FOLIA 180

Annales Universitatis Paedagogicae Cracoviensis

Studia Technica VIII (2015)

Dominik Rzepka

Simulation of level-crossing sampling of stochastic processes

Introduction

The level-crossing was proposed by Mark and Todd [1] as a compression technique for acquisition of images, and more recently, by Sayiner et al., as an energy-efficient sampling method [2]. The samples of x (t) signal are taken at the time instants, when this signal crosses any of the predefined levels $\mathbf{t} = \{L_1, L_2,\ldots,L_K\}$, disposed in the amplitude domain. The result is a series of samples $\mathbf{x} = \{x(t_1), x(t_2),\ldots,x(t_N)\}$, taken at the sampling instants $\mathbf{t} = \{t_1, t_2,\ldots,t_N\}$, which are usually nonuniform. The idea of level-crossing sampling is presented in Figure 1.

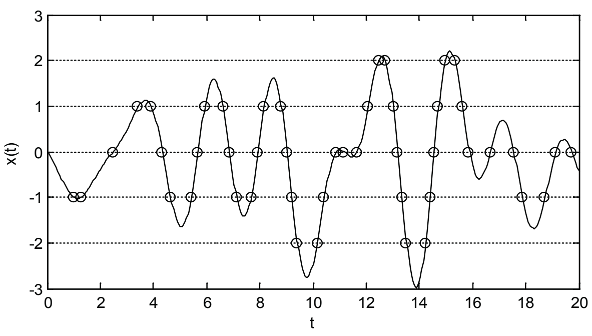

Fig. 1. Level-crossing sampling

The samples can be used for the reconstruction of the original signal x (t). This, however, cannot be done using the classical Shannon theorem [3], because the samples are non-periodic, and one of the reconstruction methods for irregular samples must be used [4, 5]. Another area of study is a distribution of a level-crossing instants $\mathbf{t} = \{t_1, t_2,\ldots,t_N\}$ for the signal x (t), which is modelled as a realization of the stationary Gaussian process X (t) [6, 7]. To compare the theoretical results (which can be only approximate) with the real properties of the level-crossing sampling, the simulation can be useful. Here, we will provide the procedure of such simulation in the MATLAB environment.

Simulation of random Gaussian process

Shannon sampling theorem [3] was also generalized to the stochastic processes [8]. The Ω-band-limited stochastic Gaussian process $X(t)$ is characterized by the autocorrelation function $R_X(t)$, so that the power spectrum

$$S_X(\omega) = \int_{-\infty}^{+\infty} R_X(t) e^{-j\omega t} dt \tag{1}$$

contains no spectral components with pulsation above Ω

$$S_X(\omega) = 0, |\omega| > \Omega \tag{2}$$

Ω pulsation is related to the spectral component of $X(t)$ with the highest, Nyquist frequency $W = \frac{\Omega}{2\pi}$. The corresponding sampling period $T = \frac{\pi}{\Omega}$ is related to the Nyquist rate $f_s = 2W = 1/T$.

The Shannon reconstruction formula of a stochastic process $X(t)$ with autocorrelation function $R_X(t)$ can be written as

$$X(t) = \sum_{n=-\infty}^{+\infty} X(nT) \frac{\sin(\Omega(t-nT))}{\Omega(t-nT)}, \tag{3}$$

where the sequence $\{X(nT)\}_{n\in\mathbb{Z}}$ is a discrete-time stochastic $X[n]$ process with the autocorrelation $R_X[n] = R_X(nT)$. Alternatively, the process $X(t)$ can be represented as filtering of the process $Y(t)$ with $R_Y(t) = \frac{\sin(\Omega t)}{\Omega t}$. Hence, we have

$$X(t) = h(t) * Y(t), \tag{4}$$

where the filter impulse response function h(t) also defines the following relationship between the covariance functions:

$$R_X(t) = h(t) * R_Y(t) * h(-t), \tag{5}$$

then

$$X(t) = \sum_{n=-\infty}^{+\infty} Y(nT) \left(h(t) * \frac{\sin(\Omega(t-nT))}{\Omega(t-nT)} \right) \tag{6}$$

and $Y(nT) = Y[n]$, where $Y[n]$ is uncorrelated discrete-time sequence of random variables, $R_Y[n] = \delta[n]$.

