Speech Enhancement Strategy for Speech Recognition Microcontroller under Noisy Environments

Kit Yan Chan[1], Sven Nordholm[1], Cedric Yiu[2], Roberto Togneri[3]

[1]Curtin University, Department of Electrical and Computer Engineering, Western Australia, Australia
[2]The Hong Kong Polytechnic University, Department of Applied Mathematics, Western Australia, Australia
[3]The University of Western Australia, School of Electrical, Electronic and Computer, Western Australia, Australia

***Abstract* — Industrial automation with speech control functions is generally installed with a speech recognition sensor which is used as an interface for users to articulate speech commands. However, recognition errors are likely to be produced when background noise surrounds the command spoken into the speech recognition microcontrollers. In this paper, a speech enhancement strategy is proposed to develop noise suppression filters in order to improve the accuracy of speech recognition microcontrollers. It uses a universal estimator, namely a neural network, to enhance the recognition accuracy of microcontrollers by integrating better signals processed by various noise suppression filters, where a global optimization algorithm, namely an intelligent particle swarm optimization, is used to optimize the inbuilt parameters of the neural network in order to maximize accuracy of speech recognition microcontrollers working within noisy environments. The proposed approach overcomes the limitations of the existing noise suppression filters intended to improve recognition accuracy. The performance of the proposed approach was evaluated by a speech recognition microcontroller, which is used in electronic products with speech control functions. Results show that the accuracy of the speech recognition microcontroller can be improved using the proposed approach, when working under low signal to noise ratio conditions in the industrial environments of automobile engines and factory machines.**

***Index Terms* —** Speech recognition microcontroller, noise suppression filters, background noise, speech control, acoustic signal enhancement, neural networks, particle swarm optimization

I. INTRODUCTION

Speech recognition microcontrollers have benefited speech control of industrial automation processes such as factory automation [30], logistic automation [29], robotic manufacture [1, 7] for over two decades [14]. The industrial automation processes can be controlled directly based on speech commands without physical contact. These industrial automation processes involve a speech recognition microcontroller which is used as an interface and consists of two components: 1) a microphone array which is used to collect acoustic speech commands from users; and 2) a speech recognition microcontroller which is used to recognize acoustic speech commands collected from the microphone. These speech recognition microcontrollers are generally developed based on a limited number of speech signals contaminated by certain types of acoustic noise. As the time and costs for training the microcontrollers are limited, it is impractical to manufacture a microcontroller which can work ideally within every noisy environment. Hence, inaccurate recognitions are likely to occur when microcontrollers are required to work within unknown noisy environments which have not been pre-trained. Also, it is impossible to change or tune any inbuilt parameters in microcontrollers, which are often purchased commercially and are encapsulated. Therefore, a practical and convenient way to improve the recognition accuracy of the microcontrollers is to pre-process the noisy speech by using an effective noise suppression filter. To enhance the recognition accuracy of the microcontroller, multi-channel beamformers [11, 36] can be used to enhance noisy speech which is contaminated by near-end noise. However, those approaches have the common limitations that the signal source needs to be tracked according to the difference between the signal spectrums collected from multi-channels. Although recognition accuracy can be improved, their mechanisms are computationally complex, and also their inbuilt parameters need to be calibrated with respect to the location of noise sources and speech sources.

Therefore, simpler noise-suppression filters, which involve only a single channel, are commonly used [8, 21] for speech enhancement, before performing speech recognition by the microcontroller. They first identify both active and inactive periods of the speech signal, and then estimate the noise spectrum based on

the inactive periods. When the noise spectrum is available, the speech spectrum can be identified by removing the noise spectrum from the original signal. Therefore, the most crucial aspect of noise-suppression filters is to estimate the noise spectrum. More specifically, when the noise spectrum is under-estimated, annoying residual noise or musical noise is perceivable. When the noise spectrum is over-estimated, original speech is distorted leading to a loss of speech quality and intelligibility [4]. Yiu et al. [34] also mention that enhancing the recognition accuracy of a microcontroller is a multi-criteria problem, where distortion and noise suppression need to be optimized. Even though the noise spectrum can be estimated correctly, signal spectrum matched to the training conditions which is critical for speech recognition may not be maintained [19]. Hence, inaccurate speech recognition may still occur, when a single noise suppression filter is used to pre-process the noisy speech before performing speech recognitions by microcontrollers.

In this paper, a speech enhancement strategy is proposed to maximize recognition accuracy of speech recognition microcontrollers working within noisy environments. It intends to optimally combine various noise suppression techniques which all have various features by performing the following operations:

(i) An optimization problem is formulated to determine optimal noise suppression filters used to enhance the recognition accuracy of microcontrollers. As the recognition accuracy is maximized by optimizing the noise suppression filters, the inbuilt and usually inaccessible parameters of the microcontrollers do not need to be tuned to enhance the recognition accuracy. Also, the limitation of noise suppression filters can be overcome, as they are designed to handle only limited degrees of acoustic criteria which may not directly relate to the optimization of recognition accuracy.

(ii) After generating the optimal noise suppression filters, the enhanced signals generated by the noise suppression filters are integrated using a universal estimator, namely a neural network [28], which can be used effectively to develop filters for speech enhancement [15, 43, 23] and performing speech recognition [33, 13]. Hence, the signals refined by the integration are more likely to produce accurate recognitions than those enhanced solely using a single noise suppression filter.

(iii) A global optimization algorithm, namely intelligent particle swarm optimization (IPSO) [27], is used to solve the optimization problem formulated for enhancing the recognition accuracy of microcontrollers, which is non-differentiable and discrete, as it is less likely to be trapped in local optima [3, 6, 22, 24]. When the IPSO starts converging prematurely, the approach of injecting activating components into the particles is used in order to increase the diversity of the particles [35]. Hence, the IPSO is more likely to search for the global optimum.

The rest of the paper is organized as follows. Section 2 presents the formulation of the optimization problem for developing noise suppression filters which are intended to enhance the accuracy of speech recognition microcontrollers. Section 3 discusses the mechanisms of the noise suppression filters aimed at improving the accuracy of speech recognition microcontrollers working within extremely noisy environments. Section 4 presents the mechanism of the IPSO, which is developed to optimize the noise suppression filters for enhancing recognition accuracy. In Section 5, the effectiveness of the noise suppression filters is evaluated based on a commercial microcontroller, which is commonly-used in electronic products involving speech control functions. Results show that significant improvement in terms of recognition accuracy can be achieved when the microcontroller is working in non-stationary noisy environments (e.g. factory noise conditions). Section 6 concludes the paper.

II. ENHANCEMENT OF SPEECH RECOGNITION MICROCONTROLLER USING NOISE SUPPRESSION FILTERS

Figure 1 shows the mechanism of the speech recognition microcontroller which is used in electronic products with speech control functions. It uses the microphone to collect speech commands from the user. Then, a speech recognition microcontroller is used to recognize the speech command collected from the microphone. These speech commands embedded in the microcontroller can be a set of single words such as numerical digits, ‘yes’ or ‘no’ decisions, ‘left’ or ‘right’ directions etc. They can also be a set of phrases, such as operational commands for manufacturing processes, action commands for speech control toys etc. However, background noise is generally not avoidable which affects the accuracy of the microcontroller.

