Introduction to Signal Smoothing

Signal smoothing is the process of reducing noise and irregularities in a signal to highlight important patterns or trends. In electronics, signals often suffer from high-frequency noise and other distortions, making it essential to apply smoothing techniques.

1 Why Smooth Signals?

Smoothing pulses in electronic signals serves several purposes:

- **Noise Reduction**: Reduces high-frequency noise, making the signal clearer.
- **Improved Signal Quality**: Enhances the overall quality of the signal, making it easier to analyze.
- **Signal Integrity**: Preserves the integrity of the signal for accurate data transmission and processing.
- **Preventing Signal Artifacts**: Reduces artifacts that might interfere with the accurate detection of signal transitions.

By smoothing electronic signals, we can improve their clarity and reliability, which is crucial for various applications in signal processing, communication, and data analysis.

2 Methods of Smoothing

Several methods can be used to smooth signals, including:

2.1 Moving Average Filter

A simple and commonly used method. It calculates the average of a fixed number of samples around each point in the signal.

2.2 Gaussian Filter

A weighted moving average where the weights are determined by a Gaussian function. This method preserves the signal better than a simple moving average.

2.3 Exponential Smoothing

Gives more weight to recent observations, making it useful for signals where recent data points are more relevant.

2.4 `tanh` Smoothing

Using the hyperbolic tangent function to smooth transitions in the signal, as discussed earlier. It provides a smooth and continuous transition between states.

3 `tanh` Smoothing

The tanh function is particularly useful for smoothing because of its smooth and continuous nature. It can be applied to pulses as follows:

3.1 Smoothing a Rectangular Pulse

A rectangular pulse with sharp transitions can be smoothed using the tanh function to create smoother edges.

3.2 Smoothing a Trapezoidal Pulse

A trapezoidal pulse with linear rise and fall times can be smoothed by applying the tanh function to the rising and falling edges.

4 Introduction

In this paper we will talk about tanh and quadratic smoothing since they are most popular in the field of electronics and circuit simulations.Figures , examples , code and schematics as well as simulation results will be provided as well.The simulation was done is Spectrum's MicroCap12. The following chapter is about tanh(x) smoothing including mathematical derivations and examples. The chapter after will talk about quadratic smoothing.

5 The $\tanh$ Function

The hyperbolic tangent function is defined as:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

It smoothly transitions from -1 to 1 as x moves from $-\infty$ to $+\infty$, with a steep slope around $x = 0$.

6 Smoothing a Pulse

To smooth a rectangular pulse starting at $t = t_0$ and ending at $t = t_1$, we can use the following formula:

$$f(t) = \frac{1}{2}\left(\tanh\left(k(t - t_0)\right) - \tanh\left(k(t - t_1)\right)\right)$$

Here, k is a constant that controls the sharpness of the transition.

7 Smoothing a Trapezoid Impulse

For a trapezoid impulse with linear rise and fall, we smooth the edges as follows:

$$f(t) = \begin{cases} 0 & t < t_0 \\ \frac{1}{2}\left(1 + \tanh\left(k(t - t_0)\right)\right) & t_0 \leq t < t_1 \\ 1 & t_1 \leq t \leq t_2 \\ \frac{1}{2}\left(1 - \tanh\left(k(t - t_3)\right)\right) & t_2 < t \leq t_3 \\ 0 & t > t_3 \end{cases}$$

In this formulation:

- t_0 is the start of the rise.
- t_1 is the end of the rise.
- t_2 is the start of the fall.
- t_3 is the end of the fall.

8 Example

In this part, we will smooth a trapezoid impulse with specific parameters using the tanh function. The parameters are as follows:

- Pulse period: 5 μs
- Pulse width: 100 ns
- Rise time: 50 ns
- Fall time: 50 ns
- Start time (time delay): 100 ns
- Amplitude: 5 V

A trapezoid impulse is characterized by its linear rise and fall times. Given the parameters:

- The pulse starts at $t_0 = 100$ ns.
- The rising edge ends at $t_1 = t_0 + 50$ ns.
- The falling edge starts at $t_2 = t_1 + 100$ ns.
- The pulse ends at $t_3 = t_2 + 50$ ns.

