Proceedings of the Institute of Acoustics

OBTAINING ACCURATE FREQUENCY INFORMATION FROM POWER SPECTRA

R Parks

University of Edinburgh, Department of Physics, King's Buildings, Edinburgh EH9 3JZ

1. INTRODUCTION

For many applications in musical acoustics, the power spectrum is a most effective way of describing the component frequencies present in a sound, together with their relative amplitudes. It is derived from the Discrete Fourier Transform (DFT), for which in practice the Fast Fourier Transform (FFT) is almost universally employed (simply a special case of the DFT giving efficient processing in return for a minor constraint on transform size).

The Fourier Transform processes a number of samples, n, taken at regular intervals over a total time, T, and determines the amplitude and phase for n/2 calculation frequencies, each being an integral multiple of the frequency interval 1/T. The output from the DFT can be considered as n/2 'frequency bins' at intervals of 1/T, each containing a calculated sum of the total amplitude of components lying within a band around its centre frequency.

In the ideal case of a periodic signal where the portion of signal analysed spans an exact number of cycles of the fundamental, then the fundamental and its harmonics each lie exactly on a calculation frequency without affecting others. If this ideal condition is not present, the signal frequencies lie between the calculation frequencies, causing the analysis to attribute them in a widespread pattern which varies according to the frequency mismatch, an effect termed "leakage". The ideal situation is often unattainable, as the signal frequency may not be known in advance, or the sample rate may not be adjustable to the precise value. More importantly, analysis should cope with several signals combined, at unknown frequencies.

The remedy is to multiply the data time-series by a "window function" which is unity in the middle and tapers towards zero at each end. The effect is to give a rounded peak spanning several frequency intervals, with fairly uniform shape regardless of where the signal frequency lies within the frequency interval, and with a substantial reduction in the leakage to distant bins. Peak shape depends on the window function, but for a given function the peak always spans the same number of frequency bins even when their width is altered by other factors such as transform size.

At first sight this uniformity might appear to have been obtained at the expense of severe degradation of frequency resolution. On closer examination, however, it emerges that the data is not in fact corrupted randomly but transformed in a rigorously mathematical form from which it can be recovered by interpolation to an accuracy better than 1% of a DFT frequency interval.

2. INTERPOLATION

The result of the DFT performed on windowed data is the convolution of the DFT of the window function and the DFT of the raw data. This mathematical statement unfortunately does not offer a simple way of recovering the frequency information. Instead an empirical approach has been developed from careful study of the characteristics of the output for calibration signals. The term "interpolation interval" is convenient for the difference between the true signal frequency and the calculation frequency immediately below it, as a fraction of DFT frequency interval.

Fig 1 is a montage comparing the DFT line clusters for ten interpolation intervals at increments of 0.1, in which some features can be readily distinguished by eye.(the pattern for 1.0 is identical to that for 0.0). The pattern for 0.0 is symmetrical about a single line, while that for 0.5 is symmetrical about a pair of lines; the skewed patterns for below 0.5 are mirror images of those above 0.5, with the relative line heights changing smoothly. While an interpolation would be possible using only the relative heights of the two highest lines, the risk of degradation by spurious signals is reduced by using differences between the four highest lines. The three separate estimates are combined with subjectively assigned weights to allow for the greater risk of contamination for lower lines. With the preferred window, straight-line interpolation is adequate for differences 1-2 and 2-3, while a second-order polynomial is used for difference 3-4.

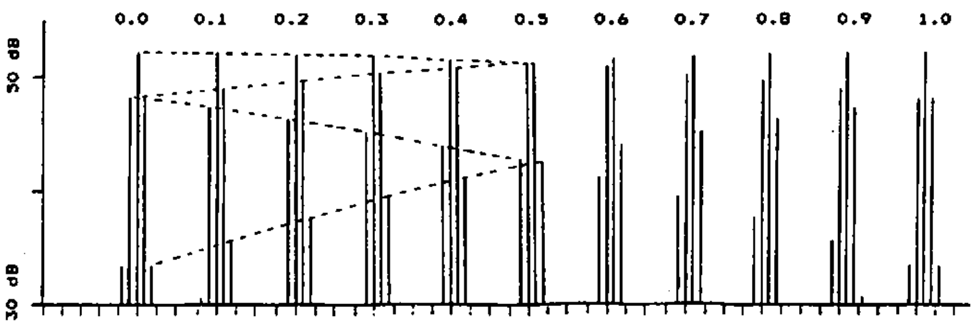

Fig 1: Variation of cluster shape with interpolation interval

3. PROGRAM

3.1 General

The version listed below is in MS GW-BASIC for use on PC. Original development was in MS BASIC with later modifications to suit BBC Basic. Output is a printed list, providing a worksheet for subsequent stages of analysis. The dB value of the highest line in each cluster is an aid for the essentially manual stage of checking for valid peaks omitted and spurious ones inserted.

