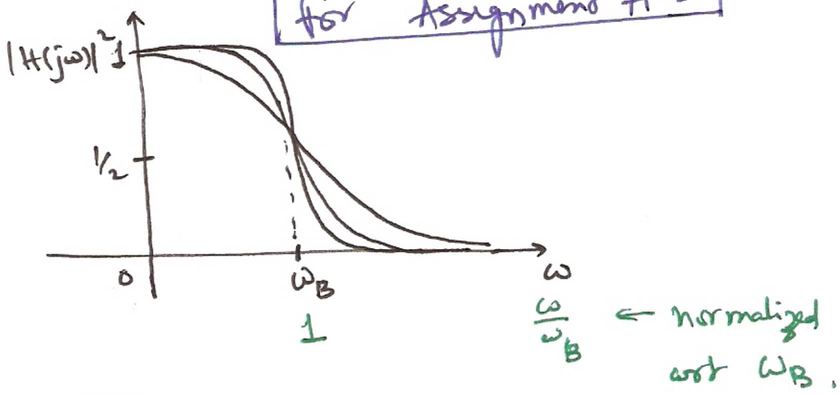
A1

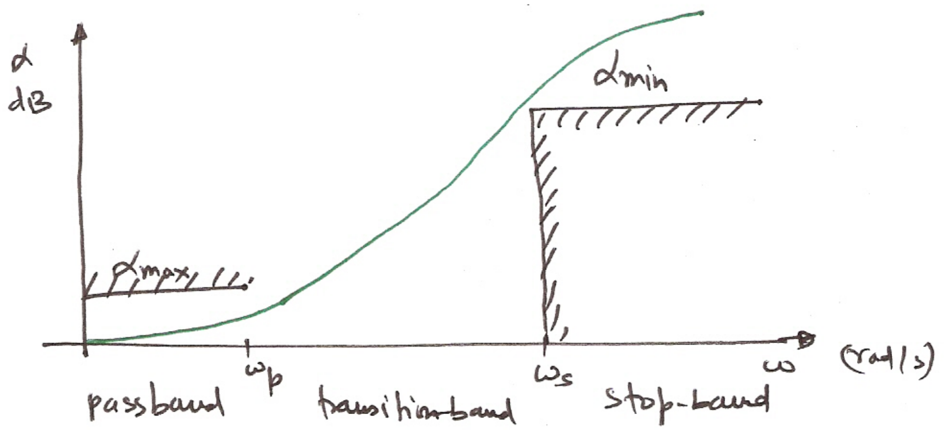
Butterworth LP filter Example

for Assignment # 1

$$H(s) = \frac{1}{1 + \left(\frac{s}{\omega_B}\right)^n}$$

$$\Rightarrow |H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_B}\right)^{2n}}$$





Attenuation:

$$\alpha(\omega) = -20 \log |H(\omega)|.$$

$$|H(\omega)| = 10^{-\frac{\alpha(\omega)}{20}}$$

Define passband corner at $\omega = 1$ $\Rightarrow$ normalize ω as $\frac{\omega}{\omega_p}$

$\varepsilon \rightarrow$ "ripple width"

$$\Rightarrow |H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_B}\right)^{2n}} = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}}$$ ← renormalized

at $\frac{\omega}{\omega_p} = 1 \Rightarrow$

$$|H(j1)|^2 = \frac{1}{1 + \left(\frac{\omega_p}{\omega_B}\right)^{2n}} = \frac{1}{1 + \varepsilon^2}$$

$$\Rightarrow \varepsilon = \left(\frac{\omega_p}{\omega_B}\right)^n$$

$$\Rightarrow \boxed{\omega_B = \varepsilon^{-1/n} \omega_s} \longrightarrow (1)$$

$\Rightarrow$ At the passband corner

$$\alpha_{max} = 10 \log (1 + \varepsilon^2)$$

$$\Rightarrow \boxed{\varepsilon^2 = 10^{0.1\,\alpha_{max}} - 1} \longrightarrow (2)$$

At the passband:

$$\alpha_{min} = 10 \log\left(1 + \varepsilon^2 \omega_s^{2n}\right) \longrightarrow (3)$$

from (2) and (3)

$$\alpha_{min} = 10 \log\left[1 + \left(10^{0.1\alpha_{max}} - 1\right)\omega_s^{2n}\right]$$

$$\Rightarrow \boxed{\omega_s^{2n} = \frac{10^{0.1\alpha_{min}} - 1}{10^{0.1\alpha_{max}} - 1}}$$

$$\Rightarrow \boxed{n = \frac{\log\left[\frac{10^{0.1\alpha_{min}} - 1}{10^{0.1\alpha_{max}} - 1}\right]}{2\log\omega_s}} \longrightarrow (4) \leftarrow \text{gives the order.}$$

