How Does OverSampling work?

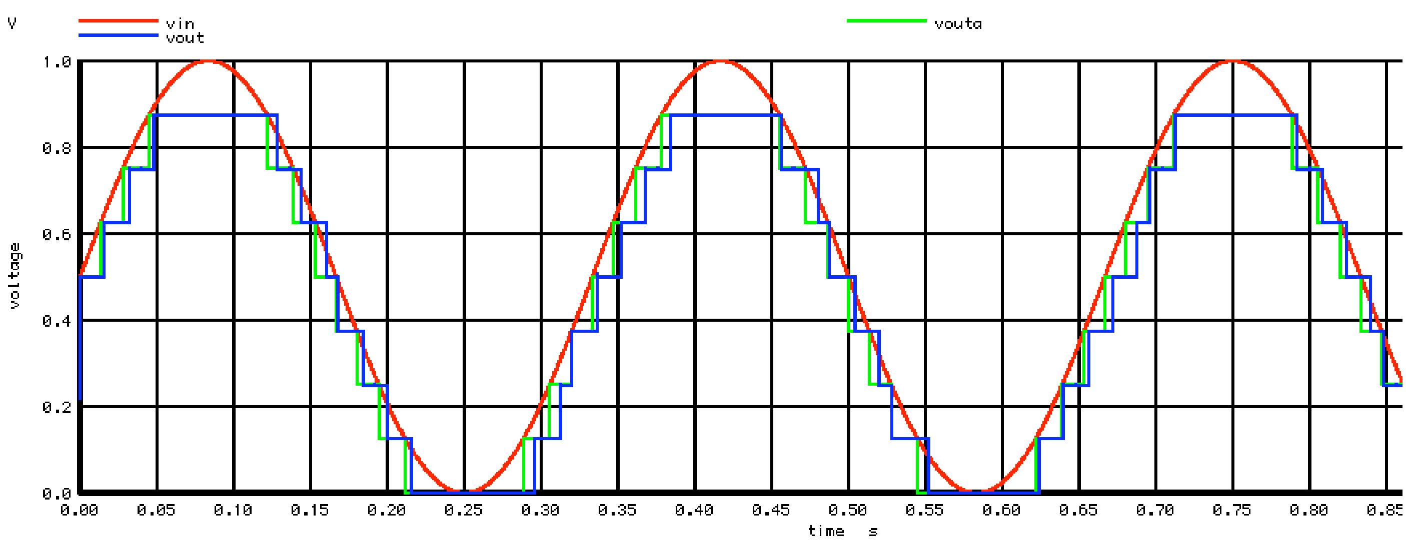
```
*        VIN
*
*      _|_______________|  |_________|           
*     / _|_ \         TH0  _|_   |   TH1     _|_      TH0
*    //   \ \         |  _/   \  |    _/   \             ______________
*    \  \ //          |\ |\___/  |\  |\___/   |\        |          |   \
*     \____/        _| \_|  |____|_|  \_|  |_______|  \_|          |    \
*       _|_           / |  OFF0     /  |  OFF1     /  |    ______  DAC  \__|VOUTA|
*       ///          |/  |          |/  |          |/   |   |          |   /   |_____|
*                        |D0|           |D1|           |D2|__|   |     __|_____/
*                        |__|           |__|           |__|      |    |
*                         |              |__________________________|    |
*                         |______________________________________________|
```

Make a simple ideal non-clocked 3bit ADC/DAC.
It only takes six lines of spice code.
It will quantize signal in terms of voltage levels, but not in time.