The tanh function provides a smooth transition between states. We define the smoothed trapezoid impulse using the following piecewise function:

$$f(t) = \begin{cases} 0 & t < t_0 \\ \frac{A}{2}\left(1 + \tanh\left(k(t - t_0)\right)\right) & t_0 \leq t < t_1 \\ A & t_1 \leq t < t_2 \\ \frac{A}{2}\left(1 - \tanh\left(k(t - t_2)\right)\right) & t_2 \leq t < t_3 \\ 0 & t \geq t_3 \end{cases}$$

Adjusting k allows you to tailor the smoothing effect according to specific requirements:

- **Higher** k: Suitable for applications where rapid transitions between signal states are needed, emphasizing clarity and immediacy in signal changes.
- **Lower** k: Ideal for scenarios where maintaining signal continuity and minimizing abrupt changes are prioritized, ensuring smoother and more natural transitions.

The images below visually represent the effect of different values of k on the smoothed trapezoidal impulse, demonstrating how parameter tuning impacts the signal's behavior.

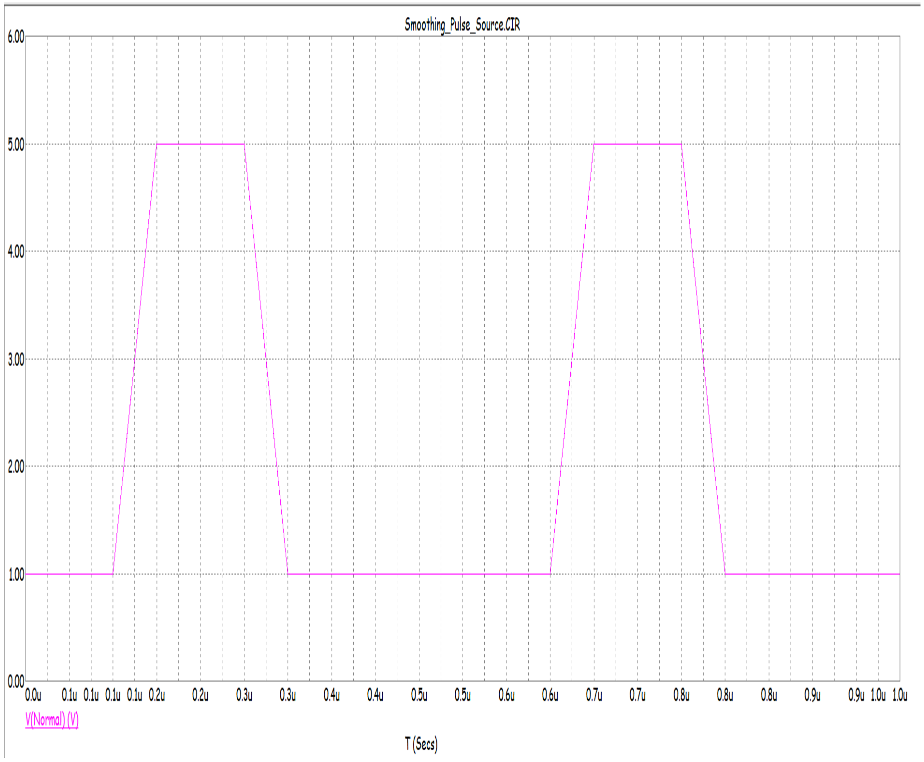

