Multichannel Filters for Speech Recognition using a Particle Swarm Optimization

Kit Yan Chan
Department of Electrical and Computer Engineering
Curtin University,
Perth, Australia

Cedric K.F. Yiu
Department of Applied Mathematics,
The Hong Kong Polytechnic University,
Hong Kong

Sven Nordholm
Department of Electrical and Computer Engineering
Curtin University,
Perth, Australia

***Abstract*— Speech recognition has been used in various real-world applications such as automotive control, electronic toys, electronic appliances etc. In many applications involved speech control functions, a commercial speech recognizer is used to identify the speech commands voiced out by the users and the recognized command is used to perform appropriate operations. However, users' commands are often corrupted by surrounding ambient noise. It decreases the effectiveness of speech recognition in order to implement the commands accurately. This paper proposes a multichannel filter to enhance noisy speech commands, in order to improve accuracy of commercial speech recognizers which work under noisy environment. An innovative particle swarm optimization (PSO) is proposed to optimize the parameters of the multichannel filter which intends to improve accuracy of the commercial speech recognizer working under noisy environment. The effectiveness of the multichannel filter was evaluated by interacting with a commercial speech recognizer, which was worked in a warehouse.**

***Keywords*—** Speech recognition, multi-channel filter, swarm optimization, speech enhancement

I. Introduction

Many electronic products involved speech control functions [1-3] are usually implemented by a commercial speech recognizer, which is developed based on a database contained a certain amount of speech commands voiced out by different kinds of people [4]. However, while using the developed commercial speech recognizer in real-world environments, original speech signals voiced out by users is usually corrupted by ambient noise, thereby causing inaccurate speech recognitions.

To enhance speech recognition, single channel approaches [5] are commonly used to filter stationary noise such as background street noise, but non-stationary noise like human conversations cannot be filtered. Multi-channel filters [6-7] are commonly used to filter non-stationary noises. They align the signals received from multi-channels in order to focus the signal power from the direction of the command source, and minimize the signal power from all the other directions. Hence, a unity from the direction of the command source can be received. However, they only intend to optimize subjective or objective acoustic criteria which are not totally related to the accuracy of the commercial speech recognizer. Speech recognition may not be enhanced, as the spectrum of the original signal is distorted by the multichannel filters [8].

To overcome the limitation, an optimization problem is formulated to maximize the accuracy of the commercial recognizer by optimizing the parameters of the multi-channel filter. With the optimization problem, an innovative particle swarm optimization (PSO) is proposed to determine the optimal multi-channel filter. It employs the mechanisms of the movement of particle swarm [9], in order to locate the global optimum. In this paper, the mechanism of enhancing the effectiveness of PSO is proposed based on a fuzzy inference system. Activating components are injected into the particles in the late generations of the search, in order to increase diversity of the particles. It aids the PSO to achieve a better solution [10]. The fuzzy inference system monitors the degree of diversity of the particles and the progress rate of the PSO. When the diversity of the particles or the progress rate is low, more activating components are injected. The fuzzy inference system helps to avoid pre-maturely converging to sub-optima, and helps the PSO to jump from the sub-optima to reach the global optimum of the speech recognition optimization problem. The proposed PSO is applied on development of multi-channel filters for a commercial speech recognizer, which worked under non-stationary noise. More convincing results can be obtained than the classical PSO and the genetic algorithm.

II. Optimization of accuracies of commercial speech recognizers

A commercial speech recognizer, $R(..)$, is used to recognize n speech commands, $\{s^1, s^2, \ldots, s^n\}$, which can be a single word or a set of phrases. The original user's command is usually corrupted by the acoustic noise as:

$$x^i = s^i + v, \quad i = 1,2,\ldots,n \qquad (1)$$

where $x^i = \left[x^i(1), x^i(2), \ldots, x^i(m)\right]$ with m samples, is the i-th corrupted command; $s^i = \left[s(1), s(2), ..., s(m)\right]$, is the i-th source command; and $v = \left[v(1), v(2), ..., v(m)\right]$, is the acoustic noise.