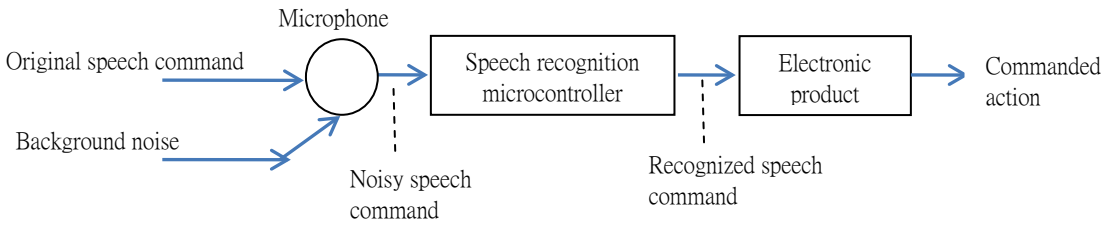

Fig. 1 Mechanism of speech recognition microcontrollers used in electronic products with speech control functions

Here, a speech recognition microcontroller, $\Re(.)$, is used to identify n inbuilt speech commands, $\{u^1, u^2, ..., u^n\}$, where the contaminated signal, $x_j^i(t)$, received by $\Re(.)$ is denoted by

$$x_j^i(t) = s_j^i(t) + v(t), \tag{1}$$

where $t = 1, 2, ..., m$ of which m is the total number of samples (i.e. duration of the signal); $s_j^i(t)$, is the i-th speech command voiced out by the j-th regular user, with j=1,2..,N of which N is the number of regular users; i=1,2,…n for which n is the number of commands; and $v(t)$, is the background noise. Let $\hat{i}$ be the recognized command from $\Re(.)$ for the contaminated signal, $x_j^i(t)$, which is given by,

$$\hat{i} = \Re(x_j^i), \tag{2}$$

where $x_j^i = [x_j^i(1), x_j^i(2), ..., x_j^i(m)]$; the n likelihoods with respect to the n inbuilt speech commands, $\{u^1, u^2, ..., u^n\}$ embedded on $\Re(.)$ are given by the likelihood vector, $\{L_1(x_j^i), \cdots, L_n(x_j^i)\}$;

$L_k(x_j^i)$ indicates the similarity between the features of the spoken i-th speech command, x_j^i and the features of the reference k-th speech command, u^k; and $\hat{i}$ is given by:

$$\hat{i} = \arg\max_i \left(\{L_1(x_j^i), \cdots, L_n(x_j^i)\} \right) \tag{3}$$

of which $\arg\max(..)$ indicates the position of the maximum likelihood of $\{L_1(x_j^i), \cdots, L_n(x_j^i)\}$. Hence, a correct recognition is obtained with respect to x_j^i, if $\hat{i} = i$. Otherwise, an incorrect recognition is obtained,

if $\hat{i} \neq i$. When the signal to noise ratio (SNR) is low due to the presence of background noise, recognition errors are likely to be produced by $\Re(.)$. A speech enhancement filter, $\mathrm{F}(.)$, is interfaced between $\Re(.)$ and the regular users with respect to the specified range of SNRs, in order to enhance the recognition accuracy. $\mathrm{F}(.)$ is developed by solving the following optimization problem which aims to train the $\mathrm{F}(.)$ to work within multi-conditions with respect to multi-SNRs and multi-users:

$$\max_{\mathrm{F}(.)} J = \sum_{j=1}^{N}\sum_{i=1}^{n}\sum_{k=1}^{K+1}\theta_j^i(\sigma_k) \text{ with } \theta_j^i(\sigma_k) = \begin{cases} 0 & \text{if } \Re\left(\mathrm{F}\left(x_j^i(\sigma_k)\right)\right) \neq i \\ 1 & \text{if } \Re\left(\mathrm{F}\left(x_j^i(\sigma_k)\right)\right) = i \end{cases} \quad (4)$$

where $x_j^i(\sigma_k) = s_j^i + v_k$;

$\sigma_k = \dfrac{|s_j^i|}{|v_k|}$ is the SNR given by $\sigma_k = \sigma_{\min} + \dfrac{(\sigma_{\max} - \sigma_{\min})}{K} \cdot (k-1)$ with $|s_j^i|$ and $|v_k|$ the power of the original speech signal, s_j^i, and the noise, v_k, respectively; $\sigma_{\max}$ and $\sigma_{\min}$ are the specified maximum and minimum SNRs of $x_j^i(\sigma_k)$ respectively which are specified with respect to the operational environment of the speech recognition microcontrollers; and K is the number of iterations within the SNR range.

A correct recognition is obtained with respect to the *i*-th speech command voiced out by the *j*-th user, if $\Re\left(\mathrm{F}\left(x_j^i(\sigma_k)\right)\right) = i$; an incorrect recognition is obtained, if $\Re\left(\mathrm{F}\left(x_j^i(\sigma_k)\right)\right) \neq i$. By solving the optimization problem formulated in (4), an optimal speech enhancement filter, $\mathrm{F}(.)$, can be produced with respect to the speech commands, the regular users and the specified SNR range. Prior to solving the optimization problem formulated in (4), the appropriate configuration of $\mathrm{F}(.)$ needs to be determined. Section III introduces the single-channel and hybrid architectures for $\mathrm{F}(.)$.

III. NOISE SUPPRESSION FILTER DESIGN

A. Single-channel filter

Development of a single-channel filter, $\mathrm{F}(.)$, with respect to the speech recognition microcontroller is illustrated in Fig. 2, where $\mathrm{F}(.)$ is intended to filter background noise by using the spectral information in the frequency domain in order to enhance accuracy of the speech recognition microcontroller.

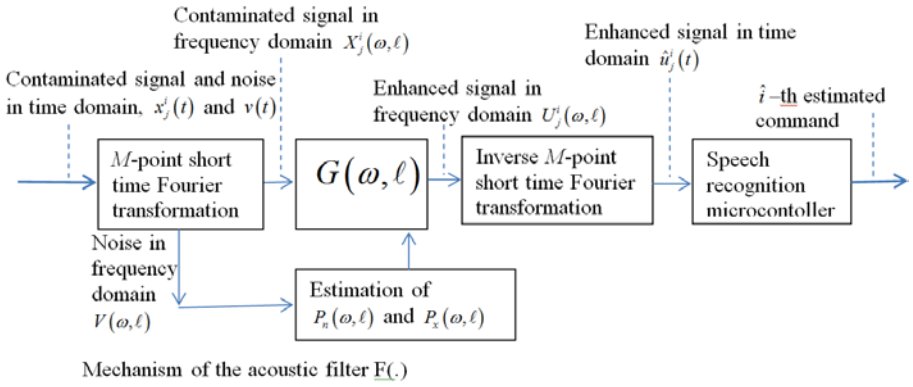

Fig. 2 Mechanism of a single-channel filter, $\mathrm{F}(.)$, for speech recognition microcontroller

In $\mathrm{F}(.)$, the *M*-point short time Fourier transformation is first used to map the contaminated signal, $x_j^i(t)$ and the noise $v(t)$ formulated in (1) from time domain into frequency domain, as $X_j^i(\omega,\ell)$, and $V(\omega,\ell)$ respectively, where $v(t)$ need to be represented from non-speech activity; ω denotes a real angular center frequency given by $\omega \in [\omega_0, \omega_1, ..., \omega_{K-1}]$; ℓ is the time frame index given by $\ell \in [1, 2, ..., L]$; and *K* is the number of bands and *L* is the number of frames. The command signal in frequency domain, $S_j^i(\omega,\ell)$, is denoted as,

$$X_j^i(\omega,\ell) = S_j^i(\omega,\ell) + V(\omega,\ell). \tag{5}$$

Then, the smoothed spectrum of the contaminated signal, $P_x(\omega,\ell)$, and the noise spectrum, $P_n(\omega,\ell)$, can be estimated based on the statistical characteristics of $X_j^i(\omega,\ell)$, and $V(\omega,\ell)$, using Boll's extended spectral subtraction [3] as,

$$P_x(\omega,\ell) = \alpha P_x(\omega,\ell-1) + (1-\alpha)\left|X_j^i(\omega,\ell)\right|^2 \quad (6)$$

and $P_n(\omega,\ell) = \beta P_n(\omega,\ell-1) + (1-\beta)P_{nn}(\omega,\ell)$ (7)

respectively, where $|.|$ is the absolute value operator, and α and β are the smoothing parameters. $P_{nn}(\omega,\ell)$ in (7) is estimated by the minimum statistics approach [20] in which the peaks of the noisy signal represent the active speech period and the valleys of the noisy signal represent noise power levels. Hence, the statistics of the noise can be estimated by tracking the minimum power within a finite window which is large enough to bridge high power speech periods. Based on $P_x(\omega,\ell)$ and $P_n(\omega,\ell)$, the enhanced command signal, $\hat{S}_j^i(\omega,\ell)$, can be generated based on a gain function, $G(\omega,\ell)$ as:

$$\hat{S}_j^i(\omega,\ell) = G(\omega,\ell)\cdot X_j^i(\omega,\ell), \quad (8)$$

where $G(\omega,\ell)$ is a function of $P_n(\omega,\ell)$ and $P_x(\omega,\ell)$ which depends on the mechanism of the filter. When $\hat{S}_j^i(\omega,\ell)$ illustrated in (8) in the frequency domain is available, the enhanced command signal, $\hat{s}_j^i(t)$, in time domain can be generated by the inverse *M*-point short-time Fourier transformation, where $\hat{s}_j^i(t)$ is the enhanced speech outcome from $\mathrm{F}(.)$, which can be fed into the speech recognition microcontroller.

