An Efficient Method for the Use of Overlapped Analysis Windows Employing the Goertzel Algorithm

Julian Penketh, MSc, AMIEE, University College London

Abstract: *For increased bandwidth savings in packet networks, speech compression algorithms are applied to reduce the amount of data that needs to be transferred to adequately reconstruct a telephone signal at the receiver. However, such algorithms do not process pure tones (such as dual-tone multi-frequency (DTMF)) with high fidelity, hence such in-band signalling tones must be detected before packetization occurs. To minimise the amount of DTMF tone distortion, and detect valid tones which may contain impairments, the tone recognition algorithm needs to decode tone-pairs in as short a time as possible. Many existing systems already use the Goertzel algorithm for DTMF tone detection. This paper presents a modification to the algorithm which allows an effective increase in detection frequency by overlapping, without the need for re-processing of input data*[1].

1 Introduction

1.1 DTMF Digit Packet Relay System

The DTMF decoder and DTMF tone generation sub-system developed formed part of the core digital signal processing (DSP) software for the Nortel Networks Passport Voice Gateway (PVG) switch. PVG provides telecoms network operators with an efficient transport mechanism for voice between public switched telephony network (PSTN)-based terminals via packet-switched networks, such as asynchronous transfer mode (ATM). A diagram of the typical set-up showing detection of a DTMF tone-pair, transmission of an ITU AAL2 (type-3) packet [1] containing the incoming ‚digit', and regeneration of a DTMF signal is given in Figure 1.1:

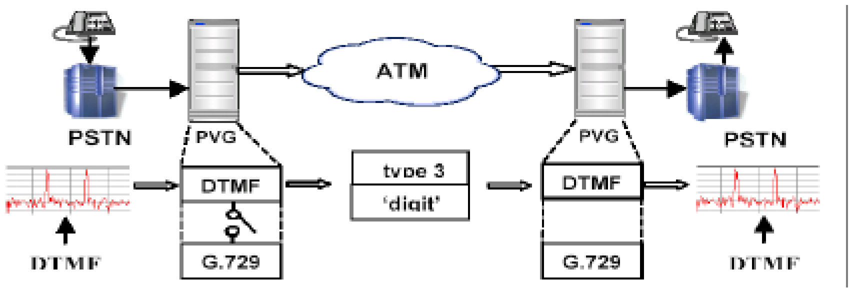

Figure 1.1 System-level view of DTMF digit relay

As can be seen from Figure 1.1, there exist several major blocks to create the DTMF digit relay system: DTMF signal detector/decoder, selection between the G.729 speech compression algorithm [2], packet transmission/reception, and DTMF signal regeneration. Standards exist from the ITU stating within which criteria a DTMF receiver is to operate [3], and to what specifications a DTMF signal generator is to re-create tones [4]. The next section discusses the heart of the receiver system, and the focus of this paper – the DTMF digit decoder.

1.2 DTMF Digit Detection

1.3 The Goertzel Algorithm (GA)

In 1958 G.Goertzel published a paper which revealed a time-domain solution that yielded the same value for the magnitude of a spectral line as that which would have been achieved (for that particular frequency) if a complete Fourier analysis had been performed [5]. The GA has become is an industry-standard method of extracting spectral information from an input signal when only a small number of spectral lines are to be analysed. For DTMF tone detection it is an highly effective and efficient choice of algorithm employing pairs of two-pole resonators to detect the dual tones making up the incoming DTMF signal. It is thereby, a recursive infinite impulse response (IIR) filter-type structure and thus has advantages of lower computational memory requirements than an equivalent finite impulse response (FIR) filter. Unlike the FFT or DFT, the GA does not need a complete block of data before it can begin processing.

[1] Note: This work was carried out at the Nortel Networks Harlow Laboratories. All intellectual property pertaining to this modification to the Goertzel algorithm is Nortel Networks proprietary.

1.3.1 Motivation for Using Overlapped Windows

How long it takes to verify the validity of a received DTMF signal has a serious impact on performance within a packet voice network scenario. A previous method developed by the Author incorporated an early warning indicator (EWI) block in the sub-system which performed high-speed scanning of the input data for the 'likely' presence of valid DTMF tone, whereupon the full GA detector would be called [6]. However, this meant that the same data used for the EWI would be re-processed by the GA if the EWI sensed a tone (inefficient in terms of mega instructions per second, or MIPS), and made the system/software more complex (uses more DSP memory also). The optimal solution is one where the DTMF decoder itself is of high enough quality (i.e. does not provide too many false detects) and can operate with very few incoming data points. The original implementation using the GA required two successive windows of 12.75ms (102 samples each) to validate a digit for DTMF digit relay. Even in the case of basic DTMF upspeed[2], where it was decided to make a DTMF decision after only a single window, waiting a full 102 samples was still not desirable.