The accuracy of speech recognition decreases, if the magnitude of the signal to noise ratio increases. A commonly used multi-channel filter with M microphones is used to filter v from x_j^i, received by the j-th microphone, where x_j^i is given by:

$$x_j^i = s_j^i + v_j, \; j\text{=}1,2,\ldots M; \qquad (2)$$

s_j^i is the i-th source command received by the j-th microphone; $x_j^i = \left[x_j^i(1), x_j^i(2), ..., x_j^i(n)\right]$; and $s_j^i = \left[s_j^i(1), s_j^i(2), ..., s_j^i(n)\right]$.

The enhanced signal, y^i, is given by:

$$y^i(k) = \sum_{j=1}^{M} \sum_{l=0}^{L-1} w_j(l) x_j^i(k-l) \qquad (3)$$

where $y^i = \left[y^i(L), y^i(L+1), ..., y^i(n)\right]$; $w_j(l)$ is the element at the j-th row and l-th column of the multi-channel filter matrix **w**, and is given by:

$$\mathbf{w} = \begin{pmatrix} w_1^T & w_2^T & \ldots & w_M^T \end{pmatrix}^T, \qquad (4)$$

with $w_j^T = \begin{pmatrix} w_j(0) & w_j(1) & \ldots & w_j(L-1) \end{pmatrix}$; j=1,2,...,M; and L is the filter length

The resulting recognition given by $R(..)$ is given by estimate , y^i, which is given the following formula:

$$\hat{i} = R\left(y^i\right) \qquad (6)$$

The optimal multi-channel filter matrix **w** is determined by solving (7), in order to maximize the correct recognition rate estimated by the commercial speech recognizer:

$$\max_{\mathbf{w}} J = \frac{100}{m} \cdot \sum_{i=1}^{m} f_i \qquad (7)$$

where $f_i = \begin{cases} 0 & \text{if } \hat{i} \neq i \\ 1 & \text{if } \hat{i} = i \end{cases}$. The PSO discussed in Section III is proposed to determine **w**.

III. Particle swarm optimization

The PSO first creates a random initial swarm which consists of N_s particles, where each particle is represented by the multi-channel filter matrix formulated in equation (4). The position of the i-th particle at the g-th generation is given by:

$$\begin{aligned} P_i^g &= \left(p_{i,1}^g, p_{i,2}^g, \ldots, p_{i,n_p}^g\right) \qquad (8) \\ &= (w_1(0), w_1(1), \ldots, w_1(L-1), w_2(0), w_2(1), \ldots, w_2(L-1), \\ &\quad \ldots, w_M(0), w_M(1), \ldots, w_M(L-1)) \end{aligned}$$

where P_i^g consists of n_p elements, $p_{i,k}^g$ with k=1,2,…, n_p ; $p_{i,k}^g$ is randomly generated in the range between $p_{\min}$ to $p_{\max}$, where $p_{i,k}^g \in \{p_{\min} \ldots p_{\max}\} = \{0 \ldots 1\}$; $n_p = L \times M$; L is the filter length of the multi-channel filter; and M is the number of microphones in the multi-channel filter.

Then each particle, P_i^g , is evaluated by the correct recognition rate based on the optimization problem (7) that determines the accuracy of the resulting commercial speech recognizer. The fitness of P_i^g at generation g, is denoted as J_i^g.