Design procedure:

(1) using Eq^n (2) determine parameter $\varepsilon \rightarrow$ sets α_{max}

(2) Using Eq^n (4) calculate the degree n and round it to the next largest integer

(3) Calculate the normalizing frequency $\boxed{\omega_B = \varepsilon^{-1/n}\omega_p}$

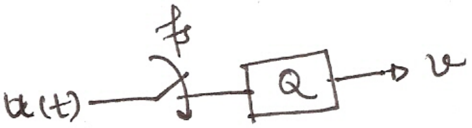
$\hookrightarrow$ 3-dB Bandwidth

Matlab functions:

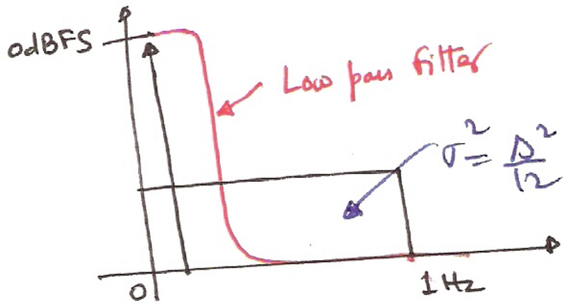
buttord

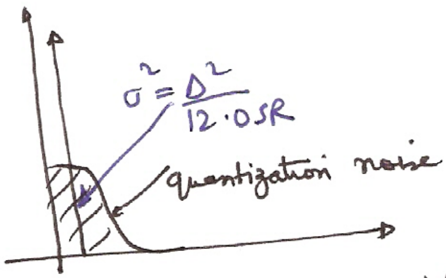
butter

Oversampling :



$f_{in} = \frac{7}{1024} f_s$

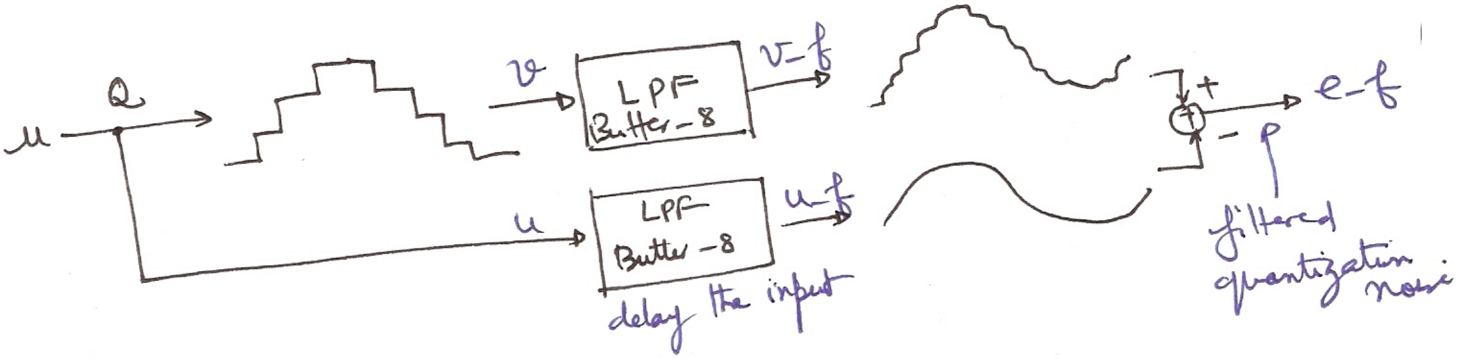




Noise now confined to the bandwidth of the LPF.

