Decoding random-noise encoded data from a DataSocket server application.

In this lab a DataSocket server application is running on a remote machine. The server broadcasts two separate variant waveforms. One variant waveform (dstp://yossi.physics.utah.edu/signal) contains a noise-encoded signal waveform. The other variant waveform (dstp://yossi.physics.utah.edu/noise) contains a unity-scaled noise waveform which is used to encode the signal. Each variant waveform has two variant attributes attached: one is called “id” and it contains a unique timestamp number which is common to both variant waveforms, the other is called “factor” and it contains a scaling factor which is used by the signal generator to generate the noise-encoded signal waveform (dstp://yossi.physics.utah.edu/signal). The original signal is noise encoded by multiplying the unity scaled noise waveform (dstp://yossi.physics.utah.edu/noise) by “factor” and adding the to the true (non-noisy signal). Figure 1 below shows you how this is done.
You are to build an application to connect to the DataSocket server, retrieve DataSocket packets of data that are being served, and decode the signals to recover a sine wave signal. The signal waveform is decoded to remove the noise in the following way:

1) Get a pair of signal and noise waveforms from the two data sockets. The id numbers should be the same for both (if not the algorithm will NOT work). The Signal Generation vi is running slow enough that generally if you read a DataSocket packet in parallel from both data Socket sources (i.e. both DataSocket reads in the same while loop structure) the signal and noise id numbers will be the same.

2) Strip out the signal waveform from the variant signal waveform (using the Variant to Data function), and also read the “id “ and “factor” attributes (using Get Variant Attribute function) like we did in class. The “id” and “factor” default values can be wired to a constant = 1.

3) Strip out the noise waveform from the variant noise waveform (using the Variant to Data function), and also read the “id “ and “factor” attributes (using Get Variant Attribute function) like we did in class. The “id” and “factor” default values can be wired to a constant = 1

4) Multiply the noise waveform by “factor”. Subtract the resulting waveform from the signal waveform. The result should be a nearly noise free waveform

Figures 2, 3, 4 below illustrate my solution and show the two waveforms attached to the DataSocket variant waveforms (dstp://yossi.physics.utah.edu/signal) and (dstp://yossi.physics.utah.edu/noise). The figure also shows the recovered waveform after processing/decoding (Waveform 3). You should recover a sine wave signal, but the amplitude and frequency of the sine wave will be random.