To obtain maximum recognition accuracy under multi-conditions, the smoothing parameters α and β need to be optimized with respect to (4), as both parameters determine the noise spectrum which can be removed from the contaminated signal by $\mathrm{F}(.)$. If the noise spectrum is underestimated, the noise still engages with the original signal after filtering. Hence, inaccurate recognition can still be produced. If noise spectrum is overestimated, the $\mathrm{F}(.)$ not only removes the noise from the contaminated signal, but it also distorts the spectrum of the command signal. Inaccurate recognition occurs. Hence, α and β in $\mathrm{F}(.)$ need to be optimized with respect to the multi-conditions by maximizing the accuracy of the speech recognition microcontroller as defined in (4). The optimization mechanism is illustrated in Figure 3.

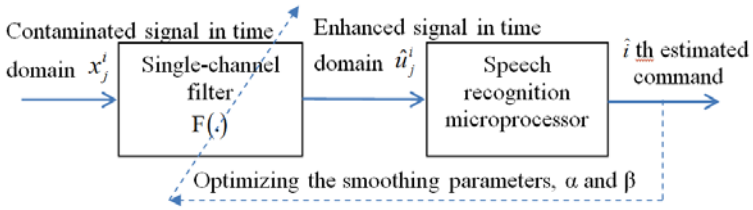

Fig. 3 Design of a single-channel filter with respect to a speech recognition microcontroller

B. Hybrid filter

Although the single-channel filter can enhance the contaminated signal, the various single-channel filters have unique properties in terms of noise reduction and signal distortion. It is difficult to find a single filter that works well against various SNRs and various types of noise. Also, they have little success in improving or maintaining speech quality and intelligibility of the original signal which is critical for speech recognition [19]. Speech quality and intelligibility could be destroyed after filtering, although we can clearly hear that the filtered signals are significantly enhanced. Hence, the accuracy of the speech recognition microcontroller can still be poor. In order to adjust the distortion and noise suppression differently, a hybrid filter which uses the mechanism of a three-layer feed-forward neural network is developed to combine the outcomes of various single-channel filters to the original contaminated signal, after the smoothing parameters of the single-channel filters are optimized. Hence, the outcomes of the various single-channel filters which have different properties in noise reduction and signal distortion can be fused. It also intends to overcome the limitation of solely using a single-channel filter that is likely to distort the original user's speech spectrum, as the original contaminated signal is used as one of the inputs to the hybrid filter.

The mechanism of the hybrid filter, $F_{NN}(.)$, is illustrated in Figure 4. It fuses the outcomes of the n_s optimal single-channel filters and the contaminated signal, where the smoothing parameters, α and β, have been optimized. The outcomes of the n_s single-channel filters, $\mathrm{F}_1(.)$, $\mathrm{F}_2(.)$, …. and $\mathrm{F}_{n_s}(.)$, are

denoted as $\hat{u}^i_{1,j}$, $\hat{u}^i_{2,j}$, …, and $\hat{u}^i_{n_s,j}$ respectively, and x^i_j is the contaminated signal. The outcome of $F_{NN}(.)$ is denoted by the following equation:

$$\hat{u}^i_{NN,j} = F_{NN}\left(x^i_j, \hat{u}^i_{1,j}, \hat{u}^i_{2,j}, ..., \hat{u}^i_{n_s,j}\right) \tag{9}$$

where $\hat{u}^i_{NN,j} = \left[\hat{u}^i_{NN,j}(L+1), \hat{u}^i_{NN,j}(L+2), ..., \hat{u}^i_{NN,j}(m)\right]$; $x^i_j = \left[x^i_j(1), x^i_j(2), ..., x^i_j(m)\right]$;

$\hat{u}^i_{1,j} = \left[\hat{u}^i_{1,j}(1), \hat{u}^i_{1,j}(2), ..., \hat{u}^i_{1,j}(m)\right]$; $\hat{u}^i_{2,j} = \left[\hat{u}^i_{2,j}(1), \hat{u}^i_{2,j}(2), ..., \hat{u}^i_{2,j}(m)\right]$;…;

$\hat{u}^i_{n_m,j} = \left[\hat{u}^i_{n_m,j}(1), \hat{u}^i_{n_m,j}(2), ..., \hat{u}^i_{n_m,j}(m)\right]$;

t=L+1, 2,…, m; and L denotes the filter length of the hybrid filter. $F_{NN}(.)$ is represented as the architecture of the universal estimator namely neural network as:

$$\hat{u}^i_{NN,j}(t) = \sum_{k=1}^{n_h} w_k f_t\left[v_{0,k} \cdot x^i_j(t) + \sum_{p=1}^{n_s}\sum_{l=0}^{L-1} v_{(p-1)\cdot L+l+1,k} \cdot \hat{u}^i_{p,j}(t-l) - b_k\right] - b \tag{10}$$

where n_h denotes the number of the hidden nodes; w_k denotes the weight of the link between the hidden node and the output; v_{ij} with $i > 0$ denotes the weight between the hidden node and the outcome of the filtered signal of $\mathrm{F}_i(.)$; v_{ij} with $i = 0$ denotes the weight between the hidden node and the contaminated signal, $x^i_j(t)$; b_j and b, denote the biases for the j-th hidden nodes and output nodes respectively; and the transfer function, $f_t(.)$, is used to compress each sample of $\hat{u}^i_{m,j}(t)$, and $x^i_j(t)$. The transfer function can be the ‘logsig’ function, ‘tansig’ function and ‘purelin’ function. A judicious choice of $f_t(.)$ enables the neural network to approximate a nonlinear filter [15], which has been shown to enhance signals for speech recognitions [17].

