APPLICATIONS IN SIGNAL PROCESSING

Section 10 Delta Modulation

This document carries out delta modulation and adaptive delta modulation of an input signal and plots the step-function output and corresponding errors. You provide:

- **F(t)**, the input signal
- **d**, the sampling interval
- **f**, the sampling frequency

The document modulates the given function, and then illustrates recovery of the input signal by filtering.

References

The Electronic Communications Problem Solver, Research and Education Association (Piscataway, NJ, 1984). See also William McC. Siebert, *Circuits, Signals, and Systems*, The MIT Press (Cambridge, 1986).

Background

Delta modulation generates a step-function approximation to a continuous signal by taking a step up if the current value of the step function is less than the current signal sample, and a step down otherwise. In adaptive delta modulation, the step size is variable and the change in step size is controlled by the sign of the current error.

The binary sequence representing whether a step up or a step down is taken can also be used to relay information. The sequence is integrated and then lowpass filtered to recover the signal. This is an attractive scheme for speech digitization because it is simple to implement, and can be made as accurate as required by changing the sampling rate.

For more information about signal digitization and sampling rates, see **Section 9: Digitizing a Signal**.

Mathcad Implementation

Mathcad's seeded iteration provides a convenient way of implementing delta modulation and adaptive delta modulation. Enter the definition of a continuous function **F(t)** to be sampled and modulated, and define the sampling frequency, length of the sample, and step size **d** for the delta modulation.

input waveform: $F(t) := \sin(2 \cdot \pi \cdot t)$

sampling frequency: $f := 30$ length of sample: $L := 2$

step size: $d := .2$ sampling interval: $T := \frac{1}{f}$

Delta Modulation

Equations defining the delta modulation:

$$k := 0 .. f \cdot L \qquad \qquad sign(a) := (a > 0) - (a < 0)$$

Starting matrix values:

$$\begin{bmatrix} S_0 \\ M_0 \end{bmatrix} := \begin{bmatrix} d \\ -d \end{bmatrix}$$

$$n := 0 .. f \cdot L + 2 \qquad \qquad signal_n := F(n \cdot T)$$

$$A := \max(signal) \qquad \qquad B := min(signal)$$

An iterative vector generates the next steps in the modulation by finding the value of the input waveform at the next sample point, checking to see if the value is dropping or rising, and assigning the new modulated height. The matrix equation simultaneously generates the step-function approximation **M** and the sequence **S**, which gives the size of each step as either **d** or **-d**.

$$\begin{bmatrix} S_{k+1} \\ M_{k+1} \end{bmatrix} := \begin{bmatrix} S_0 \cdot sign\left(signal_{k+2} - \left(M_k + S_k\right)\right) \\ M_k + S_k \end{bmatrix}$$

The resulting vector, **M**, is the stepped approximation to the input signal **F(t)**.

