

## An Adaptive Algorithm for Morse Code Recognition

by **Cheng-Hong Yang** Dept of Electronic Engineering National Kaohsiung Institute of Technology  
Kaohsiung, Taiwan 807

Ching-Hsing Luo

### ABSTRACT

*The Morse code is an international communication language that is simple, speedy, and low cost. The automatic recognition of Morse code is difficult because maintaining a stable typing rate is not easy. In this paper, a suitable adaptive automatic recognition method, the variable degree variable step size Least-Mean-Square algorithm, is provided and its procedure is divided into three modules: character separation, character recognition, and adaptive processing. Experimental results showed that the proposed method obtained high recognition rate.*

**Keywords:** Morse code, Adaptive signal processing, Least-Mean-Square .

### 1. Introduction

Samuel F. Morse is best known for the coding scheme that carries his name. Morse code, the most general means of transmitting signals, is transmitted as a tone-silent time series. A dot represented as a short beep or a dash represented as a longer beep are defined as tone intervals (switch down). A dot-space, which is a short pause between dots and dashes, a character-space, which is a longer pause between characters, or a word-space, which is a much longer pause between words, are defined as silent intervals (switch up). Actually, the word-space is represented by a tone-silent code; therefore, a much longer pause need not be used. Subsequently, Morse code is very simple and can be transmitted by just a switch. Therefore, in some circumstances, it could become a useful communication tool for the disabled who can handle a single switch. In general, one switch Morse code can be utilized by persons whose hand coordination and dexterity are impaired, but mental and cognition levels are at least fair to good. Morse code has alternatively been proposed as an efficient auxiliary method [1-6]. Many papers have been presented to discuss how to perform the typing of text on a reduced set of switches with an efficient approaching one key press per selected character [7-10]. However, in the recognition of Morse code, a stable typing rate is strictly required. This restriction is a major hindrance for disabled persons to have Morse code as a useful tool.

The Least-Mean-Square (LMS) algorithm [11] is one of the most popular algorithms in adaptive signal processing. A lot of variants of LMSs have been extensively analyzed in the literature [13-15]. In this paper, the variable degree variable step size LMS algorithm [12] was applied to solve the problem of Morse code recognition. Experimental result shows that the proposed method provided high recognition rate. The rest of this paper is organized as follows. In the next section, the new Morse code recognition method is presented. The experimental result is shown in Section 3. Finally, concluding remarks are made in Section 4.

### 2. Method

A tone ratio (dot to dash) has to be 1:3 according to the definition of Morse code. That means if the

duration of a dash is taken to be one unit, then that of a dash is three units. In addition, the silent ratio (dot-space : character-space : word-space) has to be 1:3:7. In another words, the space between the components of one character is one unit, between characters is three units and between words is seven units.

In this paper, the Morse code recognition method is divided into three modules: character separation, character recognition, and adaptive processing. The space between characters is first identified in the character separation section so that the codes of an unknown character can be regarded as a set of messages to be translated into its corresponding character in the character recognition section. The character recognition is based on the Euclidean distance between the codes of the unknown character and the codes set in the Morse code table. If the character in the table has the minimum Euclidean distance, it is chosen as the unknown character. In order to match with the typing speed, the average of the space length in a character is sent to the adaptive processing section, and the character separation section then identifies the space between characters based on the processing output. A Morse code character,  $x_i$ , is represent-ed as follows:  $e_1(x_i), b_1(x_i), \dots, e_j(x_i), b_j(x_i), \dots, e_n(x_i), b_n(x_i), 1 \leq i \leq n$ ,

where

$e_j(x_i)$ : when a key is pressed down, it is presented as 'dot' or 'dash', which is the duration of ith Morse code element of the input character  $x_i$ .

$b_j(x_i)$ : when a key is held up, it is presented as one of three spaces: the space between components of one character, the space between characters, or the space between words, which is the duration of ith space of the input character  $x_i$ .

$n$ : the total number of keyed-in Morse code elements in  $x_i$ .

## Character Separation

Character separation is used for identification of the space between characters and isolation of the Morse code elements of a character. For example, if a data stream of characters is encoded in Morse code elements, these elements are then identified as either space between characters or isolated elements of a character.

