Niels Moseley's bloggy bits

Mathematics, Signal processing, Electronics and RF/Radio.

Home
About

Search …

Recent Posts

Noise shaper system equations
Micro Pascal: from code to p-code
Writing software for the HD6309 computer: Micro Pascal
HD6309 computer design files
HD 6309 Bootloader v1.1

Archives

May 2017
April 2017
March 2017
January 2017
July 2016
June 2016
July 2015
February 2015
April 2014
May 2013
October 2012
September 2012

Categories

A simple sin(x) approximation

Introduction

When implementing digital oscillators or filters, a simple way of calculating $\sin(2\pi \cdot x)$ for $x = [0..1)$ is needed. Efficiency, in terms of the number of CPU cycles or hardware complexity, is favoured above accuracy. Several methods are known that produce good approximations to $\sin(\pi \cdot x)$. These methods can be divided into three categories: table-lookup algorithms, approximation functions and recursive resonator algorithms.

Table-lookup algorithms are cycle efficient and their accuracy can be easily increased by increasing the number of table entries. Accuracy can also be increased by introducing interpolation between table entries.

In some memory limited situations, such as small microcontroller and DSP targets, a table is undesirable and an approximation based on functions is preferred.

Here, I will show how to approximate $\sin(2 \cdot \pi \cdot x)$ based on a simple polynomial.

Approximating sin(2*pi*x)

The well known Taylor series seems a good candidate for obtaining a polynomial that approximates $\sin(2\pi \cdot x)$ well. However, the approximation gets worse the further away the function argument gets from the point from which the series was derived. This generates a discontinuity when $\sin(2\pi \cdot x)$ "wraps around". To avoid such discontinuities, the approximation must produce the same value for $x = 0$ and $x = 1$.

An interesting candidate is the polynomial $-16x^2 + 8x$, which approximates $\sin(2\pi \cdot x)$ quite well for $x = [0..\frac{1}{2}]$. The other half of the sinusoid is simply generated by mirroring the polynomial and adjusting for the offset, resulting in $16x^2 - 24x + 8$.

How good is it?

The approximation is accurate to within +/- 6% of the total amplitude as is shown by the graphs below:

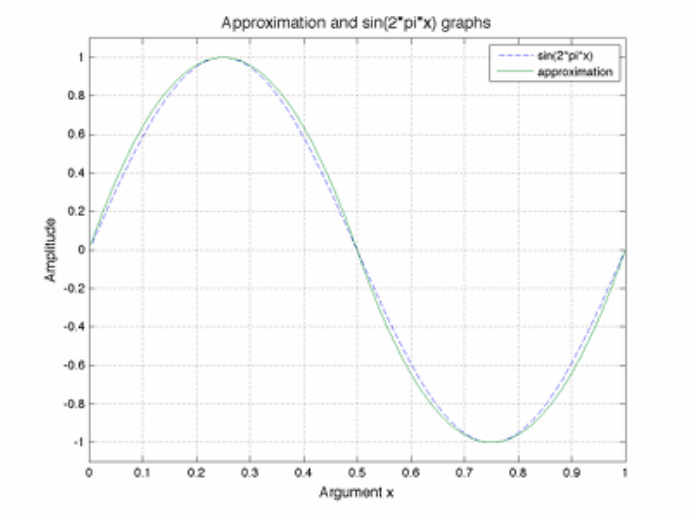
Algorithms
Embedded
Ham radio
Micro Pascal
Retrochallenge
Signal processing
Uncategorized

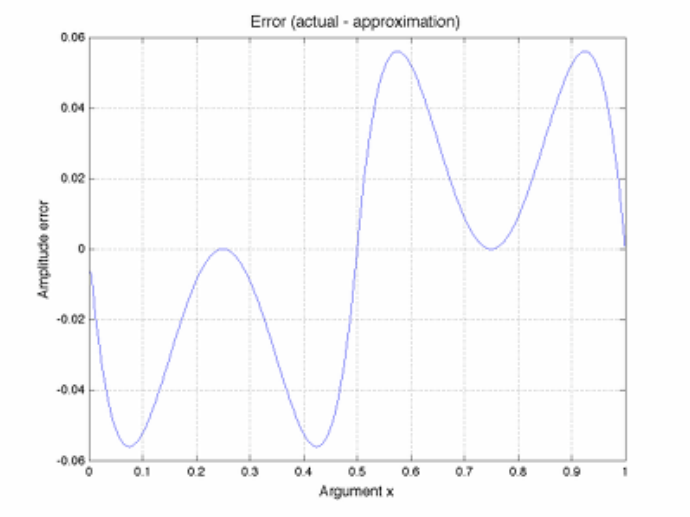
Blogroll

AA7EE – Dave Richards
EEVBLOG – David Jones
From orbit – Alan Garfield
LFT – Linus Akesson
PA3FWM – Pieter-Tjerk de Boer
The ryg blog – Fabian Giesen
VK2ZAY – Alan Yates
VK3YE – Peter Parker

