FPGA Implementation of Spectral Subtraction for Automotive Speech Recognition

Jim Whittington, Kapeel Deo, Tristan Kleinschmidt, Michael Mason

Abstract— **The use of speech recognition in noisy automotive environments requires the application of speech enhancement algorithms to improve recognition performance. Deploying these enhancement techniques necessitates significant engineering to ensure algorithms are realisable in electronic hardware. This paper describes advances in porting the popular spectral subtraction algorithm to a Spartan-3A DSP field-programmable gate array (FPGA) device suitable for integration in automotive environments. Resource analysis shows the final design uses only 13% of the total available general logic resources making it suitable for integration with other in-car devices on a single FPGA. Speech recognition experiments have been used to verify the effectiveness of the FPGA implementation for in-car speech recognition in comparison with an equivalent floating-point implementation.**

I. Introduction

A key challenge of deploying automatic speech recognition (ASR) in vehicular environments is the requirement to perform well in the presence of high levels of noise. Most current speech recognition systems are trained for use in controlled scenarios (e.g. office environments or telephone-based systems) – as a result these recognisers fail to produce satisfactory recognition performance under more adverse conditions such as in automotive environments.

Speech enhancement is one of the most common methods for making ASR systems more robust. Such techniques aim to reduce the levels of noise present in the speech signals, allowing clean speech models to be utilised in the recognition stage. This is a popular approach as little-or-no prior knowledge of the operating environment is required for improvements in recognition accuracy.

Complete speech enhancement systems for use in automotive environments have been proposed in [1], [2]. Cheng *et. al.* [1] implement an adaptive beamformer with most of the processing performed on a PC, while Yu *et. al.* [2] propose the software implementation of a dual microphone least mean square (LMS) algorithm running on an Analog Devices Blackfin Digital Signal Processor (DSP). Neither system provides a low cost, single chip, single microphone solution, which is of greatest interest to automotive manufacturers.

Spectral subtraction is an appropriate enhancement method for in-car speech recognition as it requires simple processing and installation of only a single microphone. This technique was originally proposed by Steven Boll in 1979 [3] and since this time there have been numerous reviews of the algorithm. There are only limited examples where spectral subtraction has been specifically applied to noisy signals recorded in an automotive environment. Lockwood *et. al.* [4] and Wahab *et. al.* [5] have both concluded that spectral subtraction techniques can be successfully used to enhance speech signals in the presence of automotive environment noise, however no hardware implementations were proposed.

The majority of current automotive electronics are powered by low-cost embedded processors that perform multiple tasks including CAN network communications and HMI. Currently only a small amount of automotive electronics are based on field programmable gate arrays (FPGAs), primarily due to their higher single-unit cost compared to an embedded processor. The market is changing since the cost deferential is insignificant considering the much higher performance of an FPGA. This performance is coupled with the fact that even modest-sized FPGAs may contain multiple instantiations of embedded processors as well as other specialised hardware elements such as a speech processing and enhancement system. This eliminates the need for multiple devices, simplifying overall design and cost. Recognising this market opportunity, Xilinx, a leading FPGA vendor, has developed the Xilinx Automotive (XA) product family specifically for automotive applications [6], [7].

The suitability of FPGAs for the implementation of speech enhancement processing has been demonstrated. For example, Yiu *et. al.* [8] have implemented a multi-microphone sub-band adaptive beamformer for speech enhancement in a high-end Virtex-4 FPGA. This system showed "very similar" enhancement performance to an equivalent floating-point implementation and a very large improvement in processing performance. Halupka *et. al.* [9] implemented a dual-microphone, phase-based time-frequency masking speech enhancement system on an Altera Stratix EP1S40 FPGA. The FPGA implementation produced similar real-time speech enhancement quality to equivalent floating-point software. In both these cases no automotive test data was included in the work, nor was the suitability for use in automotive environments discussed.

This paper focuses on an FPGA implementation of spectral subtraction optimised for in-car speech recognition. Section II provides an overview of the spectral subtraction algorithm and its optimisation for in-car speech recognition systems. Section III describes the advances of a fixed-point FPGA implementation of the algorithm since our previously published

Jim Whittington and Kapeel Deo are with the Department of Electronic Engineering, LaTrobe University, Melbourne, Australia. Tristan Kleinschmidt and Michael Mason are with the Speech and Audio Research Laboratory, Queensland University of Technology, Brisbane, Australia. (email: {j.whittington, kapeel.deo}@latrobe.edu.au, {t.kleinschmidt, m.mason}@qut.edu.au).

This work was supported in part by the Australian Cooperative Research Centre for Advanced Automotive Technology (AutoCRC).

work [10]. In-car speech databases used for verification and evaluation are presented in Section IV. Experimental results verifying the FPGA design are presented in Section V. Discussion of these results and possible improvements to the FPGA design are provided in Section VI.

