Noname manuscript No.
(will be inserted by the editor)

A parallel beamforming system with real-time implementation

K.F.C. Yiu

Abstract For voice control applications, it is common to employ a microphone array to enhance received signals via beamforming techniques. In designing beamformers, different criteria will lead to different signal performance. It is known that speech recognition accuracy relies heavily on the trade-off between signal distortion and noise reduction. In this paper, we propose a novel beamformer structure which can give a continuous profile in signal distortion and noise reduction. The proposed structure combines two existing optimal beamformers to form the final filter. Moreover, since both optimal beamforming filters can be executed in parallel, a method is proposed to implement the noise reduction algorithm in the frequency domain. By studying the accuracy and efficiency of different modules, a hybrid fixed-floating point arithmetic is proposed within an FPGA hardware architecture to form an embedded system for industrial applications.

1 Introduction

The Internet of Things is impacting on our daily life. A variety of sensors are distributed in our environments and are also embedded in many electrical and electronic equipment, which can be deployed for data collection. Many researchers have explored sensor fusion techniques to detect and recognize activities [1–4, 6]. Smartphone sensors are particularly popular [5, 7] for collecting useful data. When microphone sensors are applied, acoustic signals can be collected which can induce the development of voice input systems. Indeed, these kind of systems have been widely applied in industry and everyday life, including factory automation [8], robot control [9], voice telematics systems in

K.F.C. Yiu
Department of Applied Mathematics,
The Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong, PR China
Tel.: +852-34008981
E-mail: cedric.yiu@polyu.edu.hk

cars [10], and many other speech recognition devices including the iPhone siri system. These advances play a part in modern cyber-physical systems with multimedia applications [11–13]. For such applications, it is crucial to control noise and to enhance the received signal. An essential technique is to employ a microphone array so that beamforming techniques can be applied to filter noise and enhance the speech signal.

Beamforming techniques exploit fundamental properties about spatial and/or temporal distribution of both speech and noise sources in order to enhance perception [15,16]. An example is the delay and sum beamformer where received microphone signals are aligned in time to extract signals from the direction of interest [17]. An alternative to the delay and sum beamformer is adaptive beamforming, where noise signals from all directions are suppressed continuously and adaptively, whilst the direction of interest is maintained at around the same power level [18,19]. In situations where calibration signals can be collected, certain optimal beamformer designs can be obtained as described in [25,26]. More comprehensive review of beamforming techniques can be found in [20,21].

When it comes to employ filtered signals for voice control, one needs to concern about accuracy in the speech recognition and not much about the actual speech quality. In fact, it was found that the trade-off between the level of signal distortion and the level of noise suppression is the determining factor in enhancing speech recognition accuracy for voice control devices [22–24]. In most current beamformer designs, this bi-criteria requirement has not been taken into account.

In this paper, the first contribution is to develop a beamforming algorithm which can adjust for the level of signal distortion and noise suppression. We explore the most commonly used optimal beamformer designs including least-squares technique (LS) [27] and signal-to-noise ratio (SNR) [28]. It is known that least-squares technique tends to concentrate on distortion control with deficiency in noise suppression [29]. Similarly, using signal-to-noise ratio, distortion is usually significant, although high level of noise suppression can be achieved. Here, we propose a novel parallel beamformer structure which combines performance of both LS and SNR filters. We show that a continuous speech quality profile can be constructed.

The second contribution of the paper is to design and develop a real-time implementation for the proposed parallel beamforming system. In this way, if both filters can be adaptive to the changing environment and give real-time response. Since optimization algorithms for both filters are independent from each other and they share a common structure, they can be implemented efficiently if subband processing is employed. The structure of a subband processor consists of a multichannel analysis filter-bank and a set of adaptive filters, each adapting on the subband signals. A synthesis filter-bank will gather subband signals and re-create a time domain output signal. Adaptive algorithms for the changing noise are executed continuously. For the design of the resulting embedded system, since we need to adapt two filters at the same time, in

order to achieve real-time performance, it is advantageous to implement it on a machine which allows massive parallelism.

The third contribution of the paper is the design and build a FPGA hardware architecture to implement the proposed parallel beamformers. In general, microprocessor is not fast enough and ASIC is too inflexible when the filter coefficients are adaptively changing. FPGA is an excellent alternative for this kind of implementation. In the literature, it has been reported the implementation of a time-delay sonar beamformer on reconfigured devices [30]. The beamformer achieved six times speed up over dedicated DSP systems. Another beamformer implementation involves delta-sigma modulation, and the beamformer is applied to medical ultrasonic application [31]. There are also implementation for antenna signals [32] or for audio applications [33]. However, these studies do not consider subband processing and have not considered parallel filter structure.