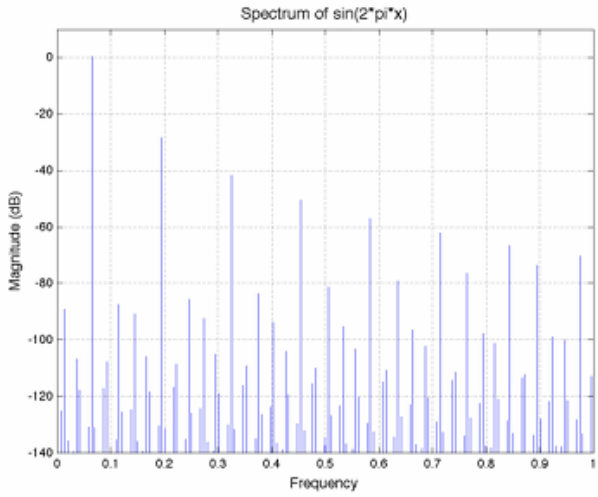
Meta

Register
Log in
Entries RSS
Comments RSS
WordPress.com





In sound synthesis and radio applications it is desirable to have low spurious tones. The following graph shows the spectrum of the approximated sinusoidal wave. It was generated using an 65536-point FFT and 2129 sine wave periods.



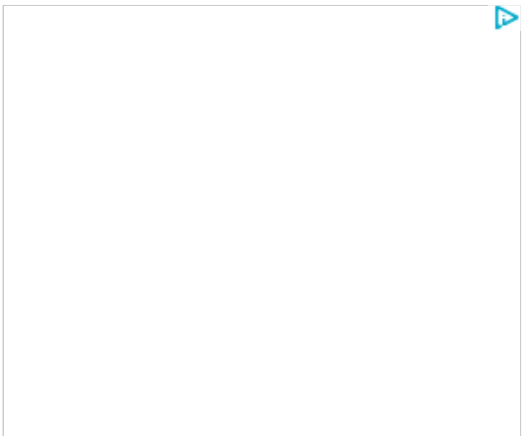
The spectrum is free of all even harmonics as the approximation is point symmetrical and the “grass” is due to high frequency components aliasing to lower frequencies. The first major spurious response is the third harmonic, which is found at -28.6 dBc.

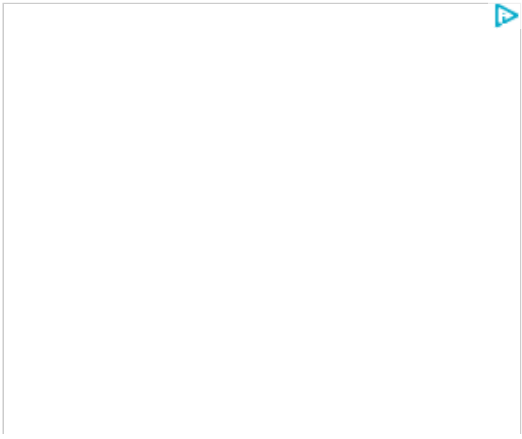
Show me the code!

```
/*
  Fast sin(2*pi*x) approximation.
  The caller must make sure that the argument lies
  within [0..1).

  Note: this can be optimized further by using Horne
  rule for evaluating polynomials, see:

  http://en.wikipedia.org/wiki/Horner%27s_method

  Omitted here for clarity.
*/

float fast_sin(float x)
{
  if (x < 0.5f)
  {
    return -16.0f*x*x + 8.0f*x;
  }
  else
  {
    return 16.0f*x*x - 24.0f*x + 8.0f;
  }
};
```

Conclusion

This may not be the world's best sin approximation, but it sure is simple. Next time we'll look at a more accurate polynomial-based sin approximation.





Share this:

Twitter Facebook

Like

Be the first to like this.

Related

Sin(x) using Taylor series: better than expected
In "Algorithms"

A better sin(x) approximation
In "Algorithms"

Sin/cos generation using table lookup and interpolation.
In "Algorithms"

Posted on September 18, 2012 by namoseley. This entry was posted in Algorithms and tagged Oscillator. Bookmark the permalink.

A better sin(x) approximation →

Leave a Reply

Enter your comment here...