II. Spectral Subtraction

A. Algorithm Description

In a noisy environment, speech $s(n)$ is assumed to be corrupted by additive background noise $d(n)$ to produce corrupted speech $y(n)$ as follows:

$$y(n) = s(n) + d(n) \tag{1}$$

where $y(n)$ is the signal captured via microphone. This signal is pre-emphasised, and then broken into frames using a Hamming window. The Discrete Fourier Transform (DFT) is taken for each frame i to produce the short-time frequency-domain representation:

$$Y(i,\omega) = S(i,\omega) + D(i,\omega) \tag{2}$$

In spectral subtraction algorithms, a scaled estimate of the magnitude (or power) spectra of the noise signal $\hat{D}(i,\omega)$ is subtracted from the corresponding spectra of the noisy signal $Y(i,\omega)$ to give an estimate of the clean speech $\hat{S}(i,\omega)$:

$$|\hat{S}(i,\omega)|^\gamma = |Y(i,\omega)|^\gamma - \alpha(i,\omega)|\hat{D}(i,\omega)|^\gamma \tag{3}$$

where γ is the exponent applied to the spectra, with $\gamma = 1$ used for magnitude spectral subtraction or $\gamma = 2$ for power spectral subtraction [11]. The noise estimate can be determined through time-recursive or moving averages, minimum statistics or histogram techniques [12]. Since the noise signal is estimated, frequency-dependent subtraction factors, $\alpha(i,\omega)$, are included to compensate for underestimating or overestimating the instantaneous noise spectrum.

Should the subtraction in (3) give negative values (i.e. the scaled noise estimate is greater than the instantaneous signal) a flooring factor is introduced. This leads to the following formulation of spectral subtraction:

$$|\hat{S}_t(i,\omega)|^\gamma = |Y(i,\omega)|^\gamma - \alpha(i,\omega)|\hat{D}(i,\omega)|^\gamma$$

$$|\hat{S}(i,\omega)|^\gamma = \begin{cases} |\hat{S}_t(i,\omega)|^\gamma & |\hat{S}_t(i,\omega)|^\gamma > \beta|Z(i,\omega)|^\gamma \\ \beta|Z(i,\omega)|^\gamma & \text{otherwise} \end{cases} \tag{4}$$

where $|Z(i,\omega)|$ is either the instantaneous noisy speech signal magnitude or the noise magnitude estimate, β is the noise floor factor, and $0 < \beta \ll 1$ [11]. Common values for this parameter range between 0.005 and 0.1 [11], [13].

The enhanced magnitude spectrum is recombined with the unaltered noisy speech phase spectrum. Each frame is transformed to the time domain using an inverse DFT, and adjacent frames are overlapped and added to resynthesise an enhanced time-domain signal. The enhanced signal can then be used for playback or as input to further speech processing such as automatic speech recognition.

More details on the speech processing elements used in this algorithm can be found in [10].

B. Optimising for In-Car Speech Recognition

The subtraction process indicated by (4) requires a lot of real-time multiplications since the frequency-dependent subtraction factors – and potentially the noise floor – are calculated on a frame-by-frame basis. We seek to simplify this equation through the following steps:

1) We assume the noise estimate $|\hat{D}(i,\omega)|^\gamma$ is sufficiently accurate, and therefore over/undersubtraction factors are not required (i.e. set $\alpha(i,\omega) = 1$ for all frames i and frequencies ω).
2) We assume the initial N frames of each recording contain noise only, and we average these frames to produce a noise estimate which remains constant for the remainder of the recording (i.e. the noise estimate is calculated prior to signal enhancement and can be represented as $|\hat{D}(\omega)|^\gamma$).
3) We utilise the constant noise estimate for calculation of the noise floor (i.e. the noise floor is also constant for the entire utterance).

Following these simplifications to (4), the spectral subtraction equation used in the following FPGA implementation is:

$$|\hat{S}_t(i,\omega)|^\gamma = |Y(i,\omega)|^\gamma - |\hat{D}(\omega)|^\gamma$$

$$|\hat{S}(i,\omega)|^\gamma = \begin{cases} |\hat{S}_t(i,\omega)|^\gamma & |\hat{S}_t(i,\omega)|^\gamma > \beta|\hat{D}(\omega)|^\gamma \\ \beta|\hat{D}(\omega)|^\gamma & \text{otherwise} \end{cases} \tag{5}$$

Equation (5) leaves only two parameters (γ and β) to be further optimised for FPGA implementation of the spectral subtraction algorithm.

C. Selection of Enhancement Parameters

Common values for γ and β are those noted in Section II-A. The values of γ are typically used for their conceptual meanings as opposed to recognition performance whilst β is often chosen to optimise SNR given a particular value of γ. Previously it was established [14] that in-car speech recognition performance differs greatly with various combinations of γ and β; therefore these values must be chosen carefully.

In order to reduce processing requirements of the FPGA implementation detailed in Section III and [10], magnitude spectral subtraction (γ = 1) was chosen. This avoids the need for resource-intensive square and square root operations in the FPGA. Further, previous experiments in [14] showed that performing magnitude spectral subtraction provided better speech recognition accuracy than power spectral subtraction (if the β values were optimised for both values of γ).