The simulation of $X(t)$ is possible only for a finite number of samples $n \in \{1,2,\dots N\}$, so that the signal

$$\tilde{X}(t) = \sum_{n=1}^{N} X(nT)\frac{\sin(\Omega(t-nT))}{\Omega(t-nT)} = \sum_{n=1}^{N} Y(nT)\left(h(t) * \frac{\sin(\Omega(t-nT))}{\Omega(t-nT)}\right) \tag{7}$$

is only an approximation of the Gaussian process. The practical representation of the process $X(t)$ is a time series $\tilde{X}[m] = \tilde{X}\left(m\frac{T}{M}\right)$, which is an M-fold oversampled $\tilde{X}(t)$. The MATLAB code for the generation of such approximation is included in the Appendix.

Search for the level-crossing instants

The time instants $\{\dots,t_{-1},t_0,t_1,\dots\}$, when the signal crosses the levels $\{L_1,L_2,\dots,L_K\}$ cannot be determined analytically, but it is possible to approximate the finite subset $\{t_1,t_2,\dots,t_N\}$ numerically, with the desired precision. As the input of the algorithm we have a sequence $\{\tilde{X}[1],\tilde{X}[2],\dots\tilde{X}[M \cdot N]\}$. The crossings of the level L_k by the sequence $\tilde{X}[m]$ can be transformed into a sequence of pulses using the following formula:

$$\varnothing\,[m] = \left|\mathrm{Sign}\left(\tilde{X}[m]-L_k\right) - \mathrm{Sign}\left(\tilde{X}[m+1]-L_k\right)\right|. \tag{8}$$

At first, the sign of the $\tilde{X}[m]-L_k$ is taken, then discrete differentiation transforms the square pulses into Kronecker's deltas, and finally the sign of the deltas is removed by the absolute value. Then it is straightforward to find the time-position of such deltas. This procedure results with approximate sampling instants $\{\tilde{t}_1,\tilde{t}_2,\dots,\tilde{t}_N\}$ which are always smaller than the true crossing instants

$$\tilde{t}_n < t_n. \tag{9}$$

The time-error of such search is equal to

$$\Delta_1 = \frac{1}{M} \tag{10}$$

and can be set to arbitrarily small value by setting high oversampling M. However, this can lead to the unnecessarily high computational effort, as the formula for the signal generation (7) must be evaluated for the $M \bullet N$ points, while the high resolution is required only in the vicinity of the level crossings $\{t_1, t_2,\dots,t_N\}$. Higher precision without need for heavy computation can be obtained by refining the search for true crossing instants, which from (9) and (10) are known to be placed in the intervals $\left(\tilde{t}_n, \tilde{t}_n + \frac{1}{M}\right)$. We used a bisection method [9] with R steps to improve the time-error of t_n approximation to the value

$$\Delta_2 = \frac{1}{M(R+1)}, \tag{11}$$

while the number of evaluations of (7) is $M \bullet N + R$ Increasing R at the expense of decreasing M is beneficial in the terms of computational effort, but it must be remembered, that if the interval $\left(\tilde{t}_n, \tilde{t}_n + \frac{1}{M}\right)$ contains more than one level-crossing, then bisection method will find only one of them. However, the distance between neighboring crossings $\tau = t_{n+1} - t_n$ can be arbitrarily small, so there is no value of M which would assure finding all crossings.

