**Software Proposal to Relieve Voltage Spikes**

This section outlines software-based solutions to mitigate voltage spikes caused by sensor noise or mechanical disturbances in the potentiometer signal. These solutions are intended to be implemented within the door-zone module ECU. Three methods are considered:

1. **Averaging Filter**  
   The averaging method replaces each current sample with the average of a window of past and current samples. This technique is effective in reducing small-magnitude, high-frequency noise and is commonly used in signal processing for potentiometer-based sensors. In this report, we evaluate the performance of moving average filters with window sizes of 4, 10, 20, and 50 samples. The filtered output is then compared to the raw signal to assess noise reduction effectiveness.

### Regarding window size there are rules of thumb: **General Rule of Thumb:**

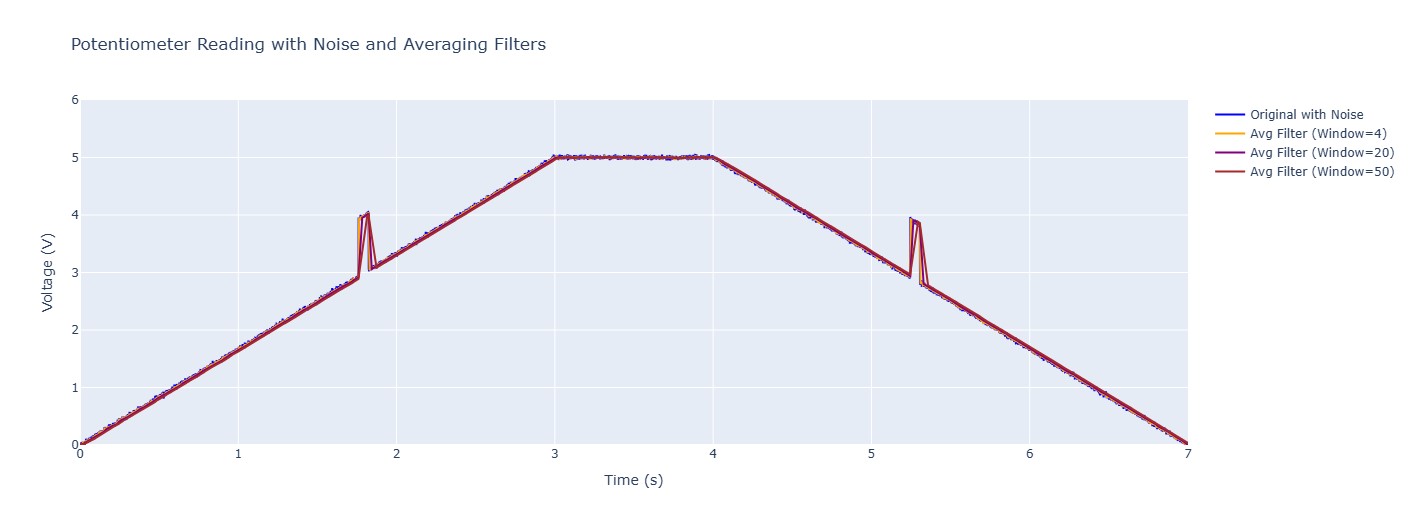
* **Smaller window (e.g., 3–5 samples)**:
  + Less delay (good responsiveness)
  + Less smoothing, may not fully suppress noise
* **Medium window (10–20 samples)**:
  + Good compromise between noise reduction and signal responsiveness
  + Slight delay introduced
* **Large window (30–50+ samples)**:
  + Strong noise reduction
  + High delay — not suitable for real-time control

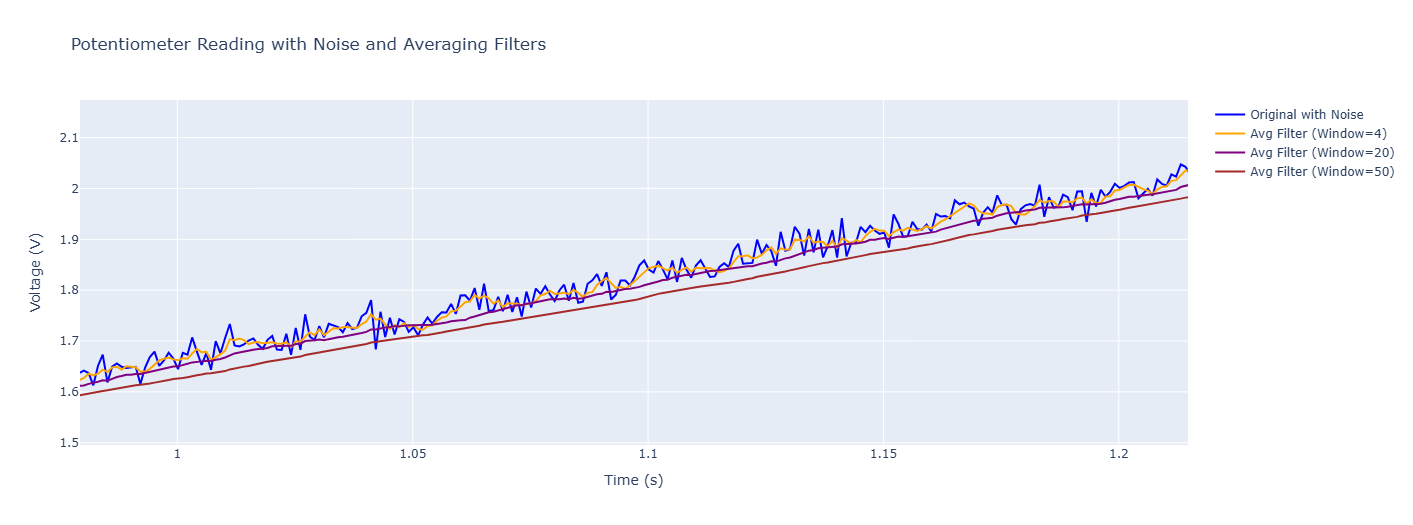
For a window size N, the filtered signal at time i is:

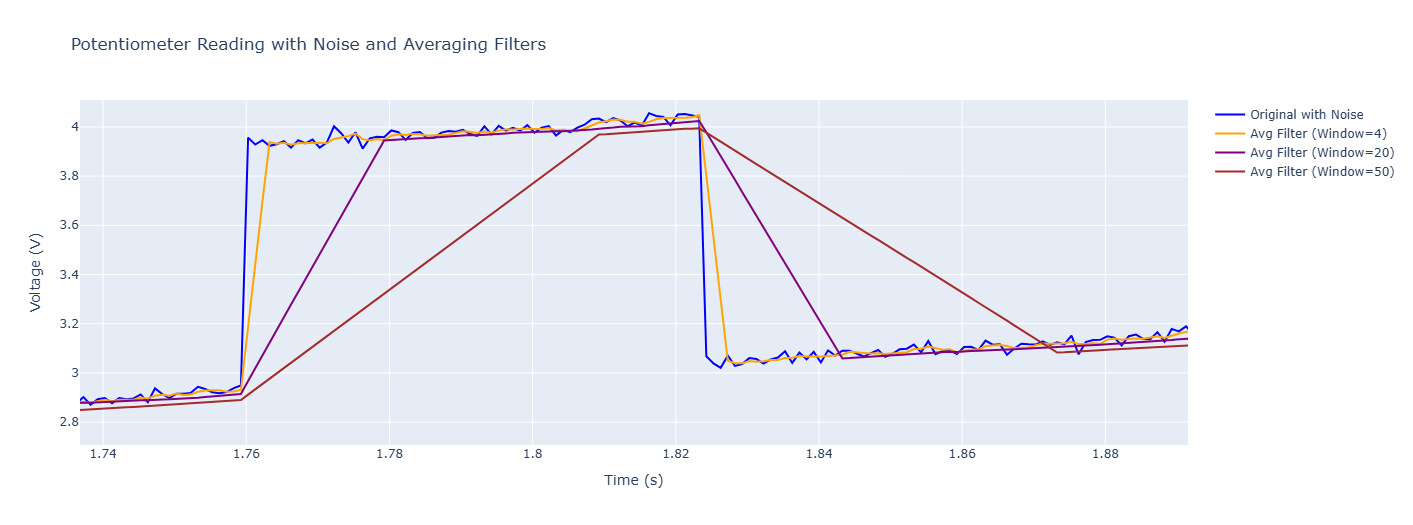
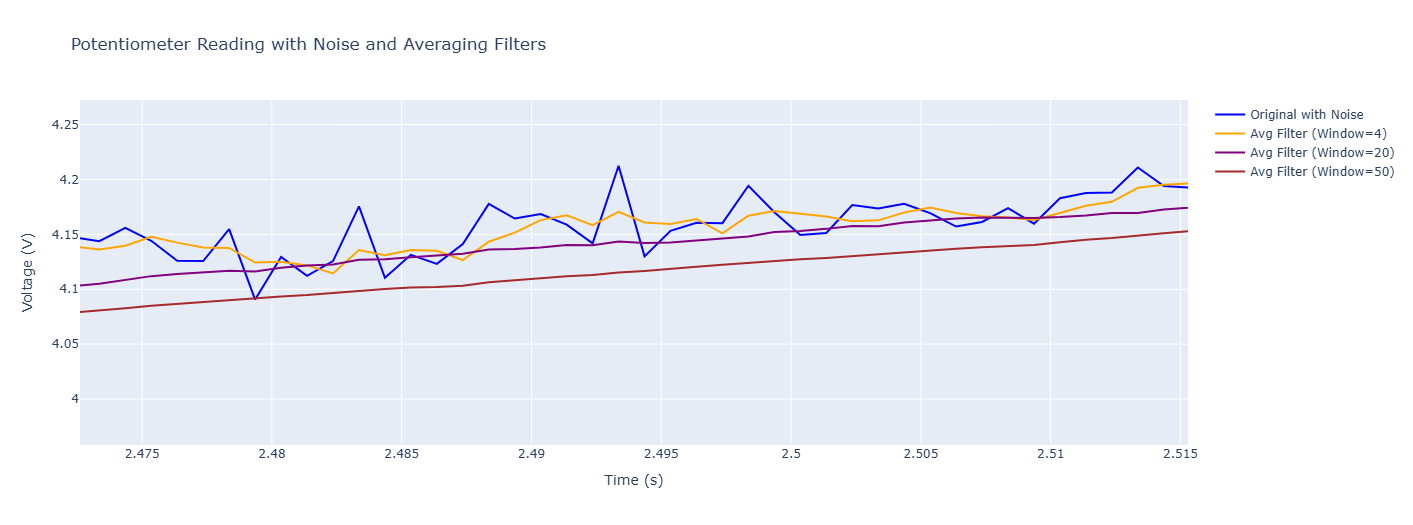
avgi=1N∑k=i−(N−1)/2i+(N−1)/2xk\text{avg}\_i = \frac{1}{N} \sum\_{k = i - (N-1)/2}^{i + (N-1)/2} x\_kavgi​=N1​k=i−(N−1)/2∑i+(N−1)/2​xk​