Preliminary experiments using floating-point software were performed to determine the optimal value of β to use in the FPGA implementation. Using the first 5 experimental folds from the evaluation protocol for the AVICAR database

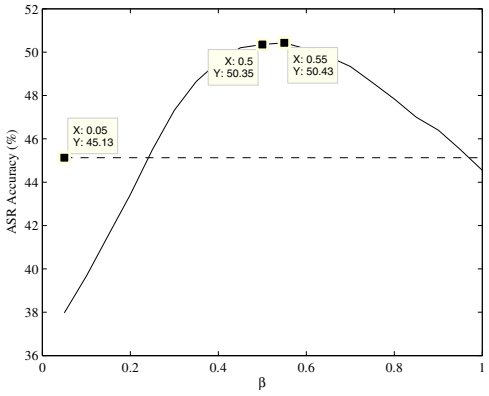

Fig. 1. *The effect of the noise floor scaling factor, β, on ASR accuracy averaged over a range of automotive noise conditions.*

[15], [16], values of β were varied in linear increments through the range [0, 1] with γ = 1. The averaged results across all the noise conditions in the AVICAR database are shown in Fig. 1. It can be seen that a wide range of β values ensure improvements over a system with no enhancement, and that whilst the maximum recognition accuracy can be obtained by setting $\beta = 0.55$, the performance is only marginally better than at $\beta = 0.5$ (approximately 0.1%). It should also be noted that individual noise conditions exhibit very similar characteristics to Fig. 1 allowing a constant value to be applied to all in-car noise scenarios. We choose $\beta = 0.5$ for the FPGA implementation as this value can be easily and accurately represented in fixed-point notation.

III. FPGA Implementation

For our implementation, Xilinx devices and development tools were chosen since they offer a clear pathway to an automotive standard commercialisable platform. Cost is a key factor to eventual widespread adoption in the automotive field, while the spectral subtraction algorithm relies on considerable DSP power. Thus, target devices must be cost-effective while still providing relatively high-performance DSP. With well over one million system gates, plus memory and XtremeDSP™ slices, Xilinx XA Spartan-3A DSP FPGAs fit this requirement well [7], [17], [18]. In previous work [10] we realised the spectral subtraction algorithm on a higher-end Xilinx Virtex-4 SX FPGA device. This work details the porting of the previous design onto a lower-cost Spartan-3A DSP 1800A device. This device is a general production equivalent to its Xilinx Automotive cousin. Successful implementation in this device will demonstrate capability for implementation on the other.

A. Design Process

Moving from an algorithmic description to a quality, cost-effective FPGA solution is anything but trivial as was outlined in the previous report on this work [10]. In summary, this work consisted of the following broad steps:

1) Development of a MATLAB version of the spectral subtraction algorithm using high-precision, complex floating-point arithmetic.
2) Conversion to a fixed-point (data and operations) implementation in MATLAB, mirroring the major blocks expected in the FPGA implementation.
3) Comprehensive testing of the fixed-point MATLAB design against the floating-point version, block-by-block and at complete system level.
4) Implementation of the fixed-point design as Xilinx System Generator™ (XSG) models.
5) Comprehensive testing of each major block of the XSG design against its fixed-point MATLAB equivalent, and testing of the complete XSG model against both the fixed-point and floating-point MATLAB versions.
6) From the completed XSG model a hardware description language was generated, synthesised using Xilinx ISE 9.2 tools, and implemented on a high-end Xilinx Virtex-4 SX FPGA.
7) Following a check of the FPGA resource usage of the design, the XSG model was analysed block-by-block to identify resource inefficiencies and refined to use more appropriate resources.
8) The performance of the Virtex-4 realisation was checked against the XSG and floating-point models by comparing output waveforms for a common input.

Once the Virtex-4 implementation was validated as appropriately equivalent to the spectral subtraction floating-point algorithm, the design was synthesised for implementation on the Spartan-3A DSP device. The outputs of the Virtex-4 and Spartan-3A DSP FPGAs were tested against each other sample-by-sample for a variety of input waveforms including basic ramps, modulated chirps, and various speech samples. In all cases the outputs of the two FPGA designs were identical, demonstrating the equivalence of the two hardware implementations, and that the spectral subtraction algorithm can be implemented in an automotive-rated FPGA.

B. Hardware Implementation of Enhancement Algorithm

A block diagram of the spectral subtraction algorithm is provided in Fig. 2. Input signals consist of 16-bit speech waveforms sampled at 16 kHz. For this specific implementation, a frame size of 512 samples was used with 50% overlap.

The pre-emphasis filter is a simple design consisting of a delay, constant multiplication and sum. Framing and windowing are achieved using a buffer, predefined Hamming window, multiplication, and appropriate control logic.

For implementation of the DFT, a forward and inverse FFT block is used. This block provides both real and imaginary data outputs. As the FFT process is not required continuously (only when a full frame of 512 samples is available), the same block performs the IFFT after spectral subtraction. To generate frequency-domain magnitude and phase data from the FFT block output, a cordic arctan block is used.