Appendix

The following program is the MATLAB implementation of the aforementioned procedure for generating level crossings of a signal modelled as the stationary Gaussian process.

```
disp('Finds level crossings of a signal with high accuracy.');
% length of signal
N = 200;
% oversampling
M = 20;
% level-crossing position refinements
R = 22;
% levels
L = [-2 -1 0 1 2];

% time instants for periodic samples
ts = (0:N-1)-N/2;
% vector of Gaussian random variables X[n]
xs = randn(N,1);

% M-oversampling time instants
t = ( (1/M:1/M:N)-N/2 )';
% M-fold oversampled X(t)
x = sinc(tsh(ts, t))*xs;

% search for level crossings
tn = [];
for i = 1:length(L)
```

```
% find the level crossings from the sequence X[m]
id = find(abs(diff(sign(x-L(i)))) > 0);

% refine the level crossings instants
if ~isempty(id)
    % sampling instants prior to crossings
    t_prior = t(id);
    % sampling instants after crossings
    t_after = t(id+1);
    for r = 1:R
        % take value from the middle of the interval
        t_mid = (t_prior+t_after)/2;

        % samples values prior to crossings
        xn_prior = sinc(tsh(ts, t_prior))*xs;
        % samples values at the middle of the interval
        xn_mid = sinc(tsh(ts, t_mid))*xs;

        % from the two intervals choose the one which contains the
        crossing
        id_prior = ( sign(xn_prior-L(i)) == sign(xn_mid-L(i)) );
        id_after = ~id_prior;

        % set new endpoints of the interval;
        t_prior(id_prior) = t_mid(id_prior);
        t_after(id_after) = t_mid(id_after);
    end;
    % gather the sampling instants
    tn = [tn; t_mid];
  end;
end;

tn = sort(tn);
xn = sinc(tsh(ts, tn))*xs;

t = sort([t; tn]);
x = sinc(tsh(ts, t))*xs;
levels = L'*ones(1, length(t));

%% visualisation
plot(t, x, '.-', tn, xn, 'ro', t, levels, 'r--');
```

References

[1] Mark J.W., Todd T.D., *A nonuniform sampling approach to data compression*, IEEE Transactions on Communications. 1981, 29(1), p. 24–32.

[2] Sayiner N., Sorensen H.V., Viswanathan T.R., *A level-crossing sampling scheme for a/d conversion*, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on. 1996, 43(4), p. 335–339.

[3] Shannon C.E., *Communication in the presence of noise*, Proceedings of the IRE. 1949, 37(1), p. 10–21.
[4] Yen J., *On nonuniform sampling of bandwidth-limited signals*, IRE Transactions on Circuit Theory. 1956, 3(4), p. 251–257.
[5] Feichtinger H.G., Gröchenig K., Strohmer T., *Efficient numerical methods in non-uniform sampling theory*, Numerische Mathematik. 1995, 69(4), p. 423–440.
[6] Munakata T., *Some unsolved problems on the level crossing of random processes*. In: C.R. Doering, L.B. Kiss, M.F. Shlesinger (eds.), *Unsolved Problems of Noise*, World Scientific, Singapore 1997, p. 213–222.
[7] Fuchikami N., Shunya I., *Statistics of level crossing intervals*. Second International Symposium on Fluctuations and Noise. International Society for Optics and Photonics, 2004.
[8] Balakrishnan A., *A note on the sampling principle for continuous signals*, Information Theory, IRE Transactions on. 1957, 3(2), p. 143, 146.
[9] Cheney E., Kincaid D., *Numerical mathematics and computing*. Cengage Learning, 2012.

Abstract

In this report we describe the procedure of generating a random Gaussian process with a given autocorrelation function and the sampling of this signal at the points where it crosses the predefined levels. To improve accuracy of the sampling, the sampling instants are refined iteratively using the bisection method. The simulation is implemented in MATLAB and the corresponding script of the developed program is provided.

Key words: level-crossing, sampling, stochastic process, simulation, MATLAB

Dominik Rzepka
Pedagogical University of Cracow
Institute of Technology
ul. Podchorążych 2
30-084 Kraków, Poland