The procedure for character separation is shown as follows:

1. Assign the index  $j=1$ .
2. If  $b_j(x_i) < 2S_1$ , then go to step 3, otherwise go to step 4.
3. Let  $j=j+1$  and save  $e_j(x_i)$ , then go to step 2.
4. If  $b_j(x_i) > 5S_1$ , then it is the end of a word, otherwise go to step 5.
5. The key-in Morse elements in  $x_i$  character are encoded as  $e_j(x_i)$  for  $j=1 \sim n$ .
6. To separate the next character go to step 1, if any.

Due to the initial value  $S_1$  is absent, the first character  $x_i$  can not be isolated immediately. The initial standard space length  $S_1$  is obtained by taking the first nine values of silent elements as the reference values; then, all of the values taken are sorted by descending order. After the sorting, the relationship

between each value is compared. If a value is larger than any other value twice, the larger value is represented as long (L) and the smaller one is represented as short (S). Once the relationship is determined, the average value of these nine values is calculated and assigned as the initial standard space length  $S_1$ . For example, assume the Morse code digital stream as follows: 423 255 156 180 297 290 79 2701 469 934 181 1557 89 351 805 360 845 309 808 1179 562, in which odd position data is defined as tone while even position data, underlined, is defined as silent. The first nine silent value were sorted as follows: 2701, 1557, 934, 360, 351, 309, 290, 255, and 180. After sorting, the first three values (2701, 1557, and 934) were presented as L and the rest of them were presented as S. The sum of long silent values is divided by 3 and the sum of short silent values are calculated, then  $S_1$  is the average value of the sum of long and short values.

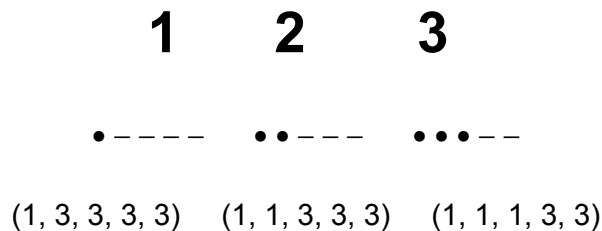
$$S_1 = (\text{sum of long}/3 + \text{sum of short}) / \text{number of elements}$$

$$S_1 = [(2701+1557+934)/3 + 360+351+309+ 290+255+180] / 9 = 386.30$$

Once the initial standard space length is obtained, the data stream is separated into character set and space. After the Morse code elements of a character is isolated from a data stream, the elements will be recognized in the character recognition section.

### Character Recognition

The Morse code table is consists of a set of Morse code elements which contains 10 digital (0-9) and 26 English characters (A-Z). These code elements are able to simplify as a set of numbers. As shown in Figure 2, '1' is coded as '.----', which is simply presented as (1, 3, 3, 3, 3).



**Figure 2. A simple representation of Morse code**

The Euclidean distance is calculated between the codes of the unknown character and the codes set in the Morse code table. The procedure for the minimum Euclidean distance method is shown as follows:

1. Read each tone value of the unknown character code set,  $e_j(x_i)$ .
2. Normalize  $e_j(x_i)$  by using the minimum  $e_j(x_i)$ ,  $e_j(x_i)/\min(e_j(x_i))$ , for  $j=1\sim n$
3. Calculate the roots of the sum of the square distances,  $d_i$ , between the normalized  $e_j(x_i)$  and the character,  $e_j(t_i)$ , of Morse code table, where

$$d_i = \left[ \sum_{j=1}^n (e_j(x_i) - e_j(t_i))^2 \right]^{1/2}$$

4. The character in the Morse code table with normalized  $e_j(x_i)$  has the shortest Euclidean distance  $\text{Min}(d_i)$ , it is determined as the unknown character.

For example, assume the Morse code elements in an unknown character as follows: 423, 156, 297, and 79, in which the minimum tone code element is 79. After normalization with the minimum tone code element, the quotients among the four Morse elements are 5.35, 1.97, 3.76, and 1.0. Following the shortest Euclidean distance is obtained from the known character 'C' (as 3, 1, 3, 1) in the Morse code Table. Thus, 'C' is chosen as the unknown character.