Now imagine we can obtain a GA result every *N/2* samples, rather than every *N* samples (where *N* is 102). Now, if say the first received window contains 90% DTMF signal, the latter 50% would contribute to our measurement of valid DTMF tone (with the *N*-sample detector, none of the initial 90% DTMF tone contributes). The detector would then appear, *on average*, to be faster. Such an increase in granularity can be achieved if we overlap successive windows, as depicted in Figure 1.2. With this method, the GA is not operating any differently. It would still require an initial *N* DTMF samples to report a digit, but from there on in during a burst of DTMF tone (typically 40ms to 50ms in duration), a DTMF frequency analysis would effectively be performed every 51 samples (6.375ms), rather than every 102 (12.75ms).

Applying such overlapping techniques is not new, and it can be seen used in conjunction with algorithms such as the FFT. Initially for the Goertzel implementation however, it was thought that the last 51 samples would have to be physically processed twice. Figure 1.2 shows the procedure for an initial segment of DTMF tone. Note that it is slightly unrealistic in that it shows the start of the detection window coinciding exactly with the start of the tone.

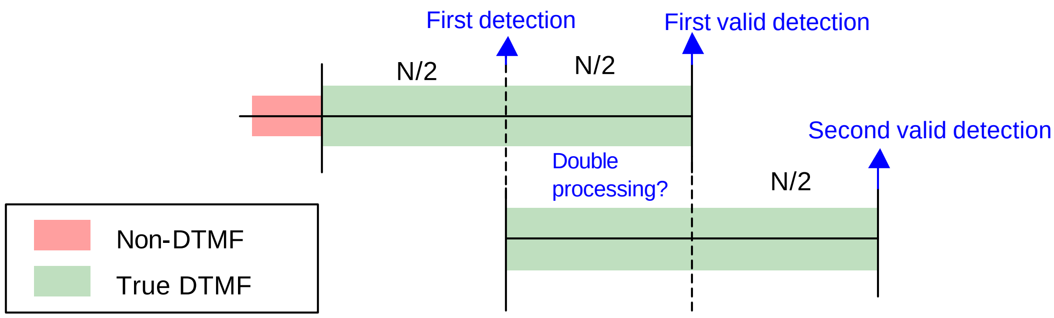

Figure 1.2 Overlapped analysis windows for GA detector

From the diagram in Figure 1.2 it is evident that the latter half of each window, being re-processed in the computation for the next window, will mean that the MIPS budget is going to reduce by approximately 50%. This impact on MIPS requirements is not desirable and in certain software builds for the PVG product was entirely untenable, given the amount of other forms of processing being carried out on incoming voice/tone data. However, it is possible to modify the GA mathematically to allow the above overlapping to be carried out with negligible impact on the MIPS budget. Instead of re-processing the last half of each *N*-sample analysis window, we effectively 'split' the GA into two equal portions to accomplish our goal of yielding an *N*-sample result every *N/2* samples (after the initial full window). The only additional processing is a few extra instructions being executed in the DSP every *N/2* samples. An analysis of this approach is given below.

2 Analysis of the Split-Goertzel Method

2.1 Background on Goertzel Resonator

Before expounding the details of the split version of the GA, some background will be given to the ordinary version. The GA is essentially an efficient way of calculating the DFT for known frequency values, where after a period of *N* samples, the filter output is equal to a spectral magnitude *X(k)*. It can do this by taking advantage of the periodicity of phase factors, which leads to the expression of the computation of the DFT as a linear filter operation utilizing recursive difference equations. Efficiency is enhanced by the fact that only a few (known) spectral lines are required, meaning that filter coefficients can be pre-calculated. Use of a second-order IIR structure for the Goertzel filter incorporating two complex-conjugate poles facilitates the computation of the

[2] Instead of sending a type-3 packet and re-generating DTMF tone, transmitter upspeeds to full 64 kb/s PCM rates to preserve tone quality (this is not efficient in terms of network bandwidth however).

difference equation by having only one real coefficient. Placement of the poles on the unit circle causes a filter to resonate when the input signal contains a spectral component at the appropriate frequency [7].

The GA difference equations will only be stated here as a starting point for the treatment of the split GA, the reader is referred to [5] and [1] for derivations of the basic supporting theory. It can be proved that the computation of the next output, $w_k(n)$, of the GA is:

$$w_k(n) = 2.\cos(2\pi k/N).\, w_k(n-1) - w_k(n-2) + x(n) \qquad (1)$$

Noting that at start-up $w_k(n-1)$ *and* $w_k(n-2) = 0$.

