SAMPLING CLOCK JITTER ESTIMATION AND COMPENSATION IN ADC CIRCUITS

Zaid J. Towfic, Shang-Kee Ting, and Ali H. Sayed
Department of Electrical Engineering
University of California, Los Angeles, CA 90095
Email: {ztowfic,tshangke,sayed}@ee.ucla.edu

***Abstract*— Clock timing jitters refer to random perturbations in the sampling time in analog-to-digital converters (ADCs). The perturbations are caused by circuit imperfections in the sampling clock. This paper analyzes the effect of sampling clock jitter on the acquired samples. The paper proposes two methods to estimate the jitter for superheterodyne receiver architectures and cognitive radio architectures at high sampling rates. The paper also proposes a method to compensate for the jitter. The methods are tested and validated via computer simulations and theoretical analysis.**

***Index Terms*— Clock jitter, analog-to-digital conversion (ADC), interpolation, compensation.**

I. INTRODUCTION

An analog-to-digital converter (ADC) is a basic building block of modern communication systems. Certain applications of modern radios, such as cognitive radios and UWB radios [1], may require ADCs operating at high sampling rates due to the use of wide frequency bandwidths. At high rates, signal distortion is introduced by clock jitters. The jitters cause the ADC to sample the input signal along a non-uniform sampling grid and introduce distortion that limits the signal fidelity and degrades the signal-to-noise ratio (SNR) [2]. Different approaches have been proposed in the literature to compensate for the effect of clock jitter. Some approaches [3], [4], [5] interleave several ADCs in order to produce an effective higher sampling rate. Since each ADC operates at a slower rate, the clock used as its reference will have lower jitter; thereby reducing the clock jitter effects. This technique, however, introduces other problems such as mismatch in the delay of the clock fed into each ADC, the gain of each ADC, and DC offset between the ADCs. Other approaches [6] transform the signal into the wavelet domain and use linear least-mean-squares estimation techniques to recover the original signal. This approach assumes that the signal in the wavelet domain has a small support, which simplifies the estimation of the covariance matrix used in the estimation step to a diagonal representation; this reduces the computational requirements of the resulting de-noising algorithm. The de-noising algorithm achieves close to 10dB SNR improvement in simulations over the original jittered signal. This approach, however, is computationally intensive to conduct in real time for a cognitive radio; especially at high sampling rates. In this work, we first examine the effect of sampling clock jitter on the SNR of the sampled signal. We then propose methods to estimate the jitter for superheterodyne architectures and UWB architectures. A compensation method based on the series expansion of the signal is proposed and analyzed. The method achieves close to 10 dB SNR improvement in simulations over the original jittered signal assuming jitter RMS of 10% of the sampling period.

This work was supported in part by DARPA contract N66001-09-1-2029.

II. EFFECT OF CLOCK JITTER

The effect of sampling jitter on a signal can be studied by modeling the sampling clock jitter as small random perturbations in the sampling time. Thus, consider a zero-mean wide sense stationary (WSS) mean-square continuous [7] random process, $q(t)$, with auto-correlation function $R_q(\tau)$, and assume it is sampled with jitter $e(t)$. Then the sampled process, $\tilde{q}(n)$, is described as:

$$\tilde{q}(n) \triangleq q(nT_s + e(n)) \tag{1}$$

where the notation $f(n)$ denotes $f(nT_s)$ and T_s is the sampling period. Assuming $|e(n)|$ is sufficiently small. A Taylor series expansion of $q(t)$ around $q(nT_s)$ leads to the approximation:

$$\tilde{q}(n) \approx q(n) + e(n)\dot{q}(n) \tag{2}$$

where

$$\dot{q}(n) \triangleq \dot{q}(t)|_{t \to nT_s} \tag{3}$$

Using (2), the SNR after the sampling operation is given by

$$\text{SNR} = \frac{E\,|q(n)|^2}{E\,|\tilde{q}(n) - q(n)|^2} \approx \frac{E\,|q(n)|^2}{E\,|e(n)|^2\,E\,|\dot{q}(n)|^2} \tag{4}$$

where $e(n)$ and $q(n)$ are assumed to be independent. For a mean square continuous WSS process, $q(t)$, it holds that [7]:

$$R_{\dot{q}}(\tau) = -\frac{d^2}{d\tau^2}R_q(\tau) \tag{5}$$

and

$$R_{\dot{q}}(0) \triangleq E\,|\dot{q}(n)|^2 = -\frac{d^2}{d\tau^2}R_q(\tau)\Big|_{\tau \to 0} \tag{6}$$

so that (4) becomes

$$\text{SNR} \approx \frac{\sigma_q^2}{-\sigma_e^2 \cdot \frac{d^2}{d\tau^2}R_q(\tau)\Big|_{\tau \to 0}} \tag{7}$$

in terms of the variances of $e(n)$ and $q(n)$. For illustration purposes, consider a bandlimited signal $q(t)$ with box-car power spectral density (PSD):

$$S_q(f) = \frac{\sigma_q^2}{2B}\text{rect}\left(\frac{f}{2B}\right) \tag{8}$$

where

$$\text{rect}(a) = \begin{cases} 1 & |a| < \frac{1}{2} \\ \frac{1}{2} & |a| = \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and $2B$ is the passband bandwidth of the signal in Hertz (Hz). Then

$$R_q(\tau) = \begin{cases} \sigma_q^2 \frac{\sin(2B\pi\tau)}{2B\pi\tau} & \tau \neq 0 \\ \sigma_q^2 & \tau = 0 \end{cases} \tag{10}$$

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

It follows that

$$-\left.\frac{d^2}{d\tau^2}R_q(\tau)\right|_{\tau\to 0} = \frac{4}{3}\pi^2 B^2 \sigma_q^2 \quad (11)$$

and the SNR expression (7) gives

$$\text{SNR} \approx \frac{3}{4\pi^2 B^2 \sigma_e^2} \quad (12)$$

Throughout the remaining analysis, σ_e^2 is chosen as:

$$\sigma_e^2 = (\alpha T_s)^2 \quad (13)$$

for some fraction α of T_s. This model allows the variance of the jitter to be defined in terms of the sampling period. For example, if $\alpha = 0.1$, this indicates that the RMS value of the jitter is 10% of T_s. For $\alpha = 0.1$ and $B = 250$MHz, expression (12) gives SNR $\approx$ 20.85 dB.

III. ESTIMATION OF CLOCK JITTER

A. LOW FREQUENCY INJECTION METHOD

Let $r(t)$ denote the signal received by a superheterodyne receiver. Assume that a training signal $w(t)$ can be injected into the lower range of the frequency spectrum immediately preceding the ADC – see Figure 1(a). Then the signal entering the ADC is

$$q(t) = r(t) + w(t)$$

When $|e(n)|$ is sufficiently small, the signal $w(nT_s + e(n))$ can be approximated using the first-order Taylor expansion:

$$w(nT_s + e(n)) \approx w(n) + e(n)\dot{w}(n) \quad (14)$$

where, as before,

$$\dot{w}(n) = \left.\dot{w}(t)\right|_{t=nT_s} \quad (15)$$

A convenient choice for $w(t)$ is a low frequency tone, which can be generated reliably by analog circuitry. Thus, let

$$w(t) = \cos(2\pi f_w t + \theta_w) \quad (16)$$

With this choice, the jittered samples become

$$\begin{aligned}\tilde{q}(n) \quad &\approx \quad r\left(nT_s + e(n)\right) + \cos\left(2\pi f_w (nT_s) + \theta_w\right) \quad (17)\\ &\quad -2\pi f_w e(n) \sin\left(2\pi f_w n T_s + \theta_w\right) \quad (18)\end{aligned}$$

Clock recovery methods such as the digital phase locked loop [8] and adaptive filters [9] can be used to recover θ_w. Multiplying $\tilde{q}(n)$ by $\sin(2\pi f_w n T_s + \hat{\theta}_w)$ and applying a low-pass filter yields:

$$\begin{aligned}\text{LPF}\left\{\tilde{q}(n)\sin(2\pi f_w n T_s + \hat{\theta}_w)\right\} \approx &- \pi f_w e(n)\cos\left(\theta_w - \hat{\theta}_w\right)\\ &- \frac{1}{2}\sin\left(\theta_w - \hat{\theta}_w\right)\end{aligned} \quad (19)$$