- Use 8th order Butterworth filter

See Matlab file : Oversampling.m

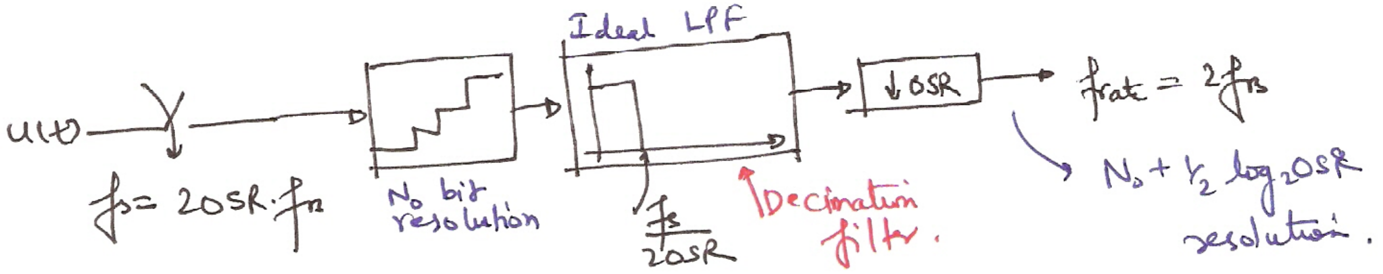


$$\text{Bandwidth} = \frac{f_s}{2 \cdot OSR}$$

oversampling ⇒ SQNR has increased by a factor close to the oversampling ratio, OSR.

$$\boxed{N_{inc} = 0.5 \log_2 OSR}$$

½ bit increase per doubling in OSR.



$$SNR = 10 \log_{10}\left(\frac{A^2/2}{\frac{\Delta^2}{12 \cdot OSR}}\right) = \frac{\frac{(2^{N-1}\Delta)^2}{2}}{\left(\frac{\Delta^2}{12 \cdot OSR}\right)}$$

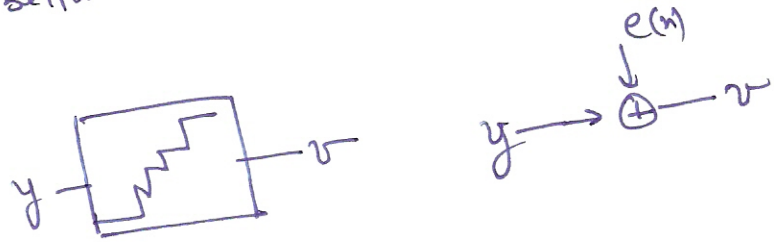
$$= 10 \log\left(\frac{3}{2} \cdot 2^{2N} \cdot OSR\right)$$

$$SNR = 6.02N + 1.76 + \boxed{10 \log_{10} OSR} \leftarrow \text{Extra term}$$

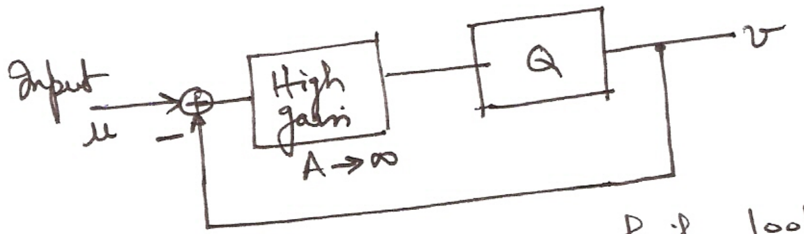
- End to end system looks like a Nyquist rate ADC with effective resolution $N_0 + \frac{1}{2}\log_2 OSR$.
  - ↳ uses only N_0 bit quantizer. (cheap quantizer)

- Trading analog complexity with digital complexity.
  - Digital Decimation filter is automatically synthesized
    - ↳ Verilog coding
  - More effort on optimizing analog portion of the circuit.

- Can we do better than ½ bit per doubling in OSR?