Figure 1: Trapezoidal input signal

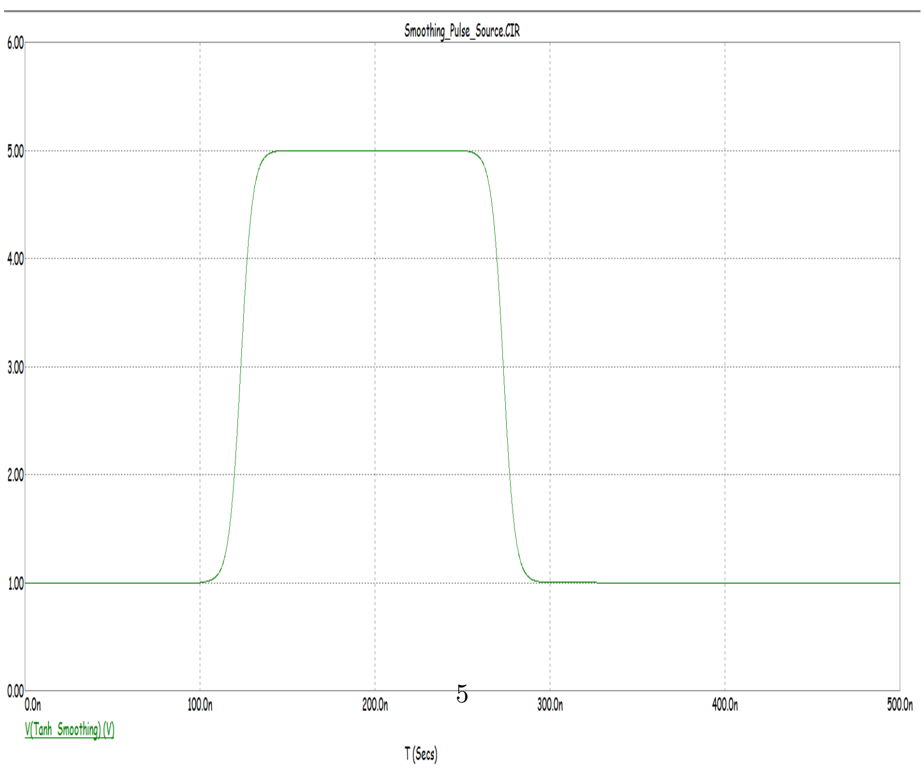

Figure 2: Tanh signal calculated in the above example

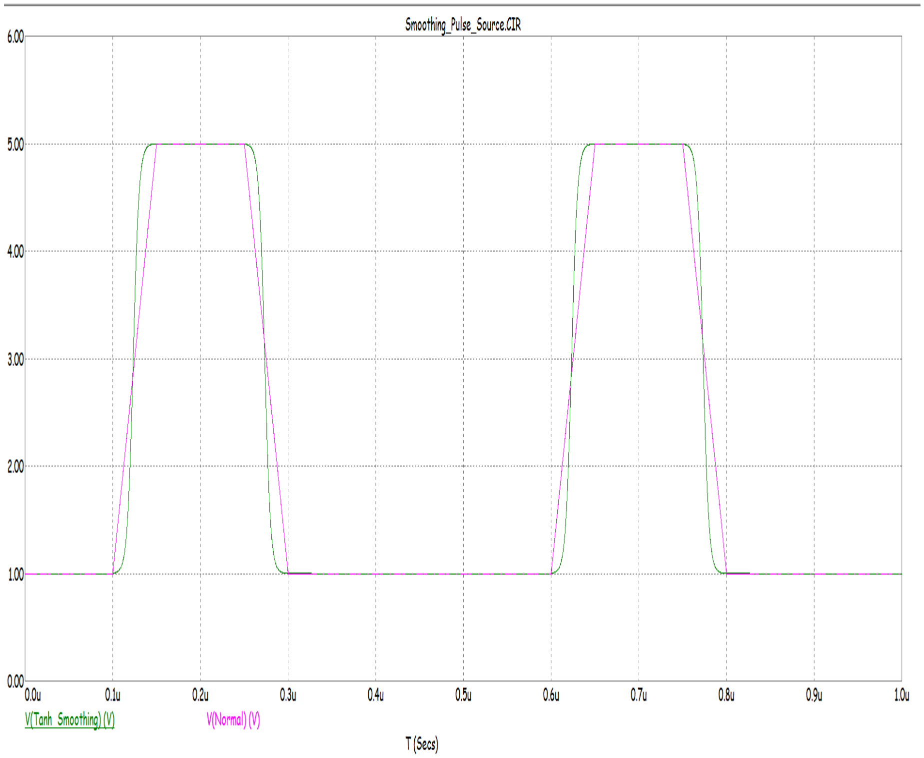

Figure 3: The above figure shows how good is our approximation

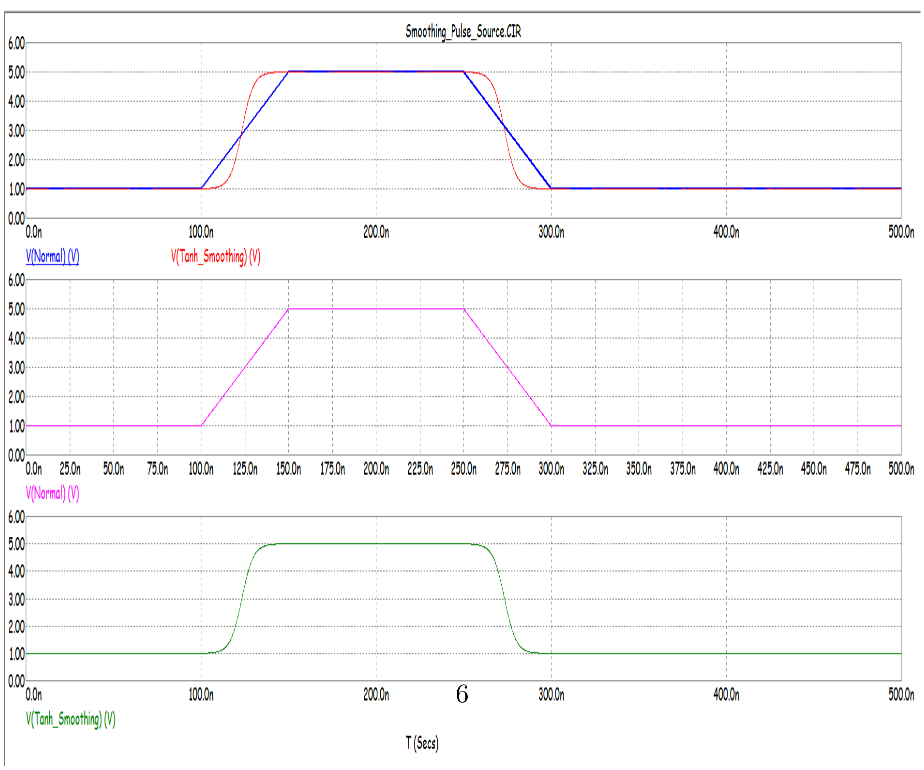

Figure 4: All signals

9 Quadratic Smoothing

Quadratic smoothing is another technique that fits a quadratic function to segments of the signal to smooth out irregularities. It offers more flexibility and can better capture the curvature of the signal compared to linear methods.

9.1 Formulation

The basic idea behind quadratic smoothing is to fit a quadratic function $f(t) = at^2 + bt + c$ to segments of the signal $y(t)$. This involves determining the coefficients a, b, and c for each segment such that the resulting smoothed signal closely follows the original data points.

Given data points (t_i, y_i) for $i = 1, 2, \ldots, n$, the quadratic smoothing approach minimizes the total variation (often the sum of squared residuals) between the original data points and the fitted quadratic functions.

9.1.1 Mathematical Formulation

For a segment of the signal between points t_i and t_{i+1}, the quadratic function can be expressed as:

$$f(t) = a_i t^2 + b_i t + c_i$$

where a_i, b_i, and c_i are coefficients to be determined.

To minimize the deviation between the smoothed signal $f(t)$ and the original signal $y(t)$, typically a least squares approach is used, where the objective function to minimize is:

$$\sum_{i=1}^{n-1} (y_i - (a_i t_i^2 + b_i t_i + c_i))^2$$

subject to continuity constraints at the segment boundaries t_i and t_{i+1}.

9.1.2 Steps in Quadratic Smoothing

1. **Segmentation**: Divide the signal into segments where quadratic functions will be fitted. This can be based on certain criteria such as time intervals or significant changes in the signal.

2. **Fit Quadratic Functions**: For each segment, determine the coefficients a_i, b_i, and c_i by minimizing the sum of squared residuals between the fitted quadratic function and the actual data points within that segment.