For tone detection, we don't need phase information - the squared magnitudes of the equivalent DFT values suffice. Equation (1), which constitutes the feedback portion of the GA, is iterated N times whereupon we then use the last two values to calculate the squared magnitude, $|y_k(N)|^2$. The resulting DFT-equivalent value, $|X(k)|^2$, is given by:

$$|y_k(N)|^2 = |X(k)|^2 = w_k^2(N) + w_k^2(N-1) - 2.\cos(2\pi k/N).w_k(N).\, w_k(N-1) \qquad (2)$$

2.2 Split GA Concept

Figure 2.1 below gives a representation of what it is we are trying to achieve with the split Goertzel method. From the diagram it is evident that a kind of half-frame processing, predicting, and combining is going on to obtain all the information necessary to yield a result for the squared magnitude the same as that obtained if using a consecutive N samples.

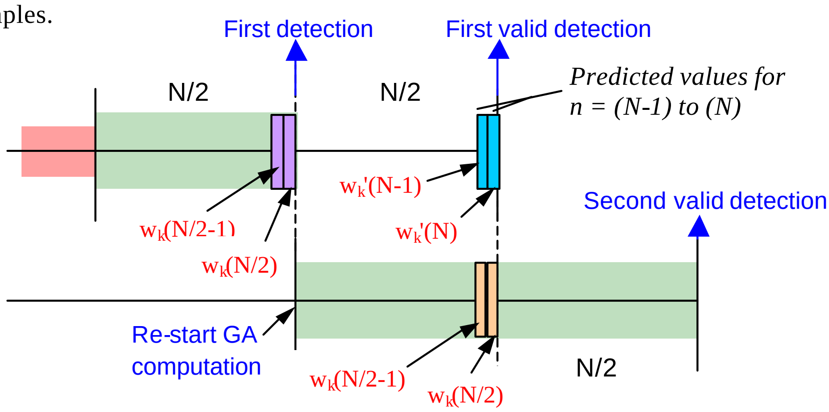

Figure 2.1 'Overlapped' analysis windows for split GA detector

The first step with the split GA is to calculate $w_k(N/2)$ and $w_k(N/2-1)$ using Equation (1), i.e. just run the GA in the usual manner from samples $n = 1$ to $N/2$. Next, the GA is essentially reset, to look just as it did $N/2$ samples previously (i.e. the current values for $w_k(N/2)$ and $w_k(N/2-1)$ are equated to zero - after being copied to memory) so the next 51 samples can be processed in exactly the same way. This process repeats indefinitely, obtaining new values for $w_k(N/2)$ and $w_k(N/2-1)$ every 51 samples. Now imagine we had *not* reset $w_k(N/2)$ and $w_k(N/2-1)$, but reset the input to zero instead and run a continuous GA for the next 51 samples (see top part of diagram). Because of linearity, we could add the output values shown as $w_k'(N)$ and $w_k'(N-1)$ to the latest ones for $w_k(N/2)$ and $w_k(N/2-1)$ (bottom part of diagram) and arrive at the same result as if we had just run the normal GA without any mid-frame resets. The mathematical trick at the core of the split Goertzel method to avoid having to run the GA every other half-frame with zero for input is to multiply the stored variables $w_k(N/2)$ and $w_k(N/2-1)$ by some function, whose value depends on the position of the required outputs to be predicted along the axis $n = N/2+1$ to N. To derive this function it is necessary to create a new solution for $w_k(n)$, one which depends on the values of the variables $w_k(N/2)$ and $w_k(N/2-1)$. A general form of the Z-transform for Equation (1) is given below:

$$W_k(z) = ZT\{A.w_k(n-1) - w_k(n-2)\}$$

where A = 2.cos (2pk/N)

$$= A.w_k(N/2) - w_k(N/2-1) + b.z^{-1} + c.\, z^{-2} + \ldots + ?.\, z^{?} \qquad (3)$$

2.3 Solution

Work was undertaken to solve Equation (3) and the following result was obtained for the time-domain:

$$w_k(n) = w_k(N/2)\,.\, f(n) - w_k(N/2-1)\,.\, f(n-1)$$

$$= w_k(N/2)\,.\, \frac{\sin(2pk/N[n+2])}{\sin(2pk/N)} - w_k(N/2-1)\,.\, \frac{\sin(2pk/N[n+1])}{\sin(2pk/N)} \qquad (4)$$

Equation (4) says that for any desired resonator output, $w_k(n)$, we need only substitute the sample number for n to calculate it (when the input is zero). Since we are dealing with samples in the range 0 to 50 for our prediction region, we would simply put a value 0 to 50 in place of index n. Since at each half-frame stage we use $w_k(N/2)$ and $w_k(N/2\text{-}1)$ from 51 samples ago, we would calculate Equation (4) for n = 50 and n = 51 to obtain our 'predicted' values. These can then be added to the latest values for $w_k(N/2)$ and $w_k(N/2\text{-}1)$ and substituted for $w_k(N)$ and $w_k(N\text{-}1)$ respectively in Equation (2) to obtain the full N-sample squared magnitude.

