Naval Submarine Medical Research Laboratory

NSMRL Report 1221 02 JULY 2001

AN ALGORITHM FOR CALCULATING THE ESSENTIAL BANDWIDTH OF A DISCRETE SPECTRUM AND THE ESSENTIAL DURATION OF A DISCRETE TIME-SERIES

by

Judi A. Lapsley Miller

Released by:
M. D. Curley, CAPT, MSC, USN
Commanding Officer
Naval Submarine Medical Research Laboratory

AN ALGORITHM FOR CALCULATING THE ESSENTIAL BANDWIDTH OF A DISCRETE SPECTRUM AND THE ESSENTIAL DURATION OF A DISCRETE TIME-SERIES

by

Judi A. Lapsley Miller

Naval Submarine Medical Research Laboratory
Report No. 1221

Approved and Released by:

M. D. Curley

M. D. CURLEY, CAPT, MSC, USN
Commanding Officer
NavSubMedRschLab

SUMMARY PAGE

THE PROBLEM

Defining bandwidth and duration for a narrow-band short-duration waveform is difficult and ambiguous due to the acoustical uncertainty principle. Landau and Pollak (1961) defined bandwidth and duration based on the energy content of the waveform. Unlike most other definitions of bandwidth and duration, these definitions are useful both theoretically and practically. A method for the calculation of these quantities for discrete-time (digital or digitized) waveforms does not exist.

THE FINDINGS

A heuristic algorithm is presented for calculating the essential bandwidth and essential duration, for a given proportion of energy constrained, for digital or digitized signals. These definitions for discrete time and frequency are based on those for continuous time and frequency by Landau and Pollak (1961).

THE APPLICATION

This algorithm may be useful anytime bandwidth, duration, or the bandwidth-duration product needs to be estimated. Such applications occur when modeling and measuring the detectability of transient (narrow-band, short-duration) waveforms (e.g., in psychoacoustics and sonar detection).

ADMINISTRATIVE INFORMATION

The views expressed in this report are those of the author(s) and do not reflect the official policy or position of the Department of the Navy, Department of Defense, or the U.S. Government. This report was approved for publication on 02 July 2001, and designated Naval Submarine Medical Research Report No. 1221.

ABSTRACT

A heuristic algorithm is presented for calculating the essential bandwidth and essential duration, for a given proportion of energy constrained, for digital or digitized signals. These definitions for discrete time and frequency are based on those for continuous time and frequency by Landau and Pollak (1961).

An Algorithm for Calculating the Essential Bandwidth of a Discrete Spectrum and the Essential Duration of a Discrete Time-series

Judi A. Lapsley Miller
Naval Submarine Medical Research Laboratory

Abstract

A heuristic algorithm is presented for calculating the essential bandwidth and essential duration, for a given proportion of energy constrained, for digital or digitized signals. These definitions for discrete time and frequency are based on those for continuous time and frequency by Landau and Pollak (1961).

Appropriate definitions of bandwidth (W) and duration (T) are needed to define and generate high quality acoustic waveforms, suitable for psychoacoustic experiments. This is particularly important if bandwidth, duration, or the bandwidth-duration product (WT) is an experimental parameter. Typically, the three-dB bandwidth, equivalent rectangular bandwidth (ERB), and equivalent rectangular duration (ERD) are used. These definitions are lacking for both theoretical and practical reasons.

Theoretically, the acoustical uncertainty principle, resulting from Fourier waveform representation, shows that a waveform cannot be simultaneously time and band limited (Landau & Pollak, 1961). Therefore, any definition of bandwidth and duration is somewhat arbitrary, because energy always exists outside what is considered to be the bandwidth or duration of the waveform. The uncertainty principle is specified by the bandwidth-duration product, and specific cases have been derived for a variety of bandwidth and duration definitions (for examples, see Slepian, 1983; Vakman, 1968).

The definitions of bandwidth and duration used in theoretical work have been criticized for not being relevant to practical, acoustical, situations (Landau & Pollak, 1961).

This article is based on an appendix from my doctoral dissertation, which was supervised by John Whitmore, at Victoria University of Wellington, New Zealand. I would like to thank Linton Miller, Vit Drga, and John Whitmore for their helpful comments.

Correspondence concerning this article should be addressed to Judi Lapsley Miller, NSMRL Box 900, Subase NLON, Groton, CT 06349-5900, USA. Email jmiller@nsmrl.navy.mil or judi@psychophysics.org.

Further details about my research are available on the Internet at www.psychophysics.org.