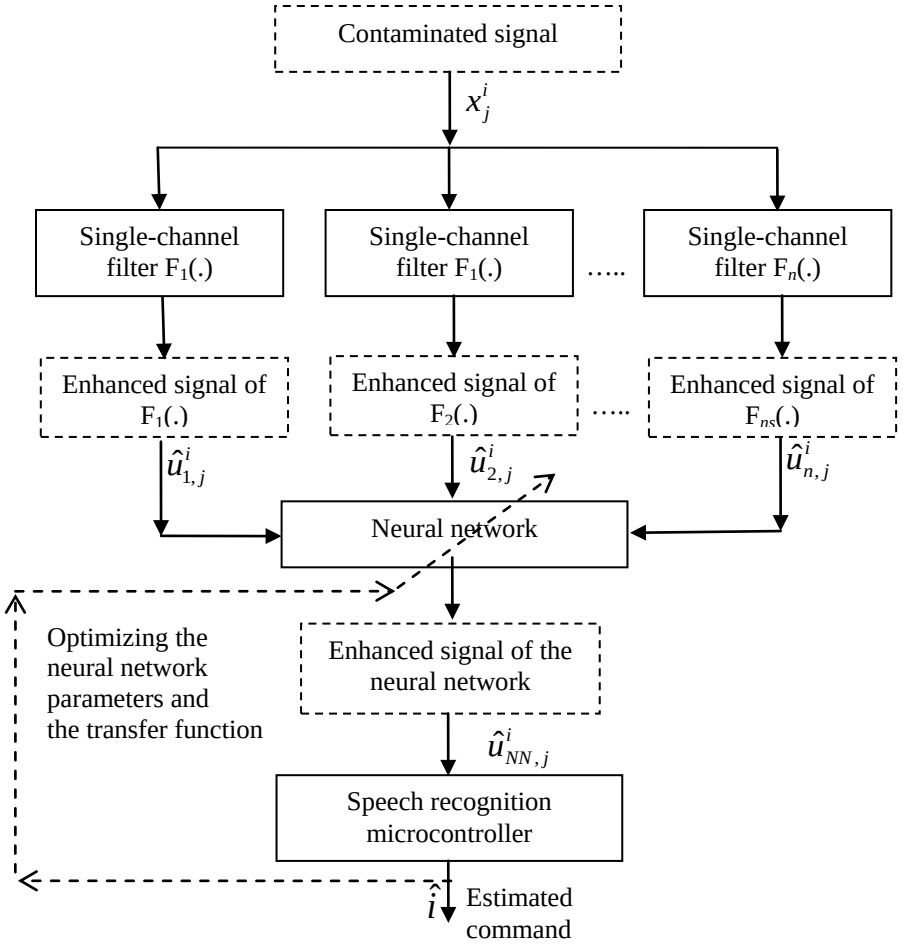

Fig. 4 Mechanism and design of the hybrid filter

$F_{NN}(.)$ is intended to produce better recognition results by combining the outcomes of the single-channel filters. It also fine-tunes the outcomes of the single-channel filters in order to further enhance the unknown acoustic criteria for the speech recognition microcontroller. The neural network parameters are optimized by maximizing the accuracy of the speech recognition microcontroller as defined in (4). However, the landscape of the formulated in (4) is discrete and discontinuous in nature, so the number of correct recognitions jumps from one level to another level. As (4) involves non-differentiable characteristics, an intelligent particle swarm optimization, namely IPSO discussed in Section IV, is proposed to maximize the recognition accuracy formulated in (4).

IV. INTELLIGENT PARTICLE SWARM OPTIMIZATION

The intelligent particle swarm optimization, IPSO, which mimics the social behavior associated with bird flocking and fish schooling [16], is used for the filter design of the hybrid filter parameters, which include those for each single-channel filter and for the neural network. The individuals in the population are called particles. Each particle is a potential solution for the optimization problem formulated in (4) which is a discrete and discontinuous function. It tries to find the best position by flying through the multidimensional space. The sociological behaviour which is controlled by the IPSO operations is used to guide the swarm, thereby probing the most promising areas of search space.

The IPSO first creates a random initial swarm which consists of N_s particles. The position of the *i*-th particle at the *g*-th generation is given by $P_i^g = \left(p_{i,1}^g, p_{i,2}^g, ..., p_{i,n_p}^g\right)$, where P_i^g consists of n_p elements, $p_{i,k}^g$ with *k*=1,2,…, n_p ; n_p is the number of parameters of the filter; and the initial value of p_{i,n_p}^g is randomly generated in the range between the parameters of the filter.

For developing the single-channel filter discussed in Section III.A, two smoothing parameters, α and β, which are formulated in equations (6) and (7) respectively, need to be optimized. Hence, n_p is set as $n_p = 2$, and the elements $p_{i,1}^g$ and $p_{i,2}^g$ are represented by α and β respectively, where $\alpha, \beta \in [0..1]$. When other smoothing parameters exist on the single-channel filter, they can also be included as the elements of the IPSO and can be optimized by the IPSO. In order to develop the hybrid filter discussed in Section III.B, the optimal smoothing parameters obtained are used directly in the hybrid filter and the neural network parameters in the hybrid filter are optimized by the IPSO. The elements, $p_{i,k}^g$ with *k*=1,2,…,$\left(n_p - 1\right)$, represent the neural network parameters, w_k, v_{ij}, b_j and *b*, where all w_k, $v_{ij} \in [-1..1]$, and all b_j, $b \in [-10..10]$. The last element, p_{i,n_p}^g, represents the transfer function, $f_t(.)$, which can either be the ‘logsig’ function, ‘tangsig’ function or ‘purelin’ function. Hence, p_{i,n_p}^g is a discrete particle with $p_{i,n_p}^g \in [0..1]$ [26], where

$$f_t(.)=\begin{cases}\log sig(.) & 1/3>p_{i,n_p}^g\geq 0\\ \tan sig(.) & \text{if} \quad 2/3>p_{i,n_p}^g\geq 1/3\\ purelin(.) & 1\geq p_{i,n_p}^g\geq 2/3\end{cases}. \tag{12}$$

The position of the *j*-th element of the *i*-th particle, $p_{i,k}^g$, is updated based on (13) at the *g*-th generation as:

$$p_{i,j}^g = p_{i,j}^{g-1} + v_{i,j}^g . \tag{13}$$

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 classical PSO, $v_{i,j}^g$ is given by:

$$v_{i,j}^g = \omega(g)\cdot v_{i,j}^{g-1} + \phi_1\cdot r_1\left(pbest_{i,j} - p_{i,j}^{g-1}\right) + \phi_2\cdot r_2\left(gbest_j - p_{i,j}^{g-1}\right), \tag{14}$$

where $pbest_i = \left[pbest_{i,1}, pbest_{i,2}, ..., pbest_{i,n_p}\right]$ is the best position of the *i*-th particle moved so far, and $gbest = \left[gbest_1, gbest_2, ..., gbest_{n_p}\right]$ 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; $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$; and $\omega(g)$ is the inertia, which is a constant value for classical PSO. Its value is in the range of [0.1...1.1] which is recommended by [9]. Then each particle, P_i^g, is evaluated by the recognition accuracy based on the optimization problem formulated in (4).

As (4) is a step function where the number of correct recognitions jumps from a level to another level, the fitness of the particles might not increase when their positions are slightly altered. Even though the particles can move randomly, they might keep moving continuously on their landing step which is the local optima. Therefore, two components, namely the varying inertia and random components, are proposed in the IPSO. They are intended to activate particles to search for a better solution, when the particles have not progressed for a long period of time. The varying inertia, $\omega(g)$, in (14) is determined based on the progress of the IPSO. $\omega(g)$ is determined based on the following formulation:

$$\omega(g) = 0.6 + 0.5\cdot\left(1 - e^{-\alpha_e\cdot\partial g}\right) \tag{15}$$

where ∂g is the number of generations for which the fitness of the best particle remains unchanged. ∂g is given by:

$$\partial g = \max_{J_{best}^{g} = J_{best}^{g'}} \left(g - g' \right), \tag{16}$$

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'*. $\alpha_e = 0.2$ is used in this research. When ∂g is very large or the IPSO has made no progress for a long time, the particles are unlikely to jump from a local optimum to a better solution with better recognition accuracy. In order to relocate the particles from the local optimum to a better solution, the velocities of the particles are increased by increasing $\omega(g)$. Hence, $\omega(g)$ is equal to the value of 1.1 which is the maximum value of inertia recommended by [9], when ∂g is very large. When the velocity of the particle is higher, the particle can move for a longer distance within a generation. Hence, the particle is more likely to jump to a better solution with better recognition accuracy. However, when ∂g is small, the IPSO keeps progressing within small generations. Small $\omega(g)$ is needed in order to let the particles explore a small vicinity by refining the positions of the particles with a better recognition accuracy.

To further avoid the particles being trapped in a local optimum with poor recognition accuracy, a random component is injected into each particle based on the following equation:

$$p_{i,k}^{g} = p_{i,k}^{g} + 0.1 \times r_3, \text{ if } r_4 < 0.3 \cdot \left(1 - e^{-\alpha_e \cdot \partial g}\right) \tag{17}$$

where r_3 is randomly generated within the range of the particle, $p_{i,k}^{g}$, and r_4 is randomly generated between 0 to 1. Hence, more random components are injected into the particles, when ∂g is large or the PSO has made no progress for a long time. When the movements of the particles are larger, the particles have more opportunity to move to a better solution resulting in better recognition accuracy for the microcontroller.

