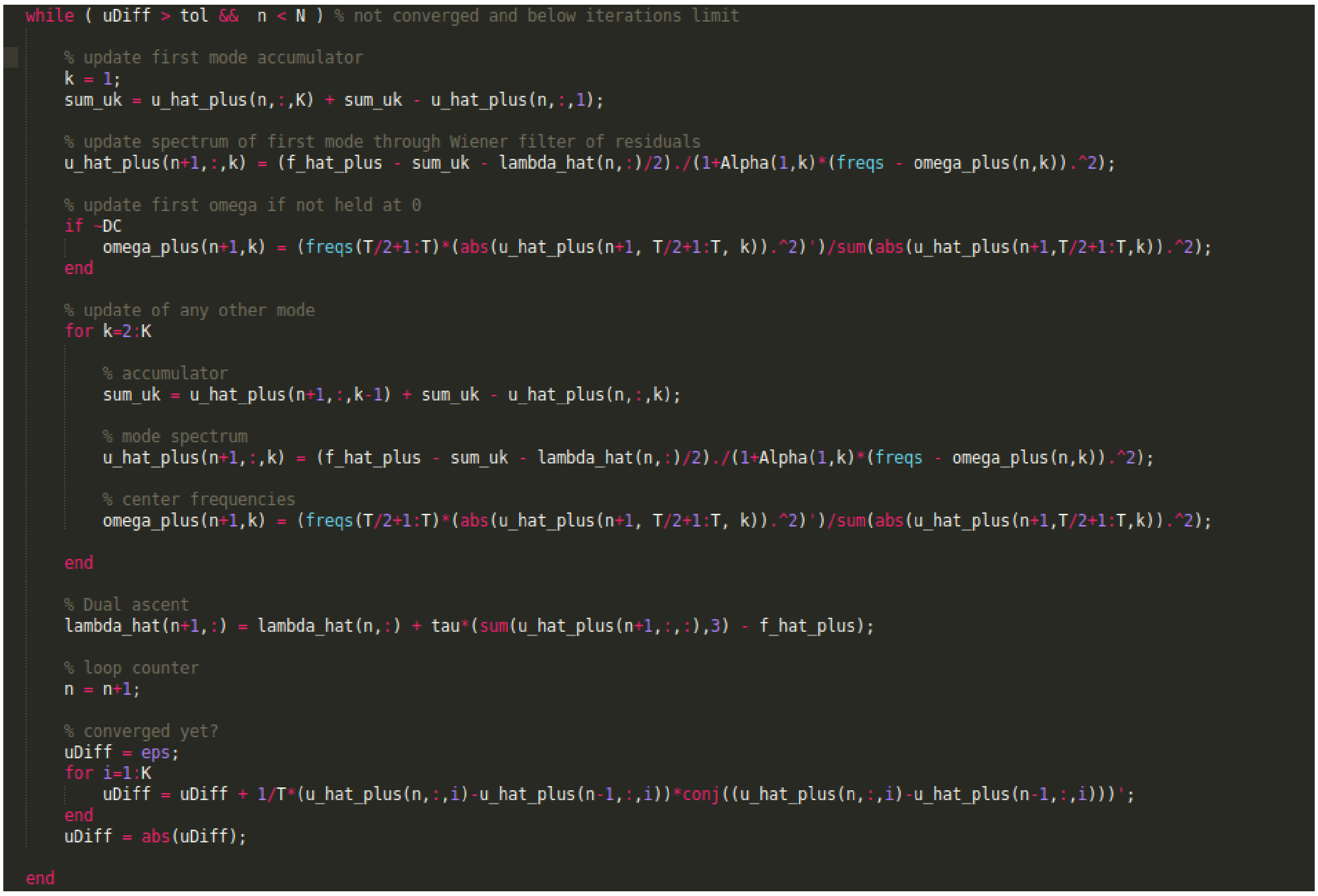
The reconstruction constraint can be addressed in different ways. Here, we suggest making use of both a quadratic penalty term and Lagrangian multipliers, in order to render the problem unconstrained. The quadratic penalty is a classic way to encourage reconstruction fidelity, typically in the presence of additive i.i.d. Gaussian noise. The weight of the penalty term is derived from such a Bayesian prior to be inversely proportional to the noise level in the data. Conversely, in a noisefree setting, the weight needs to be infinitely big in order to enforce strict data fidelity, rendering the system ill- conditioned in the process. On the other hand, Lagrangian multipliers are a common way of enforcing constraints strictly. The combination of the two terms thus benefits both from the nice convergence properties of the quadratic penalty at finite weight, and the strict enforcement of the constraint by the Lagrangian multiplier. Therefore, we introduce the augmented Lagrangian multiplier as follows :

