Comparison of Direct Digital Synthesis of Integer and Floating-Point Decimal Generation

EEE 4510 Digital Signal Processing

Dranishnikov, Peter
4/8/2019

Introduction:

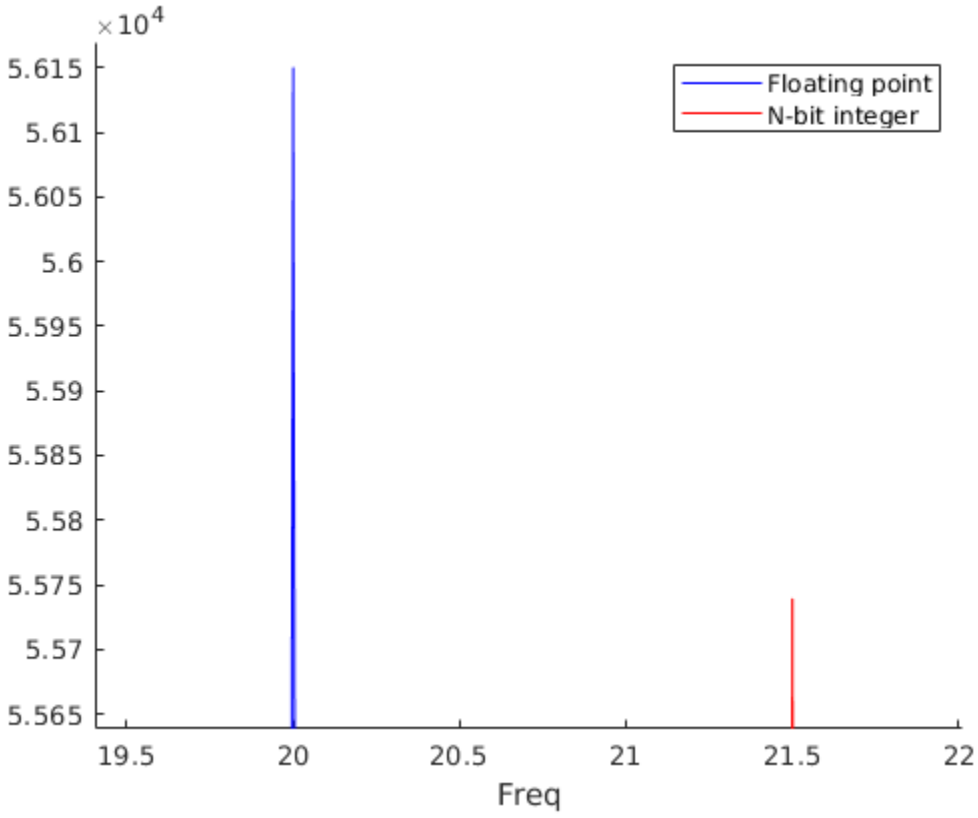
Direct Digital Synthesis (DDS) is an inexpensive way to generate signals with only simple, digital hardware or software. However, DDS has some significant drawbacks for certain applications which require more "pure" signals than the ones generated by DDS. Since DDS outputs a signal of only zeros and ones (effectively a square wave), the harmonics produced by such a wave are much greater than that of a sine wave output of the same frequency. There are several techniques to suppress such "undesired" frequencies, each with their own drawback. In this project, the effect between resulting frequencies from the desired frequency for different measurement counts and target frequencies is described. The images shown in later pages are in increasing target frequency (F_o) in alternating between 10- and 20-bit resolutions (N). The MATLAB source code executed is in Appendix A.