In practice on microcontrollers, we use a **trailing window** (past samples) instead of centered due to real-time constraints.

avgi=1N∑k=i−N+1ixk\text{avg}\_i = \frac{1}{N} \sum\_{k = i - N + 1}^{i} x\_kavgi​=N1​k=i−N+1∑i​xk​







Obvviously the averaging filter works great in smoothing out smal noise but in high voltage spike it can not be effective. Also another conclusion is that bigger windows lead to bigger delay and more smooth curve.  
  
  
Median casual filter

The median causal filter is a non-linear filtering technique that replaces each sample with the median value of a window of past and current samples. This method excels at mitigating high-magnitude voltage spikes caused by sensor noise or mechanical disturbances in potentiometer signals, as it is robust to outliers compared to the averaging filter. In this report, we evaluate the performance of a causal median filter with a window size of 11 samples, chosen to balance noise reduction and responsiveness, and compare its output to the raw signal to assess its effectiveness in spike suppression.

The median filter works by sorting the samples within the window and selecting the middle value. For a window size <math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><mrow><mi>N</mi></mrow><annotation encoding="application/x-tex"> N </annotation></semantics></math>N (which must be odd for a true median), the filtered signal at time <math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><mrow><mi>i</mi></mrow><annotation encoding="application/x-tex"> i </annotation></semantics></math>i is defined as:

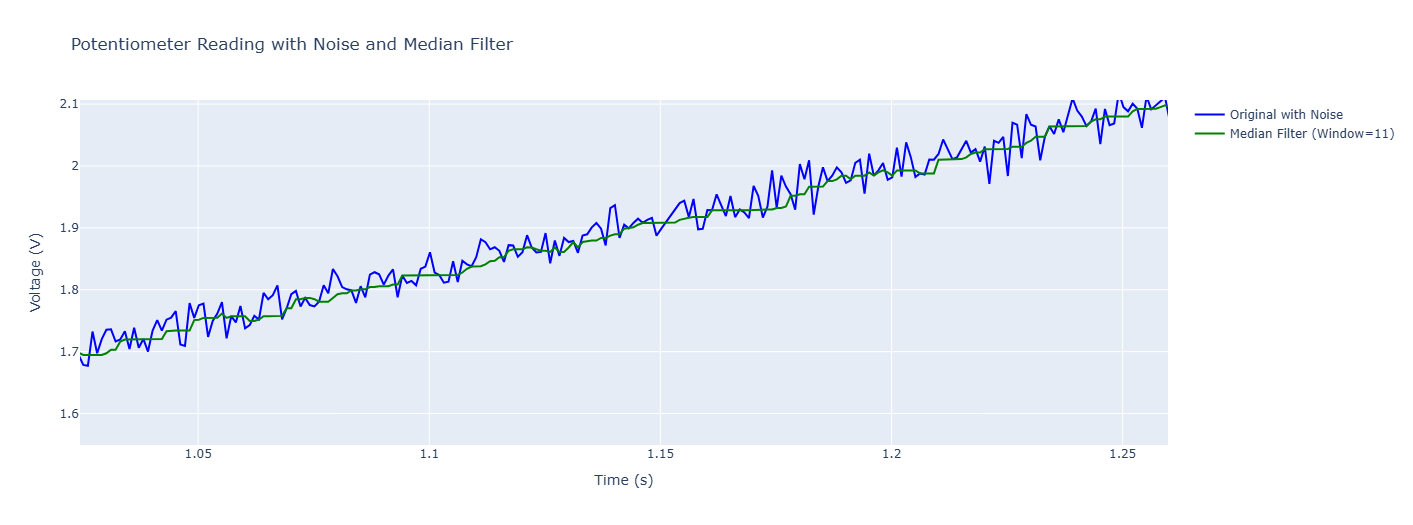
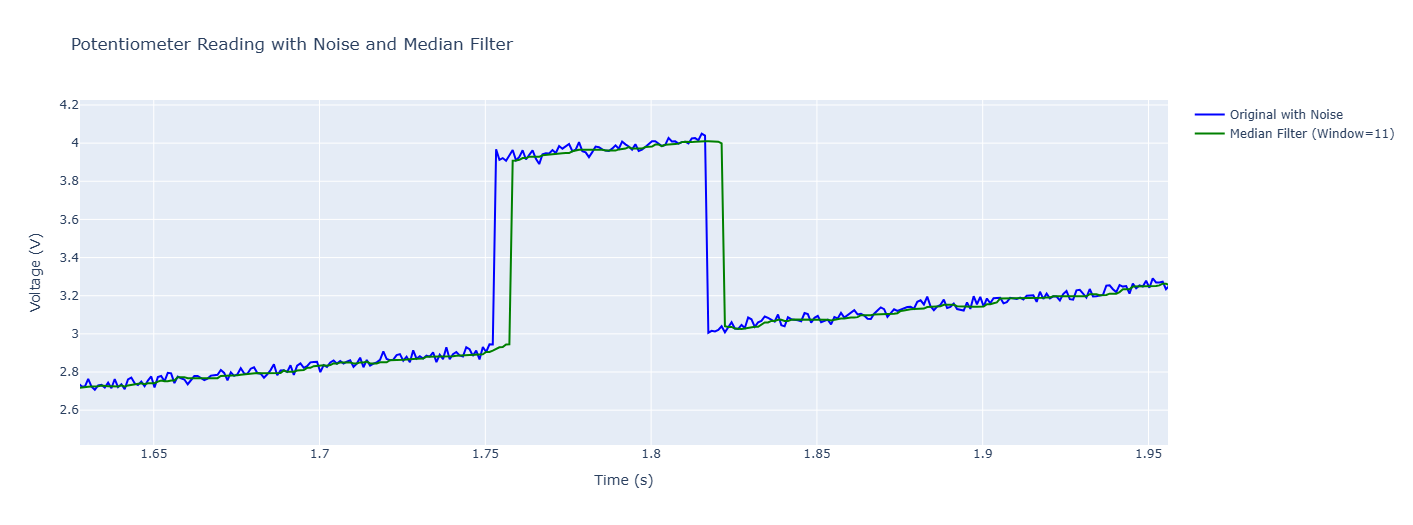
<math xmlns="http://www.w3.org/1998/Math/MathML" display="block"><semantics><mrow><msub><mtext>med</mtext><mi>i</mi></msub><mo>=</mo><mtext>median</mtext><mo stretchy="false">{</mo><msub><mi>x</mi><mi>k</mi></msub><msubsup><mo stretchy="false">}</mo><mrow><mi>k</mi><mo>=</mo><mi>i</mi><mo>−</mo><mi>N</mi><mo>+</mo><mn>1</mn></mrow><mi>i</mi></msubsup></mrow><annotation encoding="application/x-tex">\text{med}\_i = \text{median} \{ x\_k \}\_{k=i-N+1}^{i}</annotation></semantics></math>medi​=median{xk​}k=i−N+1i​

