Optimum Design of a Window Function Based on the Small-World Networks

Lingbo Kong, Zhe Chen, Fuliang Yin
School of Electronic and Information, Dalian university of technology, Dalian, P.R. China 116023
eeyin@dlut.edu.cn

Abstract

Small-world networks are the networks with high clustering as well as short average path distances, So it is a networks with efficiently searching the global optimum solutions. In this paper, we write a program the principle of the small-world to carry out the design of window function. And the results have been shown that, this method is a very efficient one. The window functions that designed by this method also have the very good performance.

1. Introduction

Windowing is a common technique to shorten data in signal processing[7]. However, in some particular cases, such as windowed data is needed to be recovered, and the rehabilitation cannot be solved by the most of good-performing classical window functions. In Enhanced Variable Rate Coder[8] and Selectable Mode Vocoder[9], trapezia window is adopted, but its spectrum is not desirable. In this paper, the small-world networks’ principle[1] is used to design window function and to solve the problem.

The Small-world network(SWN) of Watts and Strogatz is a kind of intermediate network mode transmitting from regular network to random one. It can be described as below: in a multi-dimension space, \( N \) nodes are distributed in a circle; when each node is connected with \( K \) neighboring nodes, the connection could be changed on the probability of \( p \) (the node will not be connected with itself) to connect to the far-node. When \( p = 0 \), the network is a regular one, shown in Fig.1(a); When \( p = 1 \), the network is a random one, shown in Fig.1(b); When \( 0 < p < 1 \), it is a small-world network, shown in Fig.1(c). The connection created by random selection is called “Random Long Connection”, which has significant effects on the shortest distance among overall network nodes. The effect is non-linear and can shorten the distance of network searching obviously. In a SWN, nodes have connection with not only neighboring nodes but also distant nodes, and they build up amounts of long connections with these distant nodes, thus the searching distance is shortened. According to Kleinberg’s prove[4], to ensure connection randomicity of a network, \( N >> k >> \ln N >> 1 \) is needed.

2. Searching Algorithm based on the SWN

To applying SWN theory to function optimization, there’re also two processes: locally search and random long distance connection. The searching process could be explained by the sample of plane graphic of a multi-dimension space, shown in Fig. 2. Here, A is the initialization solution of function, B is a solution of A in the neighborhood, C is a solution of A in the distant field, P is the optimum solution.

The conception of the Small-world network was firstly prompted by Stanley Milgram, the American social psychologist, in his social experiment in 1960s[2]. In 1998, Watts and Strogatz built up small-world network[1] based on Milgram’s experiment. It was caused attentions from various researchers. Until now, small-world networks have a range of successful applications in many fields.
searches should be moved to the neighborhood of C. Repeated the above processes until finding the optimum solution P. 

Suppose that a candidate solution $w_p(n) = [w_p(0), w_p(1), ..., w_p(N-1)]$, then we can define the distance between $w_p(n)$ and $w_p(n)$ as below,

$$d(w_p(n), w_p(n)) = \sum_{j=0}^{N-1} |w_p(n)-w_p(n)|$$

(1)

Suppose that $\lambda_r(w_p(n))$ and $\psi_L(w_p(n))$ are the neighborhood and far-neighborhood of the candidate solution $w_p(n)$. Where, $L>l$.

$$\lambda_r(w_p(n)) = \{w_j(n) | d(w_j(n), w_p(n)) < l\}$$

(1)

$$\psi_L(w_p(n)) = \{w_j(n) | d(w_j(n), w_p(n)) > L\}$$

(3)

The keys of using small-world principle are to build up local short-distance connection searching algorithm and global random long-distance connection searching algorithm.

(1) Local searching algorithm(n_search)

Local searching algorithm principally is the way to seek for the solution $w_p(n)$ to make criterion function $J(w_p(n))$ has greater value in the neighborhood $\lambda_r(w_p(n))$ of a candidate solution $w_p(n)$. The process is described as below:

For $w_p(n) \in \lambda_r(w_p(n))$

If $J(w_p(n)) > J(w_p(n))$, Then $w_p(n) = w_p(n)$

(2) Global searching algorithm(f_search)

$$w(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{j\omega}) d\omega = 1$$

(7)

Global algorithm principally is the way to seek for the solution $w_p(n)$, $w_p(n) \in \psi_L(w_p(n))$, to make criterion function $J(w_p(n))$ has greater value in the far-neighborhood $\psi_L(w_p(n))$ of a current candidate solution $w_p(n)$. The process is described as below:

$$r = \text{rand}(0~1)$$

If ($r < p$) 

{  
  Generate $w_p(n), w_p(n) \in \psi_L(w_p(n))$
  
  Generate $\lambda_r(w_p(n))$
  
  For $w_p(n) \in \lambda_r(w_p(n))$
  
  If $J(w_p(n)) > J(w_p(n))$, Then $w_p(n) = w_p(n)$
}

Here, $p$ is the connection changing probability. rand(0~1) denotes a random number between 0 and 1.

3 Window Function Design Algorithm

Setting $x(n)$ as long sequence, $w(n)$ is the window function with the length of $N$. If the output is $x_s(n)$ after shortened by the window function, then

$$x_s(n) = x(n)w(n)$$

(4)

The Discrete Time Fourier Transformation (DTFT) of the $x_s(n)$ is

$$X_s(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) W(e^{j(j-\omega)}) d\omega$$

(5)

Where, $X(e^{j\omega})$ and $W(e^{j\omega})$ are the DTFT of $x(n)$ and $w(n)$ separately.

