International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.10 (2015), pp.213-220
http://dx.doi.org/10.14257/ijsip.2015.8.10.23

Design of Morse-code Decoder with Filtering and Fault Correction Function Based on MATLAB

Shao Xuan and Yong De Zhang

School of Mechanical and Power Engineering, Harbin University of Science and Technology,Harbin, China 150080
shaoxuan919@sina.com

Abstract

*By using MATLAB, a kind of software decoder is designed which has filter and fault correction function and is used in aviation radio communication. Through Morse code text, using dits and dashes t**o** uniquely represent characters such as alphabets、numbers and punctuations (41)**,** those characters are coded through algorithm. A new decoding algorithm is designed which is composed of filtration, binary, difference and zero removal operation. This decoding method is robust and interference immunity, it also has such functions as filtering and fault correction. It can filter the random noise and Gaussian noise so as to reduce the error rate of aviation radio communication and thereby increase the quality of aviation radio communication. This method is checked though an example to decode the word "Hello".*

Keywords: *Morse code; Software Decoder; Signal Processing; Aviation radio communication; MATLAB*

1. Introduction

Owing to its conventional, economical and effective features, the Morse-code radio communication takes a certain part in nowadays aviation and navigation radio communication [1-3]. The main equipment in Morse radio communication is transceiver. Since 1838, through more than one century development, from analogue to digital, the transceiver has greatly advanced in many aspects such as the style of keyboard, the way of modulation, the channel coding, the transmission rate, the bulk, the weight, the hardware quality etc[4-6]. However, the existed Morse-code radio communication transceivers are different in frequency, modulation method as well as wave form. That makes it difficult to communicate between the transceivers, also that causes high rate of wrong code, low ability of anti-electromagnetic interference. Especially in aviation and navigation, there are strong random electromagnetic interferences, so, the Morse-code radio communication can not meet the need of modern high quality communication[7-13]. To solve those problems above, the MATLAB is used to design one kind of software decoder that is suitable for Morse-code radio

2. Basic Design Principle

Morse code text uses dits, dashes and their combination to uniquely represent characters such as alphabets, numbers and punctuations [14-17]. When dits or dashes appear, the voltage level is high. From the time interval of high level, the dit and dash can be distinguished. From the time interval of low level, the code interval and character interval can be distinguished. When the character interval appears, the dits, dashes and their combination are recorded, through Morse-code, the characters

ISSN: 2005-4254 IJSIP

are got. The high or low level interval can be got through filtration, binary, difference and zero removal operation of the input wave forms.

For conveniently discussing, the dit, dash, minimum time interval and space (that are called basic elements) are marked by Dit, Dah, ssp and lsp. The wave forms of Dit, Dah, ssp and lsp are illustrated as fig 1. Through filtration, binary, difference and zero removal operation of those elements, the high and low level wave forms of them be obtained as showed in fig 2. it can be found from fig2 that Dit, Dah, ssp and lsp can be easily distinguished by the length and height of the voltage level.