where <math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><mrow><msub><mi>x</mi><mi>k</mi></msub></mrow><annotation encoding="application/x-tex"> x\_k </annotation></semantics></math>xk​ are the input samples, and the window includes the current sample and the <math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><mrow><mi>N</mi><mo>−</mo><mn>1</mn></mrow><annotation encoding="application/x-tex"> N-1 </annotation></semantics></math>N−1 previous samples to ensure causality, which is essential for real-time implementation on microcontrollers. Unlike the averaging filter, which smooths the signal by blending values, the median filter selects a representative value that is minimally affected by extreme outliers, such as voltage spikes, making it particularly effective for spike rejection.

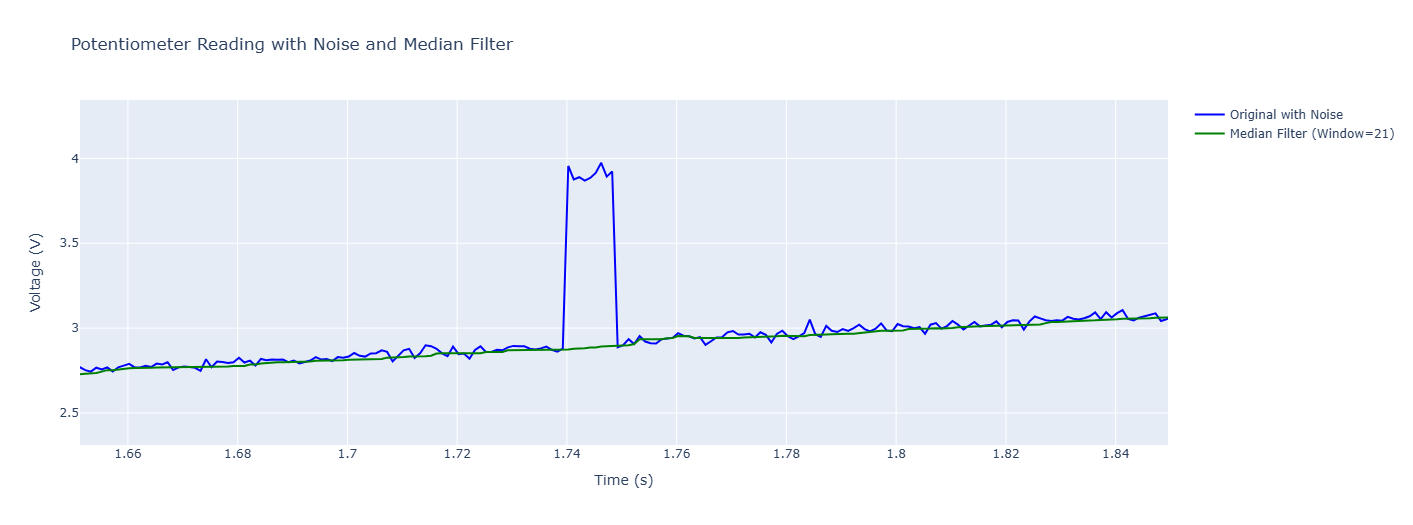
Regarding window size, the choice of <math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><mrow><mi>N</mi></mrow><annotation encoding="application/x-tex"> N </annotation></semantics></math>N impacts the filter's performance:

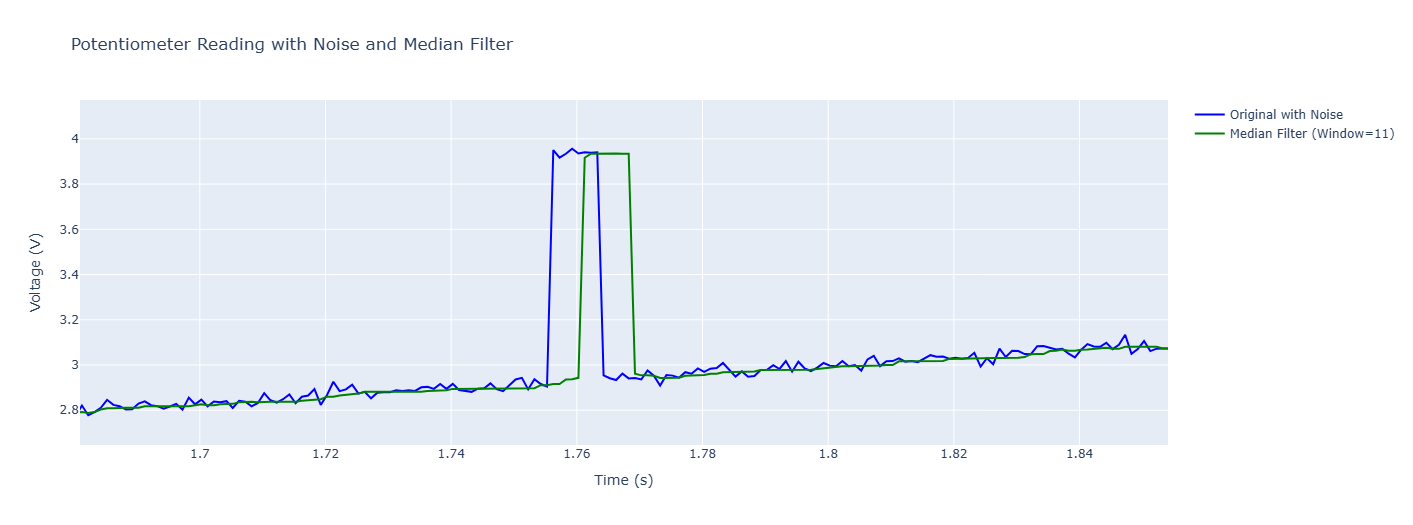
* **Smaller window (e.g., 3–5 samples):**
  + Minimal delay, suitable for real-time applications.
  + Limited spike suppression, as fewer samples reduce the likelihood of excluding outliers.
* **Medium window (e.g., 7–11 samples):**
  + Good balance between spike rejection and responsiveness.
  + Moderate delay, acceptable for most control systems.
* **Larger window (e.g., 13–21 samples):**
  + Strong spike suppression, as more samples increase the robustness to outliers.
  + Increased delay, which may affect real-time performance in fast-changing signals.

In conclusion, the median causal filter is highly effective for reducing voltage spikes in potentiometer signals, outperforming the averaging filter in scenarios with large outliers. However, its non-linear nature may preserve sharp signal transitions better but can introduce minor artifacts in smooth regions. The choice of a window size like 11 samples provides a practical compromise for the door-zone module ECU, ensuring robust spike mitigation with acceptable delay for real-time control.

The median casual filter is a good option if the length of the window is at least twice as spike duration. Let’s have a look a shorter spike with window size 21.



And if we make the window size 11, the spike will not be eliminated:   
  


So based on the fretting issue magnitude and the duration of spike we might be able to use the median casual filter and smooth out the signal

Kalman filter

1. Kalman Filter on Median-Filtered Signal The Kalman filter is a recursive, model-based algorithm that optimally estimates the state of a dynamic system from noisy measurements. In this report, we apply a Kalman filter to the output of the median causal filter (with an 11-sample window) to further refine the potentiometer signal by reducing residual noise and improving smoothness while maintaining responsiveness. This combined approach leverages the median filter’s ability to suppress high-magnitude voltage spikes and the Kalman filter’s strength in handling small-magnitude noise and tracking the underlying signal trend. We evaluate the performance of this method in the context of the door-zone module ECU, comparing the filtered output to the raw and median-filtered signals to assess its effectiveness.