$$\mathcal{L}(\{u_k\}, \{\omega_k\}, \lambda) := \alpha \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_k u_k(t) \right\rangle .$$

Algorithm 1 ADMM optimization concept for VMD

```
Initialize $u_k^1$, $\omega_k^1$, $\lambda^1$, $n \leftarrow 0$
repeat
    $n \leftarrow n + 1$
    for $k = 1 : K$ do
        Update $u_k$:
$$u_k^{n+1} \leftarrow \arg\min_{u_k} \mathcal{L}(u_1^{n+1}, \dots, u_{k-1}^{n+1}, u_k, u_{k+1}^n, \dots, u_K^n, \omega_1^n, \dots, \omega_K^n, \lambda^n) \quad (18)$$
    end for
    for $k = 1 : K$ do
        Update $\omega_k$:
$$\omega_k^{n+1} \leftarrow \arg\min_{\omega_k} \mathcal{L}(u_1^{n+1}, \dots, u_K^{n+1}, \omega_1^{n+1}, \dots, \omega_{k-1}^{n+1}, \omega_k, \omega_{k+1}^n, \dots, \omega_K^n, \lambda^n) \quad (19)$$
    end for
    Dual ascent:
$$\lambda^{n+1} \leftarrow \lambda^n + \tau \left( f - \sum_k u_k^{n+1} \right) \quad (20)$$
until convergence: $\sum_k \|u_k^{n+1} - u_k^n\|_2^2 / \|u_k^n\|_2^2 < \epsilon$.
```

(Code Snippet from VMD.m script for the above algorithm)



```
while ( uDiff > tol &&  n < N ) % not converged and below iterations limit

    % update first mode accumulator
    k = 1;
    sum_uk = u_hat_plus(n,:,K) + sum_uk - u_hat_plus(n,:,1);

    % update spectrum of first mode through Wiener filter of residuals
    u_hat_plus(n+1,:,k) = (f_hat_plus - sum_uk - lambda_hat(n,:)/2)./(1+Alpha(1,k)*(freqs - omega_plus(n,k)).^2);

    % update first omega if not held at 0
    if ~DC
        omega_plus(n+1,k) = (freqs(T/2+1:T)*(abs(u_hat_plus(n+1, T/2+1:T, k)).^2)')/sum(abs(u_hat_plus(n+1,T/2+1:T,k)).^2);
    end

    % update of any other mode
    for k=2:K

        % accumulator
        sum_uk = u_hat_plus(n+1,:,k-1) + sum_uk - u_hat_plus(n,:,k);

        % mode spectrum
        u_hat_plus(n+1,:,k) = (f_hat_plus - sum_uk - lambda_hat(n,:)/2)./(1+Alpha(1,k)*(freqs - omega_plus(n,k)).^2);

        % center frequencies
        omega_plus(n+1,k) = (freqs(T/2+1:T)*(abs(u_hat_plus(n+1, T/2+1:T, k)).^2)')/sum(abs(u_hat_plus(n+1,T/2+1:T,k)).^2);

    end

    % Dual ascent
    lambda_hat(n+1,:) = lambda_hat(n,:) + tau*(sum(u_hat_plus(n+1,:,:),3) - f_hat_plus);

    % loop counter
    n = n+1;

    % converged yet?
    uDiff = eps;
    for i=1:K
        uDiff = uDiff + 1/T*(u_hat_plus(n,:,i)-u_hat_plus(n-1,:,i))*conj((u_hat_plus(n,:,i)-u_hat_plus(n-1,:,i)))';
    end
    uDiff = abs(uDiff);

end
```

The solution to the original minimization problem (14) is now found as the saddle point of the augmented Lagrangian *L* in a sequence of iterative sub-optimizations called alternate direction method of multipliers (ADMM) [34]–[36], see algorithm 1. In the next paragraphs, we will then detail how the respectivesub-problems can be solved.