Indeed, the implementation on FPGA is not straightforward when we need to maintain accuracy and achieve sufficient computational speedup. If the architecture contains only fixed point arithmetic, we found that there are certain steps producing excessive rounding errors. As a result, a novel hybrid fixed-floating point arithmetic is proposed here where fixed point arithmetics are applied mostly except for certain part of the calculations in which floating point arithmetics are carried out due to rounding errors. Based on a careful calibration on the required numerical operations, we show that required floating point operations remain to be a very small proportion relative to fixed point operations while maintaining accuracy in the final results. In addition, optimization based on bitwidth analysis to explore suitable bitwidth of the system is carried out. The optimized integer and fraction size using fixed point arithmetic can reduce the overall circuit size by up to 80% when compared with a direct realization of the software onto an FPGA platform. The performance criteria based on distortion and noise reduction are employed to assess the accuracy in the optimized system. In achieving computational speedup, we identify common computational intensive operations for both filters and design dedicated hardware accelerators to perform the most time consuming part of the algorithm. The performance can be boosted further by optimizing on the resource to pack multiple instances of the accelerators in a single large FPGA. The acceleration is evaluated on a Virtex-4 platform, showing that the FPGA-based implementation at 184MHz can achieve real-time performance by processing a maximum of 27804 samples per second.

2 Formulation

Assume there are M elements in the microphone array. In general, signals received by microphone elements can be represented by

$$x_i(n) = s_i(n) + v_i(n), \qquad i = 1, 2, \cdots, M, \tag{1}$$

where $s_i(n)$ and $v_i(n)$ is the source signal and the noise signal, respectively. Note that noise signal could include a sum of fixed point noise sources together with a mixture of coherent and incoherent noise sources. The speech signal is located near a uniform linear array of M microphones. Each microphone has an FIR filter behind (Fig. 1). The output of the beamformer is given by

$$y(n) = \sum_{i=1}^{M} \sum_{j=0}^{L-1} w_i(j) x_i(n-j) \tag{2}$$

where $L-1$ is the order of the FIR filters and $w_i(j)$, $j = 0, 1, \cdots, L-1$, are the FIR filter taps for channel number i. Note that n denotes a continuous stream of samples for signals and a block of samples for processing is taken to be N. The signals, $x_i(n)$, are digitally sampled microphone observations and the beamformer output signal is denoted by $y(n)$.

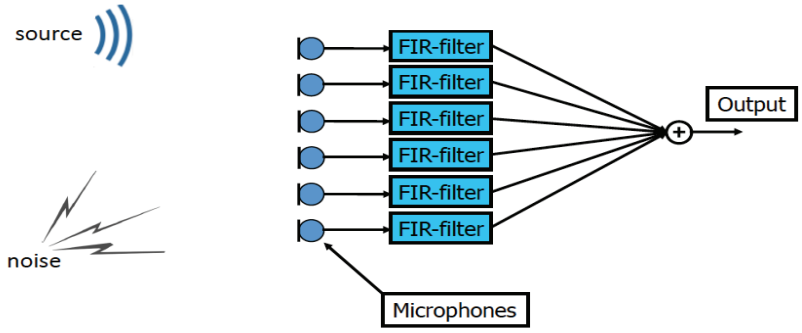

Fig. 1 The beamformer structure

By using a subband beamforming scheme, each microphone signal is filtered through a subband filter. A digital filter with the same impulse response is used for all channels so that all spatial characteristics are kept. This means that a large filtering problem is divided into a number of smaller problems and the computational burden will become substantially lower.

The signal model can equivalently be described in the frequency domain and the filtering operations will in this case become multiplications with number K complex frequency domain representation weights, $w_i^{(k)}$. For a specific frequency, k, the output is given by

$$y^{(k)}(n) = \sum_{i=1}^{M} w_i^{(k)} x_i^{(k)}(n) \tag{3}$$

where the signals $x_i^{(k)}(n)$ and $y^{(k)}(n)$ are narrow band signals containing essentially components with frequency k. The observed microphone signals are given in the same way as

$$x_i^{(k)}(n) = s_i^{(k)}(n) + v_i^{(k)}(n). \tag{4}$$

The objectives in subband levels are typically the following [34]:

$$\left.\begin{array}{l} \max_{\mathbf{w}^{(k)}} \left[y^{(k)}(n) \cong s^{(k)}(n)\right] \\ \min_{\mathbf{w}^{(k)}} \left|\left| \sum_{i=1}^{M} w_i^{(k)} v_i^{(k)}(n) \right|\right| \end{array}\right\} \forall k \tag{5}$$

where $\left[y^{(k)}(n) \cong s^{(k)}(n)\right]$ refers to some measure of resemblance. There are different ways to achieve these objectives, depending on the metric used for the objective optimization. In the following sections, we deal with the specifications of some objective functions and their corresponding solutions .

3 First filter with least-squares criterion

If a least-squares criterion is used to measure the mismatch for each subband, a least squares solution can be solved based on N samples. A calibration sequence gathered in a quiet environment is used as the reference source signal. This calibration signal (denoted by $s_r(n)$) will represent the temporal and spatial information about the source. The least-squares objective is

$$\min_{\mathbf{w}^{(k)}} \left\{ \sum_{n=0}^{N-1} \left[|y^{(k)}(n) - s_r^{(k)}(n)|^2 \right] \right\}, \tag{6}$$

which can be solved as

$$\mathbf{w}_{opt}^{(k)}(N) = \left[\hat{\mathbf{R}}_{ss}^{(k)}(N) + \hat{\mathbf{R}}_{xx}^{(k)}(N)\right]^{-1} \hat{\mathbf{r}}_s^{(k)}(N) \tag{7}$$

where the real frequency $f = F_s k/K$, with F_s the sampling frequency and K the total number of subbands, and where the array weight vector, $\mathbf{w}_{opt}^{(k)}$ for the subband k is defined as

$$\mathbf{w}_{opt}^{(k)} = [w_1^{(k)} \quad w_2^{(k)} \quad \dots \quad w_M^{(k)}]^T. \tag{8}$$