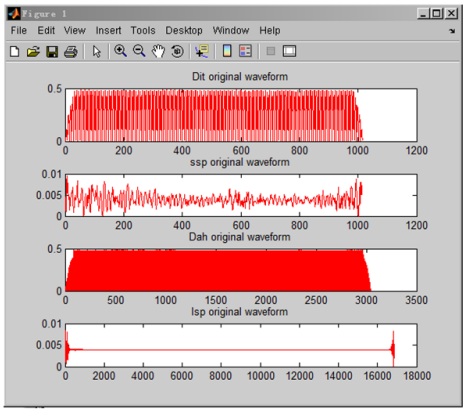

Figure 1. Wave Form of Dit, Dah, ssp and lsp

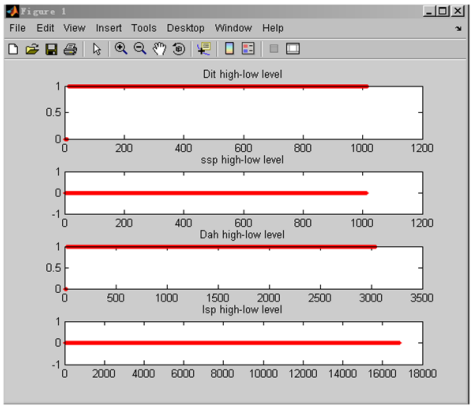

Figure 2. Wave Form of High and Low Voltage Level

For an arbitrary input wave form, through filtration, binary, difference and Zero removal operations, the time interval length of high or low voltage level can be got. To get the start and end time of each segment, according to the length of high or low voltage level of each segment, the original wave form can be divided into sequences of elements. To translate these sequences of elements into punctuations and characters, the decoding operation is finished. Because the filtration, binary, difference and zero removal operations are adopted, the decoder has the function of fault correction and Gauss noise interference immunity. In the following, as illustrated in fig 3, the Input String is “Hello”, the decoding process is described in details.

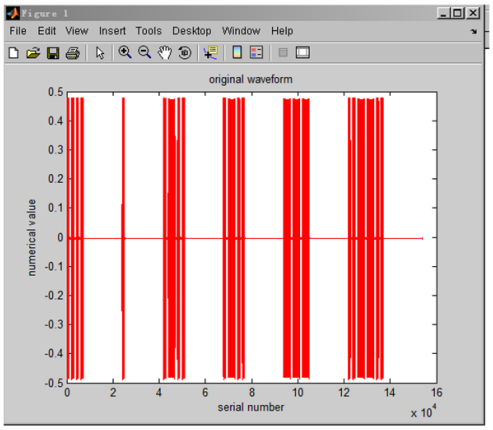

Figure 3. Wave Form of Input String “Hello”

3. Algorithm

As described above, Morse-code is composed of basic elements such as dit, dash, minimum time interval and space etc. Thus, only knowing the wave form of the basic element and the code of each string, the Morse-code of the string can be established.

3.1 Wave Form of Basic Element

Wav.mat represents the wave form of basic element, the Matlab program of the basic element code is as follows:

```
load wav;
Dit = wav(1106:2121);
ssp = wav(2121:3133);
……
lsp = wav(6176:23022)
```

3.2. Definition of Codes Basic Elements

The codes of each character are as follows: （simplified form ∑A-Z）

```
% defines letter, symbol, mark of interval, Morse-code of number.
A = [Dit;ssp;Dah];% letter
B = [Dah;ssp;Dit;ssp;Dit;ssp;Dit];
……
Y = [Dah;ssp;Dit;ssp;Dah;ssp;Dah];
Z = [Dah;ssp;Dah;ssp;Dit;ssp;Dit];
period = [Dit;ssp;Dah;ssp;Dit;ssp;Dah;ssp;Dit;ssp;Dah];% symbol
comma = [Dah;ssp;Dah;ssp;Dit;ssp;Dit;ssp;Dah;ssp;Dah];
question = [Dit;ssp;Dit;ssp;Dah;ssp;Dah;ssp;Dit;ssp;Dit];
slash_ = [Dah;ssp;Dit;ssp;Dit;ssp;Dah;ssp;Dit];
space = [Dit;ssp;Dit;ssp;Dit;ssp;Dit;ssp;Dit;ssp;Dit;ssp;Dit];
n1 = [Dit;ssp;Dah;ssp;Dah;ssp;Dah;ssp;Dah];% number
n2 = [Dit;ssp;Dit;ssp;Dah;ssp;Dah;ssp;Dah];
……
n9 = [Dah;ssp;Dah;ssp;Dah;ssp;Dah;ssp;Dit];
n0 = [Dah;ssp;Dah;ssp;Dah;ssp;Dah;ssp;Dah];
```

3.3 Morse-code of String

Supposing the input string is “text”, to get the Morse-code of it, the Matlab program is as follows:

```
text = upper(text);% function upper makes the lowercases become the capital letters in “text”
vars ={'period','comma','question','slash_','space'};
% vars ={'period','comma','question','slash_'};
……
for i=1:length(text) %circulation time
   if isvarname(text(i)) %transform character to Morse-code, function isvarname judges the character, if it
is a character , return to 1, if not, return to 0.
      temp = [];
      temp = eval(text(i));
      morsecode = [morsecode;temp];
   elseif ismember(text(i),'.,?/ ') %transform punctuation to Morse-code
      x = findstr(text(i),'.,?/ ');
      morsecode = [morsecode;eval(vars{x})];
       morsecode = [morsecode;eval(['n' text(i)])];
      ……
   elseif text(i)==' '% transform space to Morse-code
      morsecode = [morsecode;ssp;ssp;ssp];
   end
   morsecode = [morsecode;lsp];
end
```

4. Decoding Algorithm

Supposing the wave form array of the input string is X, the threshold is 0.05.