Minimization w.r.t. u_k:

To update the modes u_k , we first rewrite the subproblem (16) as the following equivalent minimization problem:

$$u_k^{n+1} = \underset{u_k \in X}{\arg\min} \left\{ \alpha \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_i u_i(t) + \frac{\lambda(t)}{2} \right\|_2^2 \right\}, \quad ($$

where for simplicity we omit .^{n} and .$^{n+1}$ for the fixed directions and , respectively, but each is implicitly understood as the most recent available update. Now, making use of the Parseval/Plancherel Fourier isometry under the norm, this problem can be solved in spectral domain:

$$\hat{u}_k^{n+1} = \underset{\hat{u}_k, u_k \in X}{\arg\min} \left\{ \alpha \left\| j\omega \left[(1 + \operatorname{sgn}(\omega + \omega_k))\, \hat{u}_k(\omega + \omega_k) \right] \right\|_2^2 + \left\| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right\|_2^2 \right\}. \quad (20)$$

We now perform a change of variables $\omega \leftarrow \omega - \omega_k$ *in the first term:*

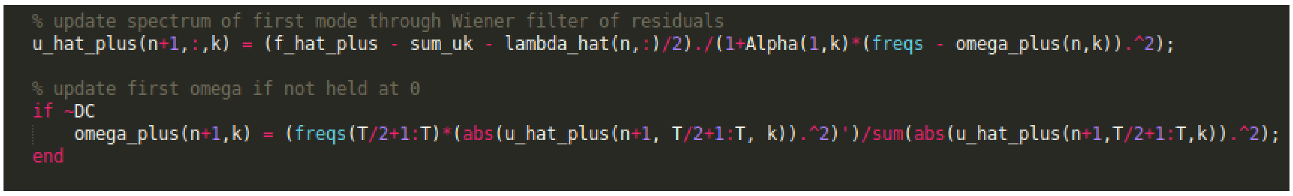
$$\hat{u}_k^{n+1} = \underset{\hat{u}_k, u_k \in X}{\arg\min} \left\{ \alpha \left\| j(\omega - \omega_k) \left[(1 + \operatorname{sgn}(\omega))\, \hat{u}_k(\omega) \right] \right\|_2^2 + \left\| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right\|_2^2 \right\}. \quad (21)$$

Exploiting the Hermitian symmetry of the real signals in the reconstruction fidelity term, we can write both terms as halfspace integrals over the non-negative frequencies:

$$\hat{u}_k^{n+1} = \underset{\hat{u}_k, u_k \in X}{\arg\min} \left\{ \int_0^\infty 4\alpha(\omega - \omega_k)^2 \left| \hat{u}_k(\omega) \right|^2 + 2 \left| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right|^2 d\omega \right\}. \quad (22)$$

The solution of this quadratic optimization problem is readily found by letting the first variation vanish for the positive frequencies:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2},$$



```
% update spectrum of first mode through Wiener filter of residuals
u_hat_plus(n+1,:,k) = (f_hat_plus - sum_uk - lambda_hat(n,:)/2)./(1+Alpha(1,k)*(freqs - omega_plus(n,k)).^2);

% update first omega if not held at 0
if ~DC
    omega_plus(n+1,k) = (freqs(T/2+1:T)*(abs(u_hat_plus(n+1, T/2+1:T, k)).^2)')/sum(abs(u_hat_plus(n+1,T/2+1:T,k)).^2);
end
```

(Code Snippet from VMD.m script)

which is clearly identified as a Wiener filtering of the current residual, with signal prior $1/(\omega-\omega_k)^2$ *. The full spectrum of the real mode is then simply obtained by Hermitian symmetric completion. Conversely, the mode in time domain is obtained as the real part of the inverse Fourier transform of this filtered analytic signal.*