The source correlation estimates can be pre-calculated in the calibration phase as

$$\hat{\mathbf{R}}_{ss}^{(k)}(N) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{s}^{(k)}(n) \mathbf{s}^{(k)\,H}(n) \tag{9}$$

$$\hat{\mathbf{r}}_s^{(k)}(N) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{s}^{(k)}(n) s_r^{(k)\,*}(n) \tag{10}$$

where the superscript * denotes conjugation while the superscript ^{H} denotes Hermitian transpose, and

$$\mathbf{s}^{(k)}[n] = [s_1^{(k)}[n], s_2^{(k)}[n], \dots, s_M^{(k)}[n]]^T$$

are microphone observations when the calibration source signal is active alone, while the observed data correlation matrix estimate $\hat{\mathbf{R}}_{xx}^{(k)}(N)$ can be calculated recursively from the received data.

3.1 Adaptive algorithm

For each subband, the correlation matrix, $\hat{\mathbf{R}}_{ss}^{(k)}$, and the source signal cross correlation vector, $\hat{\mathbf{r}}_s^{(k)}$, are estimated in the initialization phase using calibration signals. The observed data correlation matrix estimate can be decomposed into the form:

$$\left(\hat{\mathbf{R}}_{ss}^{(k)}(N) + \hat{\mathbf{R}}_{xx}^{(k)}(N)\right) = \mathbf{Q}^{(k)^H}\mathbf{\Gamma}^{(k)}\mathbf{Q}^{(k)}$$

where the eigenvectors are

$$\mathbf{Q}^{(k)} = [\mathbf{q}_1^{(k)}, \quad \mathbf{q}_2^{(k)}, \quad \dots \quad \mathbf{q}_I^{(k)}]$$

and the eigenvalues are

$$\mathbf{\Gamma}^{(k)} = diag([\gamma_1^{(k)}, \quad \gamma_2^{(k)}, \quad \dots \quad \gamma_I^{(k)}]).$$

For each subband signal, $k = 0, 1, \cdots, K-1$, where for each subband the corresponding normalized frequency is $f = 2\pi k/K$ and each sample instant n, the observed microphone signals in subband number k are denoted $x_i^{(k)}(n)$, $i = 1, 2, \cdots, M$. The number of available samples in the acquisition phase is N. The algorithm can be stated as follows [34]:

With multiple sources being active simultaneously, for $n = 1, 2, \cdots$, compute

$$\begin{aligned}
\mathbf{x}_n^{(k)} &= [x_1^{(k)}(n), \quad x_2^{(k)}(n), \quad \dots \quad x_I^{(k)}(n)]^T \\
\mathbf{P}^{(k)} &= \lambda^{-1}\mathbf{P}_{n-1}^{(k)} - \frac{\lambda^{-2}\mathbf{P}_{n-1}^{(k)}\mathbf{x}_n^{(k)}\mathbf{x}_n^{(k)^H}\mathbf{P}_{n-1}^{(k)}}{1+\lambda^{-1}\mathbf{x}_n^{(k)^H}\mathbf{P}_{n-1}^{(k)}\mathbf{x}_n^{(k)}} \\
\mathbf{P}_n^{(k)} &= \mathbf{P}^{(k)} - \frac{\gamma_p(1-\lambda)\mathbf{P}^{(k)}\mathbf{q}_p^{(k)}\mathbf{q}_p^{(k)^H}\mathbf{P}^{(k)}}{1+\gamma_p(1-\lambda)\mathbf{q}_p^{(k)^H}\mathbf{P}^{(k)}\mathbf{q}_p^{(k)}}
\end{aligned} \tag{11}$$

where index $p = (n \, mod \, M) + 1$,

$$\mathbf{w}_n^{(k)} = \alpha\mathbf{w}_{n-1}^{(k)} + (1-\alpha)\mathbf{P}_n^{(k)}\hat{\mathbf{r}}_s^{(k)}.$$

The output from each subband is

$$y^{(k)}(n) = \mathbf{w}_n^{(k)^H}\mathbf{x}_n^{(k)}.$$

In the operation phase, microphone signals are decomposed continuously into frequency subbands. The inversion of the total correlation matrix $(\hat{\mathbf{R}}_{ss}^{(k)}(N) + \hat{\mathbf{R}}_{xx}^{(k)}(N))$ and its subsequent update can be carried out sequentially by adding a rank one correction to the matrix at each sample instant [35]. The filter weights in each subband are then updated via a first order smoothing model. The output from each subband signal is finally reconstructed using a reconstruction filter-bank to yield an estimate of the sound source of interest. The algorithm is adapting continuously once the correlation estimates are placed into memory.

