This project is about constructing a Linear Time-Invariant system and another with a static non-linearity and seeing how well you can learn these systems thru a simple gradient descent rule, given just the input and the output signal.

Begin by constructing an input signal \( x(t) \) that is a gaussian white noise signal. You may want to check on the web as to how to construct such a signal.

Now consider the kernel \( h(t) = A * e^{-t/\gamma} \). Choose values for \( A \) and \( \gamma \). Produce the output \( y(t) = \int_{0}^{T} x(t - \tau) h(\tau) d\tau \) for a large enough value of \( T \).

1. (50 points) You are now set to learn this kernel. Hand the input and the output signal to a new program that attempts to learn the kernel \( h(t) \) via a simple gradient descent rule. Quantify how well your program learns the kernel by plotting the square-error on the output signal as time progresses.

2. (50 points) Now assume that the output signal passes thru the static nonlinearity \( \tilde{y}(t) = (y(t))^3 \). Ask your program to learn a kernel for this pair of input and output, i.e., \( x(t) \) and \( \tilde{y}(t) \). Quantify how well your system performs on this case.