The position of the j-th element on the i-th particle, $p_{i,k}^g$, is updated based on equation (9) at the g-th generation:

$$p_{i,j}^g = p_{i,j}^{g-1} + v_{i,j}^g. \qquad (9)$$

where $p_{i,j}^{g-1}$ is the previous position of the j-th element on the i-th particle at generation g, and $v_{i,j}^g$ is the velocity of this element. For the classical PSO, $v_{i,j}^g$ is given by:

$$\begin{aligned} v_{i,j}^g &= k \cdot \left\{\omega \cdot v_{i,j}^{g-1}\right\} + \varphi_1 \cdot r_1 \left(pbest_{i,j} - p_{i,j}^{g-1}\right) \\ &+ \varphi_2 \cdot r_2 \left(gbest_j - p_{i,j}^{g-1}\right), \qquad (10) \end{aligned}$$

where $pbest_i = \left\lfloor pbest_{i,1}, pbest_{i,2}, \ldots, pbest_{i,n_p} \right\rfloor$ is the best position of the i-th particle moved so far, and $gbest = \left\lfloor gbest_1, gbest_2, \ldots, gbest_{n_p} \right\rfloor$ is the position of the best particle among all the particles; r_1 and r_2 return a uniform random number in the range of [0,1]; φ_1 and φ_2 are acceleration constants; k is a constriction factor derived from the stability analysis of equation (11) to ensure the system converges but not prematurely [14]; $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$; ω is the inertia weight, which is in the range of [0.1...1.1] which is recommended by [12].

For the proposed PSO, the velocity of the element is given by:

$$\begin{aligned} v_{i,j}^g &= k \cdot \left\{\omega(g) \cdot \left(\left(1 - \beta(g)\right) \cdot v_{i,j}^{g-1} + \beta(g) \cdot \tilde{v}_{i,j}^{g-1}\right)\right\} \\ &+ \varphi_1 \cdot r_1 \left(pbest_j - p_{i,j}^{g-1}\right) + \varphi_2 \cdot r_2 \left(gbest_j - p_{i,j}^{g-1}\right), \quad (11) \end{aligned}$$

In equation (12), two components, namely intelligence inertia weight, $\omega(g)$, and random velocity component, $\tilde{v}_{i,j}^{g-1}$, are introduced to activate particles, when there is low diversity of particles, or the particles stop progressing for a long period of time. The intelligence inertia weight, $\omega(g)$, in equation (11) is determined by the diversity of the particles and the rate of the progress of the PSO. When the diversity of the particles is low and the progress of the PSO is low, the particles are unlikely to jump from a local optimum to the global optima. In order to relocate the particles from the local optimum to the global optima, the velocities of the particles are increased by increasing $\omega(g)$.

To further prevent the particles from being trapped in a local optimum, a random velocity component, $\tilde{v}_{i,j}^{g-1}$, is injected into the j-th element on the i-th particle. $\tilde{v}_{i,j}^{g-1}$ is determined based on (12), which is randomly generated and bound with 0.25 of the dynamic range of the particle element, and $r_3 \in [0 \quad 1]$ is an uniform random number. The resulting velocity, $v_{i,j}^{g}$, in equation (11) is determined based on the weighted sum of $v_{i,j}^{g-1}$, and $\tilde{v}_{i,j}^{g-1}$, where the intelligence weight factor, namely $\beta(g)$, is determined by the fuzzy inference system. $\beta(g)$, controls the number of random velocity components, which are injected into the regular velocity components, in order to force the particle element to escape the local optima.

$$\tilde{v}_{i,j}^{g-1} = 0.25 \cdot \left\{ r_3 \cdot \left(p_{\max} - p_{\min} \right) - p_{\max} \right\}, \tag{12}$$