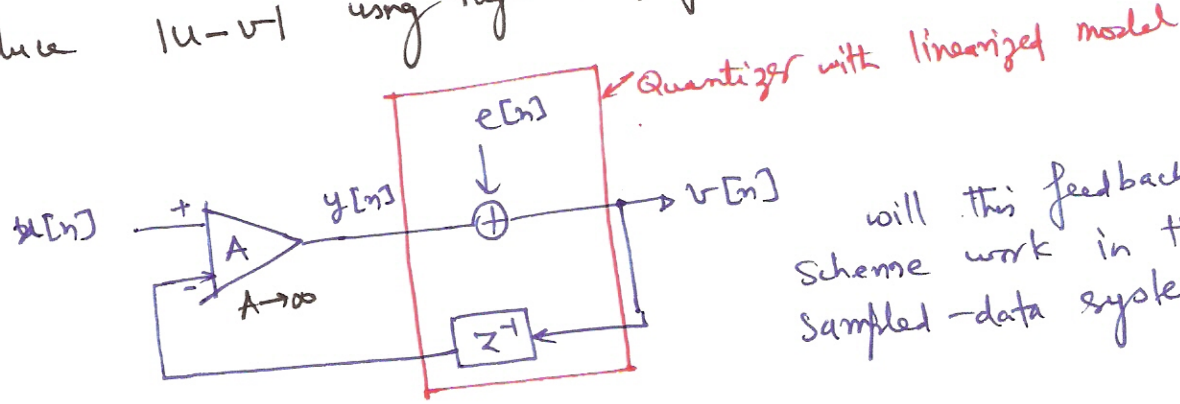


Use feedback to reduce error $e(n)$?



reduce $|u-v|$ using high loop gain?

Consider

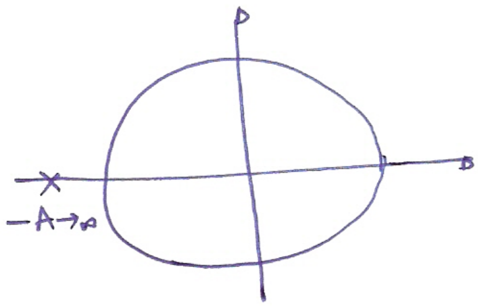


will this feedback scheme work in this sampled-data system?

$$\left(U(z) - z^{-1}V(z)\right)A + E(z) = V(z)$$

$$AU(z) + E(z) = V(z)\left(1 + Az^{-1}\right)$$

$$\Rightarrow \quad V(z) = \left(\frac{A}{1+Az^{-1}}\right)X(z) + \frac{E(z)}{1+Az^{-1}}$$

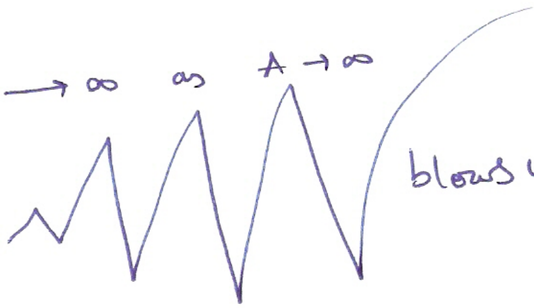


⇒ system pole at $-A \to -\infty$

⇒ system is not stable at all!

error $|u-v| \to \infty$ as $A \to \infty$

error $|u-v|$



blows up.

Ex. $v[0] = 0$

⇒ ex: $|u-v|_0 = -1 \Rightarrow y = -100$

⇒ $|u-v|_1 = -100 \Rightarrow y = +10^4$

......

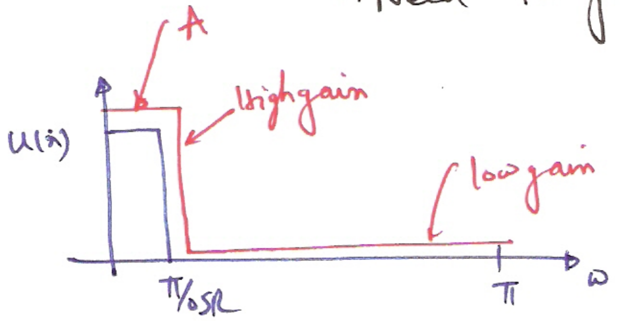
The geometric series explodes.

Doesn't work at all!

Too much delay in the loop causes instability.

$\Rightarrow$ A constant large gain doesn't work.

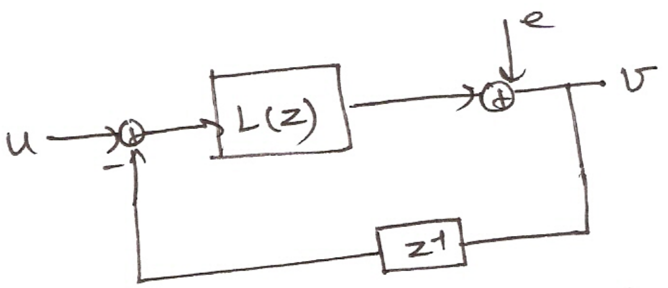
↳ Need to find another way to stabilize the loop.



- $u(t)$ has low frequency content compared to f_s.
- Apply high gain at low frequencies to reduce quantization noise.
- At high frequencies, keep the gain low to stabilize the loop.

$\Rightarrow$ Use frequency dependent gain.