3.2 Data Format

The disk file format commences with an ASCII character string for data description, followed by a sequence of numeric values giving dB for each element of the DFT output, corresponding to array indices from zero to (transform size)/2. [The zero element value is not used in interpolation, but is included in the spectrum file for other purposes]. The figures for sample rate and transform size are not read in as data, and the user must enter them when prompted. (The identifier string can include them as a reminder).

3.3 Program Operation

The program operates on a power spectrum in dB from a disk file in two passes, the first identifying lines which stand out from the background level and the second grouping these lines in clusters and interpolating a centre frequency for each cluster.

Recognition of prominent lines is based on comparison of the running mean of three lines with that of the two before and two after them. If a specified threshold is exceeded, the central line has a flag set for its subsequent treatment.

In the second pass, a cluster is defined as starting where the line flags change from zero to one, and finishing where they return to zero. Within each cluster the highest line is identified, and the lines before and after it compared to establish the polarity of interpolation, i.e. whether the centre frequency is above or below that of the highest line. This result also defines which line is to be used as the fourth highest.

Three separate interpolations are then performed, on the basis of height differences from first-to-second, second-to-third and third-to-fourth lines respectively, and a weighted mean is derived. As a rudimentary indication of the degree of agreement between the three interpolated estimates in each case, an unweighted standard deviation of the three is provided (expressed as a fraction of frequency interval), and the user is left to decide whether to accept or reject the value.

3.4 Program Variants

While the preferred window function is a specific quasi-Gauss form, an acceptable alternative is the widely-used Hanning function, for which a different set of constants and a more elaborate interpolation are required in the interpolation procedure, lines 860 - 920.

Omitting data transfer and automatic recognition of clusters, versions of the interpolation routine have been prepared for programmable calculator (Hewlett-Packard HP41, reverse Polish) and BASIC, to handle single clusters, prompting for input of the sample rate, transform size, frequency of the highest line and dB values for the four highest lines.

4. QUALITY CONTROL

The first level of selection is provided by the choice of threshold level for line identification, and the user may vary this according to the character of the data. In the final list of frequencies there is some help in the form of the standard deviations, but these can be no more than a general guide since each is derived from combining only three values.

The most dependable check comes from visual inspection of the spectrum for uniformity of cluster shape, at a resolution that enables individual lines to be distinguished. In most applications the emphasis is on accuracy and the policy is to discard readings which fail on any one of the three forms of checking.

5. WINDOW CONSIDERATIONS

The operation of windowing in used in many other differing applications of Fourier Transform analysis, and it should be recognised that the desirable properties are by no means the same for each. For present purposes, two features are important: the cluster outline shape should not vary widely with interpolation interval, and the leakage to distant frequencies should be low.

Uniformity of cluster shape is needed for visual assessment of quality, and indeed for some situations where the visual display is a sufficient end-product, as in teaching musical acoustics. The mathematical process of interpolation is made more troublesome (though not impossible) with variability of shape, and the Hanning function has this drawback as interpolation interval nears zero.

Leakage is important with sounds comprising many components. For example, low piano notes have more than fifty partials of broadly similar amplitudes; if the leakage from each is 40 dB below the source level, the combined effect almost swamps the true signals.

The key requirement for low leakage is that the window function should go down to zero at start and finish. The exact Gauss formula does not do so, and is not well suited on this account. However, by good fortune, an approximate Gaussian formula which arose from computational convenience turns out to be better in respect of both shape and leakage than any of the standard formulae examined so far. It has the form (in PASCAL):

```
FOR j := 1 TO tsize DO BEGIN
   wx := abs((j - tsize/2 - 0.5)/(tsize/2));
   window := sqr(sqr(1 - sqr(wx)));
```

6. CALIBRATION EXAMPLE

As an indication of the accuracy obtainable, the procedure has been applied to a triangular wave composed of exact harmonics. The first three columns of the table below are the standard output listing. Column 4 gives the harmonic number assigned by inspection, the first three frequencies and the last being due to mains hum and background noise.

The best estimate of the true fundamental frequency of the signal is taken as the weighted mean of the five strongest peaks, the odd harmonics up to 9, and column 6 shows the departure from this for each harmonic, expressed as a decimal of one DFT interval. In each case the departure is no more than 1% of an interval.