The intelligence inertia weight, $\omega(g)$ formulated in equation (9), and the intelligence component factor, $\beta(g)$ formulated in equation (10), are treated as the two outputs of the fuzzy intelligence system of which the mechanism of the fuzzy intelligence system is similar to those for quality evaluation [13] and fault detection [14]. Both $\beta(g)$ and $\omega(g)$ are governed by two inputs, $\zeta(g)$ and ∂g, where $\zeta(g)$ is the standard deviation of fitness of all particles, and ∂g is the number of generations in which the best particle of the PSO has made no progress. $\zeta(g)$ is given by:

$$\zeta(g) = \sqrt{\frac{1}{N_s} \sum_{i}^{N_S} \left(J_i^g - \bar{J}_i^g \right)^2}, \tag{13}$$

where

$$\bar{J}_i^g = \frac{1}{N_s} \sum_{i=1}^{N_S} J_i^g . \tag{14}$$

and J_i^g is the fitness of the *i*-th particle at the *g*-th generation. ∂g is given by:

$$\partial g = \max_{J_{best}^{g} = J_{best}^{g'}} \left(g - g' \right) \tag{15}$$

where J_{best}^{g} and $J_{best}^{g'}$ are the fitness of the best particle at the *g*-th and the *g'*-th generations respectively with *g*> *g'*.

For determination of $\omega(g)$, more momentum is injected into particles by increasing the inertia weight of the particles based on the fuzzy intelligence system, if the particles are not active. Hence, large $\omega(g)$ should be used, if $\zeta(g)$ is small or ∂g is large. While the distances between particles is greater and the PSO keeps progressing within short generations, a small momentum should be enforced into particles, so that the particles can focus on a smaller search area. Therefore, $\omega(g)$ is decreased, if $\zeta(g)$ is large or ∂g is small. The following basic principles, P_1 and P_2, regarding the determination of inertia weight, $\omega(g)$, are used:

P$_1$: If $\zeta(g)$ is small and ∂g is large, $\omega(g)$ is increased.

P$_2$: If $\zeta(g)$ is large and ∂g is small, $\omega(g)$ is decreased.

For determination of $\beta(t)$, more random velocity components are injected into the regular velocity component based on the fuzzy inference system, in order to force the particle element to escape the local optima with a random movement, if the particles are not active. If $\zeta(g)$ is large and ∂g is small, the particles of swarm are activated enough to explore the search domain. It is not necessary to add the random velocity component into the regular velocity component as formulated in equation (10). However, if $\zeta(g)$ is small and ∂g is large, more random velocity components need to be added to the regular velocity component, in order to activate the particles. The following basic principles, P_3 and P_4, regarding the random component factor, $\beta(t)$, can be used:

P$_3$: If $\zeta(g)$ is small and ∂g is large, $\beta(t)$ is increased.

P$_4$: If $\zeta(g)$ is large and ∂g is small, $\beta(t)$ is decreased.

The value of $\omega(g)$ and $\beta(g)$ is given by taking the weighted average with respect to the membership functions:

$$\omega(g) = \sum_{j=1}^{9} m_j(g) \sigma_j , \tag{16}$$

and

$$\beta(g) = \sum_{j=1}^{9} m_j(g) \chi_j \tag{17}$$

respectively, where

$$m_j(t) = \frac{\mu_{\zeta}^{j}(\varsigma(g)) \times \mu_{\partial}^{j}(\partial g)}{\sum_{j=1}^{9} \mu_{\zeta}^{j}(\varsigma(g)) \times \mu_{\partial}^{j}(\partial g)}, \tag{18}$$

$\mu_{\zeta}^{j}(\varsigma(g))$ and $\mu_{\partial}^{j}(\partial g)$ are the membership functions with respect to the *j*-th fuzzy rule. They are represented in Gaussian form.

IV. Experimental results

In this research, a commercial speech recognizer, namely RSC-4x synthesis microcontroller [15], is used. It can obtain high correct recognition rate which is greater than 99% for speaker-dependent recognition, when the signal-to-noise-ratio is high. However, when the signal-to-noise-ratio is low, only a low accurate recognition rate can be produced. A set of commands were embedded in the commercial speech recognizer to evaluate the effectiveness of the proposed speech enhancement method: It consists of five speech commands, *jingle bells*, *santa claus is coming totown*, *sleigh ride*, *let it snow*, and *winter wonderland*. Those are typical commands which can be used on the voice control device for instructing the assigned machine to delivery different song products in the warehouse.