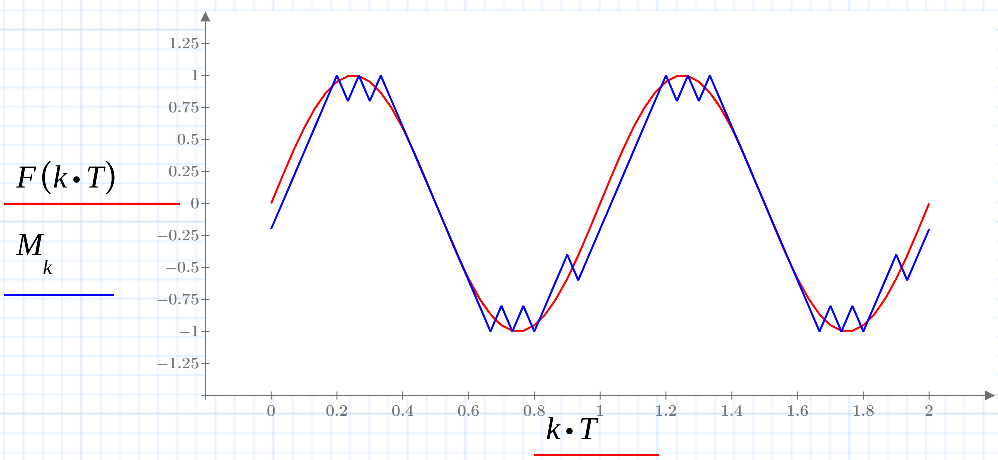

Fig. 10.1 The original input function and resulting delta modulation

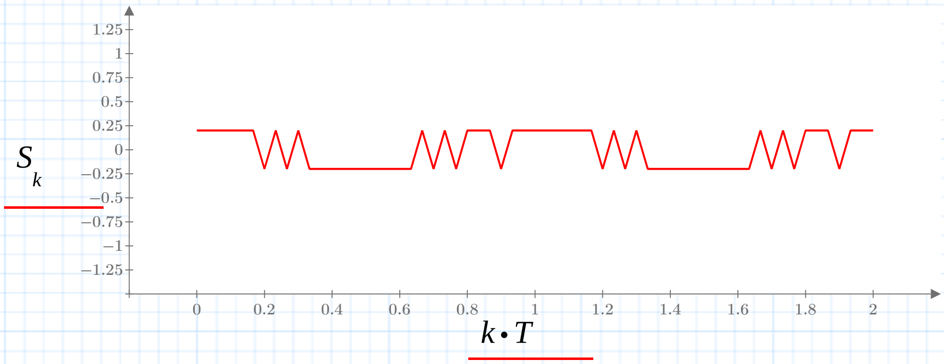

Fig. 10.2 The vector of steps, S

The errors, that is, the differences between the original signal and the step-function approximation (the error here is measured at the end of each step), are given by

$$E_k := signal_{k+1} - M_k$$

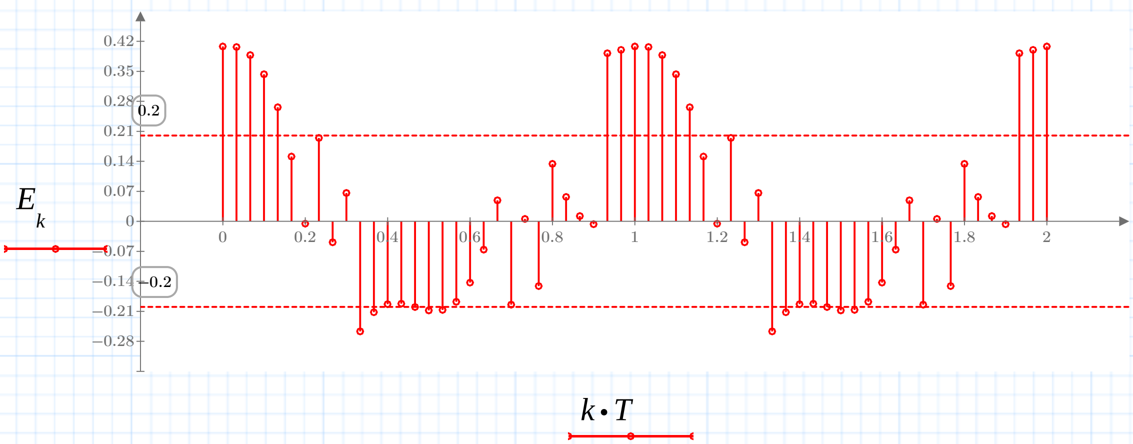

Fig. 10.3 Error bars on each of the modulated samples

Adaptive Delta Modulation

For the adaptive delta modulation scheme, set a minimum step size by editing the definition below for **dmin**. The seeded iteration now generates three sequences simultaneously: **m** is the step-function approximation to **F**, **s** is the sequence of steps, and **e** contains the sign of the error at each step.

minimum step size: $d_{min} := .12$

$$j := 0 .. f \cdot L$$

$$\begin{bmatrix} m_0 \\ s_0 \\ e_0 \end{bmatrix} := \begin{bmatrix} 0 \\ d_{min} \\ 1 \end{bmatrix} \qquad \begin{bmatrix} m_{j+1} \\ s_{j+1} \\ e_{j+1} \end{bmatrix} := \begin{bmatrix} m_j + s_j \\ \left|s_j\right| \cdot sign\left(F\left((j+2)\cdot T\right) - \left(m_j + s_j\right)\right) + s_0 \cdot e_j \\ sign\left(F\left((j+2)\cdot T\right) - \left(m_j + s_j\right)\right) \end{bmatrix}$$