3. **Continuity Constraints**: Ensure smooth transitions between segments by imposing continuity conditions at the boundaries where segments meet. This helps in achieving a continuous and smooth overall curve.

9.2 Applications of Quadratic Smoothing

Quadratic smoothing is useful in scenarios where:

- The signal exhibits nonlinear behavior or curvature.
- High precision in signal representation is required.
- Smooth transitions between signal segments are desired.

9.3 Pros and Cons of Quadratic Smoothing

Pros:

- Captures nonlinear variations in the signal.
- Provides smoother transitions compared to linear methods.
- Can be tailored to fit specific signal characteristics.

Cons:

- More computationally intensive compared to linear smoothing techniques.
- Requires careful parameter tuning to avoid overfitting or underfitting.
- May introduce slight lag in signal response due to quadratic fitting.

9.4 Example: Quadratic Smoothing of Trapezoidal Signal

Let's apply quadratic smoothing to the original trapezoidal signal defined earlier. The parameters of the trapezoidal signal are as follows:

- Pulse period: $5\mu s$
- Pulse width: $100ns$
- Rise time: $50ns$
- Fall time: $50ns$
- Start time (time delay): $100ns$
- Amplitude: $5V$

Given these parameters, the trapezoidal signal can be expressed as a piecewise function:

$$f(t) = \begin{cases} 0 & \text{for } t < 100\,\text{ns} \\ 5 & \text{for } 100\,\text{ns} \leq t < 150\,\text{ns} \\ 5 & \text{for } 150\,\text{ns} \leq t < 250\,\text{ns} \\ 0 & \text{for } t \geq 250\,\text{ns} \end{cases}$$

To apply quadratic smoothing, we fit a quadratic function to segments of the signal. For instance, between $t = 100\,\text{ns}$ and $t = 150\,\text{ns}$, the quadratic smoothing function $g(t)$ can be defined as:

$$g(t) = a(t - t_0)^2 + b(t - t_0) + c$$

where $t_0 = 100\,\text{ns}$, $t_1 = 150\,\text{ns}$, and a, b, c are coefficients determined by fitting the quadratic function to the data points.

The quadratic smoothing process involves:

- Selecting appropriate segments of the signal.
- Computing the coefficients a, b, c for each segment.
- Applying the quadratic function $g(t)$ to smooth out the transitions between segments.

The images below visually represent result of the our example.