At this point the algorithm calls for the magnitude data to be raised to the power γ. This is potentially a very complex

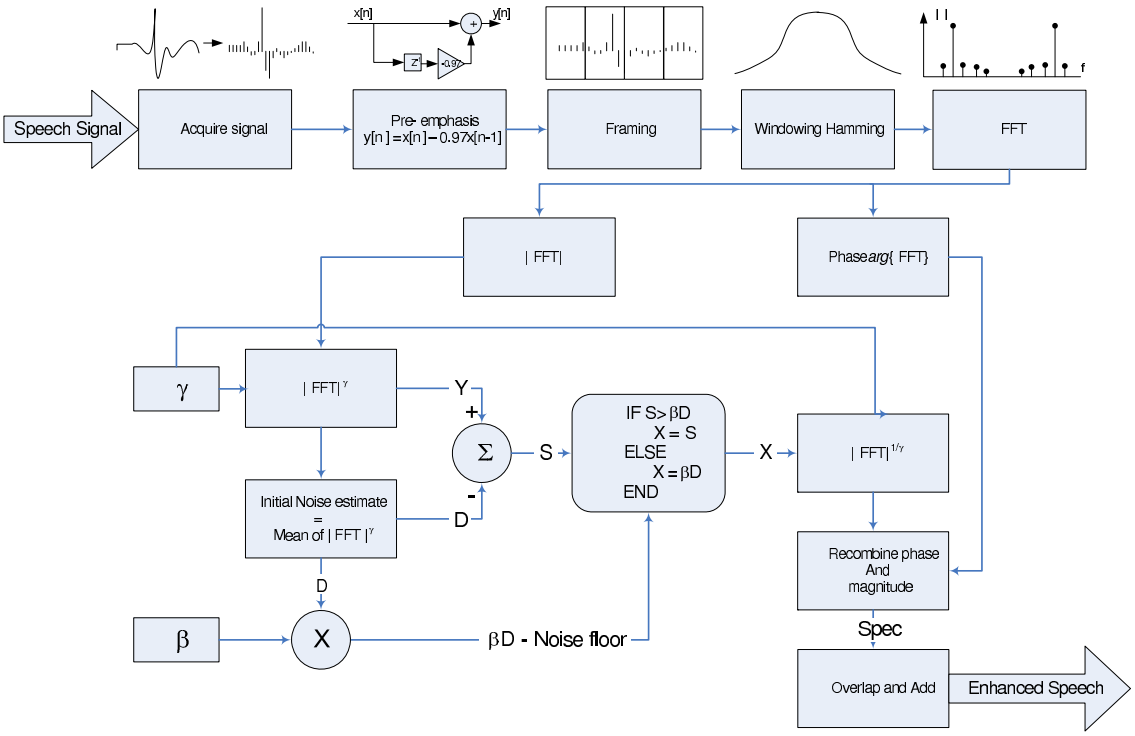

Fig. 2. *Block diagram of hardware implementation of spectral subtraction algorithm.*

hardware operation, so as outlined in Section II-B, $\gamma = 1$ is used as it greatly simplifies the design yet still provides suitable speech recognition performance.

The initial noise magnitude estimate is calculated from the first eight frames using a circular buffer, and an addition block. Each incoming frequency-bin data word is added to the previously accumulated value for that frequency. To obtain an average, the final sum must be divided by the number of frames used in the calculation. In this case using eight frames reduces this step to a simple three-bit shift.

The essence of the spectral subtraction technique occurs through subtracting the stored noise magnitude estimate $|\hat{D}(\omega)|$ from the subsequent frequency magnitude for each frame $|Y(i,\omega)|$ in the speech recording. The resulting frame S is compared with a scaled version of the average noise magnitude by the factor β which is known as the noise floor, $\beta|\hat{D}(\omega)|$. Each element of the resultant frame S is retained if it is greater than that of the noise floor, otherwise it is replaced by the noise floor value. If a noise estimate is not available (i.e. the current frame is one of the first 8 frames of the signal), the incoming data frame is ignored and all frequency elements are set to zero. As discussed previously, $\beta = 0.5$ is used; being a multiplication factor, this can be very simply implemented using a hardware shift operation as wiring between two registers. The algorithm then raises enhanced frames X to the inverse power of γ, potentially complex hardware which is avoided by using $\gamma = 1$.

To produce a time-domain frame, the new magnitude and previously retained phase frames are combined and converted to real and imaginary cartesian coordinates by a cordic sin-cos block and two multipliers, then input to the IFFT block (the FFT/IFFT block discussed earlier).

Finally, the resulting time-domain frames are overlapped and added, a reverse of the initial framing process, to produce the final reconstructed speech signal.

C. Optimisation of Hardware Design

After proving the Xilinx Virtex-4 and Spartan-3A DSP FPGA implementations produced identical results, the quality of the Spartan-3A DSP implementation was tested by running speech recognition experiments using the AVICAR database [15]. Table III (in the column 'Initial') shows test results for this design under different driving conditions. In most cases, the recognition accuracy of the FPGA design matches the floating-point model closely except for the idle condition where the accuracy is significantly below the floating-point enhancement as well as the case without enhancement. This indicated that the FPGA design was a good initial implementation of the spectral subtraction algorithm for most driving conditions, but not all.

To further optimise the FPGA implementation, each sub-block of the initial design was analysed based on the total number of bits used and the actual number of bits required. From this, an optimal number of bits was realised for each block being sufficient for processing all possible types of speech input with minimum quantization error and no arithmetic overflow. A comparison of the bit-widths in the initial and optimised designs is provided in Table I (bit-widths are represented as $X.Y$ where X is the full width and Y is the width of the fractional part).

