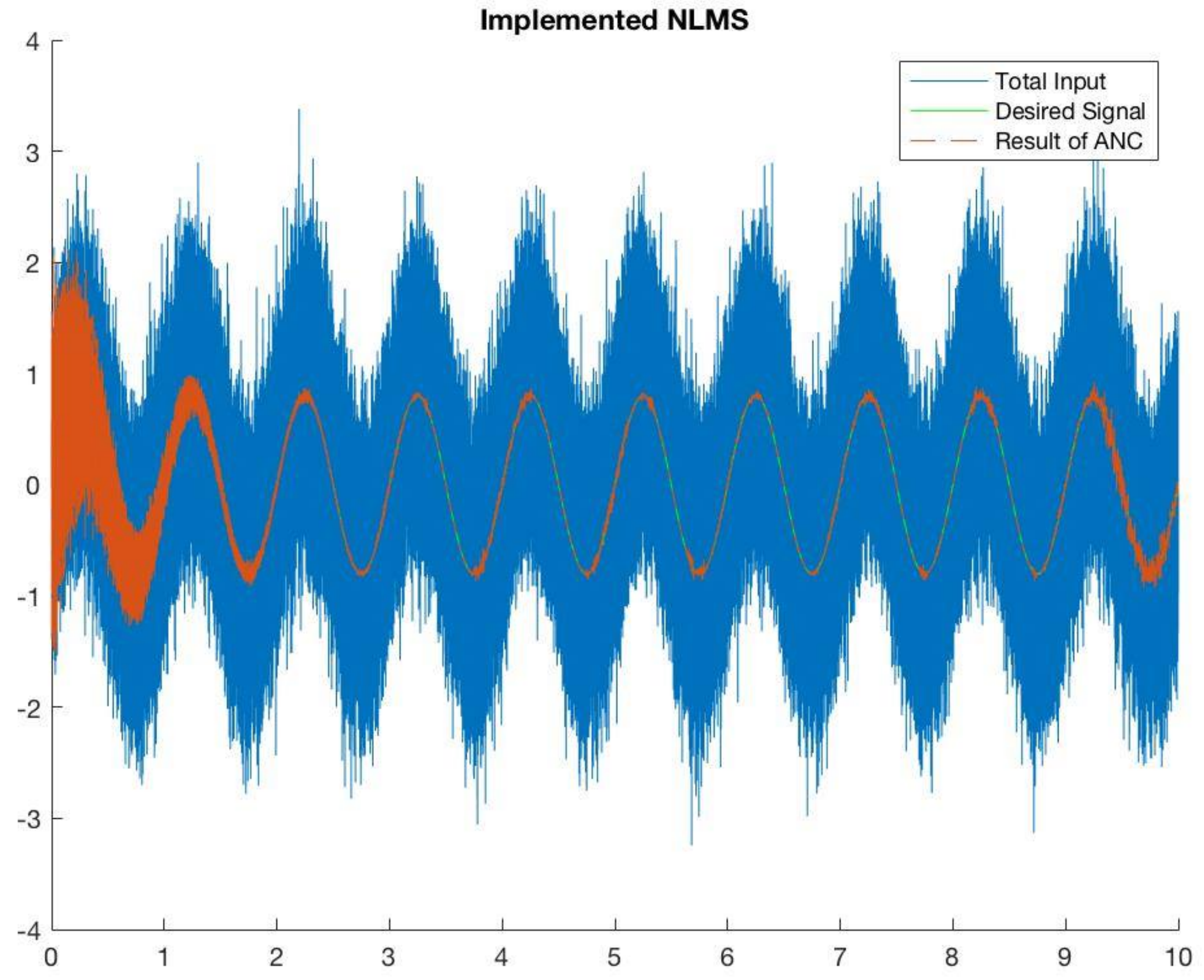
Adaptive Noise Cancellation

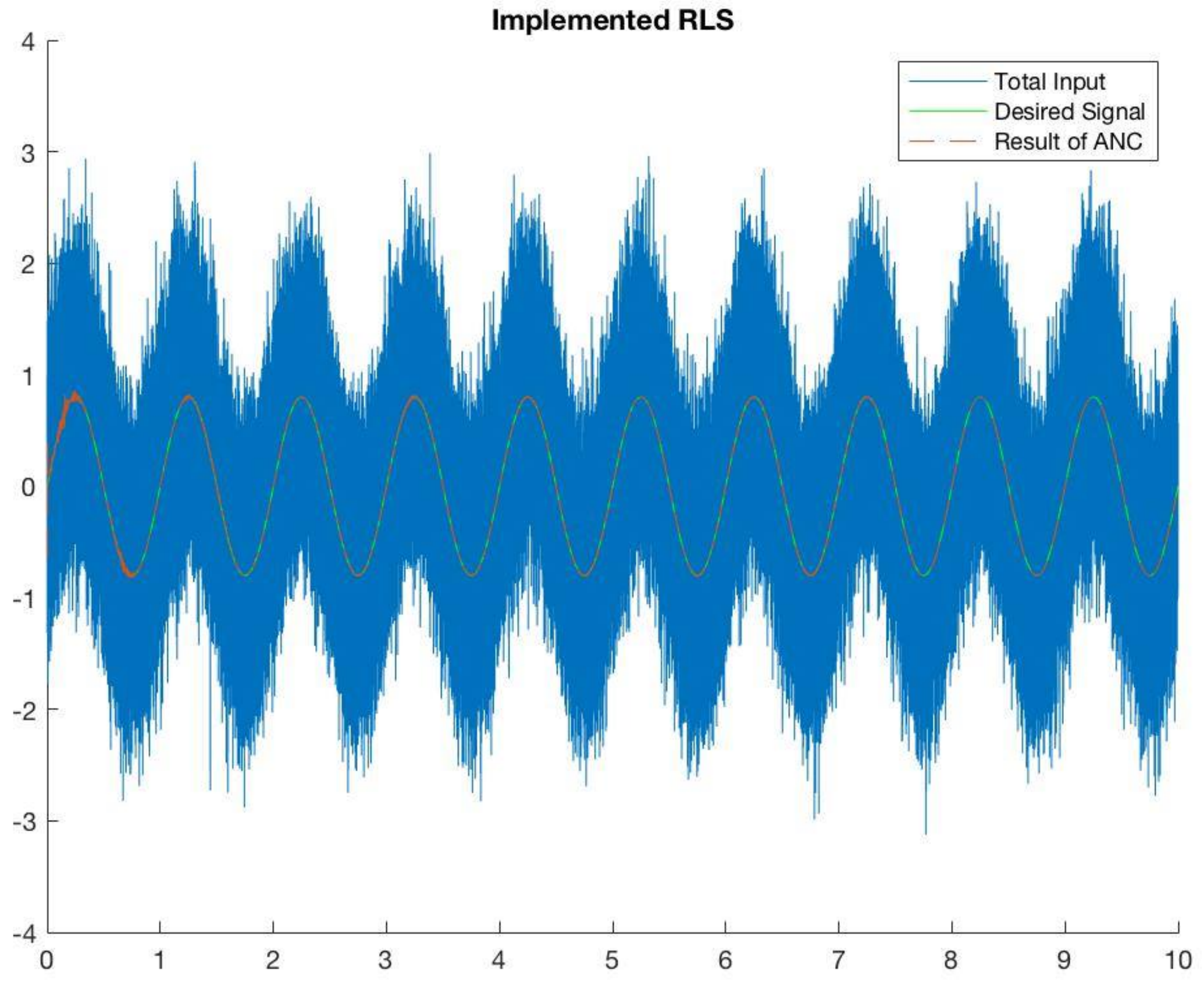
James Fosburgh, Scott Bradley, Claire Wenner

Introduction

Adaptive Noise Cancellation involves estimating a signal that has been corrupted by additive noise by using an adaptive filter. Our application of an adaptive filter is implemented in a system that is used to eliminate noise from a noisy source signal. In this application, the input signal x[n] represents the raw noise, and the input signal d[n] represents the noise filtered and combined with an unknown source signal. The weights of the adaptive filter represent the filter that is being applied to the noise before being added to the source signal, and the resulting error of the system is the raw source signal. One use of this system is for Adaptive Noise Cancellation, which is implemented in real-time in a live Simulink system.