To simulate the typical voice control device, a square microphone array with four channels (20cm x 20 cm) was used. The command source was located one metre away from the microphone array, which simulates a warehouse environment. Two different noises were collected from the warehouses. One involves no human conversations and only warehouse noise is involved, which is called *non-conversation noise*. The other noise involves both human conversations and warehouse noise, which is called *conversation noise*. All the noise and command signals were recorded in a warehouse environment with a sampling rate of 16k Hz. The correct recognition rates for the command set are 20% and 40%, where the non-conversation noise and conversation noise with signal-to-noise-ratio of 0dB was corrupted the original signal respectively. Hence, it is necessary to enhance the recognition accuracy.

The parameter settings of the proposed PSO, which is referred to [16], were utilized to develop multi-channel filters for this commercial speech recognizer: The pre-defined number of generations = 100; the number of particles in the swarm = 100; inertia upper and lower weight factors, $w_{\max}$ = 0.90 and $w_{\min}$ = 0.4 respectively; acceleration constants ϕ_1 = 1 and ϕ_2 = 1; maximum velocity $v_{\max}$ = 0.2; the number of elements in the particle = 64 (i.e. $=4\times16$), where the filter length L=16 and the number of channels is 4.

Apart from using the proposed PSO, two global optimization algorithms were employed as a comparison. They are genetic algorithm (GA) [17], and classical PSO (CPSO) [18]. They were used to develop multi-channel filters to optimize correct recognition rates by solving equation (7).

GA creates a random initial population of chromosomes based on equation (4) of which each of them is evaluated based on equation (7). Evolution of the chromosomes was performed by crossover and mutation, where intermediate crossover [18] and mutation with randomly changing one or more filter parameters within their searching ranges were used. The following GA parameters [10] were used: population size =100; crossover rate = 0.8; mutation rate =0.001; and pre-defined number generations =100.

In CPSO, each particle is represented as equation (4). The velocity, $v_{i,k}^{g}$, and the k-th element of the i-th particle at the g-th generation, $p_{i,k}^{g}$, are calculated based on equation (10) and equation (9) respectively. The following parameters recommended by [20] were used in CPSO: the inertia weight factor = 0.5; both the acceleration constants, ϕ_1 =1 and ϕ_2 =1; the inertia upper factor $w_{\max}$ =0.9 and lower weight factor $w_{\min}$ =0.4 respectively; the maximum velocity $v_{\max}$ = 0.2; the pre-defined number of generations =100; and the number of particles in the swarm = 100.

Hence, the computational efforts used on the three algorithms were the same. Those algorithms were implemented by Matlab under PC with an Intel Core 2 vPro processor. The results obtained by the multi-channel filters generated by those algorithms were recorded. As those algorithms are the stochastic algorithms, different results can be obtained by runs. Therefore, 30 runs were performed on those algorithms, in order to evaluate the effectiveness of the stochastic algorithms. Among the 30 runs, we recorded the mean correct recognition rates, the maximum correct recognition rates, the minimum correct recognition rates and the variance of correct recognition rates obtained by the filters developed by those algorithms.

Table 1 shows the results obtained by the filters developed by those algorithms for the command set, which is corrupted by non-conversation noise and conversation noise with 0dB signal-to-noise-ratio. For the command set which was corrupted by non-conversation noise, Table 1 shows that the correct recognition rate obtained by the commercial speech recognizer without any enhancement is 20%. While those algorithms were used for developing the multi-channel filters, improvement of correct recognition rate can be obtained. Mean correct recognition rate of 46.4% can be obtained by GA. The maximum correct recognition rate and minimum correct recognition rate with 60% and 40% can be obtained respectively. The variance of the correct recognition rates of the 30 runs is 21.28. The mean correct recognition rate of the proposed PSO is 88.8, which is better than the correct recognition rate, 61.8, obtained by CPSO. Also, the maximum and the minimum correct recognition rates obtained by the proposed PSO are higher than those obtained by the other three algorithms.