TABLE I

BIT-WIDTH COMPARISON OF THE INITIAL AND OPTIMISED FPGA DESIGNS OF THE SPECTRAL SUBTRACTION ALGORITHM.

Process	Initial	Optimised
Pre-emphasis	16.15	17.15
Framing & Windowing	16.15	18.15
FFT/IFFT	20.19	24.23
Spectral Subtraction	26.15	28.23
Reconstruction	16.15	18.15

Optimisation started with the pre-emphasis block. Due to the nature of this operation, the output data can be almost double the magnitude of the input, and so the data width was increased accordingly by one integer bit. The FFT/IFFT block was increased to its maximum allowed value of 24-bits, set by the Xilinx IP core block. The motivation for this was to reduce the loss of data resolution in the IFFT input data which would normally be about 27-bits wide prior to truncation. The width of the magnitude and phase extraction, noise and noise floor calculation, enforcement of the noise floor and polar to Cartesian blocks were all increased in parallel to better suit the FFT output. These blocks are referred to as the "spectral subtraction" blocks in Table I. The output of the polar-to-Cartesian block is truncated to 24.23 to match the input of the FFT/IFFT block. The framing stage (with Hamming window application) and the reconstruction stage (Hamming window reapplication and overlap-add blocks) were both increased in size by 2-bits, however the output of the overlap-add block is still truncated to a 16.15 representation to match the implementation interface.

IV. EVALUATION DATA

In order to verify the effectiveness of the FPGA implementation, speech signals from two in-car speech databases were used. These databases were the AVICAR database [15] and an Australian In-Car Speech Database collected during this work. These databases are briefly outlined in the following sections.

1) AVICAR Database: The AVICAR database used in these experiments consists of 55 native American English speakers represented by 28 female and 27 male speakers. Each speaker recorded 10 phone numbers (i.e. digit strings) per noise condition in each recording session. The five noise conditions include idle as well as driving at 35 mph and 55 mph with the windows up and down. Utterances were recorded using an array of 7 microphones placed on the sun-visor directly in front of the speaker.

More details on the AVICAR database can be found in [15]. The evaluation protocol used in these experiments is outlined in [16].

2) Australian In-Car Speech Database: The newly collected database was collected using 50 speakers represented by 30 male and 20 female speakers. Each speaker was required to speak English as their first language, and have lived in Australia for at least five years.

Each speaker recorded up to 6 utterances per noise condition, with each utterance consisting of either a string of navigation menu commands or single navigation addresses. The utterances were collected under 7 driving scenarios common to Australian conditions. These conditions are shown in Table II. In contrast to the AVICAR database, this database also consists of data collected under HVAC (heating, ventilating and air-conditioning) conditions. It should be noted that the 50-60 km/h conditions convert to approximately 35 mph, with the 90-100 km/h conditions approximately 60 mph.

TABLE II

NOISE CONDITIONS IN THE AUSTRALIAN IN-CAR SPEECH DATABASE.

Condition	Description
C0	Car idle, sealed cabin, no HVAC
C1	Medium speed (50-60 km/h), sealed cabin, no HVAC
C2	Medium speed (50-60 km/h), sealed cabin, HVAC on high fan
C3	Medium speed (50-60 km/h), driver window open, no HVAC
C4	High speed (90-100 km/h), sealed cabin, no HVAC
C5	High speed (90-100 km/h), sealed cabin, HVAC on high fan
C6	Car idle, sealed cabin, HVAC on high fan

Utterances were recorded using an array of 8 microphones mounted on the central roof console pointing downwards. This location is an industry-favoured position due to ease of integration with existing electronics whilst still providing good signal-to-noise ratios [19]. The microphones were spaced symmetrically around the midline of the vehicle with 2 cm spacing between microphones. The average location of the driver's mouth was estimated (with reference to the microphone closest to the driver) to be 35 cm to the right, 25 cm below, and 17.5 cm behind this reference microphone.

Like the AVICAR database evaluation protocol presented in [16], the Australian In-Car Speech Database was broken into 5 folds consisting of 10 speakers in order to facilitate k-fold leave-one-out speech recognition experiments.

V. VERIFICATION OF FPGA DESIGN

To test the FPGA implementation, a method was required to accept standard speech waveforms from a PC – where they can also be passed through speech recognisers – and collect the corresponding output data and pass it back to the PC for storage and comparison. This requirement was met through the use of the USB test harness detailed in [10]. The FPGA development platform used for this work was the Xilinx Spartan-3A DSP 1800A development board.