Centre Frequency	Highest Line	Standard Deviation	Harmonic Number	Relative Level	Residual, interval
50.216Hz	-12.3dB	sdi:0.0146	-		
100.230Hz	-16.4dB	sdi:0.0087	-		
151.302Hz	-27.8dB	sdi:0.0282	-		
195.824Hz	+30.0dB	sdi:0.0007	1	0.0dB	-0.0048
391.609Hz	-10.1dB	sdi:0.0260	2	-40.1dB	-0.0098
587.518Hz	+10.3dB	sdi:0.0019	3	-19.7dB	-0.0009
783.415Hz	-16.3dB	sdi:0.0288	4	-46.3dB	+0.0028
979.228Hz	+1.8dB	sdi:0.0027	5	-28.2dB	+0.0007
1174.971Hz	-19.2dB	sdi:0.0119	6	-49.2dB	-0.0037
1370.915Hz	-4.1dB	sdi:0.0063	7	-34.1db	+0.0006
1567.054Hz	-23.0dB	sdi:0.0031	8	-53.0dB	+0.0100
1762.584Hz	-8.8dB	sdi:0.0047	9	-38.8dB	0.0000
1958.177Hz	-24.8dB	sdi:0.0422	10	-54.8dB	-0.0064
2154.387Hz	-13.7dB	sdi:0.0102	11	-43.7dB	+0.0027
2545.631Hz	-18.6dB	sdi:0.0178	13	-48.6dB	-0.0064
3392.533Hz	-31.2dB	sdi:0.0097	-		

This analysis accuracy corresponds to around 1 part in 65000 for the highest frequency measured in this example. Musical sounds containing closely-spaced components present more difficulty than this triangular wave, but provided the peaks are separated by more than ten DFT intervals comparable accuracy can be achieved. The absolute accuracy depends of course on control of the sample rate, but the use of a DAT recorder with digital transfer to computer reduces timing errors to a negligible level.

7. APPLICATION EXAMPLES

Fig 2 is a detail from investigations into inharmonicity of piano strings, showing how observed mode frequencies depart from those predicted by existing theory. A sound containing some 145 partials was analysed, and the graph indicates a clear first-order curve, an order of magnitude greater than the scatter of individual points about the curve.

Fig 3 shows the variation with time for the first three mode frequencies of a plucked cello string. The ability to obtain frequency information from small transforms and consequently short data segments is valuable in tracking rapidly-changing phenomena.

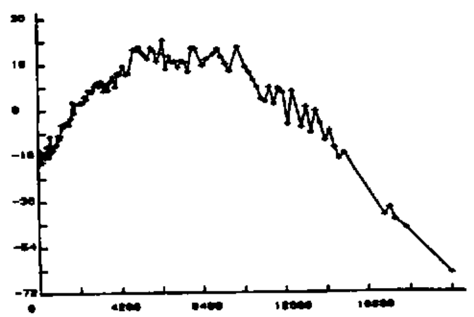

Fig 2: Piano String: Departure of mode frequency from theoretical linear relationship

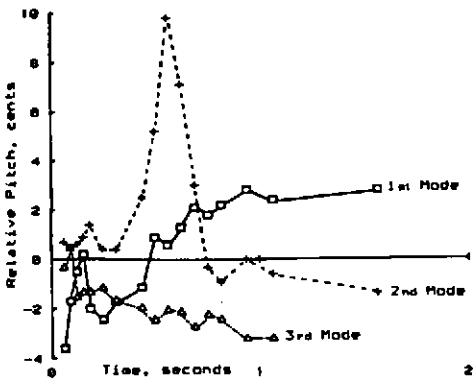

Fig 3: Cello String: Pitch Transient of first three modes of plucked note

8. ACKNOWLEDGEMENTS

A great debt is due to Dr John Davies for his meticulous spectral analysis package, and to Angus Duggan for his extensions to it, enabling the structure of the output to be studied in detail