The convergence plots of the three algorithms, GA, CPSO, and PSO are shown in Figures 1 and 2, which are regarding the commands contaminated with non-conversation noise and conversation noise respectively. They show that the GA prematurely converged to the poorest solution than the solution obtained by the other two stochastic algorithms, CPSO, and PSO. Although the proposed PSO cannot obtain the best solution among all algorithms in early generations, it can converge to the best solution compared with the other algorithms in the later generations. Hence, the proposed PSO is more likely to achieve better solutions than with the other tested methods.

Table 1 Correct recognition rates for the command set, *Phrases*, corrupted with non-conversation noise and conversation noise with 0dB

		No enhancement	**GA**	**CPSO**	**PSO**
Non-conversation noise	**Mean**	20	46.4	61.8	88.8
	Max	20	60	80	100
	Min	20	40	20	80
	Var	0	21.28	93.54	18.96
Conversation noise	**Mean**	40	46.2	68.4	98.2
	Max	40	60	80	100
	Min	40	40	60	80
	Var	0	63.15	10.97	7.48

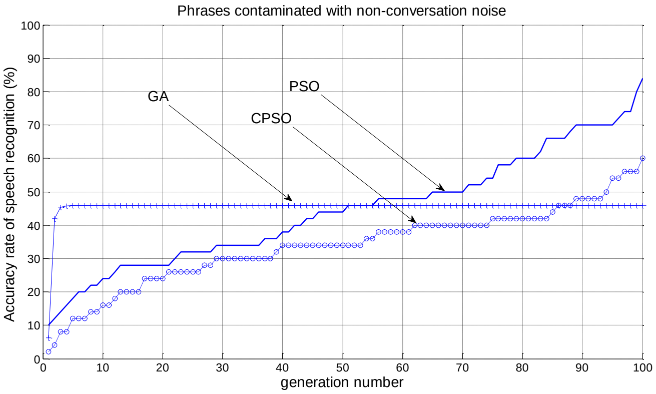

Fig. 1 Contaminated with non-conversation noise

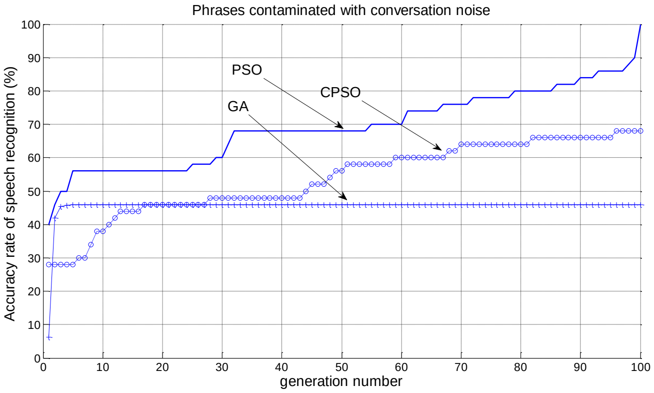

Fig. 2 Contaminated with conversation noise

V. Conclusion

In this paper, a PSO was proposed to optimize a multi-channel filter for commercial speech recognizers. A speech recognition optimization problem was formulated by developing an optimal multi-channel filter, in order to maximize accuracy of a commercial speech recognizer within a noisy environment. The approach was evaluated by implementing on a commercial speech recognizer embedded with a set of speech commands. Results show that the PSO can improve the accuracy of the commercial speech recognizer which was used under noisy environment of a warehouse. The results show that the proposed PSO outperforms GA, and CPSO in the development of the multi-channel filters for the commercial speech recognizer.

We are currently using the proposed PSO on enhancing speech signals for the commercial speech recognizer which is embedded with digital commands. Further experimental results involved more advanced PSO algorithms and different types of noise will be reported shortly.