A. Waveform Analysis

Initial testing of the accuracy of the FPGA design was performed by comparing the output generated by the hardware implementation against equivalent outputs produced by the floating-point MATLAB implementation of the spectral subtraction algorithm. Test inputs comprised in-car speech signals from the AVICAR database [15] as well as synthesised waveforms like the amplitude-modulated chirp signal used in [10]. As an example, a typical AVICAR speech sample under the 35 mph with windows down noise condition

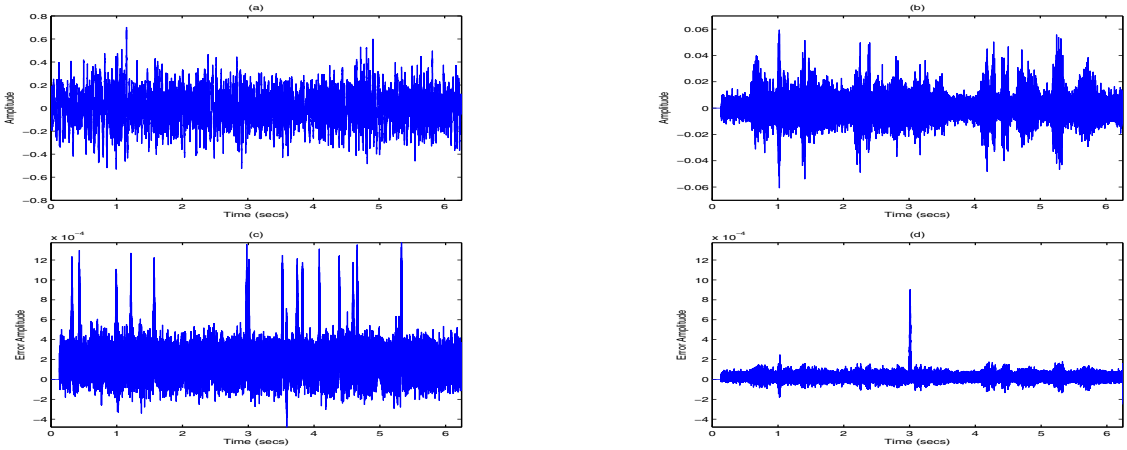

Fig. 3. *(a) Noisy speech signal from AVICAR database, (b) output of spectral subtraction algorithm, (c) difference between floating-point and initial FPGA design, and (d) difference between floating-point and optimised FPGA design.*

is shown in Fig. 3(a). The corresponding spectral subtraction output is shown in Fig. 3(b). Since the output signals from the hardware and software implementations were very similar, sample-by-sample comparisons were performed with the floating-point algorithm. These are shown in Fig. 3(c) and (d) for the initial and optimised FPGA designs respectively.

Observing the waveforms in Fig. 3(a)-(b), it can be seen spectral subtraction provides noticeable signal enhancement of the noisy in-car speech. Prior to enhancement, the time-domain structure of the speech signal is not visible – after enhancement the regions of speech are more pronounced.

Analysing the difference signals created by the two FPGA designs (Fig. 3(c)-(d)), the average gain of 13.5 dB represents an approximate improvement of 2-bits from the initial design to the optimised design. The continued presence of spikes in the difference signal can be attributed to the Xilinx FFT block outputting a quantised version of its internal scaling factor. Through design optimisation, the frequency of the spikes has been reduced to less than 1 in every 200 samples.

B. Speech Recognition Evaluation

The true test of effectiveness of the FPGA implementation for use in in-car speech recognition is to evaluate the FPGA processed waveforms using a speech recognition engine. In these experiments we applied the floating-point and FPGA implementations of spectral subtraction to the two in-car databases described in Section IV. The centre microphone (M4) is chosen for the AVICAR database, whilst the microphone closest to the driver (M0) is chosen for the Australian in-car speech database.

Context-dependent 3-state triphone hidden Markov models (HMM) were trained using the American English Wall Street Journal 1 corpus to enable speaker-independent speech recognition. The acoustic models were trained using 39-dimensional Mel-Frequency Cepstral Coefficient (MFCC) vectors – 13 MFCC (including C_0) plus delta and acceleration coefficients. Each HMM state was represented using a 16-component Gaussian Mixture Model. Utterance decoding was performed using the Hidden Markov Model Toolkit (HTK) [20] on a PC (i.e. not implemented in the FPGA). All speech recognition results quoted in this paper are word accuracies (in %).

Speech recognition results are shown in Tables III and IV for the AVICAR and Australian In-Car Speech databases respectively. Analysing these results it can be seen that all versions of the spectral subtractor provide improvements in recognition performance across a range of in-car noise scenarios. Most importantly the optimised FPGA design performs very close to that of the floating-point algorithm, proving this design is more than suitable for in-car speech recognition systems.

The exception to these observations is the 50-60 km/h with window down condition in the Australian In-Car Speech database. It can be seen that the initial FPGA design failed to deal with the noise and performance reduced by almost 20% from the no enhancement case. Further analysis showed this noise condition is highly susceptible to microphone vibration due to wind (from the open window) which causes very high amplitude values in the low-frequency range (compared to higher frequencies). These high amplitudes were originally unable to be handled by the lower-precision FPGA due to overflow in some of the hardware blocks, particularly the FFT/IFFT block. This shortfall of the original design was corrected in the optimised FPGA design where the speech recognition performance is less than 1% inferior to the floating-point version and improves on the non-enhanced case by around 7%.

C. FPGA Resource Utilisation

Table V shows the total resources required to implement the spectral subtraction algorithm design in a Spartan-3A DSP FPGA device. The 'initial' and 'optimised' designs are identical in terms of the design architecture – the only difference is the bit resolution used within the design, explained in Section III-C. Each sub-block of the original design was optimised so that the number of bits used was sufficient for processing a range of types of speech input with minimum quantization error and no arithmetic overflow. Overall, the

TABLE III

IN-CAR SPEECH RECOGNITION RESULTS (% WORD ACCURACY) ON THE AVICAR DATABASE.