### Adaptive Processing

The adaptive filtering or system identification problem being considered is to try to adjust a set of filter weights so that the system output tracks a desired signal. Let the input vector to the system be denoted by  $X_k$  and the desired scalar output be  $d_k$ . These processes are assumed to be related by the equation

$$d_k = X_k^T W_k^* + e_k$$

where  $e_k$  is a zero mean Gaussian independent sequence, independent of the input process  $X_k$ .  $W_k^*$  is randomly varying according to the equation

$$W_{k+1}^* = aW_k^* + Z_k$$

where  $a$  is less than but close to 1, and  $Z_k$  is an independent zero mean sequence, independent of  $X_k$  and  $e_k$ , with covariance  $E\{Z_k Z_k^T\} = \sigma_z^2 I \delta_{kj}$ ,  $\delta_{kj}$  being the Kronecker delta function. The input process  $X_k$  is assumed to be a zero mean independent sequence with covariance  $E\{X_k X_k^T\} = R$ , a positive definite matrix.

The LMS computes a set of weights  $W_k$  that seeks to minimize  $E(d_k - X_k^T W_k)^2$ . Each adaptive weight  $W_k$  is of the form

$$W_{k+1} = W_k + \mu_k X_k \varepsilon_k \quad \text{where } \varepsilon_k = d_k - X_k^T W_k$$

$\mu$  is the step-size parameter that controls the speed of convergence as well as the steady-state and/or tracking behavior of the adaptive filter. The selection of  $\mu$  is very critical for the LMS algorithm. A small  $\mu$  (small compared to the reciprocal of the input signal strength) will ensure small misadjustments in steady state, but the algorithm will converge slowly and may not track the nonstationary behavior of the operating environment very well. On the other hand, a large  $\mu$  will in general provide faster convergence and better tracking capabilities at the cost of higher misadjustments.

The algorithm utilized the current data to compute a new weight vector using the weight update recursion of the standard LMS with step size  $\mu$ . The new weight vector, together with the current data, are then utilized to update again the desired weight vector using the standard LMS weight update recursion with step size  $\mu$ . Each adaptive weight  $W_k$  is adjusted according to the equation

$$W_{k+1} = W_k - \alpha_2(n) \hat{\nabla}(n) \quad \text{where } \alpha_2(k) = 2\mu (1 - \mu X^T(k)X(k)) \quad \text{where the subscript on the } \alpha(n) \text{ is used to indicate the degree, and } \hat{\nabla}(n) = -2e(n)X(n)$$

is an estimate of the gradient.

The variable degree variable step size LMS algorithm used serves to cleverly change the standard 'dot' length to predict an unstable typing speed generated by the disabled. The average of space  $b_j(x_i)$  ( $i = 1 \sim n-1$ ) in  $x_i$  is the  $i$ th input data of the algorithm. The algorithm with nine weights is used in this paper.

### 3. Experimental Results and Discussion

Two groups of expert testing data, EXP1 and EXP2, were tested in order to investigate the efficiency of the proposed method. EXP1 testing data, number from Exp101 to Exp115, are collected from 15 abled peoples who are trained for a long period of time by typing 100 identical characters. EXP2 testing data, numbered from Exp201 to Exp215, are collected from 15 experts in the military wireless service by typing 100 identical characters. The experimental results are shown in Table 1. The average number of matches for the EXP1 and EXP2 are 88.73 and 90.53, respectively. As it was expected, the experts showed a little higher number of matches than the nonexperts. The experimental results indicated that the different initial  $S_1$  turned into different recognition rate.

The incorrect recognition might be generated in two main errors: character separation errors and character recognition errors. If the space between 'dot' and 'dash' within a character has unusual longer length, that will be mistaken as the space between characters. Once an incorrect character separation is generated, the character will be split into two characters so that the character recognition will be split into two characters so that the recognition will be affected.

The character recognition error is due to the typist's personality. If the typing speed is unstable, such as longer or shorter than

**Table 1. The recognition result for two types of test problems.**

Problems	Number of matches	Problems	Number of matches
Exp101	93	Exp201	95
Exp102	93	Exp202	98
Exp103	92	Exp203	97
Exp104	94	Exp204	94
Exp105	87	Exp205	80
Exp106	86	Exp206	81
Exp107	91	Exp207	94
Exp108	86	Exp208	94
Exp109	88	Exp209	89
Exp110	84	Exp210	97
Exp111	84	Exp211	85
Exp112	89	Exp212	87
Exp113	92	Exp213	90
Exp114	84	Exp214	90
Exp115	88	Exp215	87

Average	88.73	Average	90.53
---------	-------	---------	-------

the standard length, a character will be mismatched in the recognition. Usually, every one has his own typing speed. The system should provide adequate adjustment for the length of dot or dash. Because one types for a long period of time, one's typing might cause errors when the typist becomes tired. For example, it begins with 300ms to 100ms for the length of dash to dot, but it might change to 900ms to 300ms after a long period of typing. However, according to experience, a person's typing rate is generally constant over a short period, the person's present typing rate is similar to the typing rate of the previous several words. Therefore, in order to increase the recognized rate, the tone code element in the Morse code table has to be adjusted by a format which is designed for the individual. In addition, the adjustment should be based on the previous typing speed. It means that tone length has to be renewed after each character has been recognized.