The Kalman filter models the potentiometer signal as a system with a state vector comprising the voltage and its rate of change (velocity). For a sampling frequency of 1000 Hz (1 ms per sample), the state transition model assumes the voltage evolves as: [ x\_k = \begin{bmatrix} v\_k \ \dot{v}k \end{bmatrix}, \quad x\_k = F x{k-1} + w\_k, \quad F = \begin{bmatrix} 1 & 0.001 \ 0 & 1 \end{bmatrix} ] where ( v\_k ) is the voltage, ( \dot{v}\_k ) is the velocity, ( F ) is the state transition matrix (with 0.001 s as the time step), and ( w\_k ) is process noise with covariance ( Q ). The measurement model relates the observed median-filtered voltage ( z\_k ) to the state: [ z\_k = H x\_k + v\_k, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix} ] where ( H ) extracts the voltage, and ( v\_k ) is measurement noise with covariance ( R ). The Kalman filter iteratively predicts the next state and updates it based on the measurement, using a gain ( K ) to balance model predictions and measurement reliability.

To enhance spike rejection, a threshold-based mechanism is incorporated. If the residual (difference between the measured and predicted voltage) exceeds a predefined threshold (e.g., 0.1 V), the measurement is considered an outlier (residual spike not fully suppressed by the median filter), and the filter relies on the predicted state, bypassing the update step. Typical parameters include process noise covariance ( Q = 0.001 ) and measurement noise covariance ( R = 0.01 ), tuned to balance responsiveness and noise reduction.