V. ENHANCEMENT OF SPEECH RECOGNITION MICROCONTROLLER UNDER NOISY ENVIRONMENTS

In this section, the effectiveness of the noise suppression filters developed by the IPSO is evaluated based on a speech recognition microcontroller which operates within various noisy environments.

A. Speech recognition microcontroller

A commercial microcontroller, namely the RSC3X synthesis microcontroller [32], is used in the speech recognition microcontroller. This commercial microcontroller is designed with advanced audio features which have been applied widely to various electronic products and automatic systems with speech control functions. This commercial microcontroller can achieve a high recognition accuracy which can be greater than 99% for speaker-dependent recognition, when the SNRs are high. However, incorrect recognitions are likely to be produced when the SNRs are low. Therefore, this commercial microcontroller needs to be enhanced in order to upgrade the recognition accuracy within noisy environments.

In this research, the commercial microcontroller is implemented to identify five commands for five Christmas carols including '*Jingle Bells*', '*Santa Claus is Coming to Town*', '*Sleigh Ride*', '*Let It Snow*', and '*Winter Wonderland*'. For example, when the user voices the speech command, '*Let It Snow*', the commercial microcontroller determines the speech command which has the highest likelihood of being recognised among the five speech commands. The speech command is recognized correctly, if '*Let It Snow*' is recognized. Here, the noisy environments in factories are considered, as speech controls are common in factories. If the commercial microcontrollers can work accurately, factory operators can simply give verbal commands to control minor tasks in order to remain focused on the main tasks requiring both hands without taking their eyes off. Interruptions of their main task can be minimized.

B. Experimental set-up

To simulate the noisy factory environments, we first recorded the five speech commands voiced by ten people, eight males and two females, in a recording studio. The recorded speech commands were assumed to be noise free, and then were contaminated artificially with two noisy environments from two noise data files, factory noise and engine noise from the NOISEX-92 database. Both factory noise and engine noise

simulate the noisy environment in a factory, where the factory noise was recorded near plate-cutting and electrical welding equipment, and the engine noise was recorded in the engine room when the engine was running.

The recorded speech commands were contaminated by noise at different SNRs, where the SNR was determined by the energy ratio of the clean speech signal including periods of silence and the noise added within each speech command voiced by the user. Using different SNRs simulates the real environment in which the users voice the commands with different volumes and at different distances from the speech recognition microcontroller. A stronger signal is received by the speech recognition microcontroller when the user voices the commands with a stronger volume or at a shorter distance from the microcontroller. A weaker signal is received when the users voice the commands with a weaker volume or at a further distance from the speech recognition microcontroller. Cross-validations were used to evaluate the effectiveness of the noise suppression filters which are developed by the IPSO. Here the cross-validations were repeated five times. The training data was generated by the seven male speech commands and the one female speech command, which were contaminated with various SNRs. The test data was generated by the remaining one male and one female speech commands contaminated with various SNRs. By doing this, we can evaluate the performance of the noise suppression filters in terms of the recognition accuracy under multi-conditions where various SNRs and multi-users are considered. This approach also simulates a practical situation whereby the noise suppression filter is trained by regular users. Hence, one can evaluate whether the noise suppression filter can work well with respect to new users which have not been involved in training.

The performance of the speech recognition microcontroller was initially tested by decreasing the SNRs gradually until the recognition accuracy was at an unsatisfactory level with respect to a particular type of noise, where less than 80% of recognition accuracy can be considered to be unsatisfactory. For both factory noise and engine noise, the training data was contaminated with the SNRs of -4dB, -2dB, 0dB, 2dB, 4dB, 6dB and 8dB respectively, while the test data was contaminated with the SNRs of -5dB, -3dB, -1dB, 1dB, 3dB, 5dB and 7dB respectively.

C. Development of noise suppression filters using IPSO

Three commonly-used noise suppression filters, namely Boll filter [4], Wiener filter [31] and Ephraim-Malah filter [10] are considered, as they are simple and require small computational operations. These noise suppression filters remove noise from the original signal based on different approaches of noisy observation and estimates of noise spectral components. They are implemented as single-channel filters for signal enhancement as discussed in Section III.A. Their outputs are used as the input of the hybrid filter which is discussed in Section III.B.

i) **Boll filter** [4] namely $F_{bf}(..)$ is described by**:**

$$G_{bf}(\omega,\ell)=\max\left\{1-\frac{P_n(\omega,\ell)}{P_x(\omega,\ell)},\delta_{floor}\right\}, \tag{18}$$

where $G_{bf}(\omega,\ell)$ is the gain function of $F_{bf}(..)$, and $\delta_{floor}=0.1$ is used to avoid a vanishing gain function. $F_{bf}(..)$ works on the principle of subtracting the estimate of noise spectrum from the noisy signal.

ii) **Wiener filter** [31] namely $F_{wf}(..)$ is described by**:**

$$G_{wf}(\omega,\ell)=\frac{SNR_{priori}(\omega,\ell)}{SNR_{priori}(\omega,\ell)+1}, \tag{19}$$

where $G_{wf}(\omega,\ell)$ is the gain function of $F_{wf}(..)$;

$$SNR_{priori}(\omega,\ell)=\alpha'\cdot G_{wf}(\omega,\ell)\cdot SNR_{priori}(\omega,\ell-1)+(1-\alpha')\max\{SNR(\omega,\ell)-1,0\}; \tag{20}$$

$$SNR(\omega,\ell)=\frac{P_x(\omega,\ell)}{P_n(\omega,\ell)}; \tag{21}$$

and α' is the smoothing constant [4], which is needed to be optimized by the IPSO.

iii) **Ephraim-Malah filter** [10] namely $F_{\log}(..)$ is described by**:**

$$G_{\log}(\omega,\ell)=\frac{SNR_{priori}(\omega,\ell)}{SNR_{priori}(\omega,\ell)+1}\exp\left\{\frac{1}{2}\int_{\lambda(\omega,\ell)}^{\infty}\frac{e^{-\Gamma}}{\Gamma}d\Gamma\right\} \tag{22}$$

where $G_{\log}(\omega,\ell)$ is the gain function of $F_{\log}(..)$; $SNR_{priori}(\omega,\ell)$ is defined in (20), but $G_{wf}(\omega,\ell)$ in (19) is replaced by $G_{\log}(\omega,\ell)$; and the integral lower limit $\lambda(\omega, \ell)$ is defined as

$$\lambda(\omega,\ell)=\frac{SNR_{priori}(\omega,\ell)}{SNR_{priori}(\omega,\ell)+1}SNR(\omega,\ell). \quad (23)$$

iv) **Hybrid filter** namely $F_{NN}(.)$ discussed in Section III.B:

$F_{NN}(.)$ mixes the contaminated signal, x_j^i, and the outputs of the three single filters, $\hat{u}_{1,j}^i$, $\hat{u}_{2,j}^i$, and $\hat{u}_{3,j}^i$, which correspond to $F_{bf}(..)$, $F_{wf}(..)$ and $F_{\log}(..)$ respectively. The output of $F_{NN}(.)$ is given by:

$$\hat{u}_{NN,j}^i = F_{NN}\left(x_j^i,\hat{u}_{1,j}^i,\hat{u}_{2,j}^i,\hat{u}_{3,j}^i\right) \quad (24)$$

For all noise suppression filter designs, the smoothing parameters for $F_{bf}(..)$, $F_{wf}(..)$ and $F_{\log}(..)$ namely α and β discussed in Section III.A, were optimized by the IPSO. Apart from α and β, the smoothing parameter, α' existing in $F_{wf}(..)$ and $F_{\log}(..)$, was also optimized by the IPSO. For the hybrid filter, the neural network parameters of $F_{NN}(.)$ are also determined by the IPSO as discussed in Section VI. For the IPSO implementation, the parameter settings used are those recommended in [25]: the pre-defined number of generations = 100; the number of particles in the swarm = 100; inertia upper and lower weight factors, $w_{\max}$ = 0.9 and $w_{\min}$ = 0.5 respectively; acceleration constants ϕ_1 = 1 and ϕ_2 = 1; maximum velocity $v_{\max}$ = 0.2.