In this study, the defect of the new developed method is only adjusting space values and sometimes it produced some mistakes during the adaptive process. Thus, to have better performance, more efforts and adjustment should be considered in the process, such as in addition to modify into space values, tone values should be adjusted within the adaptive process. The process to modify tone values might use statistic method or similar method as the adjustment of space. Either of these two methods should provide better results.

#### 4. Conclusions

The Morse code is an international communication language that is simple, speedy, and low cost. However, automatic recognition of the Morse code is difficult, because maintaining a stable typing rate is not easy. Therefore, a suitable adaptive automatic recognition method is needed. In this paper, we presented an adaptive algorithm for Morse code recognition. The method was applied to 30 test problems. Experimental results showed that the proposed method obtained great recognition rate. In the future study, we expect to apply this method to the people with physical impairment. Moreover, Neural network and genetic search will also be used to solve the Morse code recognition problems.

#### Acknowledgements

This work was supported in part by the National Science Council R.O.C under contract NSC-88-2614-E-151-001.

#### References

1. C.-H. Luo and C.-H. Shih, Adaptive Morse-coded single-switch communication system for the disabled, *Int. J. of Biomed. Comput.* 41 (1996) 99-106.
2. C.-H. Shih and C.-H. Luo, A Morse-Coded recognition system with LMS and matching algorithms for persons with disabilities, *Int. J. of Medical Informatics* 44 (1997) 193-202.
3. C.-H. Shih and C.-H. Luo, Adaptive single-switch communication system for the disabled, *Journal of Biomedical Engineering Applications, Basis and Communications* 1994, 551-556.
4. S. P. Levine, J. R. D. Gauger, L. D. Bwera, K. J. Khan, A comparison of mouth stick and Morse code text inputs, *AAC augmentative and alternative Communication* 2, 51 (1986).
5. L. N. Goble. and H.A. Colle, High-speed Morse code training, *Proceedings of the IEEE 1985 National Aerospace and Electronics Conference, NAECON, 1985, 944-951.*
6. R. Trace and D. Center, Two switch auto-repeat Morse code, *Waisman Center, University of Wisconsin-Madison, 1984.*
7. D. Bearden, Mobile unit promotes training, hiring handicapped, *Institute for electronic and Electrical Engineering, July (1981) 150-151.*
8. D. W. Lywood and J. J. Vasa, Computer-terminal operating and communication aid for the

- severely handicapped, Medical and Biological Engineering, 12 (1974) 693-695.
9. D. A. Shannon, W. S. Staewen, J. Miller, and B. S. Cohen, Morse-code controlled computer aid for the non-vocal quadriplegic, Medical Instrumentation, 15 (1981) 341-343.
  10. A. Thomas, Communication devices for the non-vocal disabled, Computer, 14 (1981) 25-30.
  11. B. Widrow and S. D. Stearns, Adaptive Signal Processing, Englewood Cliffs, NJ: Prentice Hall.
  12. M. A. Khasawneh and K. A. Mayyas., A Newly Derived Variable Degree Variable Step Size LMS Algorithm, Int. J. Electronics, 1995, Vol. 79, No. 3, 255-264.
  13. V. J. Mathews and Z. Xie, (1993), A Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size, IEEE Transactions on Signal Processing, Vol. 41, No. 6, June, 2075-2075-2087.
  14. R. H. Kwong and E. W. Johnston, (1992), A variable Step Size LMS Algorithm, IEEE Transactions on Signal Processing, Vol. 40, No. 7, July, 1633-1642.
  15. R. W. Harris, D. M. Chabries, and F. A. Bishop, (1986), A Variable Step (VS) Adaptive Filter Algorithm, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 2, April, 309-316.
- 



Assumption University of Thailand  
Huamark, Bangkok 10240 , Thailand  
For comment, Please contact [WebMaster](#)