Minimization w.r.t. ω_k **:**

The center frequencies ω_k do not appear in the reconstruction fidelity term, but only in the bandwidth prior. The relevant problem thus reads:

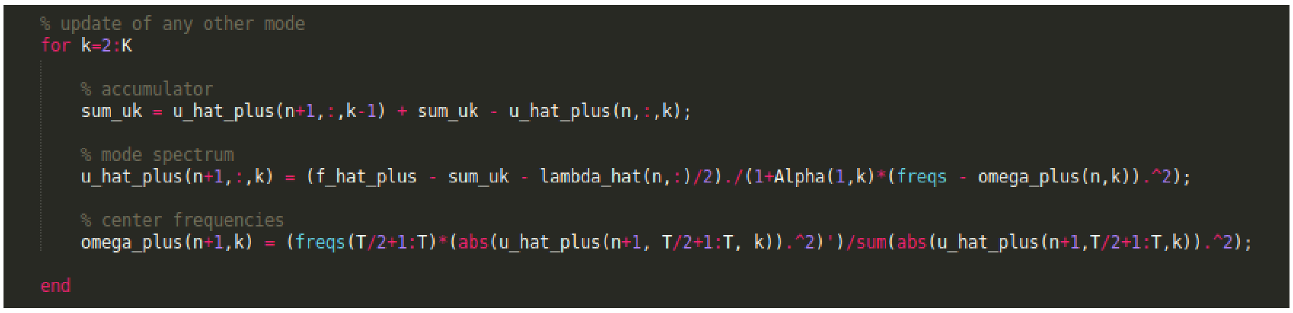
$$\omega_k^{n+1} = \arg\min_{\omega_k} \left\{ \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}.$$

As before, the optimization can take place in Fourier domain, and we end up optimizing:

$$\omega_k^{n+1} = \arg\min_{\omega_k} \left\{ \int_0^\infty (\omega - \omega_k)^2 \, |\hat{u}_k(\omega)|^2 \, d\omega \right\}$$

This quadratic problem is easily solved as:

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega \, |\hat{u}_k(\omega)|^2 \, d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 \, d\omega},$$



```
% update of any other mode
for k=2:K

    % accumulator
    sum_uk = u_hat_plus(n+1,:,k-1) + sum_uk - u_hat_plus(n,:,k);

    % mode spectrum
    u_hat_plus(n+1,:,k) = (f_hat_plus - sum_uk - lambda_hat(n,:)/2)./(1+Alpha(1,k)*(freqs - omega_plus(n,k)).^2);

    % center frequencies
    omega_plus(n+1,k) = (freqs(T/2+1:T)*(abs(u_hat_plus(n+1, T/2+1:T, k)).^2)')/sum(abs(u_hat_plus(n+1,T/2+1:T,k)).^2);

end
```

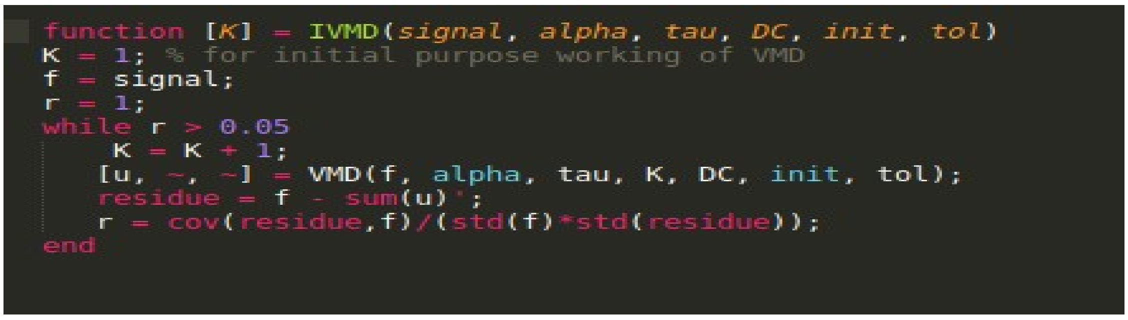
(Code snippet from VMD.m script for minimization w.r.t. omegas)

which puts the new ω_k at the center of gravity of the corresponding mode's power spectrum. This mean carrier frequency is the frequency of a least squares linear regression to the instantaneous phase observed in the mode. Plugging the solutions of the sub-optimizations into the ADMM algorithm 1, and directly optimizing in Fourier domain where appropriate, we get the complete algorithm for variational mode decomposition, summarized in algorithm.