Also, the three-dB bandwidth, ERB, or ERD have not been used to derive a specific case of the uncertainty principle. Therefore, relating experimental results to theory may be difficult.

Three-dB bandwidth, ERB, and ERD estimates of digitized waveforms also tend to be variable. This is because these bandwidth and duration definitions rely on measurements made at single points (e.g., the power at the three-dB points, and the power at the center frequency). Even with spectral or signal averaging, the bandwidth and duration estimates may not be very stable.

What is needed when psychoacoustic theories are being evaluated experimentally, are definitions of bandwidth and duration that are useful both theoretically and practically.

Continuous Essential Bandwidth and Duration

Landau and Pollak (1961) suggested that a good definition of W and T would describe the behavior of a time-series $f(t)$ in a given finite time interval and likewise the behavior of a spectrum $F(\omega)$ in a given finite frequency band. They argued that although waveforms cannot be bounded in both domains, bounds that essentially constrain the waveforms can still be specified. One way to specify these bounds is to look at a waveform's *energy* content and *energy* spread in both the time and frequency domains.[1]

Bandwidth and duration can be specified by calculating the proportion of energy constrained between two bounds relative to the total waveform energy content. This *essential bandwidth*,[2] Ω , which encloses a proportion, β^2, of the total waveform energy is defined as

$$\beta^2 = \frac{\int_{-\Omega}^{\Omega} |A(\omega)|^2 \,\mathrm{d}\omega}{\int_{-\infty}^{\infty} |A(\omega)|^2 \,\mathrm{d}\omega} \tag{1}$$

where $0 \leqslant \beta^2 \leqslant 1$, $|A(\omega)|^2$ is the energy spectrum, and ω is frequency in radians. Similarly, the *essential duration*, T, which encloses a proportion, α^2, of the total waveform energy is defined as

$$\alpha^2 = \frac{\int_{-\mathrm{T}/2}^{\mathrm{T}/2} |f(t)|^2 \,\mathrm{d}t}{\int_{-\infty}^{\infty} |f(t)|^2 \,\mathrm{d}t} \tag{2}$$

where $0 \leqslant \alpha^2 \leqslant 1$, $f(t)$ is an acoustic waveform, and t is time in seconds.

Discrete Essential Bandwidth and Duration

For practical use, with discretely represented waveforms, the continuous essential bandwidth and duration definitions must be converted into their discrete frequency and time equivalents.

There appears to be no definition of either the essential bandwidth for discrete frequency or the essential duration for discrete time in the literature. A heuristic algorithm

[1] Kharkevich (1960) has also suggested these definitions, and Chalk (1950) used a similar definition to derive the optimum pulse-shape for pulse communication.

[2] Landau and Pollak (1961) never named their new bandwidth and duration measures. I've named them "essential bandwidth" and "essential duration."

was therefore developed, based on the definitions of essential bandwidth and essential duration definitions, for continuous frequency and time, from Equations (1) and (2), respectively (pseudo-code is presented in the Appendix).

It is assumed that the spectrum or (squared) time-series is unimodal and concentrated in frequency and time. According to L. Miller (personal communication, December, 1998) the algorithm will result in the minimum bandwidth or duration, for the specified energy constrained, if these assumptions hold. Small local maxima may mean the minimum bandwidth or duration has not been found, but for large α^2 or β^2, this is unlikely to greatly affect the final estimate. The algorithm may not supply a sensible result, however, if the spectrum or time-series is not unimodal, such as a U-shaped spectrum.

The steps for calculating the essential bandwidth are:

1. Decide on the proportion of energy to be constrained, for example, 95%.

2. Calculate the total energy in the energy spectrum by summing the discrete values in the spectrum.

3. Find the frequency index with the largest energy value. Set a lower marker and an upper marker to the index. Calculate the proportion of energy in this element relative to the total energy.

4. Compare the current proportion of energy to the desired proportion of energy. If there is more than required, go to step 7.

5. If more energy is required, check the element right of the upper marker and left of the lower marker, choose the largest, and add it to the current proportion of energy. If the largest element is to the left, decrement the lower marker, otherwise if the largest element is to the right, increment the upper marker.

6. Repeat steps 4 and 5 until the energy equals or just exceeds that required.

7. The bandwidth is the width between the lower index and the upper index (or proportion of the current lower or upper index, using linear interpolation) which results in the energy required.