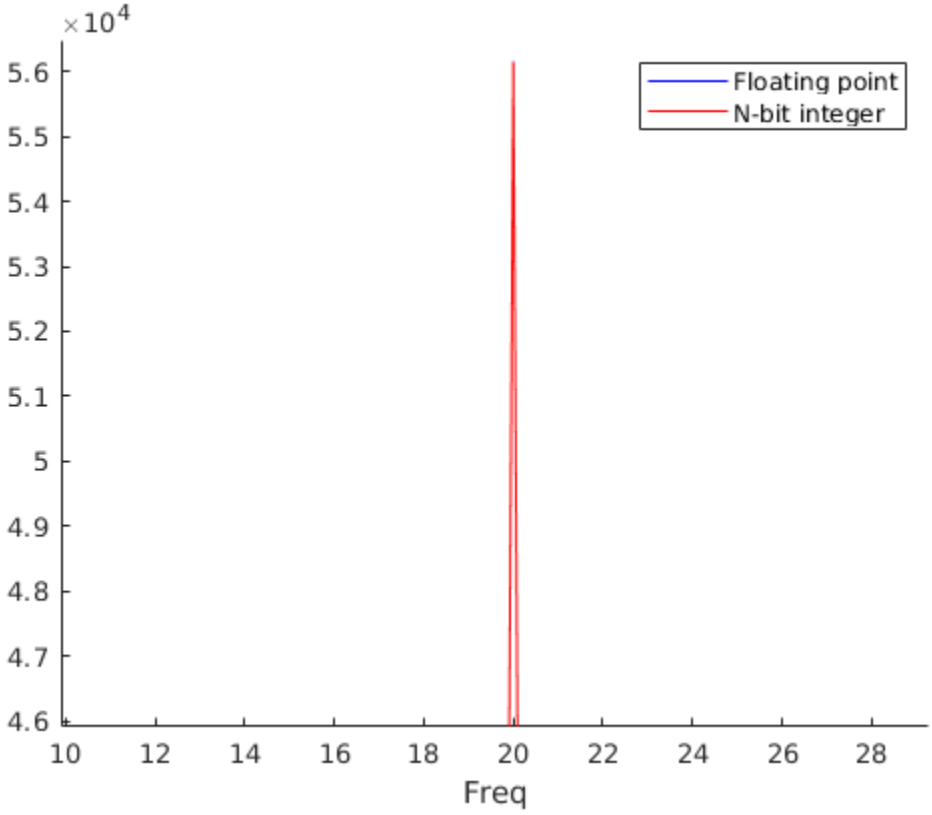
Results:

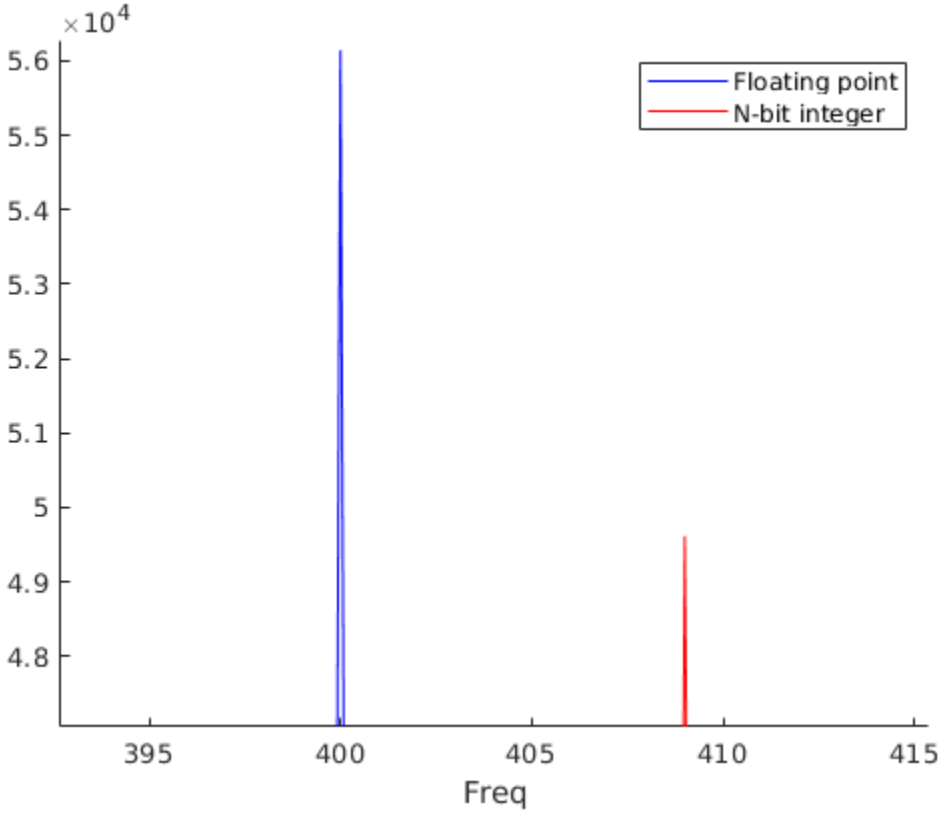
Original Frequency	Floating point frequency	Fixed point N = 10	Fixed point N = 20
20	20	21.5	20
400	400	409	400
1800	1800	1809	1800
5500	5500	5491	5500
10000	10000	9992	10001