Improved Variational Mode Decomposition :

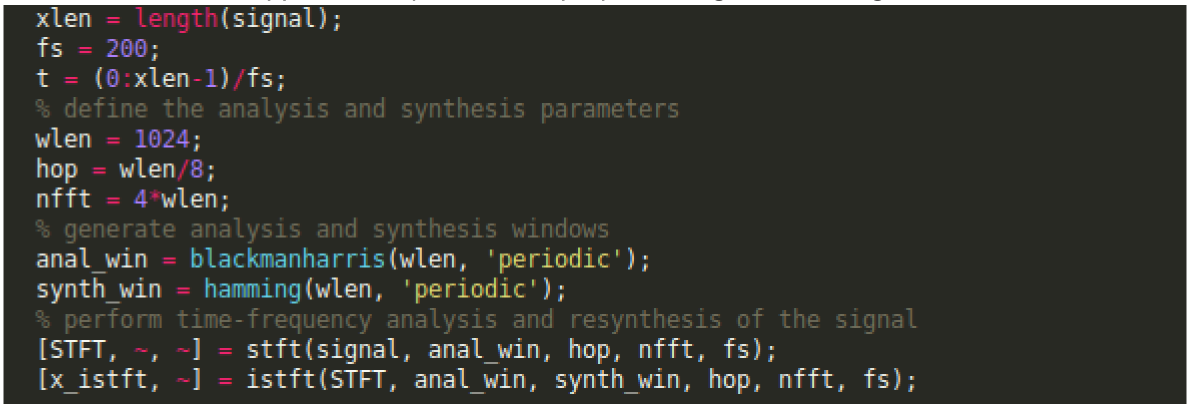
To overcome the unadaptable quadratic penalty problem, an improved VMD model via correlation coefficient and new update formulas are proposed to decompose ECG signals. To improve the denoising precision, this algorithm is combined with the interval threshold algorithm. First, the correlation coefficient is calculated, to determine quadratic penalty, in order to extract the first IMF made up of baseline drift. Then, the new update formulas derived from the variance that describes the noise level are used, to perform decomposition on the rest signal. Finally, the Interval thresholding algorithm is performed on each IMF.

(Code snippet from IVMD.m script to optimize K, number of modes)