4 Second filter with signal-to-noise criterion

By measuring the output signal-to-noise power ratio (SNR), it becomes maximizing a ratio between two quadratic forms of positive definite matrices as

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \left\{ \frac{\mathbf{w}^H \mathbf{R_{ss}} \mathbf{w}}{\mathbf{w}^H \mathbf{R_{xx}} \mathbf{w}} \right\} \tag{12}$$

is referred to as the generalized eigenvector problem, [36]. The solution can be found by solving the following relation

$$\mathbf{R_{xx}}^{-H/2} \mathbf{R_{ss}} \mathbf{R_{xx}}^{-1/2} \mathbf{v}_{opt} = \lambda \mathbf{v}_{opt} \tag{13}$$

and the final optimal weights are given by the inverse of the linear variable transformation

$$\mathbf{w}_{opt} = \mathbf{R_{xx}}^{-1/2} \mathbf{v}_{opt}. \tag{14}$$

The formulation of the optimal signal-to-noise beamformer can be carried out in each subband. The optimal weights can then be sought in the subband level. For frequency subband k, the quadratic ratio between the output signal power, and the output noise power is

$$\mathbf{w}_{opt}^{(k)} = \arg\max_{\mathbf{w}^{(k)}} \left\{ \frac{\mathbf{w}^{(k)H} \mathbf{R}_{ss}^{(k)} \mathbf{w}^{(k)}}{\mathbf{w}^{(k)H} \mathbf{R}_{xx}^{(k)} \mathbf{w}^{(k)}} \right\} \tag{15}$$

where the superscript ^{H} denotes hermitian transpose and

$$\mathbf{R}_{ss}^{(k)} = E\{\mathbf{s}^{(k)}(n) \mathbf{s}^{(k)}(n)^H\} \tag{16}$$

in which

$$\mathbf{s}^{(k)}(n) = \left[s_1^{(k)}(n) \quad s_2^{(k)}(n) \quad \dots \quad s_M^{(k)}(n) \right]^T. \tag{17}$$

Similarly, the observed data correlation matrix $\mathbf{R}_{xx}^{(k)}$ can be defined. We can then employed Eq. (14) to find the optimal for each subband.

A well-known iterative method for finding the eigenvector $v_{opt}^{(k)}$ corresponding to the largest eigenvalue of the problem (13) for each frequency band f is the power method, which makes use of the recursion

$$v_{opt}^{(k)}(p+1) = \frac{\mathbf{R_{xx}}^{-H} \mathbf{R_{ss}} v_{opt}^{(k)}(p)}{\| \mathbf{R_{xx}}^{-H} \mathbf{R_{ss}} v_{opt}^{(k)}(p) \|}, \tag{18}$$

with an arbitrary initial vector $v_{opt}^{(k)}(0)$. The eigenvector components will decay in the order of pth power of the associated eigenvalues. Thus, as long as the initial vector $v_{opt}^{(k)}(0)$ has a component in the direction of the dominant eigenvector, convergence to the desired solution will be guaranteed .

In finding the inverse of $\mathbf{R_{xx}}$, a fast and accurate algorithm is required. If a matrix is inverted using Gaussian elimination, a lot of if-branches are required

to implement the pivoting, which is very expensive in hardware implementation. Another way is to use Cramer's rule, which does not need pivoting and requires only one division (calculation of the reciprocal of the determinant). The steps are as follows:

1. Calculate all cofactors of the matrix and form a cofactor matrix.

2. Calculate the determinant of the given matrix.

3. Multiply the matrix obtained in step 2 by the reciprocal of the determinant.

Comparing with Gaussian elimination, Cramer's Rule should not be used for large matrices due to a large number of multiplication operations in the process of calculating the determinant and the cofactors. However, for inversion of matrices with small dimensions, Cramer's rule yields a performance gain. Since subband matrices are generally rather small, we adopt the use of Cramer's rule here and illustrate the performance gain over the use of Gaussian elimination.

5 FPGA Embedded System Design

Let the optimal weights for the kth subband be $\mathbf{w}_{LS}^{(k)}$ and $\mathbf{w}_{SNR}^{(k)}$. Each filter weight has its unique property in noise suppression and signal distortion. Due to the linearity of the filtering process, we attempt to form a linear combination of these two filter weights which will adjust the distortion and noise suppression continuously

$$\mathbf{w}_{\theta}^{(k)} = \theta * \mathbf{w}_{LS}^{(k)} + (1-\theta) * \mathbf{w}_{SNR}^{(k)}. \tag{19}$$

This new filter weight is then used in (3) for filtering subband signals. In order to illustrate the complete algorithm, Figure 2 depicts the parallel structure of the proposed beamformer and the basic signal flowchart.

In mapping the algorithms to hardware, we explore the synergy between the two parallel filters to achieve computational efficiency. Apart from the essential FFT/IFFT hardware module, we analyze the computational intensive steps (11) and (18) and extract the fundamental numerical operations. Since operations are essentially independent for different subbands, it is nature to attempt executing more subbands in parallel [37]. Furthermore, when the number of subband increases, the size of linear system we need to solve for each subband is reducing. Consequently, we build a functional module for complex matrix inversion at the subband level so that several subbands can be processed in parallel quickly. As for computational accuracy, there are a few numerical operations that might incur numerical errors larger than expected under fixed point arithmetic. Therefore we proposed a hybrid fixed-floating point scheme to resolve this problem without completely resorting to the use of floating point arithmetic. More details are described in the following.