4.1 Filter

Matlab codes are as follows:

```
% half-wave rectify x
……
% slow-wave filter
y = filter(ones(1,20)/20,1, x2);
```

The wave form Y after filtering is shown in Figure 4.

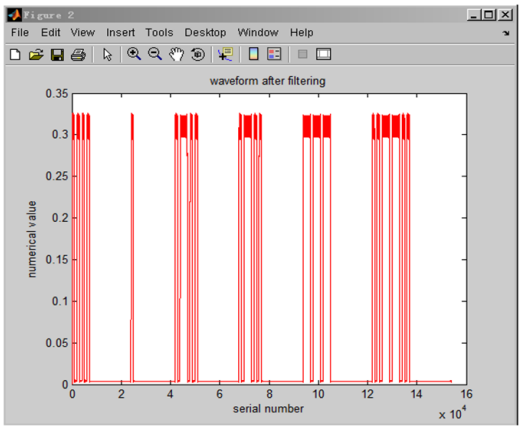

Figure 4. Wave Form after Filtering

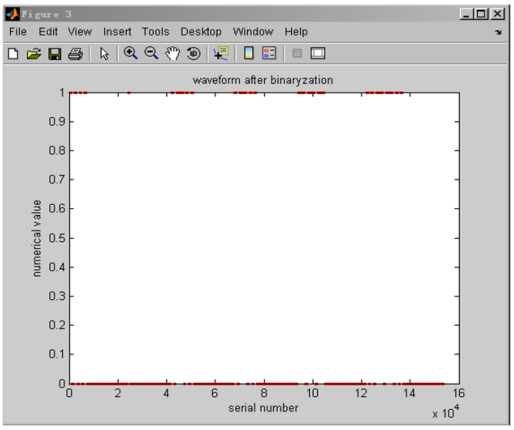

Figure 5. B*inary* Wave Form

4.2. Binaryzation

Matlab codes are as follows:

```
% threshold (digitize) y
z = y > threshold; %comparator：if the signal y（n) is greater than the threshold, z(n) is 1，others z(n) is 0
```

The wave form z after binaryzation is shown in fig 5.

4.3 Difference

Matlab codes are as follows:

```
z = [zeros(10,1); z];  %put 10 zero before array Z
% 1: change from 1 to 0
% 0: no change
% -1: change from 0 to 1
b = diff(z); %difference function
```

The wave form Y after difference is shown in fig 6.

4.4. Zero Removal and Time Interval of High and Low Voltage Level

Matlab codes are as follows:

```
c = b(b~=0);%zero removing
c2 = find(b~=0);% record the series number of the element except zero
tokens = -c .* diff([0; c2]);%calculating the wave length (time interval)
```

The time interval (tokens) of high and low voltage level is shown in fig 7.