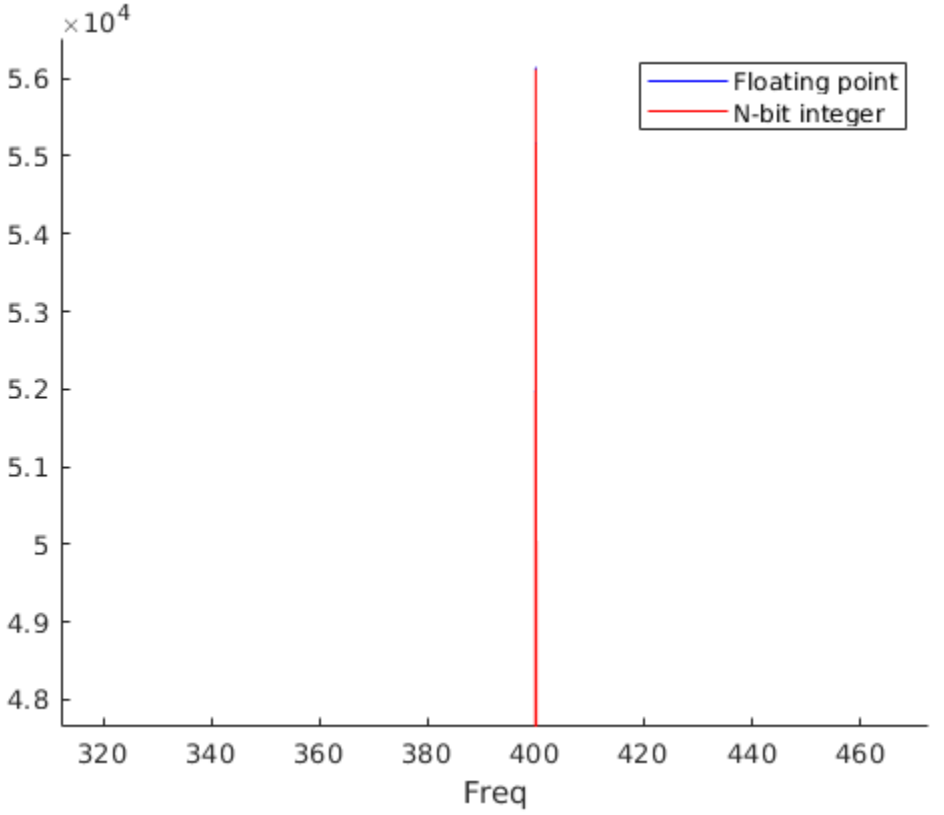
Discussion:

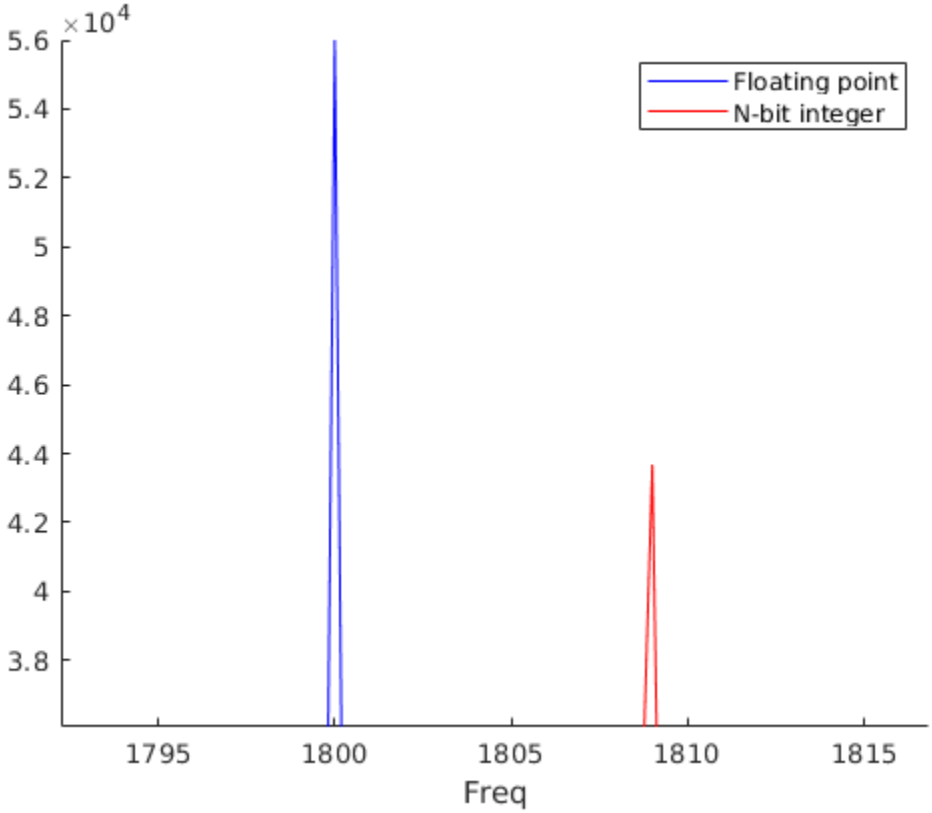
For N = 10 measurements, there is a significant deviation of the resulting frequency for the fixed-point DDS generator equation, up to 9 Hz for higher frequencies compared to the floating-point DDS generator. For N = 20 measurements, the frequency deviation compared to the floating-point is negligible, but the amplitude differs for higher frequencies.