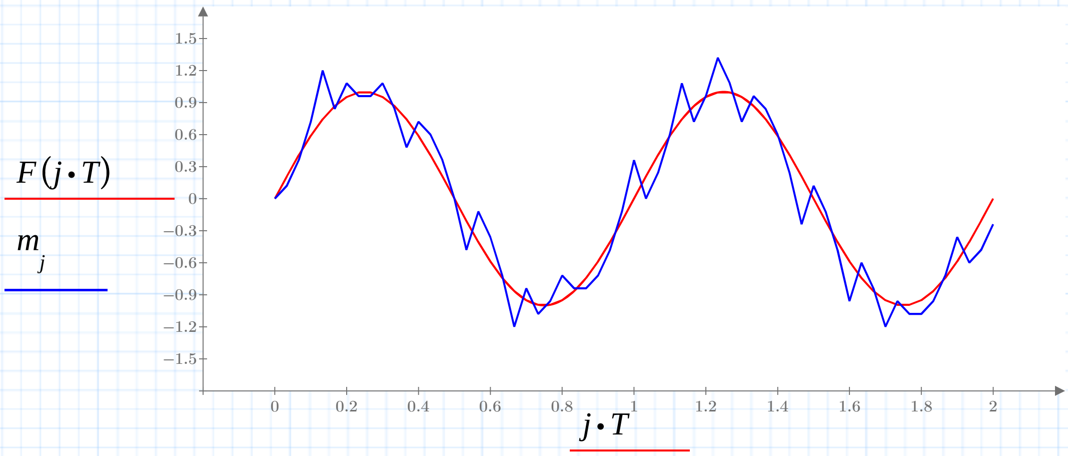

Fig. 10.4 Input and step functions

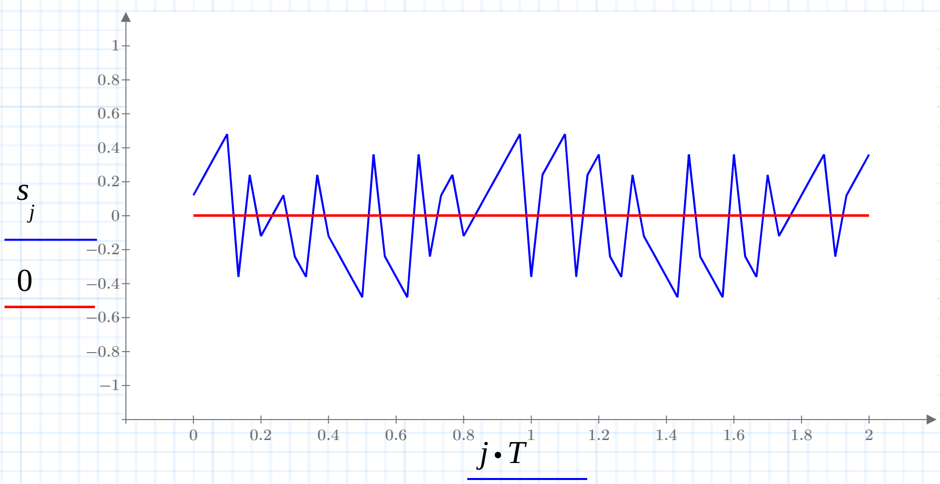

Fig. 10.5 Step sequence: integral multiples of dmin

The error function is given by:

$$err_j := F((j+1) \cdot T) - m_j$$

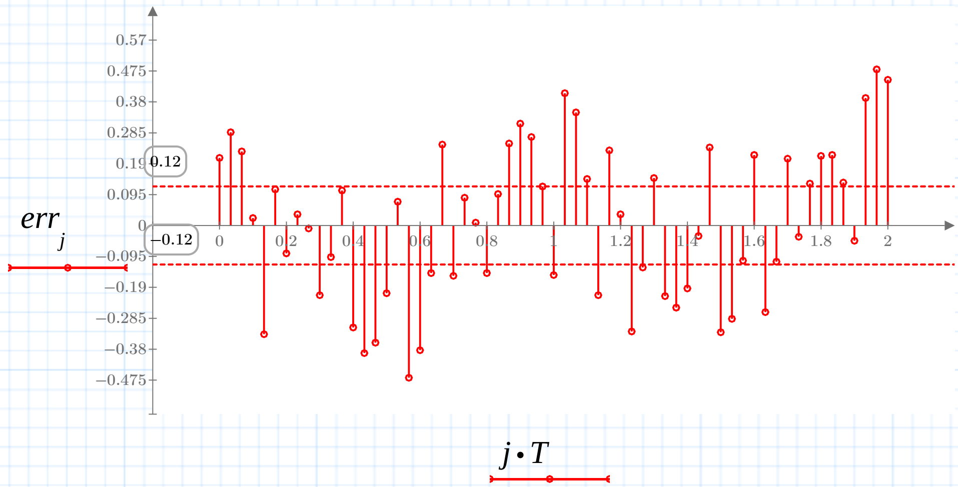

Fig. 10.6 The error signal

Reconstructing the Signal

The input signal can be reconstructed by lowpass filtering of the step approximation. Here we use the output from adaptive delta modulation.

Define the filter coefficients or load them into Mathcad from another file and store in a vector **R** :

$$R := \begin{bmatrix} 1.101 \cdot 10^{-19} \\ 0.001149 \\ 0.003691 \\ 0.002619 \\ -0.009559 \\ -0.02874 \\ -0.02846 \\ 0.02491 \\ \vdots \end{bmatrix}$$

To expand the display of the vector **R**, position your cursor at the bottom of the vector so that you see the dual-sided arrow, then drag your mouse down.

$N := \text{length}(R)$ $\quad n := 0..N-1$ $\quad N = 21$

The array **R** contains filter coefficients for a length-21 lowpass filter. **Y** is set equal to the step-approximation vector **m**. Fill out the filter and signal vectors with zeros:

$X_{N+j} := 0$ $\quad Y := m$

$K := f \cdot L + N + 1$ $\quad Y_{f \cdot L + 1 + n} := 0$

$s := 0 .. K - 2$ $\quad\quad t := 0 .. K - 2$

Convolution of filter with delta-modulated signal:

$$C_t := \sum_s \left(X_s \cdot Y_{\mathrm{mod}(t - s + K, K)} \right)$$

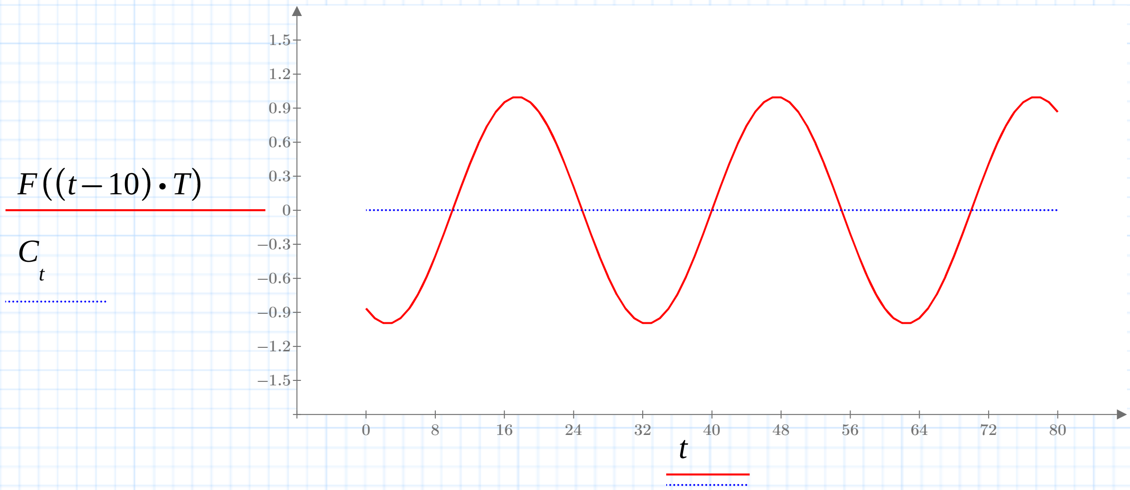

Fig. 10.7 Filtered output and the original signal, shifted to match the filter delay

See **Section 8: Convolution and Deconvolution** for other sampling and convolution techniques, including FFTs.