This suggests one method to recover $e(n)$ when $\theta_w \approx \hat{\theta}_w$, namely,

$$e(n) \approx \frac{\text{LPF}\left\{\tilde{q}(n)\sin(2\pi f_w n T_s + \hat{\theta}_w)\right\}}{-\pi f_w} \quad (20)$$

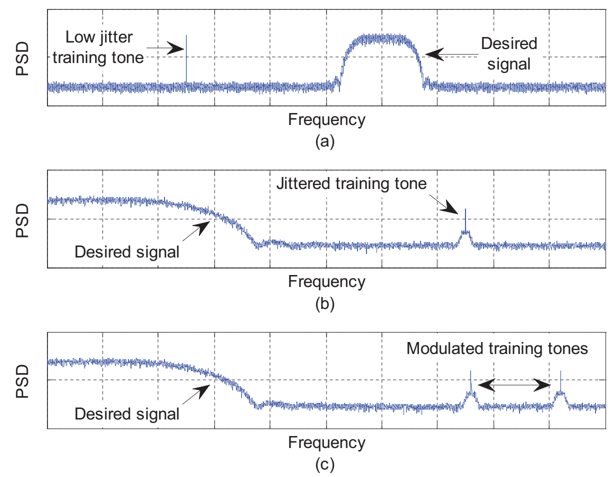

Fig. 1. (a) Spectrum of tone injection in superheterodyne receivers. (b) Spectrum of tone injection at a higher frequency band. (c) Proposed tone injection method.

B. HIGH FREQUENCY INJECTION METHOD

In some applications, such as in cognitive radio sensing, training signal injection in the low frequency band as illustrated in Figure 1a is not possible due to the signal being wideband with frequency components at baseband. In these cases, the training signal must be injected in the high frequency band. However, this is not practical from the analog circuitry standpoint since generating a jitterless high frequency tone is difficult and the result is shown in Figure 1b. For these cases, we instead propose an architecture where the low frequency training signal, $w(t)$, is modulated with a possibly jittered high frequency tone $y(t)$ as in Figure 1c — see also Figure 2.

To illustrate how the jitter of the high frequency tone can be eliminated, let the jittered oscillator, $y(t)$, be modeled as:

$$y(t) = \cos\left(2\pi f_y (t + \tau(t)) + \theta_y\right) \quad (21)$$

where $\tau(t)$ is the high frequency oscillator jitter. Then

$$\begin{aligned}x(t) &= w(t)y(t)\\ &\approx w(t)\left[\cos(2\pi f_y t + \theta_y) - 2\pi f_y \tau(t)\sin(2\pi f_y t + \theta_y)\right]\end{aligned}$$

Multiplying $x(t)$ with an in-phase cosine yields:

$$\begin{aligned}x(t)\ \cos(2\pi f_y t + \theta_y) = \frac{1}{2}w(t)\ &\left[1 + \cos\left(4\pi f_y t + 2\theta_y\right) - \right.\\ &\left. 2\pi f_y \tau(t)\sin(4\pi f_y t + 2\theta_y)\right]\end{aligned} \quad (22)$$

and it is clear that a low-pass filter applied to $x(t)\cos(2\pi f_y t + \theta_y)$ would eliminate the term that is dependent on the jitter $\tau(t)$.
Now, let $q(t)$ denote again the signal fed to the ADC with the injected training signal $w(t)y(t)$ so that

$$q(t) = r(t) + w(t)y(t) \quad (23)$$

The sampled process with jitter $e(t)$ becomes

$$\begin{aligned}\tilde{q}(n) \quad &= \quad r\left(nT_s + e(n)\right) +\\ &\quad\ w\left(nT_s + e(n)\right) y\left(nT_s + e(n)\right)\\ &\approx \quad r\left(nT_s + e(n)\right) +\\ &\quad\ w\left(nT_s + e(n)\right)\cos\left(2\pi f_y n T_s + \theta_y\right) -\\ &\quad\ 2\pi f_y\left(\tau(n) + e(n)\right) w\left(nT_s + e(n)\right) \times\\ &\quad\ \sin\left(2\pi f_y n T_s + \theta_y\right)\end{aligned}$$