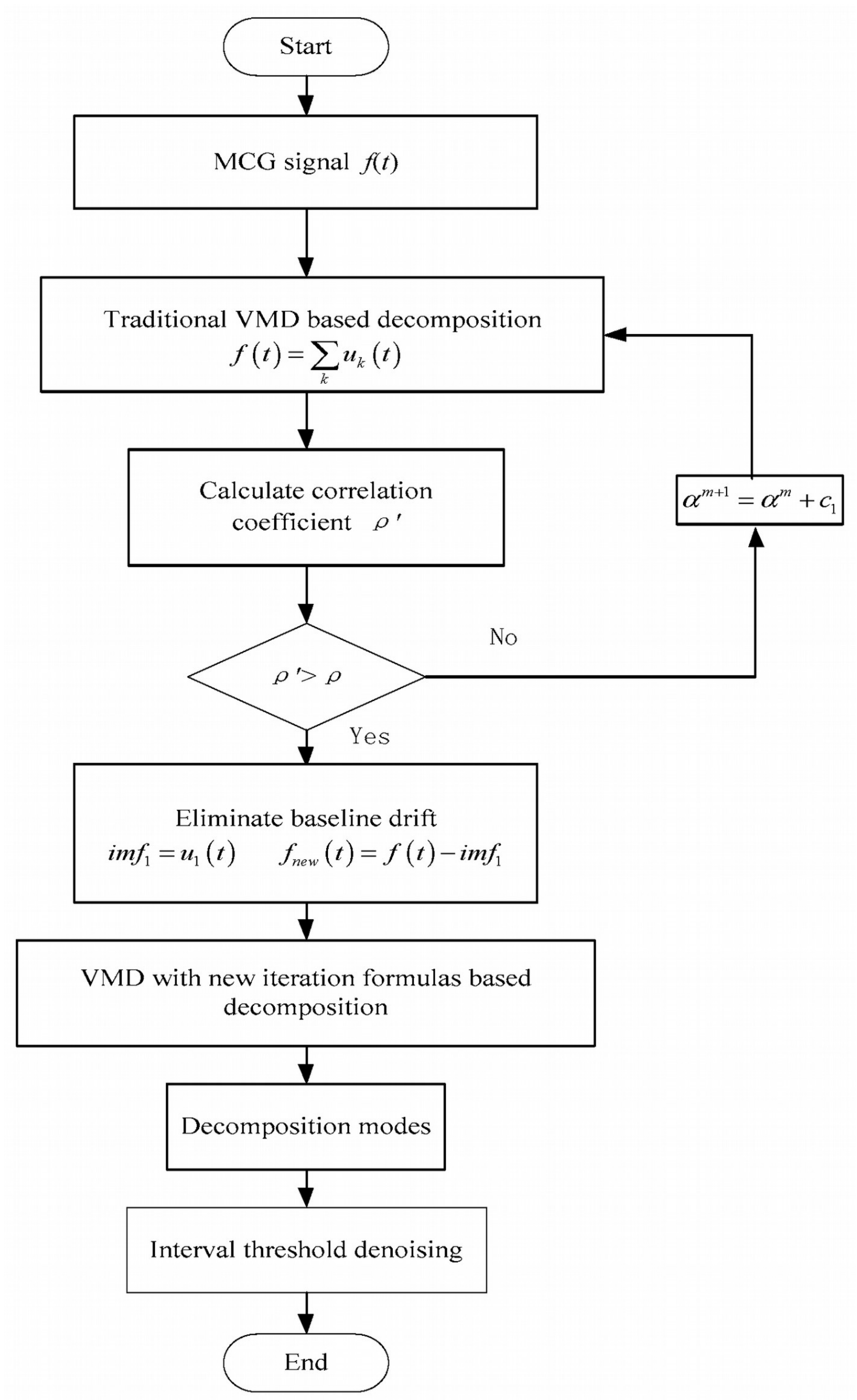


```
function [K] = IVMD(signal, alpha, tau, DC, init, tol)
K = 1; % for initial purpose working of VMD
f = signal;
r = 1;
while r > 0.05
     K = K + 1;
    [u, ~, ~] = VMD(f, alpha, tau, K, DC, init, tol);
    residue = f - sum(u)';
    r = cov(residue,f)/(std(f)*std(residue));
end
```

(code snippet from DyVMD.m for preprocessing for IVMD Algorithm)



```
xlen = length(signal);
fs = 200;
t = (0:xlen-1)/fs;
% define the analysis and synthesis parameters
wlen = 1024;
hop = wlen/8;
nfft = 4*wlen;
% generate analysis and synthesis windows
anal_win = blackmanharris(wlen, 'periodic');
synth_win = hamming(wlen, 'periodic');
% perform time-frequency analysis and resynthesis of the signal
[STFT, ~, ~] = stft(signal, anal_win, hop, nfft, fs);
[x_istft, ~] = istft(STFT, anal_win, synth_win, hop, nfft, fs);
```



Start
MCG signal $f(t)$
Traditional VMD based decomposition
$f(t)=\sum_k u_k(t)$
Calculate correlation coefficient ρ'
$\alpha^{m+1}=\alpha^m+c_1$
No
$\rho'>\rho$
Yes
Eliminate baseline drift
$imf_1=u_1(t)$ $f_{new}(t)=f(t)-imf_1$
VMD with new iteration formulas based decomposition
Decomposition modes
Interval threshold denoising
End