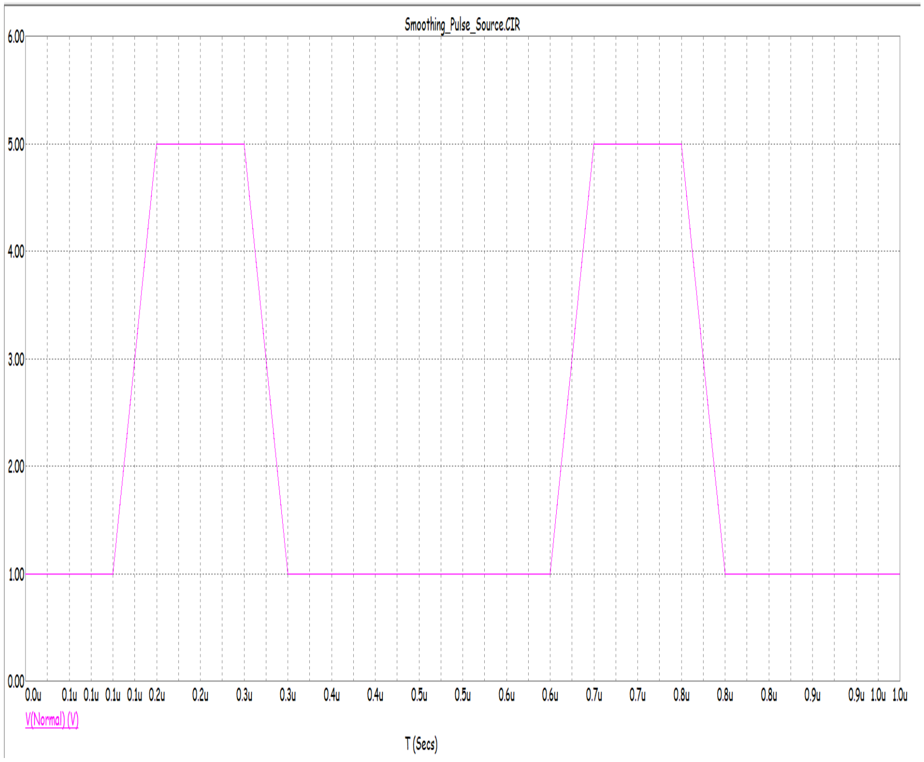

Figure 5: Trapezoidal input signal

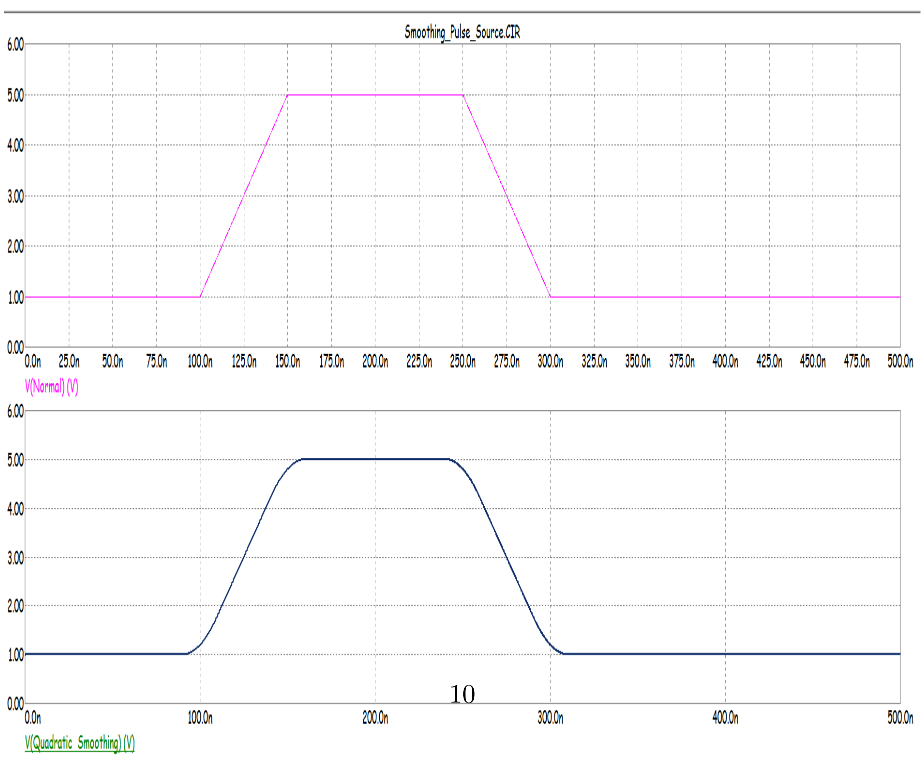

Figure 6: Trapezoidal signal above and it's quadratic approximation below

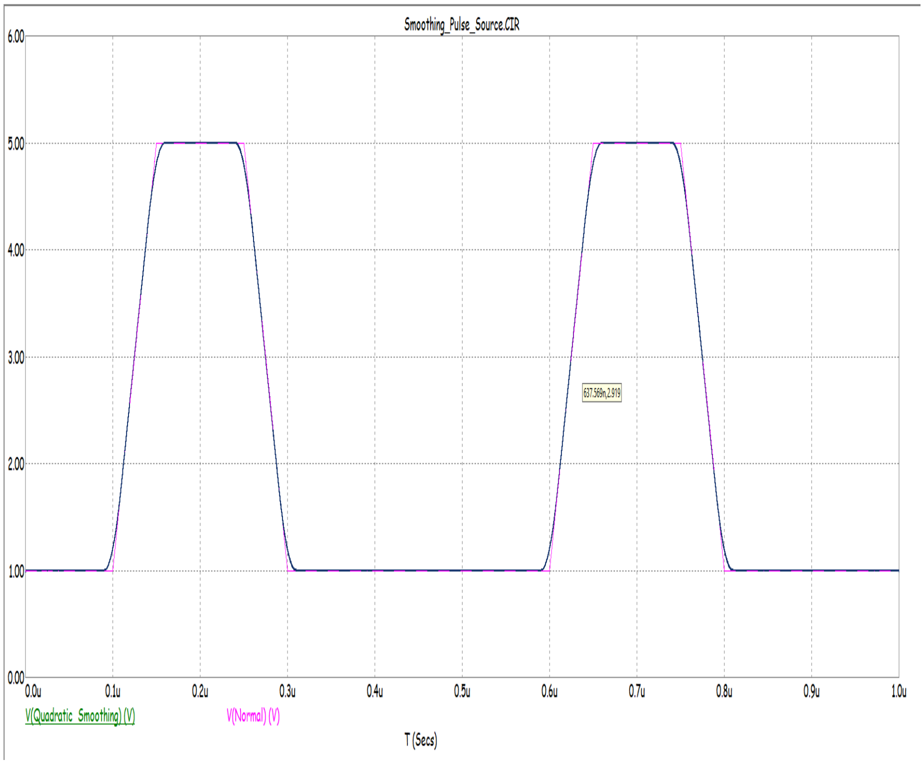

Figure 7: Both signals on the same waveform , illustrating how good approximation actually is

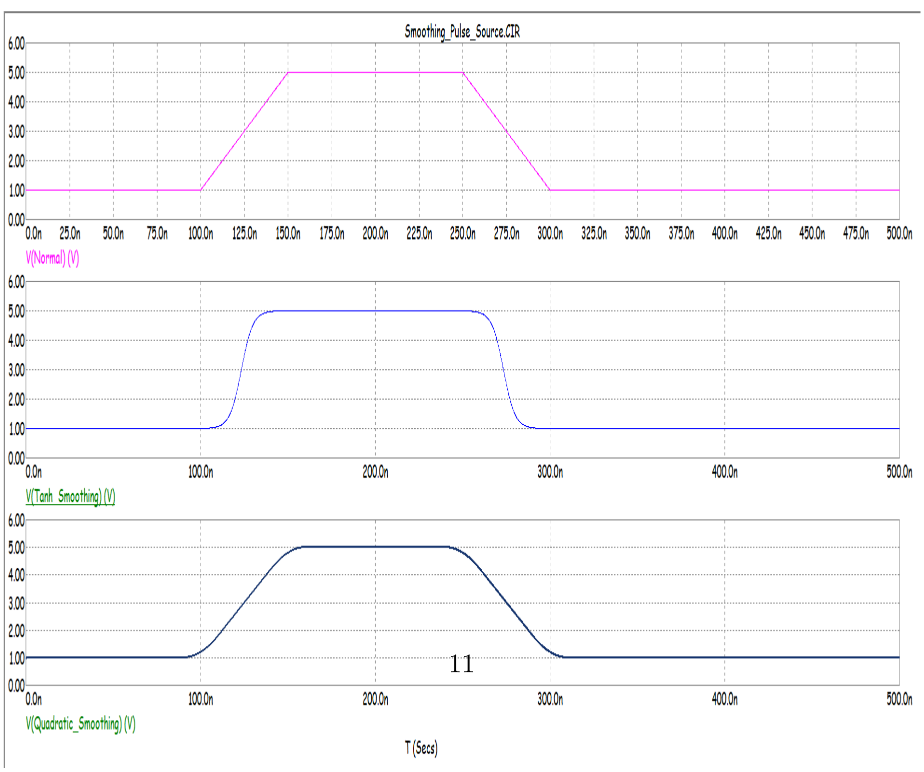

Figure 8: First waveform original signal , second waveform tanh , third waveform quadratic

10 Summary

In conclusion, both tanh smoothing and quadratic smoothing techniques offer distinct advantages depending on the characteristics of the signal and the specific requirements of the application. Engineers and researchers can choose between these methods based on factors such as signal complexity, desired smoothing effect, computational resources, and application-specific needs. Experimentation with different techniques and parameter settings can lead to optimized solutions for signal processing tasks, contributing to advancements in various fields of technology and science.