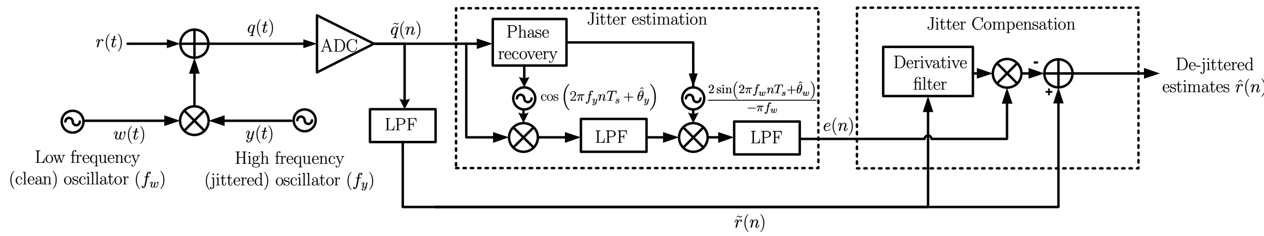

Fig. 2. Proposed architecture for jitter compensation using high frequency training signal injection.

Multiplying $\tilde{q}(n)$ by $2\cos\left(2\pi f_y nT_s+\hat{\theta}_y\right)$ and filtering using a low-pass filter yields:

$$\begin{aligned}&\text{LPF}\left\{2\tilde{q}(n)\cos\left(2\pi f_y nT_s+\hat{\theta}_y\right)\right\}\\ &\approx w\left(nT_s+e(n)\right)\cos\left(\theta_y-\hat{\theta}_y\right)-\\ &\quad 2\pi f_y\left(\tau(n)+e(n)\right)w\left(nT_s+e(n)\right)\sin\left(\theta_y-\hat{\theta}_y\right)\end{aligned} \tag{24}$$

Thus, provided $\hat{\theta}_y\approx\theta_y$, it is possible to recover the training signal as:

$$h(n) \triangleq w\left(nT_s+e(n)\right) \tag{25}$$

$$\approx \text{LPF}\left\{2\tilde{q}(n)\cos\left(2\pi f_y nT_s+\hat{\theta}_y\right)\right\} \tag{26}$$

Subsequently, the jitter $e(n)$ can be removed from $w(nT_s+e(n))$ using a method similar to (20), i.e,

$$e(n)\approx\frac{\text{LPF}\left\{h(n)\sin(2\pi f_w nT_s+\hat{\theta}_w)\right\}}{-\pi f_w} \tag{27}$$

IV. Compensation of Clock Jitter

Once the jitter, $e(n)$, has been estimated, it is still necessary to re-interpolate the jittered samples $\tilde{r}(n)$ to recover $r(n)$. First, $\tilde{r}(n)$ is recovered from $\tilde{q}(n)$ via a low-pass filter with L_0 taps:

$$\tilde{r}(n)\triangleq r(nT_s+e(n))\approx\text{LPF}\left\{\tilde{q}(n)\right\} \tag{28}$$

The recovery can be approximated via a Taylor expansion on $r(n)$:

$$\begin{aligned}r(n)&=r\left(nT_s+e(nT_s)-e(nT_s)\right)\\ &\approx\tilde{r}(n)-e(n)\dot{r}(nT_s+e(nT_s))\end{aligned} \tag{29}$$

where $\dot{r}(nT_s+e(nT_s))=\dot{r}(t)|_{t=nT_s+e(nT_s)}$. The approximation above can be extended to include higher-order derivatives to improve the accuracy of the compensation step. The derivative of $r(t)$ is not available after sampling. However, consider the derivative of $\tilde{r}(t)\triangleq r(t+e(t))$:

$$\frac{d\tilde{r}(t)}{dt}=\frac{dr(t+e(t))}{dt}=\dot{r}(t+e(t))\left(1+\dot{e}(t)\right) \tag{30}$$

where

$$\dot{r}(t+e(t))\triangleq\left.\frac{dr(x)}{dx}\right|_{x=t+e(t)} \tag{31}$$

If $e(t)$ is assumed to be small and slowly varying (in comparison to the sampling frequency), it is possible to ignore the term $\dot{e}(t)$ and write

$$\dot{r}(nT_s+e(n))\approx\left.\frac{d\tilde{r}(t)}{dt}\right|_{t=nT_s} \tag{32}$$

and the derivative can be approximated using a discrete filter [10].