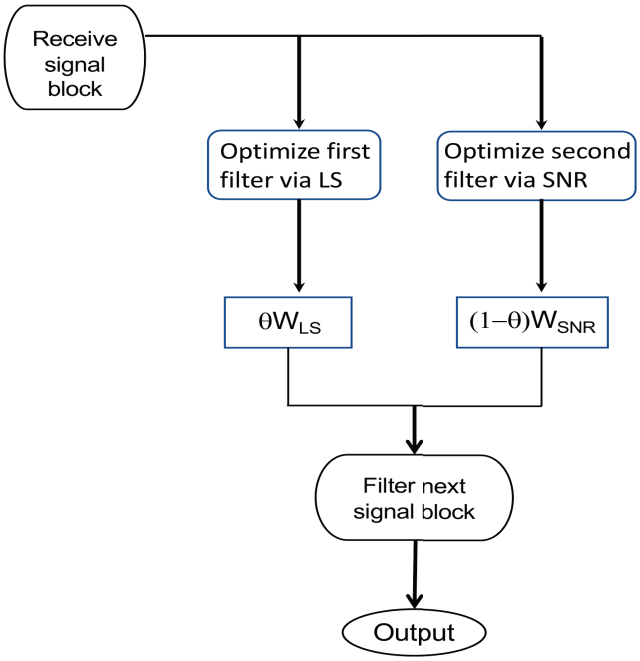

Fig. 2 A parallel filter system

5.1 Hardware Architecture

The optimized beamformer algorithms described in Section 3.1 and 4 are feasible to be implemented into reconfigurable hardware. In order to reduce the size of the circuit and increase the performance, several techniques have been applied which exploits the flexibility of reconfigurable hardware.

In the time domain, the main operations of the beamformers are the computation of the equations (7) and (11). The computation time is greatly reduced by implementing the actual filtering in the frequency domain. It involves the signal transformations from time domain to frequency domain and vice versa. The main steps including the following calculations:

1. Transform the input signals to their frequency domain representations via FFT;
2. Filter the subband signals by the subband impulse response estimates;
3. Synthesize the impulse response estimates back to the time domain via IFFT (inverse FFT).

The algorithms are analyzed to determine an optimized way to translate them to the reconfigurable hardware. The translation guarantee computational efficiency by exploiting the independence of subband processing as well as matrix and vector operations, which can be optimized at several levels:

– Loop level parallelism, consecutive loop iterations can be executed in parallel;
– Task level parallelism, that entire procedures inside the program can be executed in parallel;
– Data parallelism

The algorithms involves control components and computation components. To determine suitable components to be implemented by hardware, we first identify computationally intensive kernels in the algorithms by profiling. When profiling is carried out, time consuming operations can be determined and will be implemented by hardware. The profiling results of the main operations are shown in Table 1, indicating that the FFT/IFFT and two UPDATE operations are the best candidates. They occupy 80% of the CPU time. These kernels are mapped on dedicated processing engines of the system, optimized to operate on large amounts of data, while the remaining parts of the code is implemented by software running on the PowerPC processor. In our experiment, we find that FPGA device embedded with processors is a suitable platform for this system. For instance, we choose Xilinx Virtex-4 FX FPGA device is the target platform. The Auxiliary Processor Unit (APU) interface in the device simplifies the integration of hardware accelerators and co-processors. One can easily offload computational intensive tasks from the CPU to the hardware accelerators.

Function	**%Overall Time**
LS UPDATE	31.8%
24-bit FFT/IFFT (32pt)	28.8%
SNR UPDATE	19.4%
OTHERS	20.0%

Table 1 Profiling Results of the Main Operations

5.2 Fixed-floating point arithmetic

Different from traditional software development, designing a system on an FPGA platform involves an estimation of the length of bitwidth. This has a significant effect on the circuit size and the accuracy of calculations. Typically fixed point arithmetic is employed together with saturation arithmetic [38] to handle the overflow case, due to its inherent speed advantage over floating point arithmetic. Therefore, a set of fixed point library is developed which allows bitwidth analysis to identify suitable hardware configuration which can retend signal quality with less area consumption. We run an experiment using a 32-bit fixed point representation to vary the integer size to learn a suitable integer size in this system. The integer size is finally determined as 12-bit,

and suitable scaling have been employed in the hardware implementation to minimize the impact of overflow.

In the adaptation of the SNR beamformer, it is needed to compute the eigenvectors and eigenvalues of the estimated source covariance matrix. This is a generalized complex non-Hermitian matrix eigenvalue problem. Due to computational complexity, it is hard to compute eigenvalues and corresponding eigenvectors in such 32-bit fixed point arithmetic. As we can see from Table 2, the maximum and mean absolute errors are rather large when calculating eigenvalues and eigenvectors by the QR method. Consequently, a hybrid fixed-floating point data representation is adopted here, where floating point arithmetic is employed for computing the complex matrix inversion. In calculating the actual eigenvectors using the Power method, fixed point arithmetic can be applied for the other steps.

Complex matrix inversion is the most time-consuming process in the Power method. As discussed earlier, the use of pivoting in Gaussian elimination involves many branches, which slow down the overall calculations. Since our matrices are of rather small dimensions, it's suitable to implement the inversions using Cramer's Rule. Table 3 shows that using Cramer's Rule is 3.7 times faster than Gaussian elimination while performing 300000 times matrix inversion in the beamformer program. Since it is the only operation which requires floating point arithmetic, simulation shows that floating point computation involvement is still relatively small in the whole iterative process and does not impact the overall performance.

