Sequential Change Detection of Simulated Gaussian Signals using the CuSum Algorithm

Date: 10/10/2021
Authors: Suleyman BARTHE-SUKHERA
Instructor: P. GRANJON

I. Abstract

This research project applied the CuSum algorithm to a simulated sensor with Gaussian white noise and analyzed the algorithm's performance with regards to a False Alarm Rate (FAR) and Conditional Average Delay Detection ($CADD$). With a theoretical $FAR = 1 \times 10^{-3}$, a threshold of $\cong 6.91$ was derived. Validating this threshold on a data set composed of two signal states H_0 and H_1 which only have a different mean value ($\mu_0 = 0$ and $\mu_1 = 1.5$), was found to have an experimental FAR of 1.08×10^{-4}, and $CADD$ of 20.6. It is interesting to note that the FAR is an order of magnitude lower than defined by the user, implying that the algorithm is robust.

The $CADD$ value is unfortunately meaningless in this context as there is no relation to a time-dependency, but that translation is available to the user depending on the sample rate of the sensor the user chooses. It is important to note that this algorithm assumes the user already knows the variance of the sensor and mean value of the two signals.

II. Introduction

Detecting the change in a noisy signal can be a useful tool in classifying system behavior patterns. A change in signal can occur due to a theoretically infinite number of factors but will generally present themselves as a change in either mean value, or variance. This experiment assumes that the noisy signal utilized for classification is Gaussian and that the mean value changes abruptly, but that the variance of both signals stay the same. It is further assumed that the mean values of both states, as well as the variance, is known. For this experiment, $\mu_0 = 0$ and $\mu_1 = 1.5$, and $\sigma^2 = 4$.

The goal of this experiment is to analyze and visualize the efficacity of the CuSum algorithm in detecting the change in mean values, μ_0 and μ_1. The CuSum algorithm works by iteratively calculating the instantaneous log-likelihood ratio, integrating over all known log-likelihood ratios to decide if a change has occurred, and if so, calculating an estimation of when the abrupt change occurred.

III. Equations and Assumptions

We assume that the sensor can be modeled as:

$$\text{Equation 1: } H_i[x] = \mu_i + b_i[x]$$

Where $H_i[x]$ is the value provided by the noisy sensor in state i and at sample instance x; μ_i is the mean value of the sensor in state i; and $b_i[x]$ is the Gaussian noise of the sensor in state i, as defined below:

$$\text{Equation 2: } b_i[x] \sim \mathcal{N}(\mu_i, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Let the difference between μ_0 and μ_1, δ, be defined as:

$$\text{Equation 3: } \delta = \mu_1 - \mu_0$$

The instantaneous (natural) log-likelihood ratio is defined as:

$$\text{Equation 4: } s[x] = \ln\left(\frac{p_H(H[x], \mu_0+\delta)}{p_H(H[x], \mu_0)}\right) = \frac{\delta}{\sigma^2}\left(H[x] - \mu_0 - \frac{\delta}{2}\right)$$

Where $H[x]$ is the sensor value at sample x.
The discretized integral of the instantaneous log-likelihood ratios, S, is defined as:

$$\text{Equation 5: } S[x] = S[x-1] + s[x]$$

Let the decision function, g, be defined as:

$$\text{Equation 6: } g[x] = \max(s[x], 0)$$

Let the False Alarm Rate (FAR) be defined as:

$$\text{Equation 7: } FAR = \frac{1}{E_{H_0}(s[x])}$$

Let the threshold be defined as:

$$\text{Equation 8: } threshold = |\ln(FAR)|$$

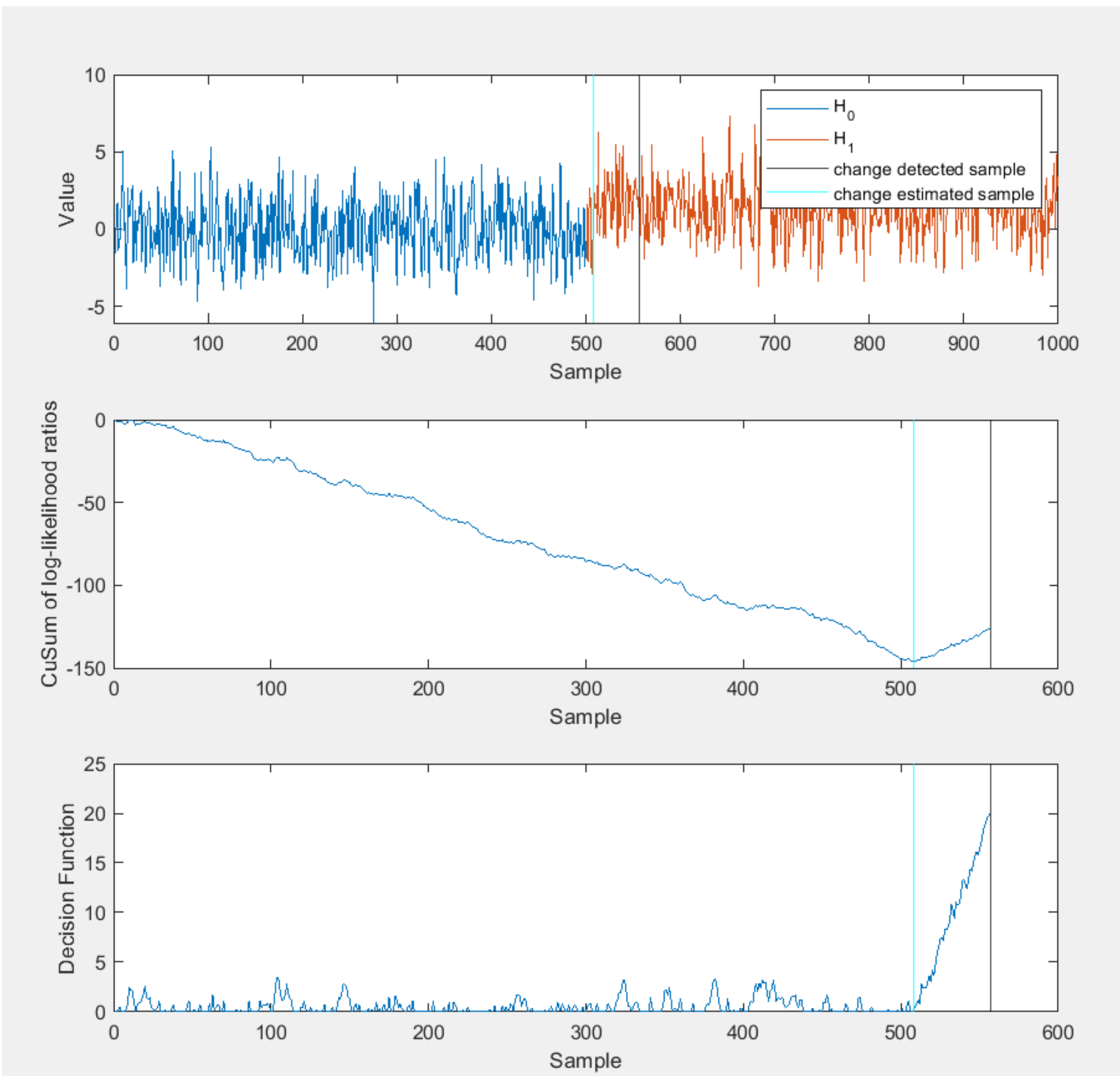
Let the Conditional Average Delay Detection (CADD) be defined as:

$$\text{Equation 9: } CADD = \frac{threshold}{E_{H_1}(s[x])}$$

IV. Results

2: CuSum Algorithm

An online algorithm was constructed using Equations 4, 5, and 6. A change would be detected once $g[x]$ surpassed a certain threshold, which was arbitrarily set to 20 for this example. Signal values of a noisy sensor was simulated in two states, state "0" $H_0[x]$ and state "1" $H_1[x]$, using $\mu_0 = 0$ and $\mu_1 = 1.5$ respectively, along with a common variance of $\sigma^2 = 4$. 500 samples of H_0 and H_1 were respectively and sequentially generated, yielding the following results:

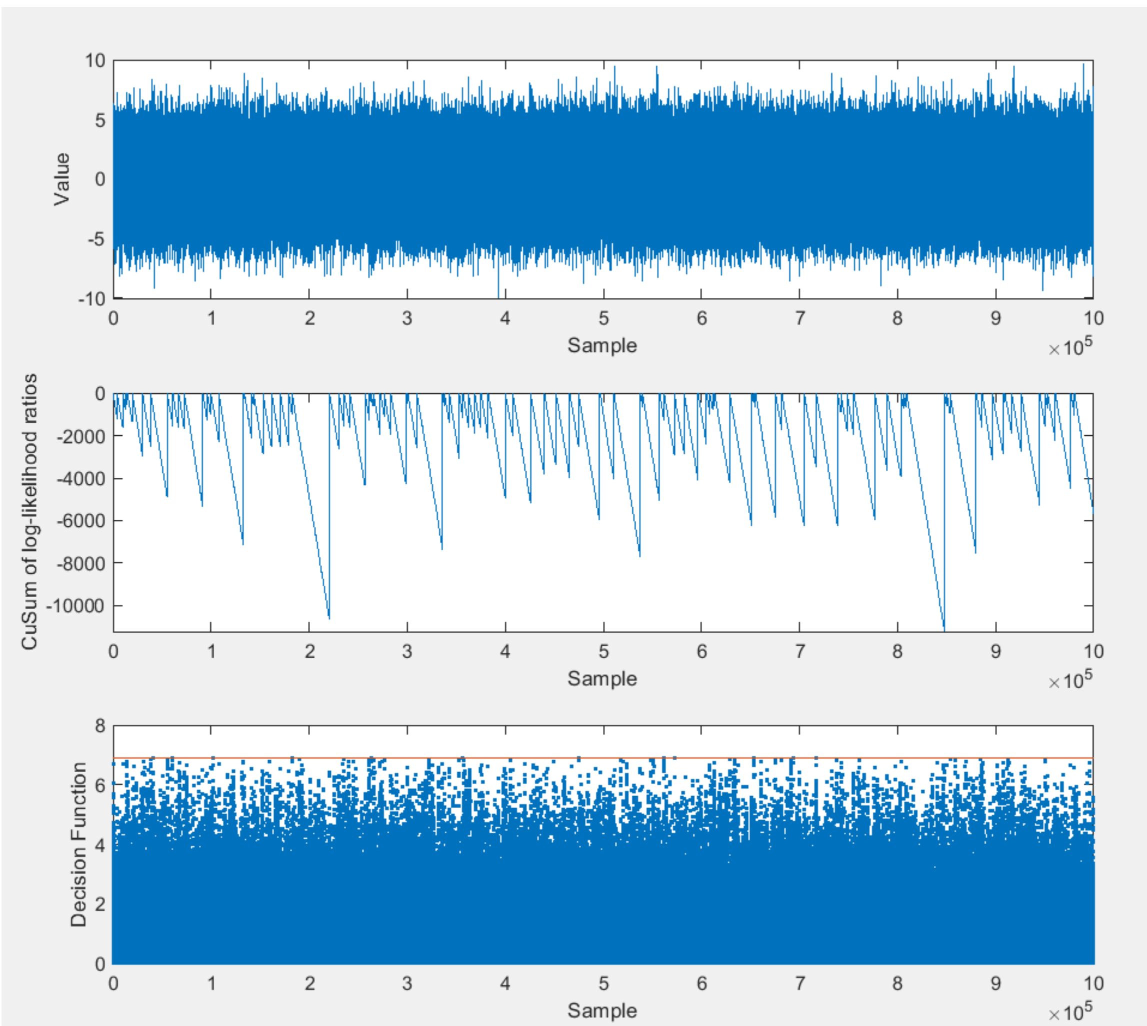


The CuSum algorithm detected a change between H_0 and H_1 states at sample number 557 and estimated that the abrupt change occurred at sample 508; a result that was accurate within 8 samples of the true change and took 57 samples to arrive at the conclusion that a change occurred.