APPENDIX: PROGRAM LISTING

```
10 C$="Auto Spectral Freq Interpoln, QUASI GAUSS. R Parks 15.1.93"
30 PRINT C$: PRINT
40 CLOSE
50 GOSUB 230:                                  REM  (Find spectrum file)
60 INPUT "Sample Rate, kHz:        "; SR
70 GOSUB 300:                                  REM  (Set Transform Size)
90 N=TS/2: DIM U(N), Z(N), CORD(N): XW=TS/1024: NM=512: LG$=""
110 CT=9:  REM  db diff threshold for cluster recognition
120 GOSUB 470:                                      REM  (Set Threshold)
130 GOSUB 350:                           REM  (Load spectrum from disc)
140 GOSUB 520:                     REM  (Prominent line identification)
150 GOSUB 640:                                 REM  (Freq interpolation)
160 PRINT "HOWZAT?! Another at same parameters?": INPUT A$
180 GOSUB 230: GOSUB 350: GOSUB 520: GOSUB 640: GOTO 160
230 INPUT "
                                          Spectrum File: "; LF$
250 OPEN "I", 1, LF$: INPUT#1, T$, FFTO: LPRINT "          "; T$
290 RETURN
300 TS=2048: PRINT "Current Transform Size: ";TS
320 INPUT "     Enter new or <RET> "; TS$
330 IF TS$="" THEN 340 ELSE TS=VAL(TS$)
340 RETURN
350 FOR I = 1 TO N:  REM ---- Load Spectrum from disc file  ----
370 IF EOF(1) THEN 400
380 INPUT#1, U(I)
390 NEXT
400 PRINT"
                              (Data Loaded)": BEEP: CLOSE #1
420 RETURN
470 PRINT "Current db Threshold:   ";CT: REM -- Set Threshold  ----
490 INPUT "   Enter new or <RET>   "; DB$
500 IF DB$="" THEN 510 ELSE CT=VAL(DB$)
510 RETURN
520 PRINT "
                                       (Scanning for lines)"
540 PRINT "          FFT lines within clusters:"
550 FOR I=5 TO N-8
560 RMF=(U(I-3)+U(I-2)+U(I+2)+U(I+3))/4:RMC=(U(I-1)+U(I)+U(I+1))/3
580 IF RMC-RMF>CT THEN 600 ELSE Z(I)=0
590 GOTO 610
600 Z(I)=1: PRINT I;
610 NEXT: PRINT
630 RETURN
640 PRINT: BEEP: REM      ---- Auto Freq Interpoln  ----
650 SP=1
660 PRINT "
                             (Scanning for interpoln)": PRINT
670 FOR I=SP TO N-4:               REM find beginning of cluster
680 IF Z(I)=0 AND Z(I+1)=1 THEN BC=I+1: GOSUB 710
690 NEXT
```

```
700 RETURN
710 FOR J=BC TO BC+4:                          REM find end of cluster
720 IF Z(J)=1 AND Z(J+1)=0 THEN SP=J+1: GOSUB 750
730 NEXT
740 RETURN
750 REF =-100:                          REM find line no of local max
760 FOR K=BC-1 TO SP
770 IF U(K)>REF THEN REF=U(K): LMXI=K
780 NEXT
790 CORD(1)=LMXI:      REM local max; now find 3 next highest lines
800 P=1: REM polarity of interpoln
810 IF U(LMXI-1)>U(LMXI+1) THEN P=-1
820 CORD(2)=CORD(1)+P: CORD(3)=CORD(1)-P: CORD(4)=CORD(1)+2*P
830 DEL12=U(CORD(1))-U(CORD(2)):  REM the three height differences
840 DEL23=U(CORD(2))-U(CORD(3))
850 DEL34=U(CORD(3))-U(CORD(4))
860 IF DEL12>4 THEN NINT12=0: GOTO 880
870 NINT12=P*(.5 - DEL12/8):                    REM interpoln for 1-2
880 IF DEL23>8.600001 THEN NINT23=P*.5: GOTO 900
890 NINT23=P*.058*DEL23:                        REM interpoln for 2-3
900 IF DEL34<0 THEN NINT34=P*.5: GOTO 930
910 IF DEL34>15 THEN NINT34=0: GOTO 930
920 NINT34=P*(.502-.0365*DEL34+.000184*(DEL34^2)): REM int for 3-4
930 W12=10^((U(CORD(2))-U(CORD(1)))/20)
940 W23=10^((U(CORD(3))-U(CORD(1)))/20)
950 W34=10^((U(CORD(4))-U(CORD(1)))/20)
960 SUMW=W12 + W23 + W34
970 NINTW=(W12*NINT12 + W23*NINT23 + W34*NINT34)/SUMW
980 CF=(LMXI+NINTW)*SR*1000/TS
990 MN=(NINT12 + NINT23 + NINT34)/3
1000 SDN=SQR(((NINT12-MN)^2 + (NINT23-MN)^2 + (NINT34-MN)^2)/3)
1020 LPRINT: LPRINT "          ";
1030 LPRINT USING "####.###"; CF;: LPRINT "Hz: ";
1050 LPRINT USING "+##.#"; U(LMXI);: LPRINT "dB.    sdi:";
1070 LPRINT USING "#.####"; SDN;
1080 IF SDN > .05 THEN LPRINT "   ---"; ELSE LPRINT
1090 RETURN

860 IF DEL12>6 THEN NINT12=0: GOTO 880 ' Alternative for Hanning
870 NINT12=P*(.5-.0867*DEL12+.00006*(DEL12^2)+.000088*(DEL12^3))
880 IF DEL23>14 THEN NINT23=P*.5: GOTO 900
890 NINT23=P*(.0382*DEL23+.0000334*(DEL23^2)-.0000148*(DEL23^2))
900 IF DEL34<0 THEN NINT34=P*.5: GOTO 930
910 IF DEL34>56 THEN NINT34=0: GOTO 930
920 NINT34=P*(.502-.0205*DEL34+.0002*(DEL34^2)+1.05E-07*(DEL34^3))
```