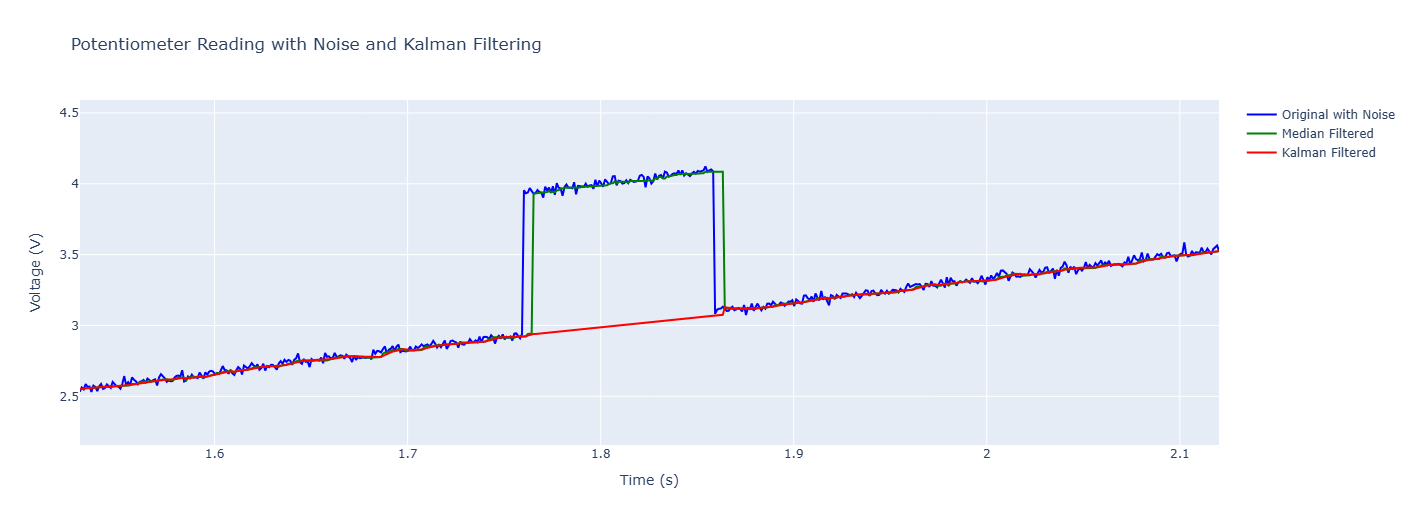
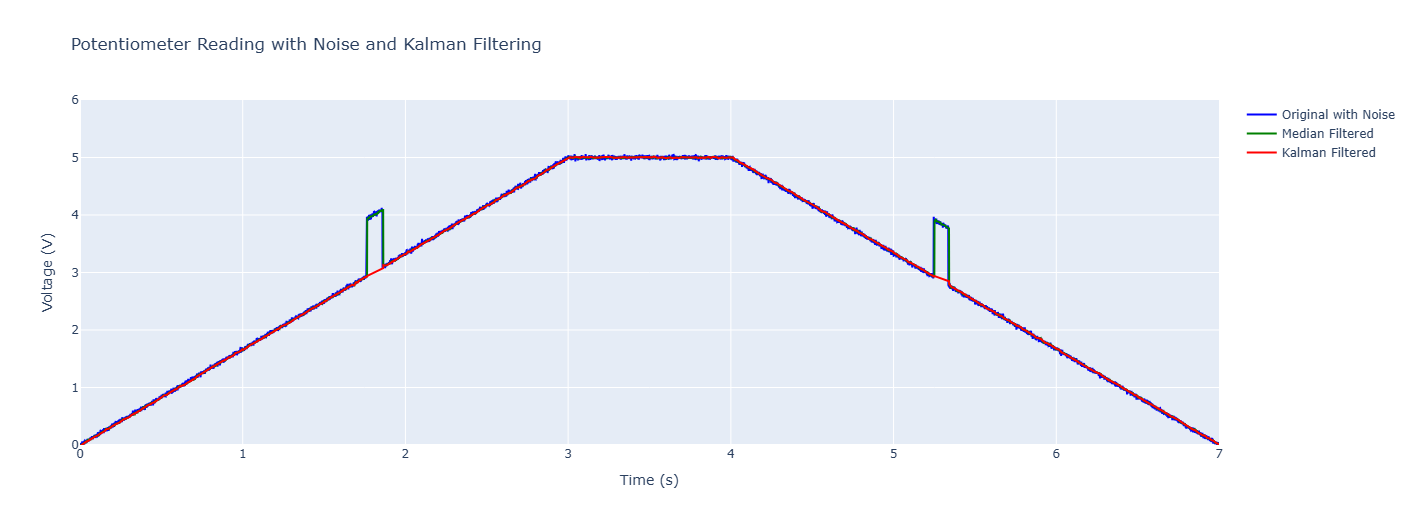
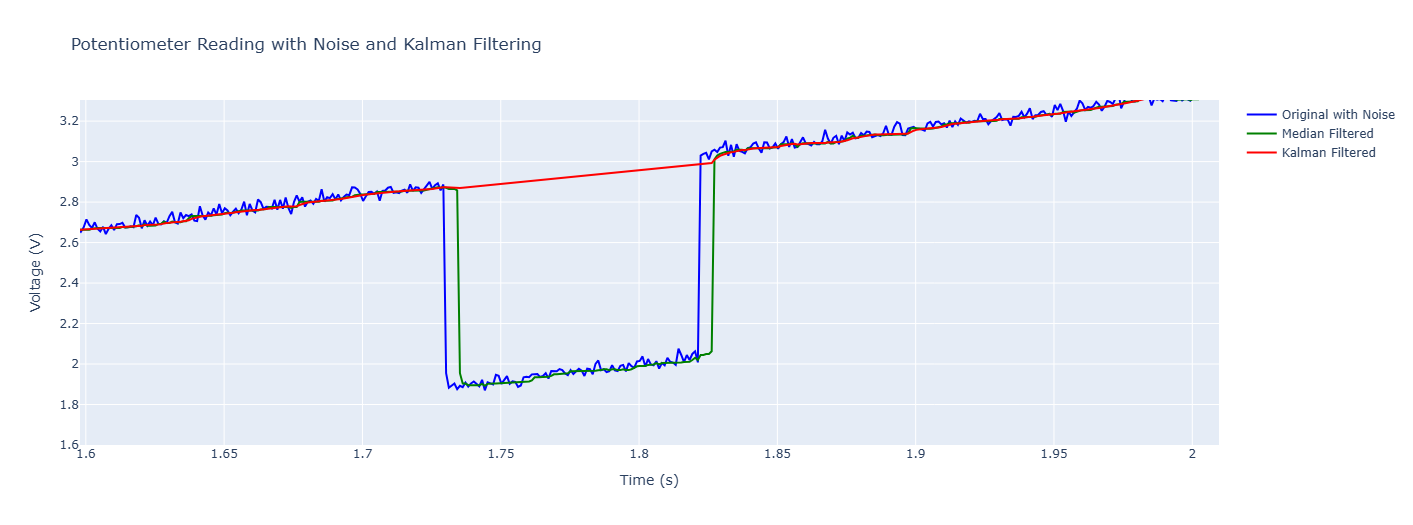
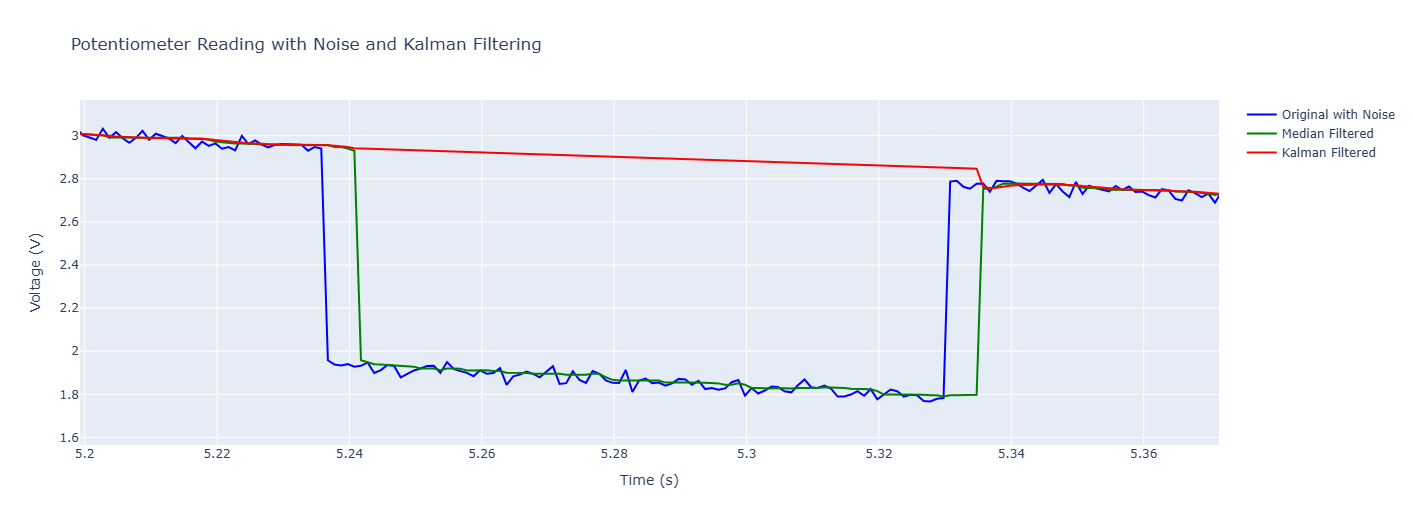
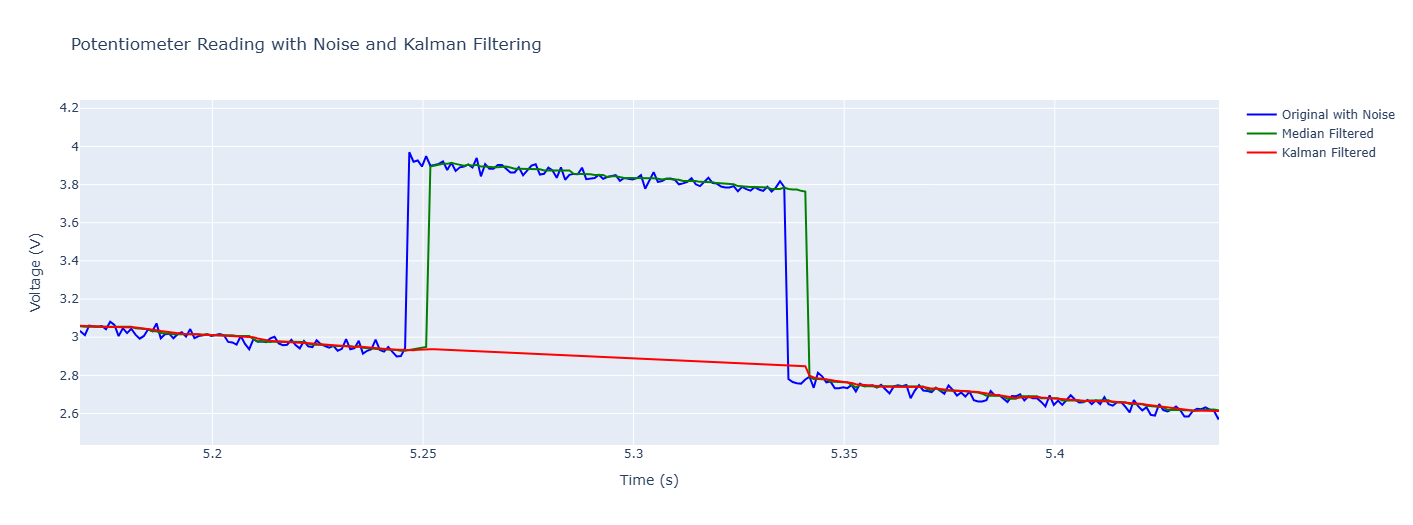
The Kalman filter’s performance depends on the quality of the median-filtered input and the tuning of ( Q ), ( R ), and the spike threshold:

* **High ( Q )**: Allows the filter to adapt quickly to signal changes but may pass more noise.
* **Low ( Q )**: Produces smoother output but may lag during rapid signal transitions.
* **High ( R )**: Reduces trust in measurements, relying more on the model, improving smoothness.
* **Low ( R )**: Trusts measurements more, improving responsiveness but potentially retaining noise.
* **Spike threshold**: A value like 0.1 V ensures rejection of residual outliers while accepting normal signal variations (noise standard deviation of 0.02 V).

For the evaluated signal (spike duration 50–100 ms, 1 V magnitude), the 11-sample median filter (11 ms) partially suppresses spikes, as it is smaller than twice the spike duration (100–200 samples). The Kalman filter further smooths the signal, reducing residual noise and minor artifacts introduced by the median filter. However, for long spikes (e.g., 100 ms), some distortion may remain if the median filter fails to fully eliminate the spike, as the Kalman filter assumes a relatively clean input.

In a test case with a shorter spike (e.g., 10 ms or 10 samples), the 11-sample median filter is closer to the ideal window size (twice the spike duration = 20 samples), improving spike suppression. The Kalman filter then effectively smooths the remaining noise, producing a signal closely aligned with the true potentiometer trend. The combined approach introduces a minimal additional delay (on the order of 1–2 ms due to the recursive nature of the Kalman filter), making it suitable for real-time control.

In conclusion, applying a Kalman filter on top of an 11-sample median filter significantly enhances signal quality for the door-zone module ECU. The median filter reduces high-magnitude spikes, while the Kalman filter smooths residual noise and tracks the signal’s dynamic behavior. This method outperforms the averaging filter for spike rejection and provides smoother output than the median filter alone. However, for optimal performance with 50–100 ms spikes, a larger median window (e.g., 101–201 samples) is recommended, though this increases delay. For shorter spikes (e.g., 10 ms) or fretting-induced disturbances, the 11-sample window with Kalman filtering offers a practical compromise, balancing spike suppression, noise reduction, and real-time responsiveness.



How to implement it  
  