Acknowledgement

The second author was supported by the RGC Grant PolyU. (5365/09E), NSFC Grant (10901170) and the Research Committee of the Hong Kong Polytechnic University.

References

1. F. Alonso-Martin and M.A. Salichs, Integration of a voice recognition system in a social robot, Cybernetic and Systems, vol. 42, pp. 215-245, 2011.
2. Y. Qian, J. Liu and M.T. Johnson, Efficient embedded speech recognition for very large vocabulary mandarin car-navigation systems, IEEE Transactions on Consumer Electronics, vol. 55, no. 3, pp. 1496-1500, 2009.
3. Spelta, V. Manzoni, A. Corti, A. Goggi and S.M. Savaresi, Smartphone-based vehicle-to-driver/environment interaction system fro motorcycles, IEEE Embedded systems letters, vol. 2, no. 2, pp. 39-42, 2010.
4. Woodland P., Speech recognition, IEE Colloquium on Speech and Language Engineering, 1–5, 1998.
5. P.C. Loizou, Speech enhancement: theory and practice, series of Signal Processing and Communications, Florida, USA: CRC Press, 2007.
6. M. Brandstein and D. Ward, Microphone arrays: signal processing techniques and applications, Springer-Verlay, 2001.
7. Van Veen and K.M. Buckley, Beamforming: A versatile approach to spatial filtering, IEEE Signal Processing Magazine, vol. 5, pp. 4-24, 1988.
8. M. Seltzer, B. Raj, and R. Stern, Likelihood-maximizing beamforming for robust hands-free speech recognition, IEEE Transactions on Speech and Audio processing, vol. 12, no. 5, pp. 489–498, 2004.
9. J. Kennedy and R. Eberhart, Particle swarm optimization, in Proc. 30th IEEE Conf. Decision and Control, vol. 4, 1995, pp. 1942-1948.
10. Z.H. Zhan, J. Zhang and H.S.H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man, and Cybernetic – Part B: Cybernetics, vol. 39, no. 6, pp. 1362-1381, 2009.
11. A. Hyvarinen and E. Oja, Independent component analysis: algorithms and applications, Neural Networks, vol. 13. no. 4, pp. 411-430, 2000.
12. R.C. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in Evolutionary Programming VII. New York: Springer-Verlag, LNCS, vol. 1447, 1998, pp. 611-616.
13. G. Acciani, G. Fornarelli and A. Giaquinto, A fuzzy method for global quality index evaluation of solder joints in surface mount technology, IEEE Transactions on Industrial Informatics, vol. 7, no. 1, pp. 115-124, 2011.

14. C.H. Lo, E.H.K. Fung and Y.K. Wong, Intelligent automatic fault detection for actuator failures in aircraft, IEEE Transactions on Industrial Informatics, vol. 5, no. 1, pp. 50-55, 2009.
15. Sensory, INC, Company, Making electronic devices talk and hear, 2010.
16. M.O. Neill and A. Brabazon, Grammatical Swarm: The generation of programs by social programming, Natural Computing, 5, 443-462, 2006.
17. K.Y. Chan, C.K.F. Yiu, S.Y Low, S. Nordholm and S.H. Ling, Speech recognition enhancement using beamforming and a genetic algorithm, Proceedings of IEEE International Conference on Network and System Security, pp. 510-515, 2009.
18. J. Kennedy and R. Eberhart, Particle swarm optimization, in Proc. 30th IEEE Conf. Decision and Control, vol. 4, 1995, pp. 1942-1948.
19. A. A. E. Ahmed, L. T. Germano and Z. C. Antonio, A hybrid particle swarm optimization applied to loss power minimization, IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 859-866, May 2005.
20. R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, In Proc. Of IEEE Congress on Evolutionary Computing, Jul 2000, pp. 84-88.
21. G. E.-P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for Experiments: Design, Innovation, and Discovery*, 2nd ed. New York: Wiley, 2005.