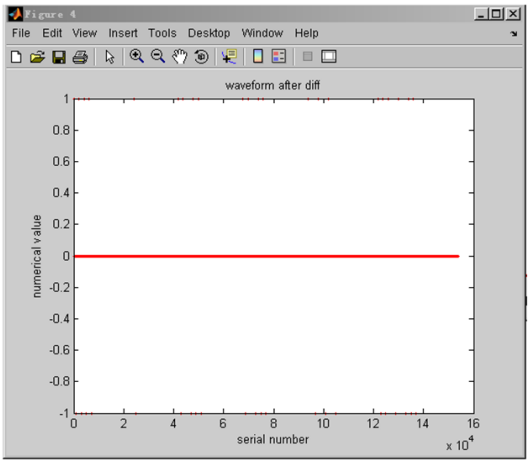

Figure 6. Wave Form after Difference

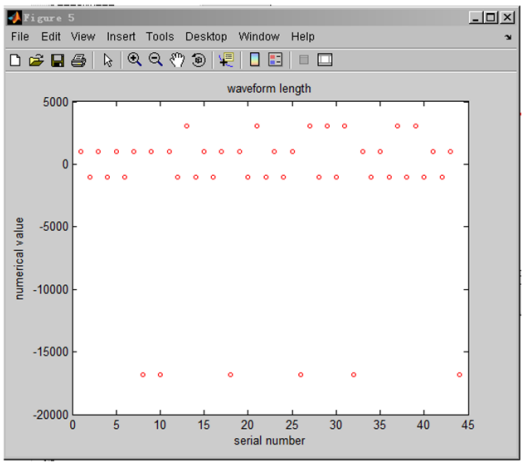

Figure 7. Time Interval Length of High and Low Voltage Level Tokens

4.5. Wave Forms Compare, Obtaining dit, dash, Minimum Time Interval and Space

Matlab codes are as follows:

```
% 1: short, 2: long, +: tone, -: space
tokens2 = tokens;% comparing the wave forms, get dit, dash, minimum time interval and space
% cutoff tones, cutoff spaces;
cut_t = mean(tokens2(tokens2>0));
cut_s = mean(tokens2(tokens2<0));
……————————
% now tokens 2 is a string of -1s, -2s, 1s, 2s, can trim first known space;
% put final endstop at end
tokens2 = [tokens2(2:end); -2];
```

The element sequences (tokens2) are shown in figure 8.

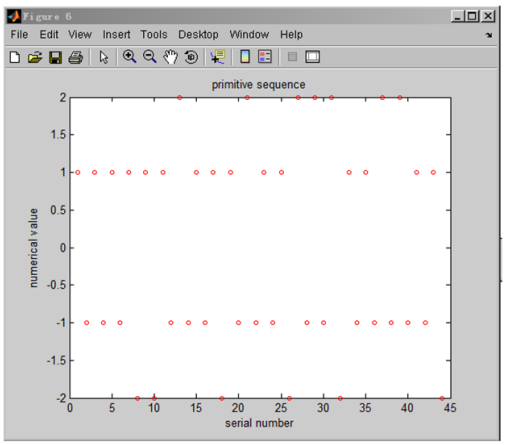

Figure 8. Element Sequences

4.6. Get Character from Morse-code

Matlab codes are as follows:

```
% can drop little spaces, b/c they don't matter when parsing;
tokens2(tokens2 == -1) = [];
tokens3 = tokens2;
tokens4 = {};
ctr = 1;
start_idx = 1;
%parse
```

```
    ……————————
    start_idx = a+1;
end
% now tokens4 is de-codeable tokens... proceed to setup lookups
% letters
∑A-Z
% comparing tokens to tables
out1 = [];
for j = 1:length(tokens4)%output decipher
    out1(j) = '_';
    %to check if it is one of the 41 basic elements
    ……————————
            %display(decode{k})
        end
    end
    % if didn't find a match
    if isempty(out1)
        out1(j) = '_';
    end
    if isempty(out1(j))
        out1(j) = '_';
    end
  end %if(length(tokens4{j})==0 && length(tokens4{j})>7)
end
```

4.7 Result

The operation result is HELLO.

5. Program and Results with Gauss Noise and Wrong Code

5.1 Matlab Codes with Gauss Noise and Wrong Code

```
%% constructing Gauss noise and wrong code, input string, corresponding array is X.
%normalNoise  is Gauss noise, wrongCode is wrong  code sin wave
%noise_orimorsecode  is wave form with Gauss noise and wrong code
% constructing Gauss noise
noiseMean = 0;
noiseStd = std(ori_morsecode)/10; %add 0, normal difference is 1/100 of signal
normalNoise = normrnd(noiseMean,noiseStd,length(ori_morsecode),1);
noise_orimorsecode = X + normalNoise;
%constructing wrong code
wrongCode = 0.05*sin([0:0.001:2*pi]);
noise_orimorsecode  = [noise_orimorsecode(1:80000); wrongCode'; noise_orimorsecode(80001:end)];
```

5.2 Test Results Added with Gauss Noise and Wrong Code

To repeat operation “noise_orimorsecode”, the corresponding wave forms are illustrated in Figures 9-14.