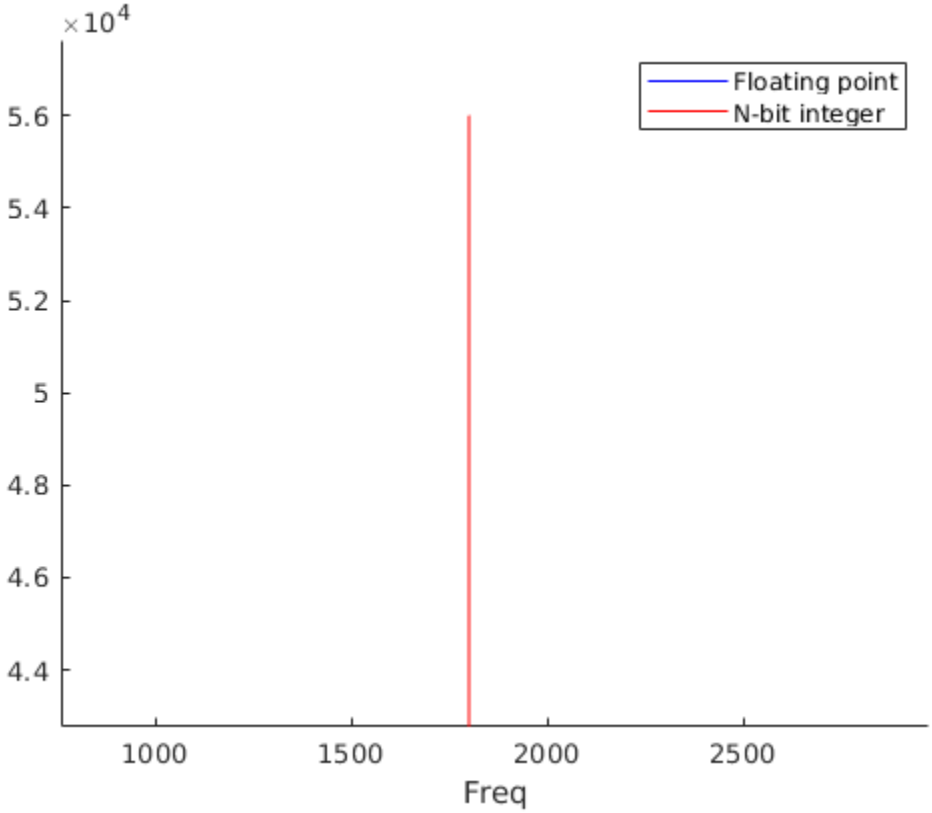

×10^4
5.615
5.61
5.605
5.6
5.595
5.59
5.585
5.58
5.575
5.57
5.565
19.5
20
20.5
21
21.5
22
Freq
Floating point
N-bit integer