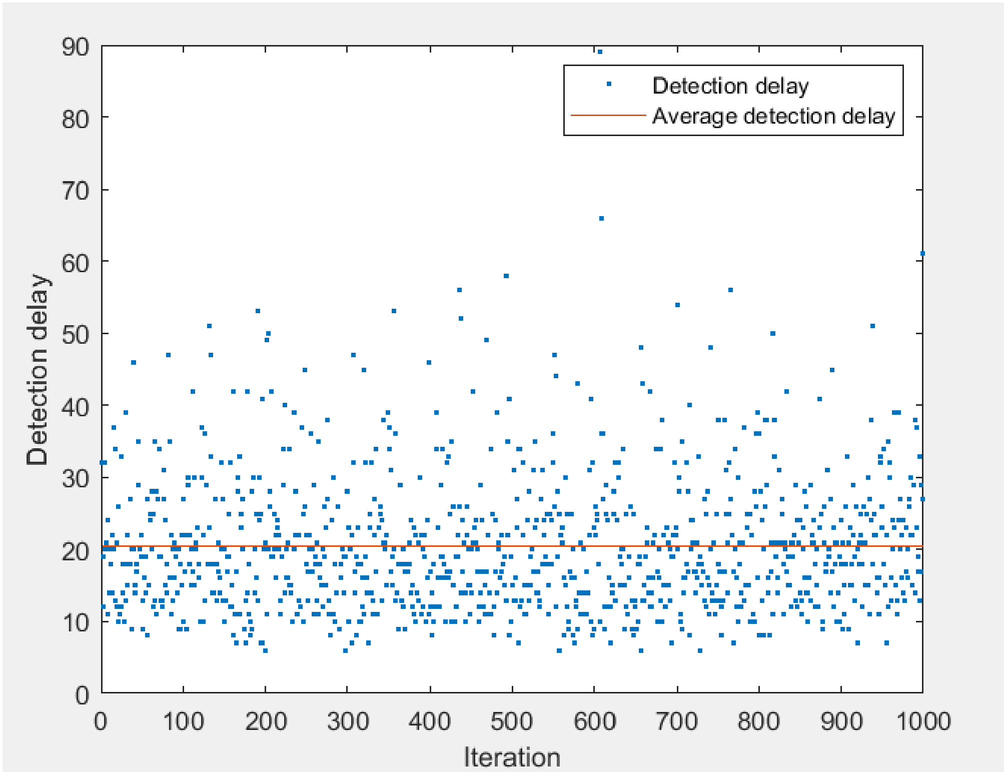
3: Settings and Performance Analysis

This part of the experiment involved using equations 7, 8, and 9 to theoretically define a threshold value and Conditional Average Delay Detection (CADD) value, as well as to experimentally validate the results. By starting with a user defined False Alarm Rate of $FAR = 1 \times 10^{-3}$, Equation 8 was used to calculate the threshold as $\cong 6.91$. Equation 9 was then used to calculate the theoretical $CADD$ using a threshold of 6.91 and was found to be $\cong 24.56$.

Experimental validation of the FAR was done by generating $1{,}000{,}000$ H_0 samples and calculating the number of times the decision function surpassed the threshold. The FAR value was experimentally determined to be 1.08×10^{-4}, which is impressively an entire order of magnitude lower than the user-defined value.



Experimental validation of $CADD$ was done by generating 500 samples for 1,000 iterations and calculating the average delay the CuSum algorithm had in determining an abrupt change. The $CADD$ value was experimentally determined to be 20.6 samples.



V. Conclusion

The CuSum appears to be a mathematically simple and effective tool at detecting signal changes. However, it should be noted that effective use of this tool requires the prior knowledge of both signal means, μ_0 and μ_1. This technique can also be applied to detecting an abrupt variance change, as well as for non-Gaussian probability functions and smooth changes. The effectiveness of this algorithm was not determined for these later variation types, and it can be safely assumed that while this algorithm will most likely not give an optimal solution, it will work well and maintains the desirable quality that it is a very simple and computationally inexpensive detection algorithm.