5.3 Result

The operation result is HELLO.

To compare fig 8 with figure 14, you can see that they are absolutely identical. That is to say that by operation above, the added Gauss noise is filtered correctly [18].

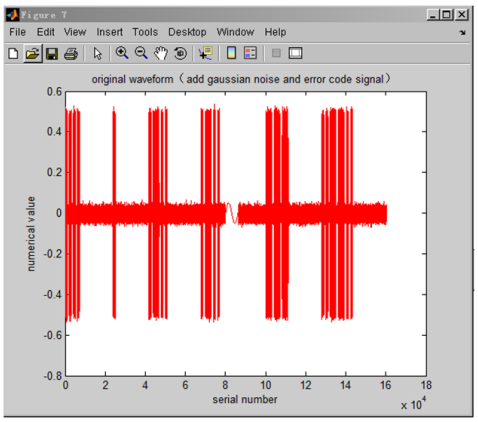

Figure 9. Noise_orimorsecode

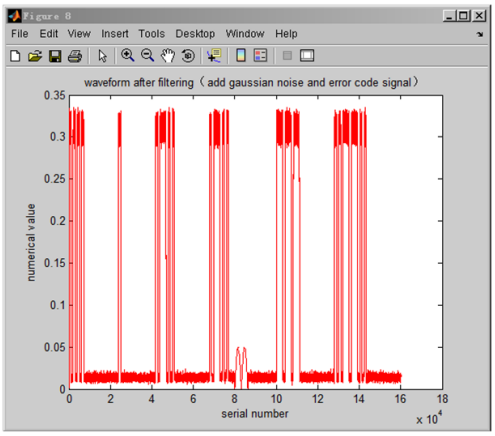

Figure 10. Wave form after filtering

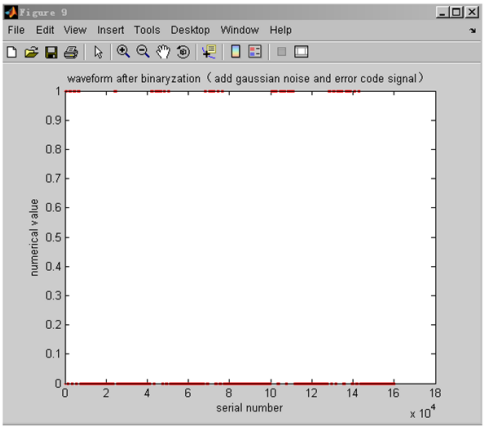

Figure 11. B*inary* Wave Form

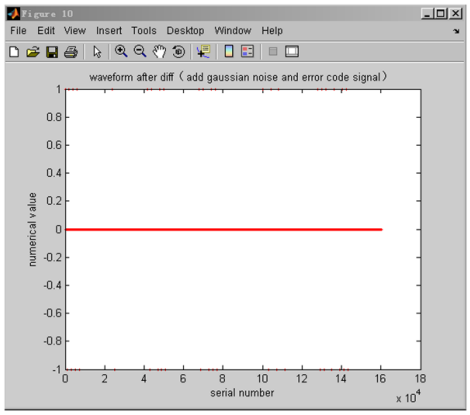

Figure 12. Wave Form after Difference

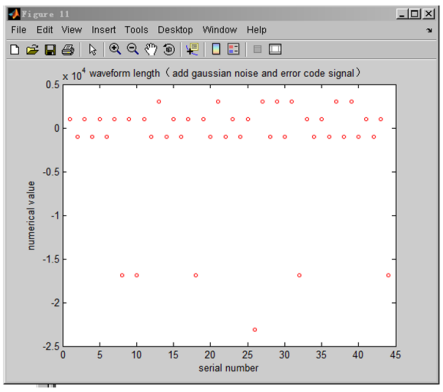

Figure 13. Time Interval Length of High and Low Voltage Level Tokens

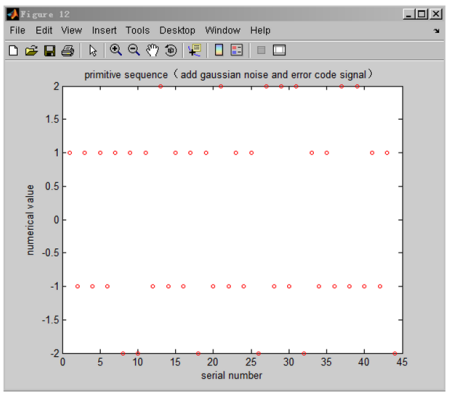

Figure 14. Element Sequences Tokens2

6. Conclusion

A new and practical Morse-code software decoder is designed for aviation radio communication. Owing to using filtration, binary, difference and zero removal operations, the decoder is robust and immunity, it also has such abilities as filter and wrong code correction. Running program “morse_hello2.m” can get the string HELLO and the fig 14. Comparing fig 3 -8 with fig 9-14, the fact that fig 8 and fig

14 are absolutely same shows that this new decoder has excellent electromagnetic interference immunity and wrong code correction function. By using this decoder, the rate of wrong code can be reduced and the quality of aviation radio communication can be improved.

Acknowledgments

This work was supported in part by NSF Heilongjiang Province of China under Grant Nos.E201130

References

[1] D. A, Shannon, W. S. Staewen, J. T. Miller and B. S. Cohen, "Morse-code-controlled computer aid for the nonvocal quadriplegic", Medical Instrumentation, vol. l, no. 15, (**1981**), pp. 341-343.
[2] Bau, "The exist value of Morse code", http://bau.blog.hexun.com/23126533_d.html.2008
[3] F. Changyuan and W. Bin, "TPU Morse decoder for civil aviation [J]", Journal of Electronic Measurement and Instrument, vol. 09, (**2004**), pp. 45-48.
[4] W. Ma and J. Zhang, "Morse-code decoding system [J]", Armament Automation, (**2007**), vol. 26, no. 06, pp. 51－52, 55.