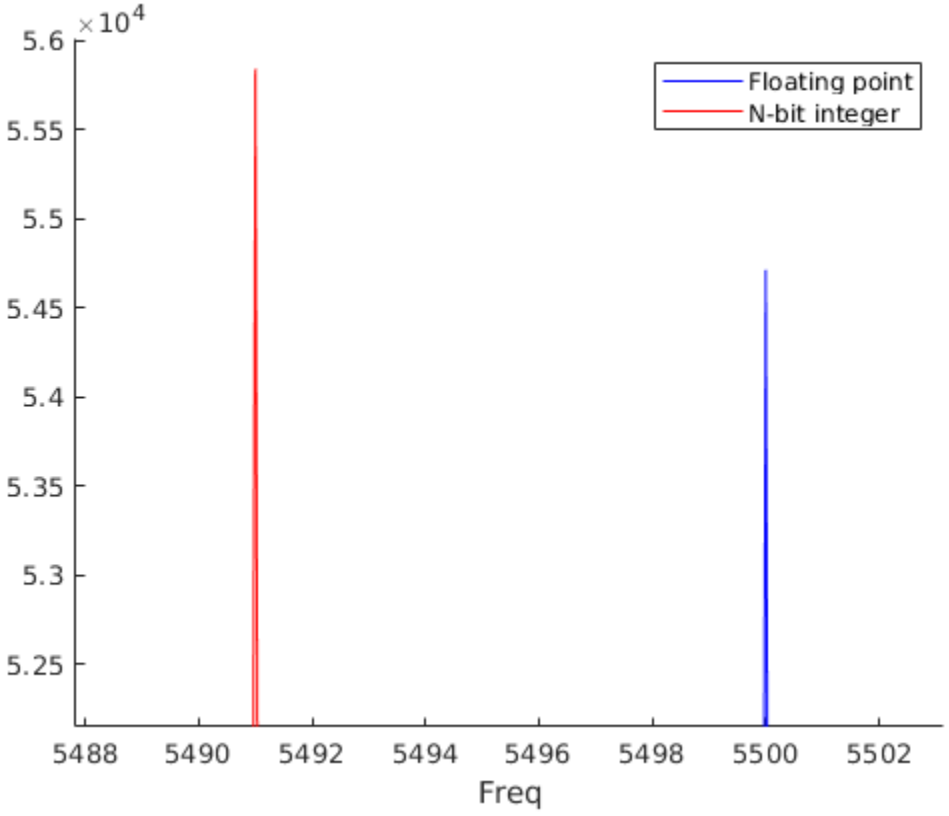

×10^4
5.6
5.5
5.4
5.3
5.2
5.1
5
4.9
4.8
4.7
4.6
10
12
14
16
18
20
22
24
26
28
Freq
Floating point
N-bit integer