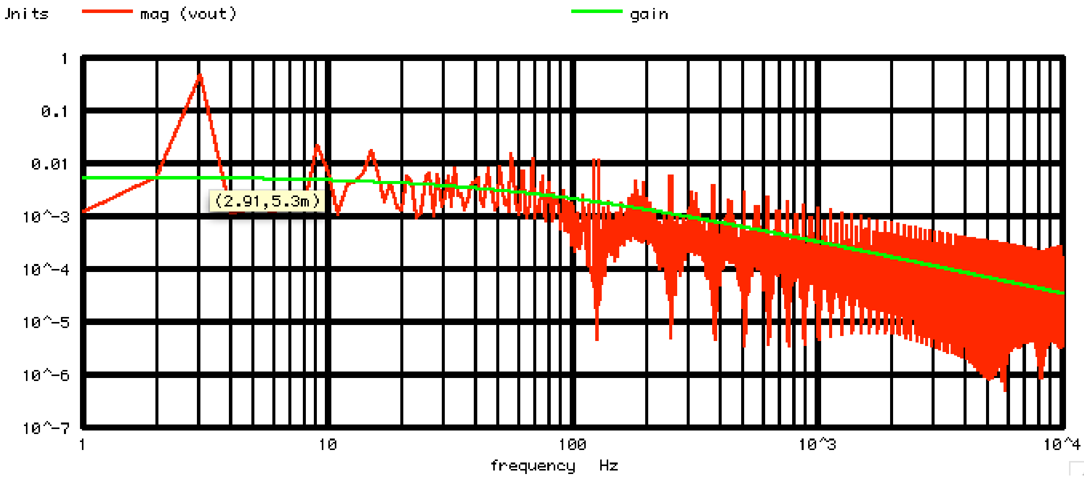
```
BTH0     D0      0      V =    u(V(VIN)  -1/2)
BOFF0    VIN     OFF0   V =    V(D0)/2
BTH1     D1      0      V =    u(V(OFF0) -1/4)
BOFF1    OFF0    OFF1   V =    V(D1)/4
BTH2     D2      0      V =    u(V(OFF1) -1/8)
BDAC     VOUTA   0      V =    (V(D0)/2+V(D1)/4+V(D2)/8)
```

```
*
*   VOUTA     IN2
*          |\      /           IN3  |\
*   |IN |_ |1\__/   ____________    |2\____|OUT|
*   |___|  |  / S1    _|_           |  /   |___|
*          |/         ___           |/
*               C1     |
*                     _|_
*                     ///
```

Now add a sample and hold to quantize the output now in both voltage and time.
In this case, sample period is set to a 8msec rate.



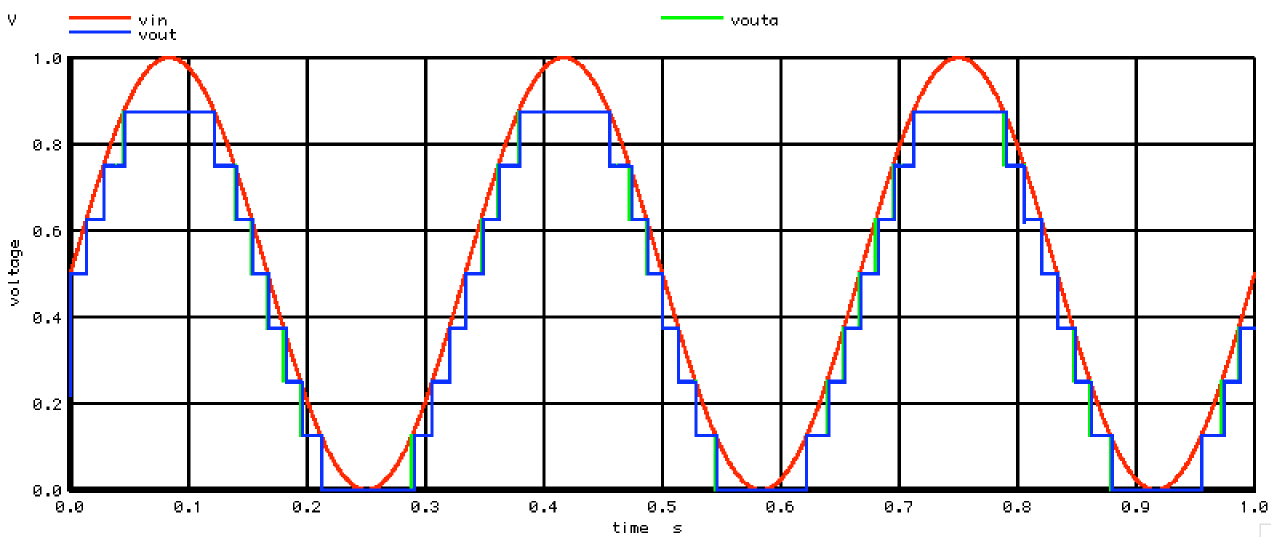
The sampling rate is low enough such that the difference between the clocked and clock-less output values are visible.