Type of LA	Cases	Max Err	Mean Abs Err
Matrix Multiply	20	0.000156	0.0000002
FFT	20	0.000235	0.0000004
QR Method	20	33.0899467	0.2255622
Power Method	20	0.0000769	0.0000010
Matrix Inverse	20	42.2291222	0.0659830

Table 2 Error Estimation for Linear Algebra

Type of Matrix Inversion	Dimension	Cycles	Time (s)
Gaussian elimination	4x4	300000	23.8922
Cramer's Rule	4x4	300000	6.4573

Table 3 Performance Comparison for Matrix Inversion

6 Results

In the simulation, the total number of subbands is chosen as $M = 128$ with a decimation factor of $D = 64$ and the size of a subband matrix equal to 4.

Four microphones are employed in this simulation. Another setup has been introduced for bitwidth analysis.

The fixed point and fixed-floating point structures are studied in Section 5.2, also by varying the integer size in order to determine the suitable integer size in this system. Speech distortion and noise suppression performance measures [29,34] are applied in order to quantified the difference in performance for different integer sizes. The normalized distortion quantity, $\mathcal{D}$, is introduced as

$$\mathcal{D} = \frac{1}{2\pi} \int_{-\pi}^{\pi} |C_d \hat{P}_{y_s}(w) - \hat{P}_{x_s}(w)| dw \tag{20}$$

where $w = 2\pi f$, and f is normalized frequency. The constant, C_d, is defined as

$$C_d = \frac{\int_{-\pi}^{\pi} \hat{P}_{x_s}(w) dw}{\int_{-\pi}^{\pi} \hat{P}_{y_s}(w) dw} \tag{21}$$

where $\hat{P}_{x_s}(w)$ is a spectral power estimate of a single sensor observation and $\hat{P}_{y_s}(w)$ is the spectral power estimate of the beamformer output, when the source signal is active alone. The constant C_d normalizes the mean output spectral power to that of the single sensor spectral power. The single sensor observation is chosen as the reference microphone observation. In order to measure the noise suppression the normalized noise suppression quantity, S_N, is introduced as

$$S_N = C_s \frac{\int_{-\pi}^{\pi} \hat{P}_{y_N}(w) dw}{\int_{-\pi}^{\pi} \hat{P}_{x_N}(w) dw} \tag{22}$$

The results are summarized in Table 4. The appropriate integer size is 12 and further increase does not improve the results significantly. Table 5 represents the implementation results of the proposed hardware design on both Xilinx XC4VSX55-12 and Xilinx XC2VP30-7-FF896 FPGA devices.

For each θ, the pair (S_N, $\mathcal{D}$) can be calculated. In order to understand the trade-off between signal distortion and noise suppression, a range of θ is applied and a solution set can be constructed by employing Eq. (19). The result is depicted in Figure 3. From the Figure, it is observed that different speech quality can be achieved in a continuous manner by varying the value of θ.

An estimation has been made to evaluate the performance of the FPGA-based LS and SNR beamformer that is equipped with one FFT/IFFT and one filter update hardware accelerator. Assuming one block of data contains 64 samples under a 16kHz sampling rate, the number of clock cycle required for processing the block of data in the frequency domain is measured as 823600. Therefore, given that the period of one clock cycle is $1/(184MHz) = 5.43ns$ on a Virtex4 FPGA, the FPGA-based beamformer can perform one step of speech enhancement in 0.0045s, or equivalently 14311 samples per second.

Multiple instances of LS and SNR beamformers can be packed in a single large FPGA to boost the performance further. This is very useful here because

Bitwidth	LS (Fixed point)	
	Speech Distortion [dB]	Noise Suppression [dB]
I=12, F=20	-31.2943	4.5154
I=12, F=18	-31.2958	4.5149
I=16,F=16	-31.3020	4.5127
I=14,F=14	-31.2972	4.5020
I=12,F=12	-31.2197	4.4431
I=10,F=10	-28.9885	4.2677
I=8,F=8	overflow	overflow
Bitwidth	SNR (Fixed-Floating point)	
	Speech Distortion [dB]	Noise Suppression [dB]
I=12, F=20	-24.2352	29.3429
I=12, F=18	-24.2352	29.4348
I=16,F=16	-24.2354	29.6685
I=14,F=14	-24.2362	28.8212
I=12,F=12	-24.2408	23.7736
I=10,F=10	-24.2382	10.4378
I=8,F=8	overflow	overflow
Bitwidth	SNR (Fixed point)	
	Speech Distortion [dB]	Noise Suppression [dB]
I=12, F=20	-28.1280	15.3718
I=12, F=18	-28.1284	15.3726
I=16,F=16	-28.1296	15.3741
I=14,F=14	-28.1364	15.3244
I=12,F=12	-28.1770	14.8667
I=10,F=10	-28.3267	12.1612
I=8,F=8	overflow	overflow

Table 4 Performance for different integer sizes

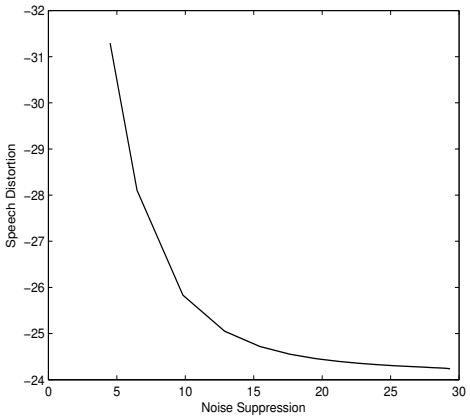

Fig. 3 Trade-off between noise and distortion levels.