D. Evaluations of noise suppression filters

1) Training and testing performance

Table 1 shows the recognition accuracy amidst factory noise, where the recognition accuracies are obtained based on the training data for the cross validation. It shows the results obtained solely by the commercial microcontroller, and also those obtained by $F_{bf}(..)$, $F_{wf}(..)$, $F_{\log}(..)$ and $F_{NN}(.)$. It indicates that the commercial microcontroller can work satisfactorily with more than 75% recognition accuracy against the

SNR of 8dB, while its performance decreased to lower than 70% against 6dB. Less than 50% recognition accuracy is obtained only when the SNR is at 4dB. Much poorer recognition accuracies, which are lower than 20%, are obtained when the SNRs are lower than -2dB.

When the noise suppression filters were used, there was better speech recognition. For the single-channel filters, the recognition accuracies obtained by $F_{bf}(..)$ are better than those obtained by $F_{\log}(..)$, which are better than those obtained by $F_{wf}(..)$. The hybrid filter, $F_{NN}(.)$, is the best filter among the four, where $F_{NN}(.)$ is generally better than $F_{\log}(..)$ and $F_{wf}(..)$, and it is slightly better than $F_{bf}(..)$. This result clearly indicates that the single-channel filters developed by the IPSO can significantly enhance the accuracies of the commercial microcontroller. It also shows that $F_{NN}(.)$ developed by the IPSO is able to align the outcomes obtained by the best single-channel filters, as $F_{NN}(.)$ integrates the outcomes of all single-channel filters. Hence, of all the filters, $F_{NN}(.)$ is the best noise suppression filter for enhancing the recognition accuracies. As all the noise suppression filters were developed using the training data to maximize the recognition accuracy formulated in (4), this result shows that the enhanced signals produced by those noise suppression filters can fit well into the commercial microcontroller.

Table 1 Training recognition accuracies for factory noise

SNR	No enhancement	$F_{wf}(..)$	$F_{\log}(..)$	$F_{bf}(..)$	$F_{NN}(..)$
8	75.5	84.5	88.5	96.5	97.0
6	66.0	80.0	80.0	93.5	95.0
4	28.5	52.5	76.0	92.0	94.0
2	23.5	45.0	82.0	86.0	88.5
0	21.5	38.5	72.0	83.0	84.0
-2	17.5	29.5	60.0	64.0	65.5
-4	13.5	25.5	50.5	38.0	38.5
Mean accuracy	35.1	50.8	72.7	79.0	80.4
Relative improvement	0	0.45	1.07	1.25	1.29

Table 2 shows the recognition accuracies with respect to the test data which is independent of the training data. This result indicates poor speech recognitions were obtained by the commercial microcontroller with

no enhancement, where the recognition accuracy was below 30% when SNRs are less than 3dB. Significantly poor recognition accuracy was obtained at -5dB. When the single channel filter is used, improvement can be obtained. The recognition accuracies obtained by $F_{bf}(..)$ are better than those obtained by $F_{\log}(..)$, which are better than those obtained by $F_{wf}(..)$ for the test data. $F_{NN}(..)$ is the best filter among the four filters for the test data, as $F_{NN}(..)$ integrates the outcomes of all single-channel filters. As all those noise suppression filters were developed based on the test data, this result indicates that the commercial microcontroller can still work well under the low SNRs and the speech commands voiced by the users which have not been involved in the training. Hence, the noise suppression filters developed based on (4) can enhance the generalization capability of the commercial microcontroller.

Table 2 Test recognition accuracies for factory noise

SNR	No enhancement	$F_{wf}(..)$	$F_{\log}(..)$	$F_{bf}(..)$	$F_{NN}(..)$
7	72.0	78.0	90.0	96.0	96.0
5	62.0	72.0	88.0	92.0	94.0
3	24.0	52.0	84.0	90.0	92.0
1	22.0	42.0	78.0	75.0	76.0
-1	20.0	20.0	74.0	62.0	64.0
-3	16.0	18.0	56.0	52.0	54.0
-5	6.0	20.0	24.0	32.0	34.0
Mean accuracy	31.7	43.1	70.1	71.3	72.9
Relative improvement	0	0.36	1.21	1.25	1.30

Tables 3-4 indicate similar results in that the accuracy of the commercial microcontroller can be enhanced under the engine noise when the noise suppression filters were interfaced. Results further show that $F_{NN}(..)$ outperforms the three individual noise suppression filters, where $F_{bf}(..)$ is generally better than $F_{wf}(..)$ and $F_{\log}(..)$. These results demonstrate that $F_{NN}(..)$ intends to mix the advantages of the three individual noise suppression filters. Hence, better recognition accuracy can be achieved by $F_{NN}(..)$.

Table 3 Training recognition accuracies for engine noise

SNR	No enhancement	$F_{wf}(..)$	$F_{\log}(..)$	$F_{bf}(..)$	$F_{NN}(..)$
8	71.0	77.5	95.5	98.0	98.0
6	50.5	67.5	92.5	95.0	95.0
4	27.5	54.0	88.0	92.5	92.5
2	24.0	36.0	78.5	88.5	90.0
0	24.0	27.5	73.0	83.5	85.5
-2	23.5	26.0	60.5	63.5	65.0
-4	11.5	24.0	19.5	50.5	51.0
Mean accuracy	33.1	44.6	72.5	81.6	82.4
Relative improvement	0	0.35	1.19	1.47	1.49

Table 4 Test recognition accuracies for engine noise

SNR	No enhancement	$F_{wf}(..)$	$F_{\log}(..)$	$F_{bf}(..)$	$F_{NN}(..)$
7	67.0	71.5	94.0	96.0	96.0
5	51.5	60.5	90.0	94.0	94.0
3	26.0	52.0	86.0	88.0	90.0
1	24.0	28.0	78.0	86.0	88.0
-1	24.0	26.0	62.0	64.0	68.0
-3	20.0	24.0	54.0	54.0	54.0
-5	10.0	24.0	14.0	46.0	48.0
Mean accuracy	31.8	40.9	68.3	75.4	76.9
Relative improvement	0	0.29	1.15	1.37	1.41

2) Result discussion

The results shown in Tables 1 to 4 generally indicate that the classic filter $F_{bf}(..)$ is usually better than the modern filters $F_{wf}(..)$ and $F_{\log}(..)$, although an improved estimation approach of signal to noise ratio is used in $F_{wf}(..)$ and $F_{\log}(..)$. Equation (20) shows that $SNR_{priori}(\omega,\ell)$ is used as the estimation of signal to noise ratio of $F_{wf}(..)$ and $F_{\log}(..)$, where $SNR_{priori}(\omega,\ell)$ is a smoothed delayed version of $SNR(\omega,\ell)$. It is precisely this smoothing effect that reduces the effects of musical tones with the log spectral amplitude estimate gain function [5]. However, the delay introduced in the estimate may introduce reverberation effects especially during speech onset and offset periods. This may decrease the recognition accuracy.

Hence, these results demonstrate that using more advanced filters, $F_{wf}(..)$ and $F_{\log}(..)$, may not bring better accuracy for the commercial microcontroller than the classic filter $F_{bf}(..)$. It demonstrates that it is necessary to select an appropriate filter with respect to the accuracy of the commercial microcontroller rather than selecting those that can satisfy some acoustic citations which may not relate to speech recognition. Also, it is necessary to mix the outcomes of the single filters by $F_{NN}(..)$ in order to achieve better accuracy for the commercial microcontroller.

