SGN-21006 Advanced Signal Processing

Comparing Adaptive filters

DSP System Toolbox

DSP System Toolbox provides algorithms for designing and simulating signal processing systems e.g. specialized FIR and IIR filters, FFTs, multirate filters, DSP techniques for processing real-time data. Here, we use the toolbox for designing adaptive filters in order to compare their performance.

Browse the documentation of the toolbox by typing *doc* in MATLAB and selecting the *DSP System Toolbox*. Note, that you need the Matlab version R2012b or newer in order to use DSP System Toolbox! If you use an older version, the *Filter Design Toolbox* provides the same functionality with a different syntax.

Learn how to use the toolbox to implement different types of *Least Mean Squares* and *Recursive Least Squares* algorithms. The examples in the documentation are particularly useful.

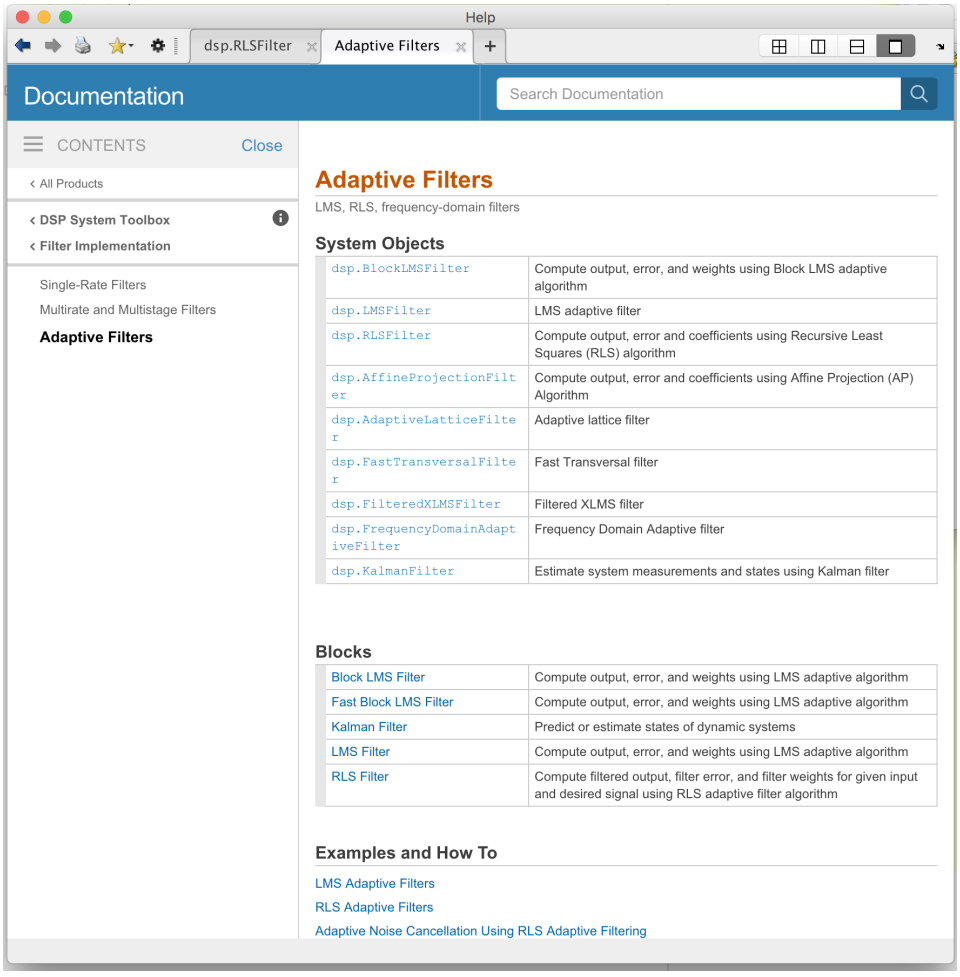

Figure 1: DSP System Toolbox → Filter Implementation → Adaptive Filters

Tasks

You are given the script *L8.m* for reading the input data signal u and desired signal d. The application in question is channel equalization, where we try to estimate the noiseless signal by inverting the effects of the channel by the adaptive filter. So, the error in the optimal case should be zero(!).

The script *L8.m* loads 10 different realizations of the input signal that only differ by the random white noise that has been added to them. So, you should calculate and plot the average error of the realizations i.e. *learning curve*.

1. Use `dsp.LMSFilter` to test different variants of *Least Mean Squares*, namely LMS, NLMS and Sign-Sign LMS. Plot the learning curve. Why is it useful to take many realizations? Can you observe the convergence time and steady state error from your plot? What is the theoretical difference of the algorithms? How and why are the learning curves different?

2. Use `dsp.RLSFilter` to test different variants of *Recursive Least Squares*, namely RLS, Householder RLS and Householder sliding window RLS. Plot the learning curve. Do you see a different result for all the methods? Why/why not? What is the effect of the forgetting factor? Is it in any way related to the filter length?

3. Select best algorithms among the *Least Mean Squares* variants and the *Recursive Least Squares* variants, and plot together the results. What are the best performing algorithms? Does it make sense theoretically? What criteria do you use to define ”best”?