	No Enhancement	Floating-Point SS	Initial FPGA SS	Optimised FPGA SS
Idle	71.52	74.81	70.54	74.66
35 mph, windows up	49.56	54.74	54.90	54.76
35 mph, windows down	37.18	40.85	41.61	40.86
55 mph, windows up	42.77	50.70	50.73	50.55
55 mph, windows down	24.61	30.71	30.82	30.67

TABLE IV

IN-CAR SPEECH RECOGNITION RESULTS (% WORD ACCURACY) ON THE AUSTRALIAN IN-CAR SPEECH DATABASE.

	No Enhancement	Floating-Point SS	Initial FPGA SS	Optimised FPGA SS
Idle, no HVAC	84.89	86.88	86.17	86.88
Idle, HVAC	41.16	52.76	52.80	52.88
50-60 km/h, no HVAC	69.68	76.17	70.56	76.21
50-60 km/h, HVAC	34.06	48.31	47.61	48.39
50-60 km/h, no HVAC, window down	53.01	60.57	34.19	59.75
90-100 km/h, no HVAC	53.88	61.51	60.50	61.63
90-100 km/h, HVAC	30.54	45.24	45.68	45.35

TABLE V

SPARTAN-3A DSP 1800A FPGA RESOURCE USAGE SUMMARY.

Resource Type	Available	Usage (%)	
		Initial	Optimised
Slices	16640	1622 (9%)	2196 (13%)
Flip Flops	33280	2581 (7%)	3093 (9%)
4-input LUTs	33280	2419 (7%)	3010 (9%)
BRAM	84	10 (11%)	10 (11%)
DCM	8	1 (12.5%)	1 (12.5%)
DSP48	84	21 (25%)	25 (29%)

'initial' design used 9% of the total (general FPGA logic fabric) slices available, and 25% of the DSP48 XtremeDSP™ blocks. The larger percentage use of the DSP48 blocks is expected due to the intensive DSP requirements of the algorithm. The percentage use of other key resources, block-RAM (BRAM) and digital clock manager (DCM) blocks is of a similar level to the slice usage. Optimisation of the design resulted in a slight increase in resource usage to 13% of slices and 29% of DSP48 blocks. The use of larger bit widths within the various sub-blocks requires the allocation of additional resources to implement the design.

VI. DISCUSSION

The performance-optimised fixed-point FPGA implementation of the spectral subtraction enhancement algorithm closely matches that of a floating-point equivalent model running on a PC. This was verified through waveform analysis and speech recognition experiments; the latter showed a maximum word accuracy variation between the two implementations of 0.82% which was in the 50-60km/h with driver's window down condition in the Australian In-Car Speech database. In all other conditions across both databases, the performance difference is less than 0.15%. Furthermore, in all cases the optimised fixed-point design shows improvements over the non-enhancement case of between 1.99% and 14.81%. The lowest levels of improvement occur in the idle noise conditions where the noise level is at a minimum and the non-enhanced speech recognition accuracy is at its peak. This clearly demonstrates that the current fixed-point FPGA design can provide a level of performance suitable for use in automotive environments.

Despite the considerable improvement in speech recognition accuracy of the optimised FPGA design, the greater deviation from the floating-point case of the 50-60 km/h with driver's window down condition warrants further investigation. As previously mentioned, this noise condition suffers from wind from the open window disturbing the microphones, causing them to vibrate. This leads to very high amplitudes in the low-frequency components of the signal. Despite the optimised design reducing this effect considerably, it appears that the output from the FFT block – which has been set to the maximum bit-width available from the Xilinx IP core used in this design – is still experiencing some arithmetic overflow, causing the noise estimation and subtraction processes to become less accurate.

The spikes observed in the difference between floating-point and FPGA designs (Fig. 3(c)-(d)) appear to be another artefact resulting from limitations of the FFT block. Using the maximum bit-width available has reduced the occurrence of these significant deviations, but has not eliminated them. To improve the design further, a new, higher resolution FFT/IFFT block would be needed – the implementation of which would require significantly more FPGA resources. Alternatively, a redesign of the pre-emphasis filter for greater

attenuation at low frequencies would lead to some improvement in quality, at a more modest increase in resources.

Having successfully implemented the spectral subtraction algorithm in a Spartan-3A DSP 1800 device, we have demonstrated its suitability for instantiation in an equivalent Xilinx Automotive FPGA (the XA3D1800A). Furthermore, the performance of the fixed-point spectral subtraction implementation has been demonstrated to provide a clear improvement in speech recognition accuracy in the presence of automotive environment noise. However, another important consideration is the FPGA resource usage, which will have an impact on the cost of such a system should it be commercialised.

Moving from the 'initial' to the current, 'optimised', implementation resulted in a slight increase in FPGA resources used: from 9% to 13% of general logic fabric slices; and 25% to 29% of the specialised DSP48 blocks. Providing the necessary performance improvement through increasing bit widths in key blocks naturally requires increased logic and DSP resources. The above increases were expected and are of an order that is acceptable given the resultant recognition accuracy improvement. Further improvements in the performance could be made as noted, however, given the current performance level, increasing the resource "cost" may not be warranted. Overall, the current design uses less than one seventh of the FPGA resources available in the Spartan-3A DSP 1800 device, apart from the specialised DSP48 blocks, of which over 70% remain free for other uses. This low resource usage enables other processes (such as CAN communications or HMI providing infotainment, driver information and possibly driver assistance) to be incorporated into a single FPGA. By amortising implementation costs over a number of applications, overall manufacturing component costs can be kept to a minimum.