The algorithm for the discrete essential duration is virtually identical to that for the essential bandwidth, except that the discrete values in the time-series are squared first. To avoid confusion, the algorithms for bandwidth and duration are presented separately.

The algorithm presented here was designed to be expository rather than efficient. In implementation, efficiencies may be gained by (a) combining the calculation of the maximum energy index and the total energy, (b) minimizing divisions, and (c) combining the functions for calculating essential bandwidth and duration, with an additional parameter to indicate whether the values should first be squared.

Summary

Besides providing a link with theoretical results related to the acoustical uncertainty principle (such as energy detection of small WT Gaussian noise waveforms), the discrete essential bandwidth and duration provide a practical benefit. Both are less prone to measurement error compared with the three-dB bandwidth, ERB, and ERD, because the summations smooth random error associated with the individual elements.

These definitions were successfully used in a psychoacoustic experiment to study the detectability of small-WT Gaussian noise (Lapsley Miller, 1999).[3] In this study, the bandwidth, duration, and bandwidth-duration product of Gaussian noise transients, for W=2.5–160 Hz, T=400–6.25 ms, and WT=1, 2, and 4, were specified using a proportion of energy constrained in each domain of 92.4%.

Because of the theoretical and practical benefits associated with the essential bandwidth and duration, psychophysicists should consider using them to define stimulus properties and for digital measurement of real waveforms.

References

Chalk, J. H. H. (1950). The optimum pulse-shape for pulse communication. *Proc. IEE*, *87*, 88-92.

Kharkevich, A. A. (1960). *Spectra and analysis.* New York: Consultants Bureau Enterprises.

Landau, H. J., & Pollak, H. O. (1961). Prolate spheroidal wave functions, Fourier analysis and uncertainty–II. *Bell Sys. Tech. J.*, *40*, 65-84.

Lapsley Miller, J. A. (1999). *The role of the bandwidth-duration product WT in the detectability of diotic signals.* Doctoral dissertation, Victoria University of Wellington, New Zealand. (Available electronically on the web at www.psychophysics.org.)

Ronken, D. A. (1970). Some effects of bandwidth-duration constraints on frequency discrimination. *J. Acoust. Soc. Am.*, *49*, 1232-1242.

Slepian, D. (1983). Some comments on Fourier analysis, uncertainty and modeling. *SIAM Review*, *25*, 379-393.

Vakman, D. E. (1968). *Sophisticated signals and the uncertainty principle in radar.* New York: Springer-Verlag.

[3]Ronken (1970) also used essential bandwidth and duration to study frequency discrimination, but specified his stimulus parameters using the continuous definitions.

Appendix

The Algorithm in Pseudo-code for Calculating the Discrete Essential Bandwidth and Duration

input

waveform : a real-valued array representing a time-series.

wavespectrum : a real valued array representing an energy spectrum.

wavesize : the number of elements in the waveform.

spectrumsize : the number of elements in the spectrum.

timebase : the time-series resolution in seconds.

freqbase : the spectrum resolution in hertz.

α^2 : proportion of energy constrained in the time domain.

β^2 : proportion of energy constrained in the frequency domain.

tolerance : an arbitrary cutoff to determine how close the current proportion of energy is to the desired bound.

output

essential_duration: a real value representing duration in seconds.

essential_bandwidth: a real value representing bandwidth in Hz.

energy_time: a real value representing the total energy in the time domain.

energy_freq: a real value representing the total energy in the frequency domain.

find_maximum_energy_time: an integer value representing the index of the maximum energy element in the time domain.