From Eq. (4) and (5), we see that the window function have the effect on the original signal both time and frequency character. Since the spectrum resolution of sequence is depended by the main valove width of window function, side valove is probable to shield part of the signal spectrum with less component values[5]. Thus generally main valove is designed to be as narrow as possible, and the value of side valove to be less, i.e. the energy mainly concentrates within the main valove. In order to satisfy this point, we can get such a criterion function $J(w(n))$

$$J(w(n)) = \frac{\int_{\pi}^{\pi} |W(e^{j\omega})|^2 d\omega}{\int_{\pi}^{\pi} |W(e^{j\omega})|^2 d\omega}$$

(6)

In order to ensure the estimation of power spectrum is asymptotic and no drift, window function should have:

Window function usually is a non-negative real even function, So $w(n)$ should have:
Where, \( M \) is the overlap length, and \( M < N/2 + 1 \).

In this paper, based on the fact situation, we define that the criterion functions is formula (6) and the restriction functions are formulas (7)–(11).

Normally there are two coding methods of SWN algorithm. One is binary method, and the other is denary one. It is similar to genetic algorithm, so the genetic coding methods[6] could be used for reference in the small-world one. This paper adopts the binary coding.

When the accuracy of function solution is determined, i.e. coding length of each parameter is fixed. The accuracy of this solution is \( 2^{-15} \), and each factor could be expressed by 15 bits (denary coding) or 15 \( \times \) 8 bits (binary coding) for factors of window functions are normally no non-negative. Since the problem in this study is special, only part of parameters needs to be coded whiles others can be figured out according the symmetry of formulas (9)–(11). For example, when the window length is 128 and the overlap length is 24, only former 12 parameters need to be coding.

When the solution accuracy and the number of coding factors are fixed, total sample numbers of candidate solutions are known. Assuming the number of coding factors is \( m \), then total sample number \( N = 32768^m \). According to the Kleinberg’s prove[4], to ensure random network is connected, the network building process should ensure: \( N >> k >> \ln N >> 1 \). So in this paper, specifying element number in neighborhood is \( k = 400. \) It can be proved that all the examples in the paper satisfy requirements of Kleinberg’s relationships.

The general processed of window function design by small-world network theory are expressed as below:

Step1: Initialize the SWN model

\[
\begin{align*}
N &= N_0, M = M_0, L = L_0, l = l_0, J_{\text{max}} = J_0 \\
w_0(n) &\leftarrow \text{random number} \\
\lambda_1(w_0(n)) &\leftarrow \text{Generate} \\
w_0(n) &= \text{n_search}(\lambda_1(w_0(n)))
\end{align*}
\]

Step2: Generate \( \lambda_i(w_0(n)) \)

\[
w_i(n) = \text{n_search}(\lambda_i(w_i(n)))
\]

Step3:

\[
\text{If } J(w_p(n)) > J_{\text{max}} \text{ goto Step 4 else} \\
\text{r = rand}(0~1) \\
\text{If (r < p) } w_p(n) = \text{f_search}(\lambda_i(w_p(n))) \\
\text{goto Step 2}
\]

Step4: end

4 Simulation and Conclusion

Assuming the length of window \( N=128 \), in variable overlap length situation, the figures of the trapezia window used in SMV and SWN are shown in Fig.3(a)–(c), the corresponding figures of the Amplitude-Frequency response are shown in Fig.3(d)–(f). From Fig.3(a)–(c), we can see that the result of SWN is a real even centrosymmetric function, but it’s not the one non-increased, because the program do not have this restriction.

From Fig. 3(d) we can see that the maximum main valoe of SWN result is better than the trapezia window’s, but the advantage is not very distinct. The main reason is that the overlap length is too short to embody some good characteristics. We can see clearly that the performance of SWN result is increased with the overlap length from Fig. 3(e) and 3(f). The compare table between trapezia window and SWN-window is shown in table1. From table1, we can see the performance of the SWN-window become better with the length of the window increased. The whole SWN-windows parameters except 3dB bandwidth of the triangle window function are all better than of the trapezia windows. We also can see that the function curve will become more and smoother with the iteration number increased as well as the performance.

In simulation experiment, all the SWN functions were got within 200 iterations except the one with overlap length \( M=64 \). And the program-window function with \( M=64 \) was also got within 500 iterations. If the initialization of the solution set is a little better, iteration number will become fewer, and sometimes, we can get such SWN-window functions in 100 iterations.
Figure 3. Function curves of trapezia windows (dot line) and program-ones (solid line)

Table 1. Performance of trapezia and SWN

<table>
<thead>
<tr>
<th>$M$</th>
<th>Method</th>
<th>3dB BW</th>
<th>Side above peak (dB)</th>
<th>Side above energy percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Triangle</td>
<td>67Hz</td>
<td>-14.71</td>
<td>4.24%</td>
</tr>
<tr>
<td></td>
<td>SWN</td>
<td>65Hz</td>
<td>-16.69</td>
<td>2.86%</td>
</tr>
<tr>
<td>36</td>
<td>Triangle</td>
<td>72.5Hz</td>
<td>-17.71</td>
<td>1.84%</td>
</tr>
<tr>
<td></td>
<td>SWN</td>
<td>69.5Hz</td>
<td>-21.83</td>
<td>1.01%</td>
</tr>
<tr>
<td>64</td>
<td>Triangle</td>
<td>80.5Hz</td>
<td>-26.52</td>
<td>0.29%</td>
</tr>
<tr>
<td></td>
<td>SWN</td>
<td>81.5Hz</td>
<td>-36.06</td>
<td>0.18%</td>
</tr>
</tbody>
</table>

In this paper, we complete the optimum design of window function with taking use of the principle of SWN. And the experiment indicates that this method can get perfect window function with only costing little time. So it is an effective method. Moreover, this method has a good transplant characteristic. We can take use of it to design difference function for other fields and other situation if only we design the difference criterion functions.

References