Figure 5 shows the simulation results for the second command '*Santa Claus is Coming to Town*' contaminated with factory noise with SNR=-2 dB, where the original speech commands, contaminated speech commands, enhanced speech commands from one of the single-channel filters, $F_{wf}(..)$, and enhanced speech commands of our proposed hybrid filter, $F_{NN}(..)$, are shown respectively. The figure shows that it is hard to distinguish the original speech commands from the contaminated speech commands, while the speech commands enhanced by the noise suppression filters, $F_{wf}(..)$ and $F_{NN}(..)$, can be recognized more easily. Hence, better recognition accuracies can be produced by the enhanced speech commands, and the effectiveness of the noise suppression filters can be demonstrated. Similar results can be found for engine noise with SNR=-2dB, which are shown in Figure 6. Hence, the effectiveness of the noise suppression filters can be further demonstrated. Based on those figures, it is hard to determine which enhanced signal is better than the other one with respect to recognition accuracy, by observing only the enhanced signals. Hence, it further demonstrates that optimizing the noise suppression filters with respect to the recognition accuracy shown in (4) is necessary rather than optimizing the limited degree of acoustic citations which is the common approach for developing noise suppression filters.

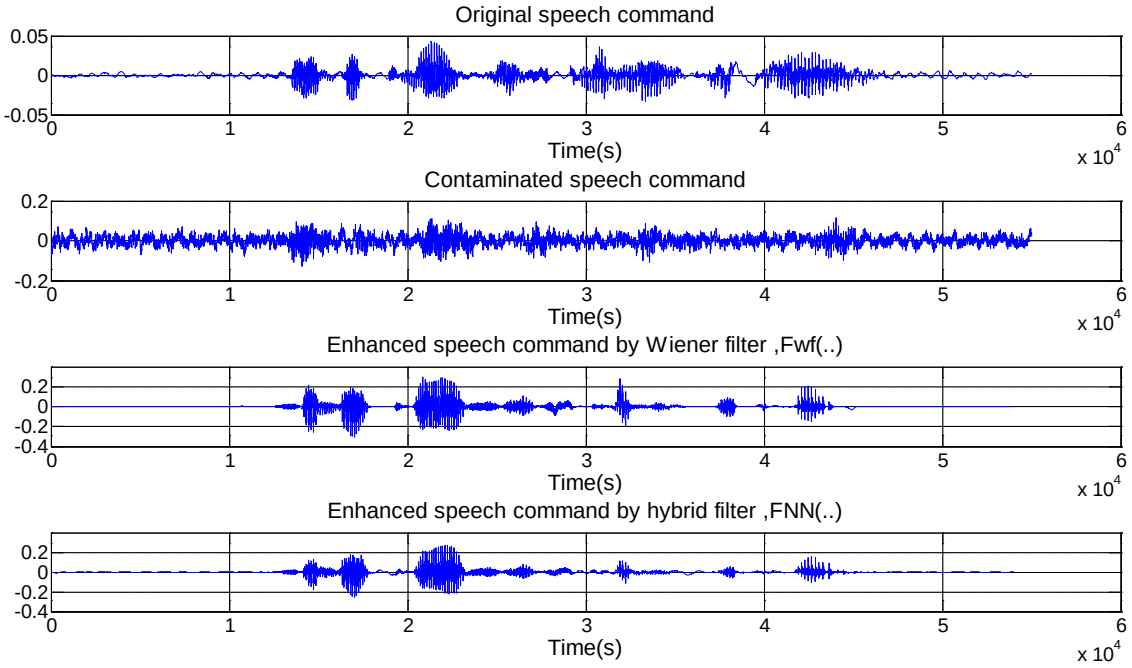

Fig. 5 Speech command contaminated with factory noise

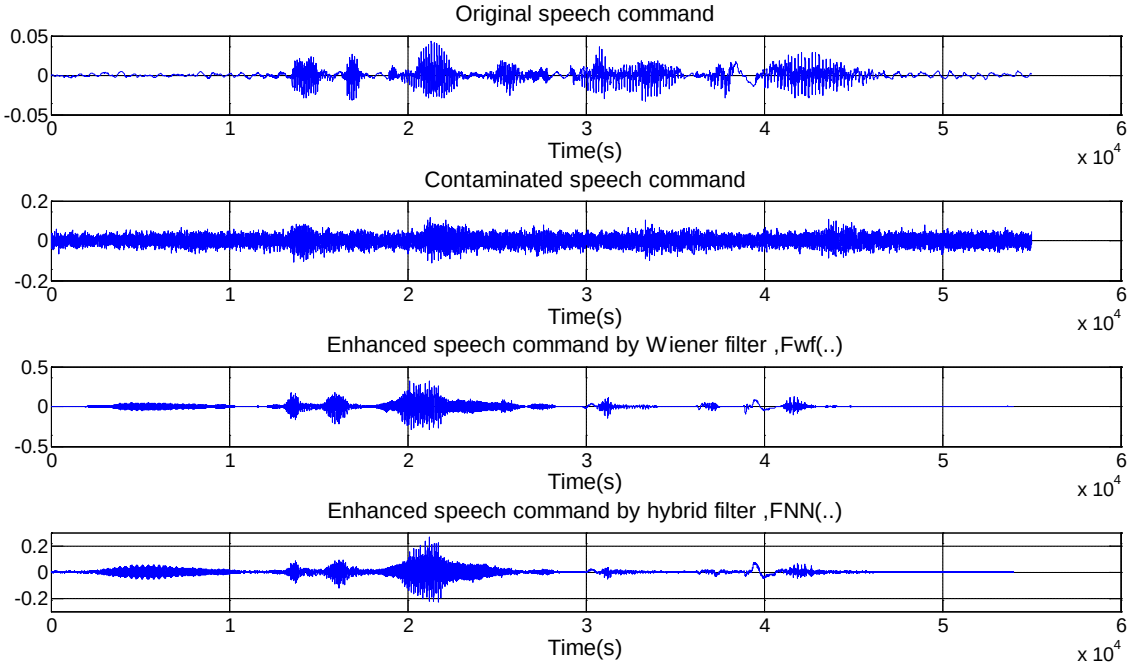

Fig. 6 Speech command contaminated with engine noise

VI Conclusion and Future works

In this paper, a speech enhancement strategy was proposed to enhance accuracies for speech recognition microcontrollers. It consists of three main operations:

(a) An optimization problem was formulated in order to optimize the in-built parameters of the noise suppression filters. It intends to maximize the accuracy of a speech recognition microcontroller under multi-conditions. It overcomes the limitation of the existing enhancement approaches where the parameters inside a microcontroller need to be tuned, which is impossible in the real world.

(b) The performance of the noise suppression filters are optimized based on the optimization problem. The outcomes of each noise suppression filter and the original speech are integrated using the neural network. Hence, various filter properties and speech intelligibilities can be included. This overcomes the limitation of using solely a noise suppression filter that optimizes only limited degrees of acoustic criteria which may not directly enhance the recognition accuracy of microcontrollers.

(c) An algorithm, namely IPSO, was developed to solve this optimization problem, which is discontinuous and discrete in nature. It overcomes the limitation of the commonly-used gradient descent methods which are likely to be trapped in local optima for noise suppression filter design.

The performance of the proposed approach was evaluated by implementing it on a commercial microcontroller, which simulated speech control in factory environments. Results show that the proposed approach significantly improved the performance of the commercial microcontroller in factory environments. Compared with the other commonly-used single-channel filters, it outperforms them in enhancing the recognition accuracy of the commercial microcontroller.

The proposed filtering approach can be extended by conducting the following research:

1) The optimization problem is formulated by solely addressing the accuracies of speech recognition microcontrollers, where the landscape of the optimization problem is non-differentiable and discrete. Hence, it may not be the most effective approach, as the global optimum of the optimization problem is hard to locate. In the future, we will reformulate the optimization problem by incorporating it with

specific acoustic criteria [12] which are related to the accuracies of speech recognition microcontrollers. By doing this, the global optimum in terms of recognition accuracies can be searched with respect to those acoustic criteria.