FPGA device	XC4VSX55-12	XC2VP30-7
Slices used	55882 (42%)	27783(71%)
Block RAM used	18 (8%)	14 (11%)
Frequency (MHz)	184.8	141.7

Table 5 Implementation results of the LS and SNR beamformer. Note that the hardware accelerator only contains one FFT/IFFT and one UPDATE modules.

of the design has multiple channels. In this way, the resource can be fully utilized on FPGA. On the other hand, with an increase in logic utilization, the speedup amount will slow down, due to the increased routing congestion and delay. In view of this, a medium size FPGA is used here to implement the hardware accelerator and can accommodate different combinations of FFT/IFFT and UPDATE within the hardware accelerator, which provides flexible solutions between speed and area trade-off.

Table 6 summarizes the implementation results when adding more instances of the filter in an XC4VSX55-12-FF1148 FPGA chip and shows how the number of instances affects the speedup. A XC4VSX55-12-FF1148 chip can accommodate at most two FFT/IFFT and UPDATE hardware accelerators, so the sampling rate will be 27804 samples per second. It achieves real-time performance.

To summarize some of the pros and cons of the proposed signal enhancement architecture and its implementation, the advantages include flexibility in achieving varying signal quality, which is important for voice control applications. Moreover, the proposed implementation is fast enough to process enough samples per seconds for most practical applications. The disadvantages are that it requires a hardware architecture that supports fast parallel processing, and also one needs to optimize on θ in order to design for dedicated applications.

Samples/s	Number of Instances		Slices Used	DSP Used
	FFT/IFFT	Filter update		
14311	1	1	42%	12%
20035	1	2	64%	19%
26169	1	3	87%	26%
19627	2	1	62%	16%
27804	2	2	84%	23%
20444	3	1	77%	21%
20853	4	1	92%	24%

Table 6 Slices and DSPs used, maximum frequency and sampling rate when implementing multiple instance on an XC4VSX55-12-FF1148 FPGA device.

7 Conclusions

In this paper, a novel beamforming structure together with an embedded system has been proposed, which involves designing a set of parallel beamforming filter weights. We have demonstrated the final beamformer is equipped with varying speech quality. By exploring the parallel nature of the beamformers and subband operations, a hardware implementation of the algorithm on an FPGA virtex-4 system has been described using the proposed fixed-float arithmetic. The algorithm have been simulated in hardware and results have shown that real-time performance can be achieved when an FPGA-based hardware accelerator performs the critical parts of the algorithm. The resulting embedded system will find applications in modern multimedia systems. As a future extension, it is possible to speed up the implementation further by designing the beamformers under the power-of-two space [39]. Also, it would certainly be of interest to optimize further on the configuration of the microphone array [40], and to include beamforming filters designed via model-based approach, such as [41]

Acknowledgements

This paper is supported by RGC Grant PolyU. 152200/14E and PolyU Grant 4-ZZGS and G-YBVQ. The author would like to thank Dr. Chun-Hok Ho, Mr. Yao Lu, Mr. Xiaoxiang Shi for carrying out the implementation on FPGA. The author would also like to thank the support of Prof Sven Nordholm and Dr Nedelko Grbic.

References

1. J. Wu, Y. Feng and P. Sun, "Sensor fusion for recognition of activities of daily living," it Sensors, 18:4029 (2018).
2. Y. Liu, L. Nie, L. Liu and D.S.Rosenblum, "From action to activity: Sensor-based activity recognition Author links open overlay panel," *Neurocomputing*, 181:108-115 (2016).
3. Y. Liu, L. Nie, L. Liu and D.S.Rosenblum, "Action2Activity: Recognizing Complex Activities from Sensor Data," *arXiv*, 1611.01872 (2016).
4. Y. Liu, L. Nie, L. Liu and D.S.Rosenblum, "Fortune Teller: Predicting Your Career Path," *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16).* February 12-17, Phoenix, Arizona USA (2016).
5. N. Roy, A. Misra and D. Cook, "Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments," *Journal of Ambient Intelligence and Humanized Computing*, 7(1):1-19 (2016).
6. S. Ranasinghe, F.A. Machot and H.C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *International Journal of Distributed Sensor Networks*, 12(8):1-22 (2016).
7. M. Shoaib, S. Bosch, O.D. Incel, H. Scholten and P.J.M. Havinga, "A survey of online activity recognition using mobile phones," it Sensors, 15:2059-2085 (2015).
8. K. Thramboulidis, "Model-integrated mechatronics - toward a new paradigm in the development of manufacturing systems," *IEEE Transactions on Industrial Informatics*, 1(1):54-61 (2005).