The architecture for the high frequency training signal injection jitter estimation method and a single derivative reconstruction is illustrated in Figure 2.

V. Compensation Analysis

As discussed in the previous section, it is possible to include higher-order derivatives in the compensation block (which leads to a higher-order Taylor series expansion) in order to improve the compensation performance. A natural question that arises is what the SNR improvement will be when N derivatives are used. Assume perfect derivatives and slowly varying and small Gaussian distributed jitter $e(n)$ with zero mean and variance σ_e^2. Assume further a mean-square continuous WSS process $r(t)$ with autocorrelation function defined by (10) sampled with $e(n)$. It can be shown that the SNR of

$$\hat{r}(n)\approx\tilde{r}(n)+\sum_{k=1}^{N}\frac{(-e(n))^k}{k!}\left.\frac{d^k}{dt^k}\tilde{r}(t)\right|_{t=nT_s} \tag{33}$$

where $r(n)$ is recovered from $\tilde{r}(n)$ using N derivatives can be approximated by

$$\frac{E\left|r(n)\right|^2}{E\left|r(n)-\hat{r}(n)\right|^2}\approx\frac{(2\sigma_e^2)^{N+1}\Gamma\left(N+\frac{3}{2}\right)}{\left((N+1)!\right)^2\sqrt{\pi}}\frac{(\pi 2B)^{2N+2}}{2N+3} \tag{34}$$

where $\Gamma(z)\triangleq\int_0^\infty t^{z-1}e^{-t}dt$ and B is the bandwidth of $r(n)$ in Hertz. Figure 3 illustrates this relationship for $B=250\text{MHz}$ and $T_s=1\text{ns}$ and perfect knowledge of $e(n)$.

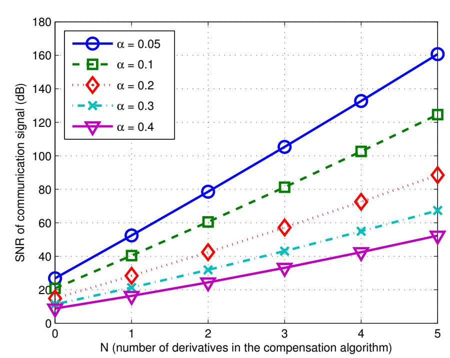

Fig. 3. SNR vs. number of derivatives in the compensation algorithm

VI. Complexity Analysis

Dejittering as described is a two step process: 1) jitter estimation and 2) jitter compensation. The FLOP count for the high frequency signal injection jitter estimation algorithm is $2(L_1+L_2)$ FLOPs per sample. In addition, the compensation algorithm with N derivative

TABLE I

FILTER TAPS USED FOR THE FIRST AND SECOND ORDER DERIVATIVE FILTERS IN SIMULATION ($\times 10^{-3}$)

Tap $(n-k)$	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
$1st$ Derivative	2	-5	-4	18	13	-193	808	0	-808	193	-13	-18	4	5	-2
$2nd$ Derivative	41	-56	80	-125	222	-500	2000	0	2000	-500	222	-125	80	-56	41

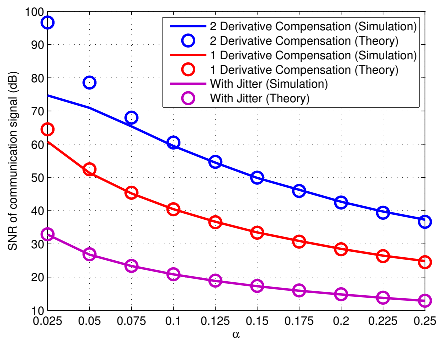

Fig. 4. SNR of communication signal vs. α

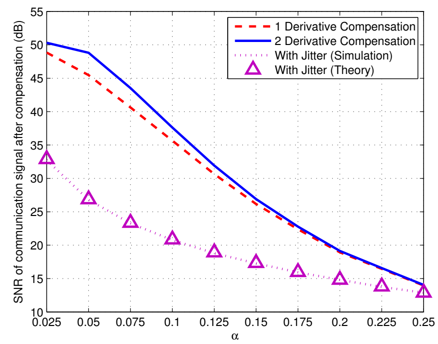

Fig. 5. SNR of communication signal before and after compensation.