2) As we used only a simple neural network with one hidden layer in this research, we will investigate enhancing the effectiveness of the neural network by integrating the mechanism of hybrid approaches [37]. Also, we can integrate the mechanism of Quantum PSO in order to help search for the global optimum of the filter parameters [38].

3) The filtering approach can be implemented on a real-time embedded system, where the filter parameters can be adapted with respect to time-varying noise. By doing this, the filter parameter and the computational effort used on filtering can be evaluated and optimized through real-time implementation. The trade-off between adaptive time and recognition accuracy can be found. It is expected that the implementation of a filtering approach for enhancing recognition accuracy is a step forward.

Acknowledgment

The second and the third authors would like to thank the support of RGC Grant PolyU. (5301/12E). The third author would also like to thank the support of 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] M. Bekrani, A.W.H. Khong and M. Lotfizad, A linear neural network based approach to stereophonic acoustic echo cancellation, IEEE Transactions on audio, speech and language processing, vol. 19, no. 6, pp. 1743- 1753, 2011.

[3] A.P. Bhondekar, R. Vig, A. Gulati, M.L. Singla and P. Kapur, Performance evaluation of a novel iTongue for indian black tea discrimination, IEEE Sensors Journal, vol. 11, no. 12, pp. 3462-3468, 2011.

[4] S. Boll, Suppression of acoustic noise in speech using spectral subtraction, IEEE Transaction on Acoustic, Speech, and Signal Processing, vol. 27, pp. 113-120, 1979.

[5] O. Cappe, Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor, IEEE Transactions on Speech and Audio Process, vol. 2, pp. 345–349 1994.

[6] C.C. Chang, C.C Chen , U. Kurokawa and B.I. Choi, Accurate sensing of LED spectra via low-cost spectrum sensors, IEEE Sensors Journal, vol. 11, no. 11, pp. 2869-2877, 2011.

[7] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, A particleswarm-optimized fuzzy-neural network for voice-controlled robot systems, IEEE Transactions on Industrial Electronics, vol. 52, no. 6, pp. 1478–1489, Dec. 2005.

[8] I. Cohen and B. Berdugo, Noise estimation by minima controlled recursive averaging for robust speech enhancement, IEEE Signal Processing Letters, vol. 9, no. 1, pp. 12-15, 2002.

[9] R.C. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in Evolutionary Programming VII. New York: Springer-Verlag, Lecture Notes in Computer Science, vol. 1447, pp. 611-616, 1998.

[10] Y. Ephraim and D. Malah, Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, IEEE Transactions on Acoustic, Speech, Signal Processing, vol. 33, 443–445, 1985.

[11] J.S. Hu, C.C. Cheng, W.H. Liu and C.H. Yang, A robust adaptive speech enhancement system for vehicular applications, IEEE Transactions on Consumer Electronics, vol. 52, no. 3, pp. 1069-1077, 2006.

[12] Y. Hu and P. Loizou, Evaluation of objective quality measures for speech enhancement, IEEE Transactions on Audio, Speech, and Language Process, vol. 16, no. 1, pp. 229-238, 2008.

[13] C.F. Juang, C.T. Chiou and C.L. Lai, Hierarchical singleton-type recurrent neural fuzzy networks for noisy speech recognition, IEEE Transactions on Neural Networks, vol. 18, no. 3, pp. 833-843, 2007.

[14] J. Noyes and A. Starr, “Use of automatic speech recognition: current and potential applications,” *Comput. Control Eng. J.*, vol. 7, no. 5, pp. 203–208, 1996.

[15] W.G. Knecht, Nonlinear noise filtering and beamforming using the perceptron and its volterra approximation, IEEE Transactions on Speech and Audio Processing, vol. 2, no. 1, part 1, pp. 55-62, 1994.

[16] J. Kennedy and R. Eberhart, Particle swarm optimization, Proceedings of 30th IEEE International Conference on Decision and Control, vol. 4, 1995, pp. 1942-1948.

[17] Y. Gao and J.P. Haton, A hierarchical LPNN network for noise reduction and noise degraded speech recognition, Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing, vol. 2, pp. 19-22, 1994.

[18] H. Lam and F. Leung, Design and training for combinational neural logic systems, IEEE Transactions on Industrial Electronics, vol. 54, no. 1, pp. 612–619, Feb. 2007.

[19] J.L. Lim and A.V. Oppenheim, Enhancement and bandwith compression of noisy speech, Proceedings of the IEEE, vol. 67, pp. 1586-1604, 1979.

[20] R. Martin, Spectral subtraction based on minimum statistics, Proceedings of the 7th European Signal Processing Conference, pp. 1182–1185, 1994.

[21] R. Martin, Noise power spectral density estimation based on optimal smoothing and minimum statistics, IEEE Transactions on Speech and Audio Processing, vol. 9, no. 5, pp. 504-512, 2001.

[22] Y. Morsly, N. Aouf, M.S. Djouadi and M. Richardson, Particle swarm optimization inspired probability algorithm for optimal camera network placement, IEEE Sensors Journal, vol. 12, no. 5, pp. 1402-1412, 2012.

[23] F. Mustiere, M. Bolic and M. Bouchard, Speech enhancement based on nonlinear models using particle filters, IEEE Transactions on Neural Networks, vol. 20, no. 12, pp. 1923-1937 ,2009.

[24] M. Naeem, U. Pareek and D.C. Lee, Swarm intelligence for sensor selection problems, IEEE Sensors Journal, vol. 12, no. 8, pp. 2577-2585, 2012.

[25] M.O. Neill and A. Brabazon, Grammatical Swarm: The generation of programs by social programming, Natural Computing, vol. 5, pp. 443-462, 2006.

[26] K. Pan, M. F. Tasgetiren, and Y. C. Liang, A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, Computers and Operations Research, vol. 35, pp. 2839–2907, 2008.

[27] K.E. Parsopoulos and M.N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 211-,224 2004.

[28] T. Poggio and F. Girosi, Networks for approximation and learning, Proceedings of the IEEE, vol. 78, no.9, pp. 1481-1497, 1990.

[29] Aberdeen Group, Inc., “Warehouse automation—What’s really working for pallet, case and piece-pick operations”, Boston, MA, Tech. Rep., 2007.

[30] K. Thramboulidis, “Model-integrated mechatronics—Toward a new paradigm in the development of manufacturing systems”, *IEEE Trans.Ind. Informat.*, vol. 1, no. 1, pp. 54–61, Feb. 2005.

[31] P. Scalart and J. V. Filho, Speech enhancement based on a priori signal to noise estimation, IEEE International Conference on Acoustic, Speech, and Signal Processing, vol. 2, pp. 629–632, 1996.

[32] Sensory, INC, Company, Making electronic devices talk and hear, 2010.

[33] J.F. Wang, C.H. Wu, S.H. Chang and J.Y. Lee, A hierarchical neural network model based on a C/V segmentation algorithm for isolated mandarin speech recognition, IEEE Transactions on Signal Processing, vol. 39, no. 9, pp. 2141-2146, 1991.

[34] K.F.C. Yiu, K.Y. Chan, S.Y. Low, and S. Nordholm, A multi-filter system for speech enhancement under low signal-to-noise ratios, Journal of Industrial and Management Optimization, vol. 5, no. 3, pp. 671-682, 2009.

[35] S.X. Zhang, A. Ragni and M.J.F. Gales, Structured log linear models for noise robust speech recognition, IEEE Signal Processing Letters, vol. 17, no. 11, pp. 945-748, 2010.

[36] M. Zeltzer, 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.

[37] M. Negoita, D. Neagu, V. Palade, Computational Intelligence: Engineering of Hybrid Systems, Springer Verlag. 2005.

[38] J. Sun, W. Fang, X. Wu, V. Palade and X. Wenbo, Quantum−Behaved Particle Swarm Optimization: Analysis of the Individual Particle's Behavior and Parameter Selection, Evolutionary Computation. Vol. 20. No. 3. 2012.