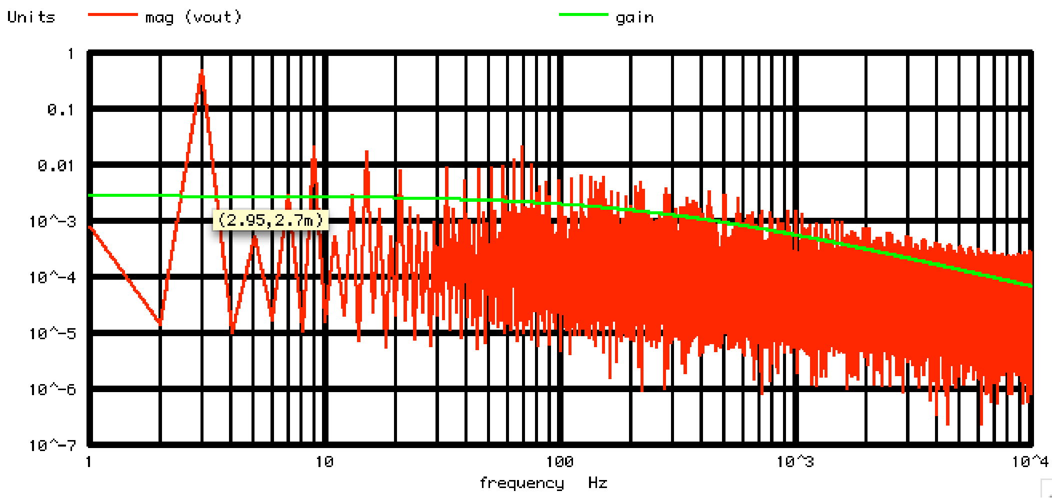
**The spectrum of the quantization error is defined as per bandwidth and power.
The bandwidth is defined by the nyquist of the 8msec sample rate.
The total quantization noise is set by the number of bits.
The typical formula is given below.**

Signal_2_Noise_Max_db = 6.02*Numb_Bits + 1.76

**This 5.3m quantization noise level will be spread out over the full bandwidth.
Since its noise, it average level drops by the square root of the bandwidth.
The spectrum shows expected quantization noise level with expected bandwidth.**



Now increase the sample rate by a factor of four to 2msec.



Now the bandwidth has been made four times wider.
But since Noise drops by the square root of the bandwidth,
the 2.7m noise voltage level is a factor of two lower.

So by oversampling,
most of the quantization noise has been moved to the higher frequencies,
where it can easily be digitally filtered.

What is really important is the noise floor in the bandwidth you care about.
It is often referred to as Effective Number of Bits or ENOB.