- Replace A by $L(z)$ $\Rightarrow$ Loop filter



$|L(z)| \to \infty$ at $z=1$

$\Rightarrow |u-v| \to 0$

* Example $\Rightarrow$ 1st order : $L(z) = \dfrac{1}{1-z^{-1}}$

$$[U(z) - z^{-1}V(z)]\,L(z) + E(z) = V(z)$$

$$\Rightarrow U(z)\,L(z) + E(z) = V(z)\,(1+z^{-1})$$

$$\Rightarrow V(z) = \underbrace{\boxed{\frac{L(z)}{1+z^{-1}L(z)}}}_{STF(z)}\,U(z) + \underbrace{\boxed{\frac{1}{1+z^{-1}L(z)}}}_{NTF(z)}\,E(z)$$

uncorrelated with $U(z)$.

$$STF(z) = \frac{\frac{1}{1-z^{-1}}}{1+\frac{z^{-1}}{1-z^{-1}}} = 1$$

$$NTF(z) = \frac{1}{1+\frac{z^{-1}}{1-z^{-1}}} = (1-z^{-1}) \longrightarrow \text{High pass response}$$

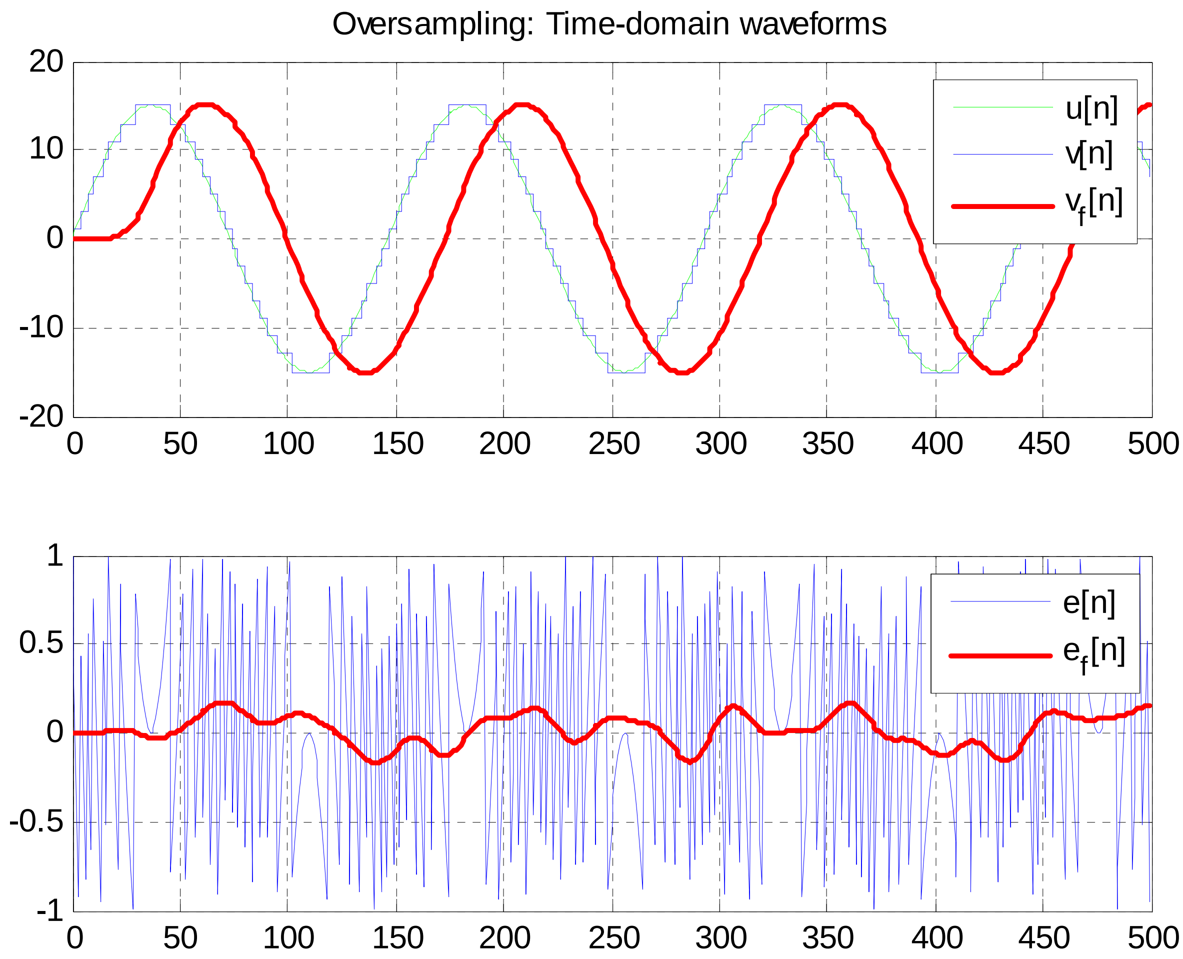
ECE 697 Delta-Sigma Converters Design

Lecture#6 Slides

Vishal Saxena
(vishalsaxena@u.boisetstate.edu)