2.4 Performance and Measured Results

Some empirical measurements were carried out at the Nortel Networks laboratory to try and quantify the amount of actual, rather than purely theoretical improvement to the DTMF detector. Figure 2.1 shows a live screen capture from a PC-based signal generator and oscilloscope connected to the PVG DSP test-bed. The first image is of six DTMF tone-bursts which have been detected, then upspeeded to 64kb/s, where the generated pulses (not shown) had minimum temporal characteristics (duration = 40ms, inter-digit gap = 53ms). The top three pulses show how the standard GA -based detector performed. Due to the latency in detect time, it is possible for the first quarter of the signal to become squelched and distorted by the G.729 (8kb/s) codec compressor, before the upspeed occurs. The lower three pulses show the yield when applying the split GA version of the software, and it can be seen that although we always lose the first three or four milliseconds of tone, detection is effective much sooner with less distortion of the signal and there is no gradual 'ramping' effect either; a clean 64kb/s signal is apparent very rapidly relative to the original system. (Note that in the case of DTMF relay where we re-generate the tone-pair at the far-end, being able to detect the end of a tone as precisely/quickly as possible is also important to guard against shortening or lengthening relayed DTMF signals.) Many tests were carried out and the picture below depicts the average case. The two small images on the right show the frequency-domain distortion to a DTMF tone-pair caused by the G.729 compressor (top), and after removal of distortion by upspeeding (bottom).

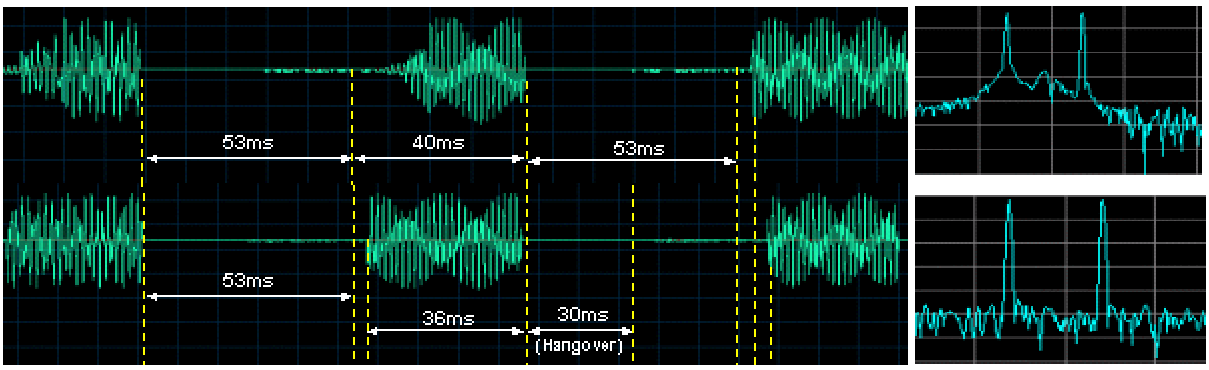

Figure 2.1 Improvements using 'Overlapped' analysis windows for split GA detector

2.5 Conclusion

The benefits of the mathematical modifications to the GA are self-evident, though more work such as studies into how many times we could go on splitting the GA to further reduce effective detect time are to be undertaken. Other aspects not discussed are frequency resolution concerns when applying the GA with fixed window lengths; the different frequency tolerances set by the different administrations cannot exactly be met when a uniform window length is used for all the Goertzel resonators. Investigation into non-uniform approaches using the split GA will also be carried out. Other methods of performing fast DTMF/signalling tone detection are also under investigation and work will include comparing performances with the split GA.

Acknowledgments

Acknowledgements are extended to Mark Stewart for originally showing how the Goertzel algorithm could be split using mathematics, and to Joachim Fouret for his assistance with some of the theoretical aspects during modelling and evaluation.

References

[1] "AAL2 Service Specific Convergence Sublayer for Trunking", Annex K., ITU-T Rec. I.366.2
[2] ITU-T Recommendation G.729 Annex B
[3] "Multi-frequency Push-Button Signal Reception", ITU-T Recommendation Q.24.
[4] "Technical Features of Push-Button Telephone Sets", ITU-T Recommendation Q.23
[5] "An Algorithm for the Evaluation of Finite Trigonometric Series", G.Goertzel, American Math Monthly, Vol.65, Jan. 1958, pp 34-35.
[6] "Providing Touch-Tone Services in Voice Over Packet Networks", J. Penketh, Proceedings of the London Communications Symposium 2000, Communications Engineering Doctorate Centre ISBN: 0-9538863-0-1
[7] "Discrete-Time Signal Processing", 2nd Ed., A.V.Oppenheim, R.W.Shafer, J.R.Buck, Prentice-Hall Inc., 1999.