[5] Y. Wenzhan and W. Tang, "Computer simulation on Morse telegram automatic sending [J]", computer technology and automation, (**2001),** vol. 20, no. 02, pp. 75－78, 86.
[6] Z. Wang, "MAD-I Morse-code automatic decoder [J]", Communication and computing technology, vol. 03, (**1984**), pp. 126－128.
[7] C. Li and D. Zhao, "Automatic recognition of Morse telegram by sounds recognition technology [J]", Information technology, vol. 30, no. 02, (**2006**), pp. 56-57, 125.
[8] R. Lin, "Morse-code decoding By using microprocessor [J]", Communication and computing technology, vol. 02, (**1983**), pp. 128－129.
[9] R. Zhang, L. He and X. Li, "Morse signal automatic detection and recognition under strong noise background [J]", Journal of Harbin engineering university, vol. 27, no. 01, (**2006**), pp. 112-116.
[10] H. Yu and Y. Zhang, "A new computation method of automatic receiver for manual Morse telegram and the manual technique recognition computation method [J]", Electronic countermeasures, vol. 11, no. 04, (**1995**), pp. 317-320.
[11] J. Lin and G. Li, "Automatic detection of high frequency CW telegram signal under strong noise background [J]", Journal of Chongqing University of Posts and Telecommunications (**2008**), vol. 20, no. 05, pp. 505-509.
[12] J. Gao, "Morse-code automatic recognition in aerospace [J]", Journal of Electronic Measurement and Instrument, vol. 08, (**2006**), pp. 5-9.
[13] R J. Lacky and D W. Upmal, "Speakeasy: The Military Software Radio", IEEE Communications Magazine, (**1995**) May, pp. 56-61.
[14] Reading Morse-code[EB/OL].http://k.pconline.com.cn/question.883698.html
[15] Morse-code table [EB/OL]. http://www.wen8.net/html/876.htm
[16] Ancient and magical Morse telegram [E B/OL]. http://bbs.rednet.cn/thread-15464110-1-1.html.2008
[17] Y.-N. Xie and J.-J. Huang, "A Novel Robust Speech Feature for Speaker Recognition", Journal of Harbin University of Science and Technology, vol. 4, (**2013**), pp. 47-51.
[18] C.-x. Deng, L. Peng, H.-y. Meng, "Gauss-free Galerkin Method for Solving Partial Differential Equations", Journal of Harbin University of Science and Technology, vol. 06, (**2013),** pp. 114-116.