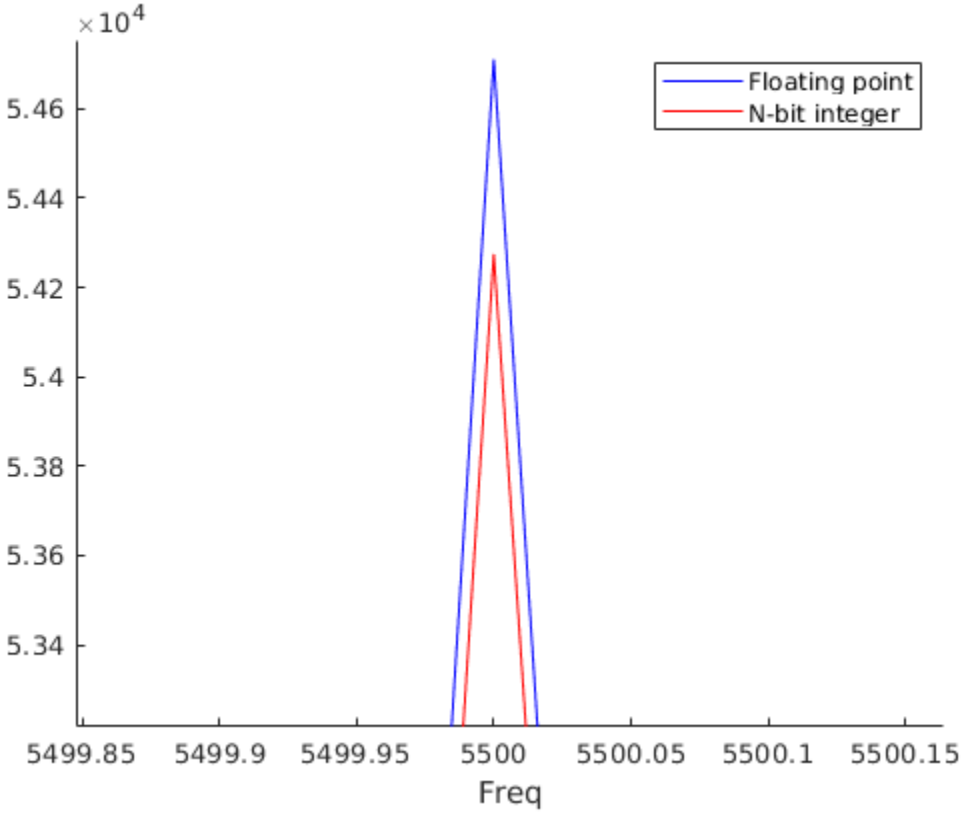

×10^4
5.6
5.5
5.4
5.3
5.2
5.1
5
4.9
4.8
395
400
405
410
415
Freq
Floating point
N-bit integer

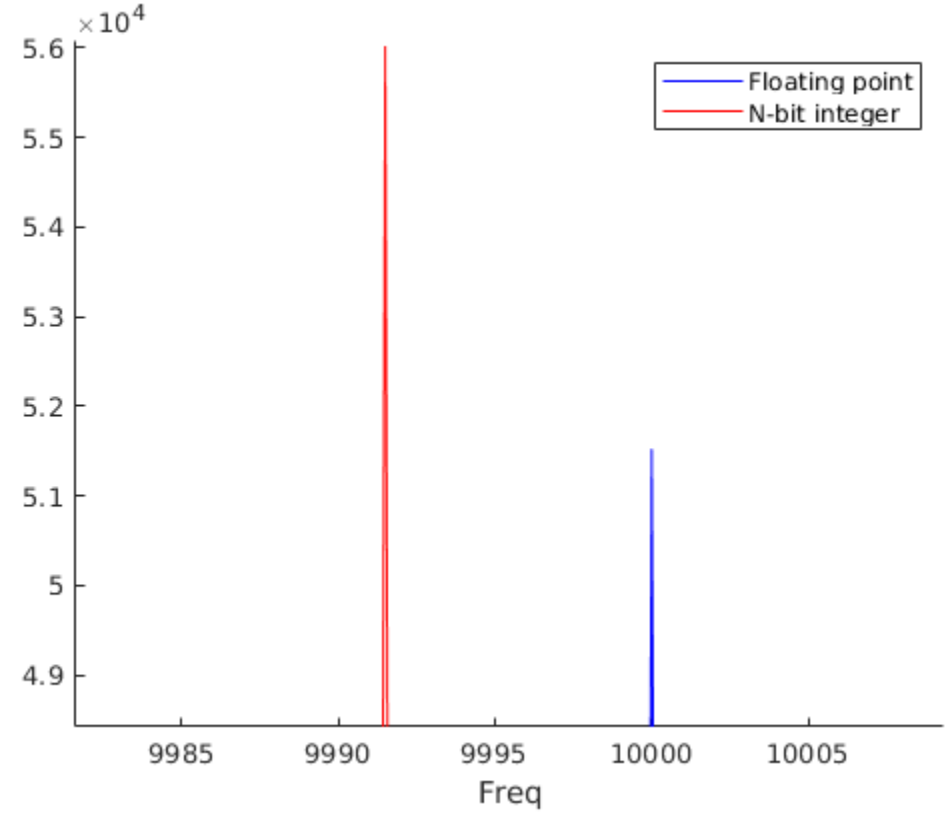

×10^4
5.6
5.5
5.4
5.3
5.2
5.1
5
4.9
4.8
320
340
360
380
400
420
440
460
Freq
Floating point
N-bit integer