ENOB = (SNR_dB-1.76dB)/6.02dB

So in terms of noise in the critical bandwidth,
oversampling by four is like adding one more bit of resolution to the ADC.

```
===================MacSpiceCode==========================================
Simple_Clocked_ADC
*       VIN
*
*    _|_________|_____TH0___|_    _TH1______|_      TH0
*   /_|_\        |          _/ \_  |       _/ \_
*  //   \\       ||\       |\___/  ||\    |\___/  |\        ______________
*  \     \//     |_| >_    |       |_| >_ |       | \      |           |  \
*   \___/          |/  |   OFF0 ___|  |/  |  OFF1 |  >_    |     DAC   |   \__|VOUTA|
*    _|_           |/  |            |/    |       |/  |  __|           |   /  |_____|
*    ///               |                  |           |  |  |          |  /
*                   |D0|               |D1|          |D2|__|   __|__________/
*                   |__|               |__|          |__|   |  |
*                     |                  |__________________|  |
*                     |________________________________________|
*
*
*  VOUTA      IN2
*          |\    /          IN3  |\
*  |IN|_|1\__/  S1  ______________|2\____|OUT|
*  |__|  |  /           _|_       |  /   |___|
*        |/                        |/
*              C1 __|__
*                   |
*                  ///
*=========Create_Signal==================
*V_SIN#   NODE_P NODE_N DC      VALUE   SIN(   V_DC    AC_MAG FREQ    DELAY   FDamp)
VIN       VIN    0      DC      0       SIN(   .5      .5     3                     )
*V_PULSE# NODE_P NODE_N DC      VALUE   PULSE( VINIT   VPULSE TDELAY  TRISE   TFALL   PWIDTH PERIOD )
VCLKL     CLK    0      DC      0       PULSE( 0       1      1n      1n      1n      1m     2m     )

XPOSE1    CLK    CNTL   POS_E
XS_H1     VOUTA  CNTL   VOUT     SH

BTH0      D0     0      V =     u( V(VIN) -1/2)
BOFF0     VIN    OFF0   V =     V(D0)/2
BTH1      D1     0      V =     u( V(OFF0) -1/4)
BOFF1     OFF0   OFF1   V =     V(D1)/4
```

```
BTH2      D2      0       V =     u( V(OFF1) -1/8)
BOFF2     OFF1    OFF2    V =     V(D2)/8
BTH3      D3      0       V =     u( V(OFF2) -1/16)
BOFF3     OFF2    OFF3    V =     V(D3)/16
BTH4      D4      0       V =     u( V(OFF3) -1/32)
BDAC      VOUTA   0       V =     (V(D0)/2+V(D1)/4+V(D2)/8)

.control
*TRAN     TSTEP   TSTOP   TSTART TMAX    ?UIC?
tran      .05m    1        0       .05m
set       pensize = 2
plot      vin  vouta vout

*plot      vin -vout xlimit 1m 1

echo "===================FFT_and_Plot====================================="
linearize

let       FFT_BandWidth_Hz = 10k
let       FFT_resolution_Hz = 1
echo     "FFT_BandWidth_Hz= $&FFT_BandWidth_Hz"
echo     "FFT_resolution_Hz= $&FFT_resolution_Hz"
set      specwindow = "rectangular"
spec     $&FFT_resolution_Hz $&FFT_BandWidth_Hz $&FFT_resolution_Hz v(vout)
let       BW = 1/2m
let       gain = 1/(1 + frequency/(0.5*BW))
let       gain = gain*((.5/8)/sqrt(BW))
plot      mag (vout) gain loglog
echo "===================Done============================================="

.endc

*=========Sample_Hold============================================
*
*             IN2
*     ___   |\    /          IN3 |\     ___
*    |IN |_|1\__/  ____________|2\____|OUT|
*    |___|  | / S1     _|_       | /    |___|
*           |/         ___       |/
*               C1      |
*                      _|_
*                      ///
.SUBCKT   SH      IN      CNTL      OUT
 B1       IN2     0       V =       v(IN )
 S1       IN2     IN3     CNTL     0     SW
 C1       IN3     0       .1u

 R1       IN3     0       100Meg
 B2       OUT     0       V =       v(IN3 )
.ENDS     SH

*=========POS_Edge=========================================
*
*         |\                VBF  ____               ___
*  ___    |3\_________________|    \              |   \
* |IN |__|  /  |          VLP|  PE  \_|OUT|  |IN |_|PE  \_|OUT|
* |___|  |/   |/\  /\  /\____|      / |___|  |___| |    / |___|
*               \/  \/    |  |____/               |___/
*              RLP       _|_
*
*                  CLP   ___
*                         |
*                        _|_
*                        ///
*
.SUBCKT   POS_E  IN      OUT
 BBUF     VBF    0       V =    u( v(IN )-.5 )
 RLP      VBF    VLP     10k
 CLP      VLP    0       1n     IC=0
 BAND     OUT    0       V =     u( u(v(VBF )-.5)*u(.5 -v(VLP ) ) -.1)
.ENDS     POS_E

.MODEL    SW      SW(     VT=.5 VH=.1  RON=1 ROFF=100MEG)


.end
```


7.22.11_1.18PM
dsauersanjose@aol.com
Don Sauer