find_maximum_energy_freq: an integer value representing the index of the maximum energy element in the frequency domain.

```
function essential_bandwidth (wavespectrum, β^2, spectrumsize, freqbase, tolerance)
(* Calculates the essential bandwidth for the discrete spectrum, wavespectrum, with
spectrumsize elements, and with resolution of freqbase Hz, for a given proportion of
energy constrained, β^2. Includes the DC component at 0 Hz. *)
  prop_energy, p, total_energy : real
  max_e_idx, lower_idx, upper_idx : integer
  total_energy ← energy_freq (wavespectrum, spectrumsize)
  max_e_idx ← find_maximum_energy_freq (wavespectrum, spectrumsize)
  lower_idx ← max_e_idx
  upper_idx ← max_e_idx
  p ← wavespectrum[max_e_idx] / total_energy
  prop_energy ← p
  while (prop_energy + tolerance) < β^2 do
    if (lower_idx > 0) and (upper_idx < spectrumsize) then
      (* Sum bin with most energy from either side of max_e_idx. *)
      if wavespectrum[lower_idx - 1] > wavespectrum[upper_idx + 1] then
```

```
        lower_idx ← lower_idx - 1
        p ← wavespectrum[lower_idx] / total_energy
      else
        upper_idx ← upper_idx + 1
        p ← wavespectrum[upper_idx] / total_energy
      fi
    else if (lower_idx = 0) and (upper_idx < spectrumsize) then
      (* If the lowest index is reached then sum only from the upper index. *)
      upper_idx ← upper_idx + 1
      p ← wavespectrum[upper_idx] / total_energy
    else if (lower_idx > 0) and (upper_idx = spectrumsize) then
      (* If the highest index is reached then sum only from the lower index. *)
      lower_idx ← lower_idx - 1
      p ← wavespectrum[lower_idx] / total_energy
    else if (lower_idx = 0) and (upper_idx = spectrumsize) then
      return spectrumsize × freqbase
    fi
    prop_energy ← prop_energy + p
  endwhile
  (* Use linear interpolation to calculate the bandwidth. *)
  return ((β^2 - (prop_energy - p))/p + (upper_idx - lower_idx)) × freqbase
end

function essential_duration (waveform, α^2, wavesize, timebase, tolerance)
(* Calculates the essential duration for the discrete time-series, waveform, with wavesize
elements, and with resolution of timebase seconds, for a given proportion of energy
constrained α^2. *)
  prop_energy, p, total_energy : real
  max_e_idx, lower_idx, upper_idx : integer
  total_energy ← energy_time (waveform, wavesize)
  max_e_idx ← find_maximum_energy_time (waveform, wavesize)
  lower_idx ← max_e_idx
  upper_idx ← max_e_idx
  p ← sqr(waveform[max_e_idx]) / total_energy
  prop_energy ← p
  while (prop_energy + tolerance) < α^2 do
    if (lower_idx > 1) and (upper_idx < wavesize) then
      (* Sum bin with most energy from either side of max_e_idx. *)
      if sqr(waveform[lower_idx - 1]) > sqr(waveform[upper_idx + 1]) then
        lower_idx ← lower_idx - 1
        p ← sqr(waveform[lower_idx]) / total_energy
```

```
        else
          upper_idx ← upper_idx + 1
          p ← sqr(waveform[upper_idx]) / total_energy
        fi
      else if (lower_idx = 1) and (upper_idx < wavesize) then
        (* If the lowest index is reached then sum only from the upper index. *)
        upper_idx ← upper_idx + 1
        p ← sqr(waveform[upper_idx]) / total_energy
      else if (lower_idx > 1) and (upper_idx = wavesize) then
        (* If the highest index is reached then sum only from the lower index. *)
        lower_idx ← lower_idx - 1
        p ← sqr(waveform[lower_idx]) / total_energy
      else if (lower_idx = 1) and (upper_idx = wavesize) then
        return wavesize × timebase
      fi
      prop_energy ← prop_energy + p
  endwhile
  (* Use linear interpolation to calculate the duration. *)
  return ((α^2 - (prop_energy - p))/p + (upper_idx - lower_idx)) × timebase
end

function energy_freq (wavespectrum, spectrumsize)
(* Input is a discrete energy spectrum, wavespectrum, with spectrumsize elements.
Includes the DC component at 0 Hz. *)
  energy : real
  energy ← 0
  for i ← 0 to spectrumsize do
    energy ← energy + wavespectrum[i]
  return energy
end

function energy_time (waveform, wavesize)
(* Input is a discrete time-series, waveform, with wavesize elements. *)
  energy : real
  energy ← 0
  for i ← 1 to wavesize do
    energy ← energy + sqr(waveform[i])
  return energy
end

function find_maximum_energy_freq (wavespectrum, spectrumsize)
(* Input is a discrete energy spectrum, wavespectrum, with spectrumsize elements. *)
```

```
  max_e_idx : integer
  max_e_idx ← 0
  for i ← 1 to spectrumsize do
    if wavespectrum[i] > wavespectrum[max_e_idx] then
      max_e_idx ← i
    fi
  return max_e_idx
end

function find_maximum_energy_time (waveform, wavesize)
(* Input is a discrete time-series, waveform, with wavesize elements. *)
  max_e_idx : integer
  max_e_idx ← 1
  for i ← 2 to wavesize do
    if sqr(waveform[i]) > sqr(waveform[max_e_idx]) then
      max_e_idx ← i
    fi
  return max_e_idx
end
```