© Vishal Saxena

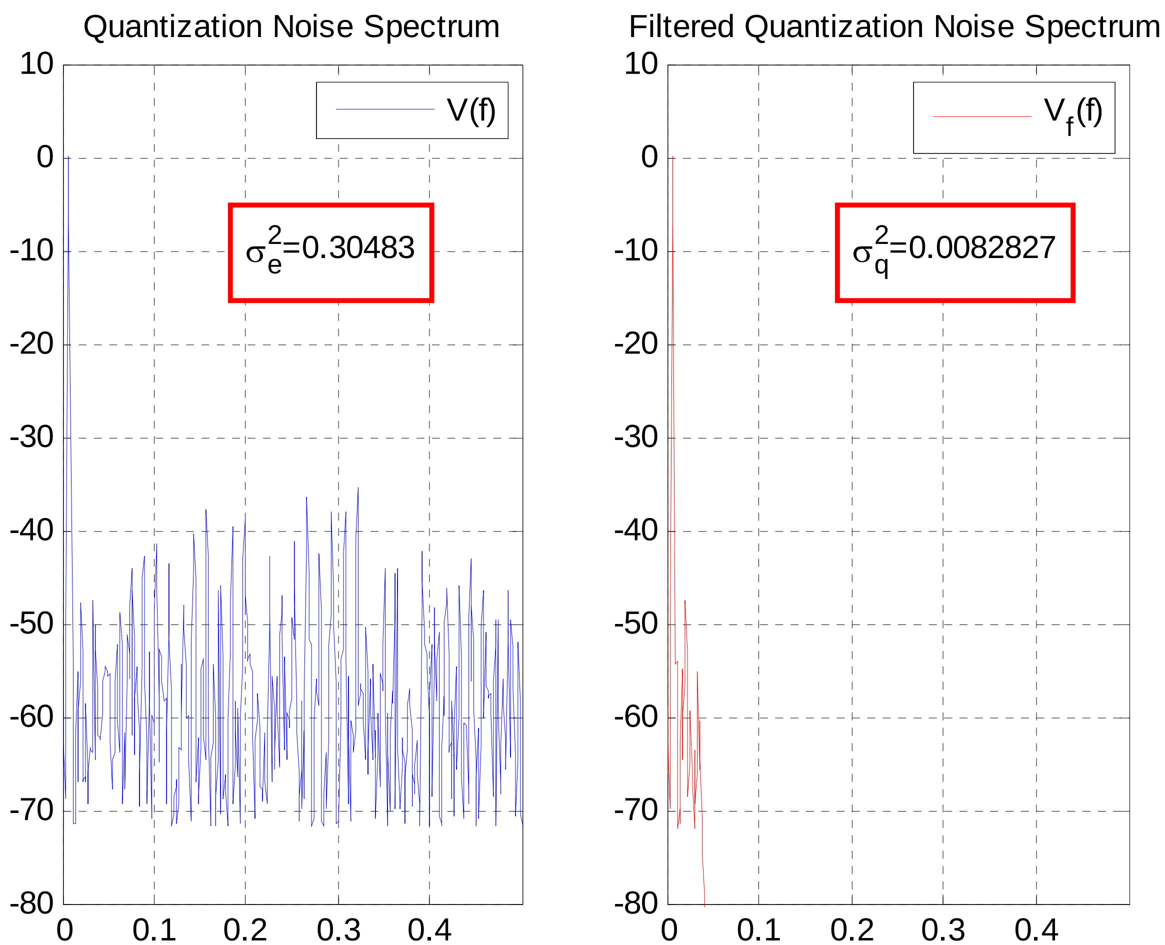
Oversampling



File: oversampling1.m

© Vishal Saxena

Oversampling



File: oversampling1.m

© Vishal Saxena