Adaptive Filters



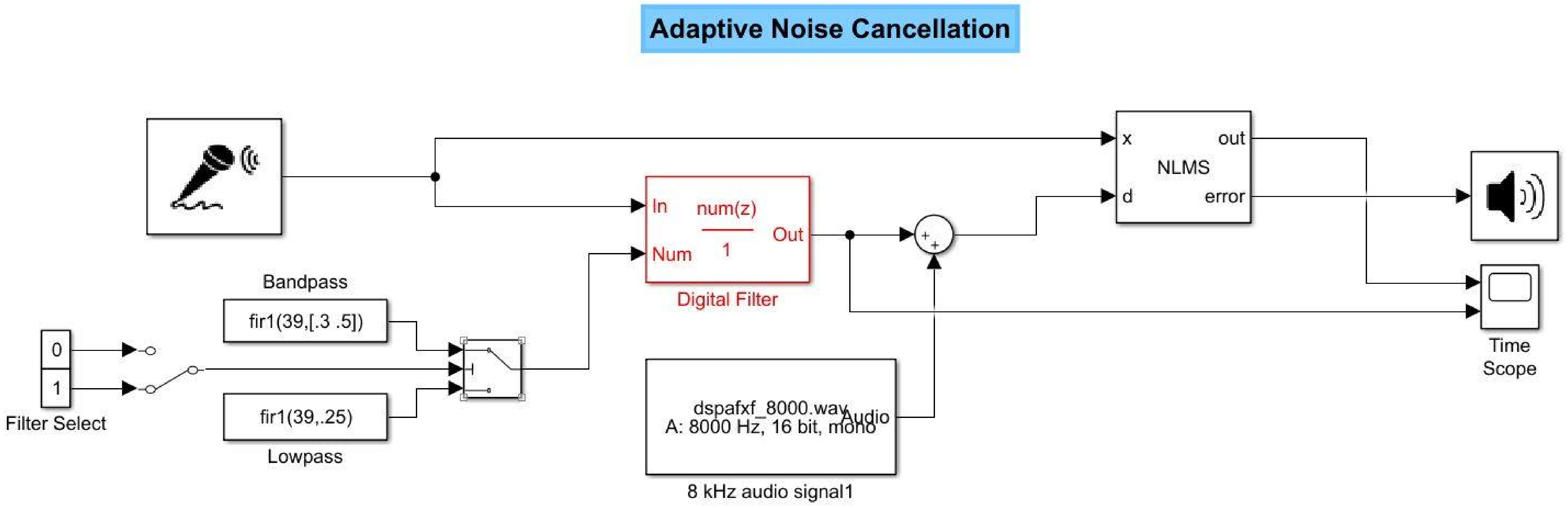


An adaptive filter is a system with a filter whose transfer function is controlled by a set of weights that change over time. It takes in two inputs, x[n] and d[n], and the weights of the filter are changed so that the difference between the two signals, the error e[n], is minimized. The output of the filter y[n], represents the filtered version of x[n] that is similar to d[n], and the error is found by simply subtracting y[n] from d[n]. The resulting weights represent the filter applied to the signal x[n], before being added to a source signal to create d[n].

The simplest adaptive noise cancellation filter is the Least Mean Square (LMS) filter. This algorithm works by finding a filter represented by coefficients that creates the least mean square when applied to the noise of the system and subtracted from the total input. One problem with LMS is that it does not take into account the power of the noise input. This can be solved by adding a normalization to the updating of the weights. We have implemented our own Normalized Least Mean Square (NLMS) algorithm in Matlab. The image above shows the result of passing a 1Hz sine wave with randomly generated noise through our implemented NLMS filter.