filters of D_n taps each requires $2L_0 - 1 + 2N + \sum_{n=2}^{N} n + \sum_{n=1}^{N}(2D_n - 1)$ FLOPS per sample where L_0 is the number of taps in the low-pass filter which extracts $\tilde{r}(n)$ from $\tilde{q}(n)$. Thus, the total number of FLOPs per sample for jitter estimation and compensation is

$$2(L_0 + L_1 + L_2) + 2N - 1 + \sum_{n=2}^{N} n + \sum_{n=1}^{N}(2D_n - 1) \quad \frac{\text{FLOPs}}{\text{sample}}$$

Some practical frequency configurations allow for the elimination of a low-pass filter from the estimation algorithm, which would reduce the complexity by either L_1 or L_2 FLOPs per sample.

VII. SIMULATION RESULTS

In simulation, the sampling period is 1ns while the signal bandwidth, B, is set to 250MHz. The jitter $e(n)$ is Gaussian with zero mean and standard deviation $\sigma_e^2 = 0.1T_s$ and is correlated with bandwidth of 5MHz. The frequency of the two sinusoids are $f_w = 40$MHz and $f_y = 420$MHz. The signal $y(t)$ was itself jittered with jitter $\tau(t)$ due to the assumption that a high frequency oscillator will have jitter of its own. The jitter $\tau(t)$ has the same stochastic properties as $e(t)$. The low-pass filters are FIR filters designed using MATLAB's FIRLS function and have 128 taps. Incidently, with the above frequency allocation, the LPF in the definition of $h(n)$ in (26) is not required. Table IV lists the filter taps of the derivative filters used in the simulation. The magnitude responses of the 1^{st} derivative filter used in the simulation and the ideal derivative filter are illustrated in Figure 6.

The compensation analysis in Section V is verified by assuming the jitter $e(n)$ is perfectly known while $\tau(n)$ is set to zero. Figure 4 shows the SNR before and after compensation assuming perfect jitter estimation. The deviation from theory, which occurs on the low α range, is due to noise, which leaks through the low-pass filter. This noise can be reduced with a higher-order filter. Figure 5 shows the SNR before and after compensation and includes $\tau(n)$ and the estimation of $e(n)$ described in Section III. The figure also confirms our derivation in Section II.

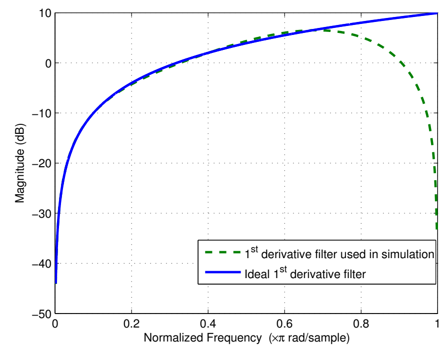

Fig. 6. Magnitude response of 1^{st} derivative filter used in simulation and close to ideal derivative filter.

REFERENCES

[1] H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. Springer-Verlag, New York, 2007.

[2] B. Razavi, *Principles of Data Conversion System Design*. John Wiley & Sons, NJ, 1995.

[3] J. Elbornsson, F. Gustafsson, and J.-E. Eklund, "Blind adaptive equalization of mismatch errors in a time-interleaved A/D converter system," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 1, pp. 151–158, Jan. 2004.

[4] C. Fu and H. Wang, "Adaptive optimization in A/D converter system," *6th IEEE International Conference on Industrial Informatics, Proc. Daejeon, Korea*, pp. 627–630, July 2008.

[5] S. Huang and B. Levy, "Blind calibration of timing offsets for four-channel time-interleaved ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 4, pp. 863–876, April 2007.

[6] N. Testoni, N. Speciale, A. Ridolfi, and C. Pouzat, "Adaptive wavelet-based signal dejittering," *Research in Microelectronics and Electronics Conference, Talence, France*, pp. 257–260, July 2007.

[7] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Upper Saddle River: Pearson/Prentice Hall, 2008.

[8] C. R. Johnson Jr and W. A. Sethares, *Telecommunication Breakdown: Concepts of Communications Transmitted via Software-Defined Radio*. Prentice Hall, 2004.

[9] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, NJ, 2008.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, January 1999.