Having verified the Spartan-3A DSP spectral subtraction implementation on speech data collected in automotive environments, the system is ready for real-time tests in a vehicle. A stand-alone Spartan-3A DSP board has been developed and these trials will be conducted in the near future.

VII. Conclusion

A fixed-point design of the frequency-domain spectral subtraction enhancement algorithm has been successfully implemented in a Xilinx Spartan-3A DSP FPGA. Speech recognition experiments using data collected in automotive environments have shown the performance of the fixed-point FPGA implementation closely matches that of an equivalent floating-point version running on a PC. In all tested cases the FPGA implementation demonstrated a clear speech recognition improvement over a system without enhancement.

The previous Virtex-4 FPGA design was ported to a Spartan-3A DSP FPGA and experimentation showed that the performance of the initial design was satisfactory under most noise conditions tested – but not all. Optimising the implementation by increasing bit-widths in various parts of the design have improved the performance such that it matches the floating-point model under all tested driving conditions whilst only utilising 4% more FPGA resources. By proving the design in a Spartan-3A DSP device, it has been demonstrated that it can be implemented in an automotive grade FPGA. Furthermore, the design uses only 13% of the general logic resources available and 29% of the specialised DSP blocks; it is clearly suitable for integration on a single FPGA with other automotive processes such as CAN communications or HMI.

References

[1] C.-C. Cheng, W.-H. Liu, C.-H. Yang, and J.-S. Hu, "A robust speech enhancement system for vehicular applications using H_∞ adaptive filtering," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2006, pp. 2541–2546.

[2] S. Yu, "Hybrid speech enhancement and speech recognition system for car telematics platform for hands-free control GPS navigator and voice dialer for handphone," ASEAN Virtual Instrument Applications Contest Submission, 2006.

[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[4] P. Lockwood, J. Boudy, and M. Blanchet, "Non-linear spectral subtraction (NSS) and hidden Markov models for robust speech recognition in car noise environments," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1992, pp. 265–268.

[5] A. Wahab, E. C. Tan, and H. Abut, "CMAC spectral subtraction for speech enhancement," in *Sixth International Symposium on Signal Processing and its Applications*, vol. 2, 2001, pp. 707–710.

[6] K. Kitagawa, "At the heart of consumer and automotive innovation," *XCell Journal*, no. 63, pp. 12–13, 2007.

[7] Xilinc Inc., "Xilinx Automotive - flexible solutions beyond silicon," 2007.

[8] K.-F. C. Yiu, Y. Lu, X. Shi, and W. Luk, "FPGA acceleration of a subband beamforming algorithm for speech enhancement," in *Congress on Image and Signal Processing*, vol. 5, 2008, pp. 742–746.

[9] D. Halupka, A. Rabi, P. Aarabi, and A. Sheikholeslami, "Low-power dual-microphone speech enhancement using field programmable gate arrays," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3526–3535, 2007.

[10] J. Whittington, K. Deo, T. Kleinschmidt, and M. Mason, "FPGA implementation of spectral subtraction for in-car speech enhancement and recognition," in *2nd International Conference on Signal Processing and Communication Systems*, December 2008.

[11] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1979, pp. 208–211.

[12] P. Loizou, *Speech enhancement: theory and practice*. Boca Raton, FL: CRC Press, 2007.

[13] R. Martin, "Spectral subtraction based on minimum statistics," in *EUSIPCO*, 1994, pp. 1182–1185.

[14] T. Kleinschmidt, S. Sridharan, and M. Mason, "A modified LIMA framework for spectral subtraction applied to in-car speech recognition," in *1st International Conference on Signal Processing and Communication Systems*, 2007, pp. 335–338.

[15] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. Huang, "AVICAR: Audio-visual speech corpus in a car environment," in *INTERSPEECH*, 2004, pp. 2489–2492.

[16] T. Kleinschmidt, D. Dean, S. Sridharan, and M. Mason, "A continuous speech recognition protocol for the AVICAR database," in *1st International Conference on Signal Processing and Communication Systems*, 2007, pp. 339–344.

[17] D. Bagni and P. Zoratti, "Block matching for automotive applications on Spartan-3A DSP devices," *XCell Journal*, no. 63, pp. 16–19, 2007.

[18] V. Sardana, "Slash your total cost by up to 50% with Spartan-3 generation FPGAs," Xilinc Inc., Tech. Rep., 2008.

[19] M. Sala, H. Wengelnik, H. van den Heuvel, A. Moreno, E. Le Chevalier, E. Deregibus, and G. Richard, "SpeechDat-Car: speech databases for voice driven teleservices and control of in-car applications," in *Proc. EAEC*, 1999, pp. 90–98.

[20] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, 3rd ed., Cambridge University Engineering Department, December 2006.