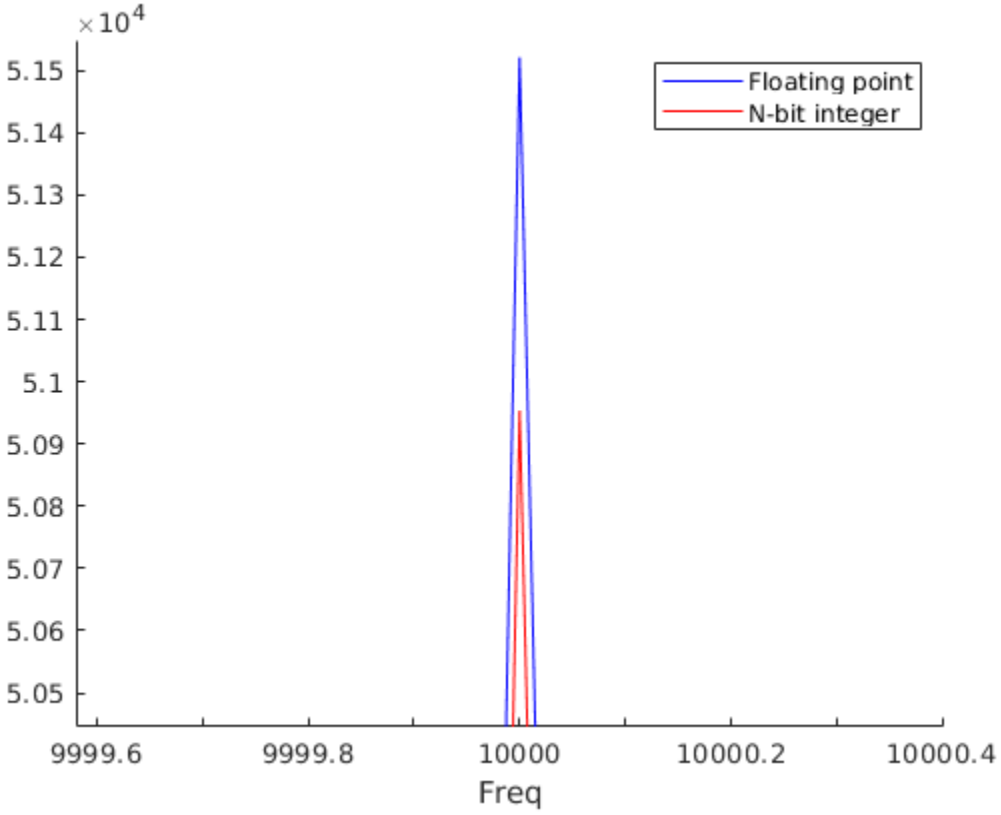

×10^4
5.6
5.4
5.2
5
4.8
4.6
4.4
4.2
4
3.8
1795
1800
1805
1810
1815
Freq
Floating point
N-bit integer


×10^4
5.6
5.4
5.2
5
4.8
4.6
4.4
1000
1500
2000
2500
Freq
Floating point
N-bit integer


×10^4
5.6
5.55
5.5
5.45
5.4
5.35
5.3
5.25
5488
5490
5492
5494
5496
5498
5500
5502
Freq
Floating point
N-bit integer


×10^4
5.46
5.44
5.42
5.4
5.38
5.36
5.34
5499.85
5499.9
5499.95
5500
5500.05
5500.1
5500.15
Freq
Floating point
N-bit integer


×10^4
5.6
5.5
5.4
5.3
5.2
5.1
5
4.9
9985
9990
9995
10000
10005
Freq
Floating point
N-bit integer


×10^4
5.15
5.14
5.13
5.12
5.11
5.1
5.09
5.08
5.07
5.06
5.05
9999.6
9999.8
10000
10000.2
10000.4
Freq
Floating point
N-bit integer

Appendix A: MATLAB Source code

```
fo_vals = [20, 400, 1800, 5500, 10000];
n_vals = [10, 20];
for i = fo_vals
    % DDS project
    fs=44100;
    Ts=1/fs;
    signal_duration=2;
    kval=[0:signal_duration*fs];
    tval=kval*Ts;
    %fo=20001.8;
    fo=i; %20, 400, 1800, 5500, 10000
    % table
    ta = [-1 1];
    % alpha = fo/(fs/2)
    alpha = fo/(fs/2);
    % (-1)^POWER formula
    xdds=(-1).^round(alpha*kval);

    % Table/array based formula
    xdds1=ta(1 + mod(round(alpha*kval),2));

    % Integer arithmetic based formula
    % alpha = M/2^N
    for j = n_vals
        N=j;
        M=round(2^N*alpha);
        xdds2=ta(1 + mod(round((M/2^N)*kval),2));
```

```
        %sound(xdds, fs);


        figure; clf; hold on;
        xdds1=xdds1-mean(xdds1);
        plot(linspace(-
fs/2,fs/2,length(xdds1)),abs(fftshift(fft(xdds1))),'b-')
        xdds2=xdds2-mean(xdds2);
        plot(linspace(-
fs/2,fs/2,length(xdds2)),abs(fftshift(fft(xdds2))),'r-')
        xlabel('Freq');
        legend('Floating point','N-bit integer');
    end
end
```