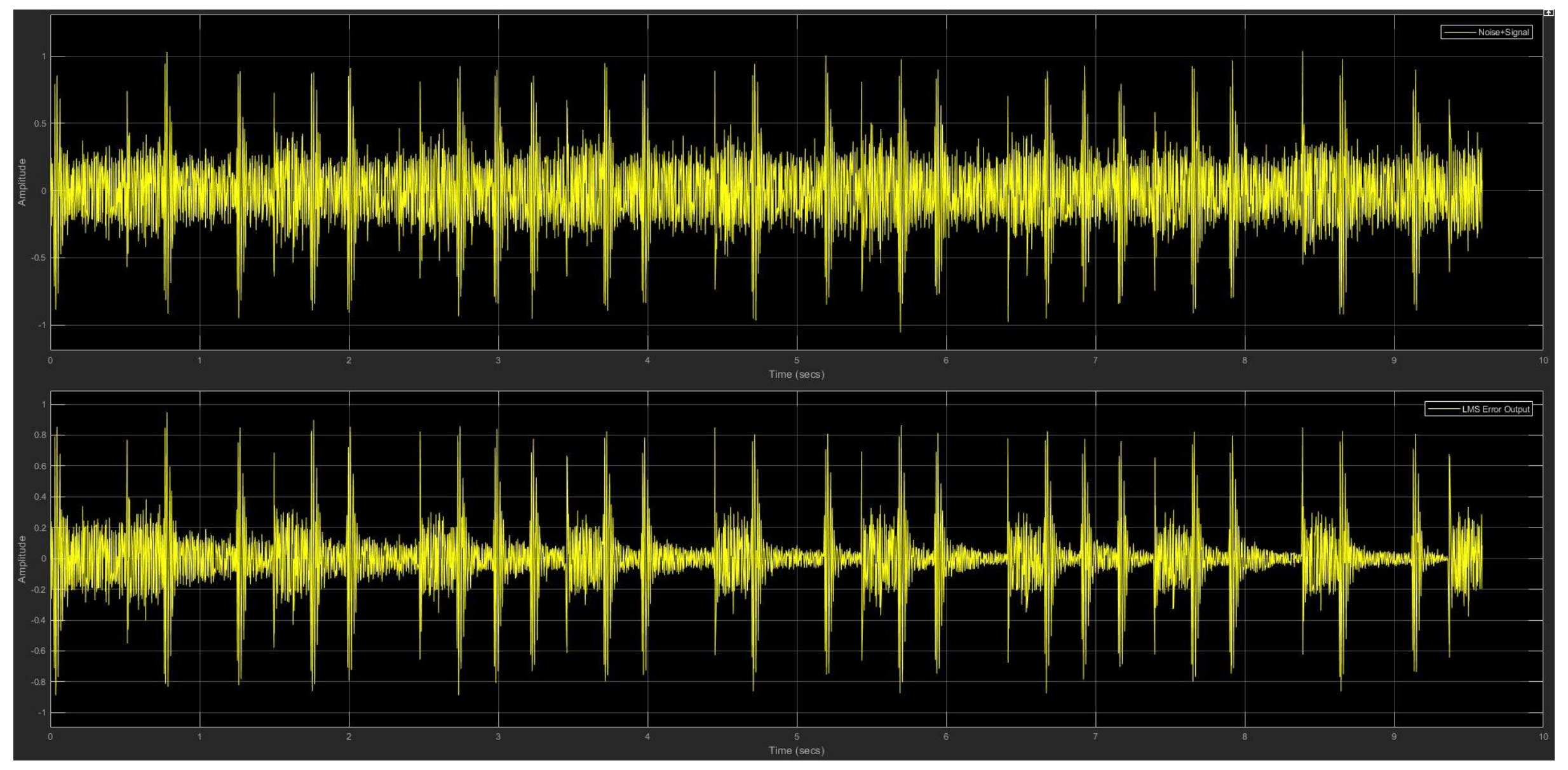
An alternative algorithm to LMS is the Recursive Least Squares (RLS) algorithm. The RLS algorithm is overall fairly similar to LMS, but updates the weight coefficients recursively. Overall, the RLS algorithm converges faster and is more precise than LMS. With LMS, there was a tradeoff between getting the filter to converge quickly and having it remain stable once it has converged. In comparing the picture above to that of the LMS algorithm learning under the same conditions, it is clear that the RLS algorithm works both more quickly and more effectively than the LMS algorithm.

Simulink Implementation



For our live implementation, we decided to use MATLAB Simulink. Our model stimulates an Adaptive Noise Cancelling system. The figure above shows our simulink system using our implemented LMS adaptive filter algorithm, which was able to significantly reduce the noise found in the corrupted signal even in different acoustic environments. In our system, we take a microphone input to record the noise, and we simulate the filter and the corrupted signal from that. The output of the system is the error of our implemented LMS adaptive filter, and should be the noise canceled input signal. Although we have this source signal being a pre-recorded audio file for demonstration purposes, this input can be replaced with another microphone representing a corrupted signal.

Results



An audio file of a drum loop was used as the source signal and played through our live Simulink system. The first graph in the figure to the left shows the corrupted signal represented as filtered noise combined with a source signal, and the second shows the LMS error, which should be the isolated source signal. The LMS algorithm had a delay period before it fully took effect - about 1 second in this example - as the filter weights need more input data before they fully represent the noise filter. After that period, you can see the noise from the signal almost completely eliminated.

References

[1] A. Singh. *Adaptive Noise Cancellation.* (2001). Avaliable at: http://www.cs.cmu.edu/ aarti/pubs/ANC.pdf.

[2] Mathworks.com. (2018). *Overview of Adaptive Filters and Applications.* [online] Available at: https://www.mathworks.com/help/dsp/ug/overview-of-adaptive-filters-and-applications.html.

[3] S.C. Douglas. *Digital Signal Processing Handbook.* Ed. Vijay K. Madisetti and Douglas B. Williams. Boca Raton: CRC Press LLC, 1999.