Band-Limited Pulse Generator

Version 1.3: Mar 17, 2008 9:31 pm GMT-5

Ed Doering

Abstract

Subtractive synthesis techniques often require a wideband excitation source such as a pulse train to drive a time-varying digital filter. Traditional rectangular pulses have theoretically infinite bandwidth, and therefore always introduce aliasing noise into the input signal. A band-limited pulse (BLP) source is free of aliasing problems, and is more suitable for subtractive synthesis algorithms. The mathematics of the band-limited pulse is presented, and a LabVIEW VI is developed to implement the BLP source. An audio demonstration is included.

Band-Limited Pulse Generator

	This module refers to LabVIEW, a software develo programming language. Please see the LabVIEW Qu umentation that will help you:
• Apply LabVIEW to Audio Signal Processing	
• Get started with LabVIEW	
• Obtain a fully-functional evaluation edition of LabVIEW	

1 Introduction

Subtractive synthesis techniques apply a filter (usually time-varying) to a wideband excitation source such as noise or a pulse train. The filter shapes the wideband spectrum into the desired spectrum. The excitation/filter technique describes the sound-producing mechanism of many types of physical instruments as well as the human voice, making subtractive synthesis an attractive method for **physical modeling** of real instruments.

A **pulse train**, a repetitive series of pulses, provides an excitation source that has a perceptible pitch, so in a sense the excitation spectrum is "pre-shaped" before applying it to a filter. Many types of musical instruments use some sort of pulse train as an excitation,

Figure 1: [video] Band-limited pulse generator in LabVIEW using "Tones and Noise" built-in subVI

notably wind instruments such as brass (e.g., trumpet, trombone, and tuba) and woodwinds (e.g., clarinet, saxophone, oboe, and bassoon). Likewise, the human voice begins as a series of pulses produced by vocal cord vibrations, which can be considered the "excitation signal" to the vocal and nasal tract that acts as a resonant cavity to amplify and filter the "signal."

Traditional rectangular pulse shapes have significant spectral energy contained in harmonics that extend beyond the **folding frequency** (half of the sampling frequency). These harmonics are subject to **aliasing**, and are "folded back" into the **principal alias**, i.e., the spectrum between 0 and $f_s/2$. The aliased harmonics are distinctly audible as high-frequency tones that, since undesired, qualify as noise.

The **band-limited pulse**, however, is free of aliasing problems because its maximum harmonic can be chosen to be below the folding frequency. In this module the mathematics of the band-limited pulse are developed, and a band-limited pulse generator is implemented in LabVIEW.

2 Mathematical Development of the Band-Limited Pulse

By definition, a **band-limited pulse** has zero spectral energy beyond some determined frequency. You can use a truncated Fourier series to create a series of harmonics, or sinusoids, as in Equation 1:

$$x(t) = \sum_{k=1}^{N} \sin(2\pi k f_0 t) \tag{1}$$

The Figure 1 screencast video shows how to implement Equation 1 in LabVIEW by introducing the "Tones and Noise" built-in subVI that is part of the "Signal Processing" palette. The video includes a demonstration that relates the time-domain pulse shape, spectral behavior, and audible sound of the band-limited pulse.

 Download the finished VI from the video: blp_demo.vi[2]. This VI requires installation of the TripleDisplay[3] front-panel indicator.

The truncated Fourier series approach works fine for off-line or batch-mode signal processing. However, in a real-time application the computational cost of generating individual sinusoids becomes prohibitive, especially when a fairly dense spectrum is required (for example, 50 sinusoids).

A closed-form version of the truncated Fourier series equation is presented in Equation 2 (refer to Moore in "References" section below):

$$x(t) = \sum_{k=1}^{N} \sin(k\theta) = \sin\left[(N+1)\frac{\theta}{2}\right] \frac{\sin\left(N\frac{\theta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} \tag{2}$$

where

[2] blp_demo.vi
[3] http://cnx.rice.edu/content/m15430/latest/

$\theta = 2\pi f_0 t$. The closed-form version of the summation requires only three sinusoidal oscillators yet can produce an arbitrary number of sinusoidal components.

Implementing Equation 2 contains one significant challenge, however. Note the ratio of two sinusoids on the far right of the equation. The denominator sinusoid periodically passes through zero, leading to a divide-by-zero error. However, because the numerator sinusoid operates at a frequency that is N times higher, the numerator sinusoid also approaches zero whenever the lower-frequency denominator sinusoid approaches zero. This "0/0" condition converges to either N or -N; the sign can be inferred by looking at adjacent samples.

3 References

- Moore, F.R., "Elements of Computer Music," Prentice-Hall, 1990, ISBN 0-13-252552-6.