9. J.N. Pires, "Robot-by-voice: experiments on commanding an industrial robot using the human voice", *An International Journal of Industrial Robot*, 32(6):1159-1320 (2005).
10. 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*, 55(3):1496-1500 (2009).
11. B. Andersson and S. Prabh, "Localizing Objects in Large-Scale Cyber-Physical Systems", in *The 4th International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*. June 11-14, Santorini Island, Greece (2008).
12. G. Healy and A. F. Smeaton, "Spatially augmented audio delivery: Applications of spatial sound awareness in sensorequipped indoor environments", in *Tenth International Conference on Mobile Data Management (MDM 2009)*. May 18-20, Taipei, Taiwan (2009).
13. M. Duchon, C. Schindhelm and C. Niedermeier, "Cyber Physical Multimedia Systems: A Pervasive Virtual Audio Community", in *The Third International Conferences on Advances in Multimedia (MMEDIA 2011)*. April 17-22, Budapest, Hungary (2011).
14. NXP, "Philips new LifeVibes family of acoustical solutions improves the quality of mobile communications," *http://www.nxp.com/news/content/file_943.html* (2003).
15. B.D. Van Veen and K.M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, 5(2):4-24 (1988).
16. D. Johnson and D. Dudgeon, *Array Signal Processing: Concepts and Applications*. Englewood Cliffs, New Jersey: Prentice-Hall (1993).
17. W. Kellermann, "A self-steering digital microphone array," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 91)*. May 14-17, Toronto, Ontario, Canada., p. 3581-3584 (1991).
18. O. Hoshuyama, A. Sugiyama and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE T. Signal Processing*, 47(10):2677-2684 (1999).
19. S. Gannot, D. Burshtein and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Processing*, 49(8):1614-1626 (2001).
20. M. Brandstein and D. Ward (Eds.), "Microphone arrays : techniques and applications," *Berlin ; New York : Springer* (2001).
21. J. Li and P. Stoica (Eds.), "Robust adaptive beamforming," *Hoboken, N.J. : John Wiley* (2006).
22. 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*, 5(3):671-682 (2009).
23. K.Y. Chan, K.F.C. Yiu, T.S. Dillon and S.H. Ling, "Enhancement of speech recognitions for control automation using an intelligent particle swarm optimization", *IEEE Transactions on Industrial Informatics*, 8:869-879 (2012).
24. K.F.C. Yiu, K.Y. Chan, N. Grbic and S. Nordholm, "A hybrid design of beamformers for voice control devices", *Pacific Journal of Optimization*, 8:533-544 (2012).
25. S. Nordholm, I. Claesson and M. Dahl, "Adaptive microphone array employing calibration signals: An analytical evaluation," *IEEE Trans. Speech Audio Processing*, 7:241-252 (1999).
26. S. Nordholm, I. Claesson and N. Grbić, "Optimal and adaptive microphone arrays for speech input in automobiles," in *Microphone Arrays*, M. Brandstein and D. Ward, Eds. Springer Verlag (2001).
27. N. Grbić and S. Nordholm, "Soft constrained subband beamforming for hands-free speech enhancement," *ICASSP-02*, I-885-888 (2002).
28. M. Dahl and I. Claesson, "Acoustic noise and echo canceling with microphone array," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1518–1526 (1999).
29. K. F. C. Yiu, N. Grbic, K. L. Teo and S. Nordholm, "A new design method for broadband microphone arrays for speech input in automobiles," *IEEE Signal Processing Leters*, vol. 9, no. 7, pp. 222–224 (2002).
30. P. Graham and B. Nelson, "FPGA-Based Sonar Processing," *Proc of Field Programmable Gate Arrays*, pp. 201–208 (1998).

31. B. G. Tomov and J. A. Jensen, "A new architecture for a single-chip multi-channel beamformer basedon a standard FPGA," *IEEE Ultrasonics Symposium*, 2:1529-1533 (2001).
32. M.D. van de Burgwal, K.C. Rovers, K.C.H. Blom, A.B.J. Kokkeler and G.J.M. Smit, "Adaptive beamforming using the reconfigurable MONTIUM TP," The 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD), 301 - 308 (2010).
33. D. Theodoropoulos, G. Gaydadjiev and G. Kuzmanov, "A reconfigurable beamformer for audio applications," . IEEE 7th Symposium on Application Specific Processors (SASP '09), 80 - 87 (2009).
34. N. Grbić, "Optimal and Adaptive Subband Beamforming," *PhD. Thesis, Blekinge Institute of Technology*, Ronneby, Sweden, 2001.
35. J. E. Hudson,"Adaptive Array Principles,", *Peter Peregrinus Ltd.* (1991).
36. G. H. Golub and F. Van Loan, "Matrix Computations," *John Hopkins University Press*, London (1989).
37. K.F.C. Yiu, Y. Lu, J.Q. Huo, S. Nordholm, C.H. Ho and W. Luk, "Reconfigurable FPGA-based robust frequency-domain echo canceller with applications to voice control device," *Digital Signal Processing*, 22(2): 376-390 (2012).
38. G.A. Constantinides, P.Y.K. Cheung and W. Luk, "Synthesis of saturation arithmetic architectures," *ACM Transactions on Design Automation of Electronic Systems*, 8(3):334–354 (2003).
39. Z.G. Feng, K.F.C. Yiu, K.L. Teo and S.E. Nordholm, "Design of broadband beamformers with low implementation complexity", *Eurasip J. Advanced Signal Processing*, 2013:62 (2012).
40. Z.G. Feng, K.F.C. Yiu and S.E. Nordholm, "Placement design of microphone arrays in near-field broadband beamformers", *IEEE Transactions on Signal processing*, 60:1195-1204 (2012).
41. Z.G. Feng, K.F.C. Yiu and S.E. Nordholm, "A two-stage method for the design of near-field broadband beamformers", *IEEE Transactions on Signal processing*, 59(8):3647-3656 (2011).