Dept. of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Fall Semester 1991

ECE 652 Adaptive Filtering
Computer Simulation Project

In this project two adaptive filtering algorithms will be investigated: the LMS and the fast RLS algorithms. Both algorithms will be analyzed theoretically and by simulation. The topology for the LMS adaptive filter is shown in Figure 1. Note that the buffer numbers and star names are also shown. The topology for the fast RLS algorithm is shown in Figure 2. The *predftf* star is used for the fast RLS algorithm. It implements the Fast Transversal Filter (FTF) algorithm which for our purposes is numerically equivalent to the RLS algorithm (only in infinite precision). In both systems, the impulse response of a system is to be estimated. The input to the system is a white guassian random process. The system is simply a linear convolution of its input with a finite impulse response stored in a file(*imp.dat*). The *convolve* star performs a linear convolution. For your reference the

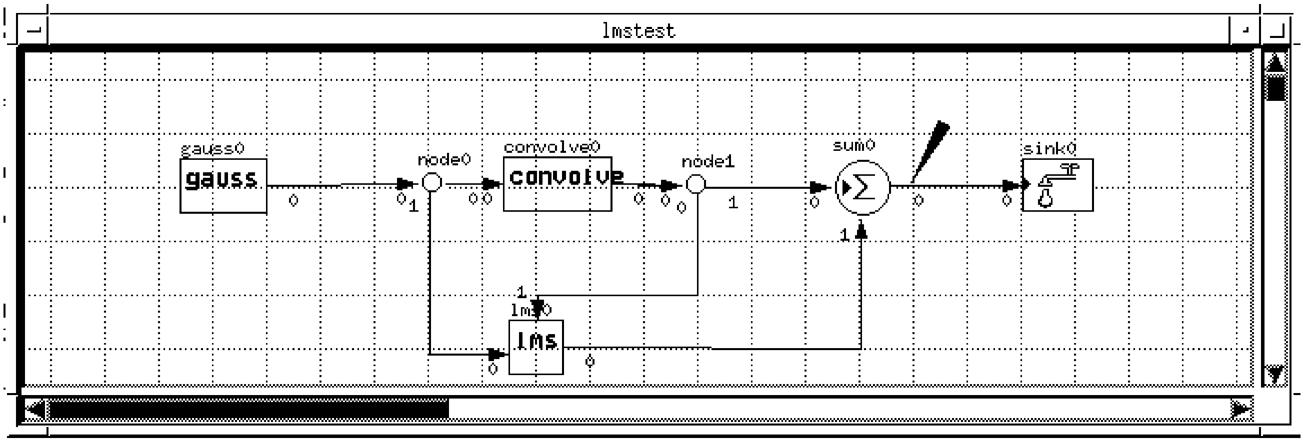

Figure 1. Topology for adaptive LMS filter

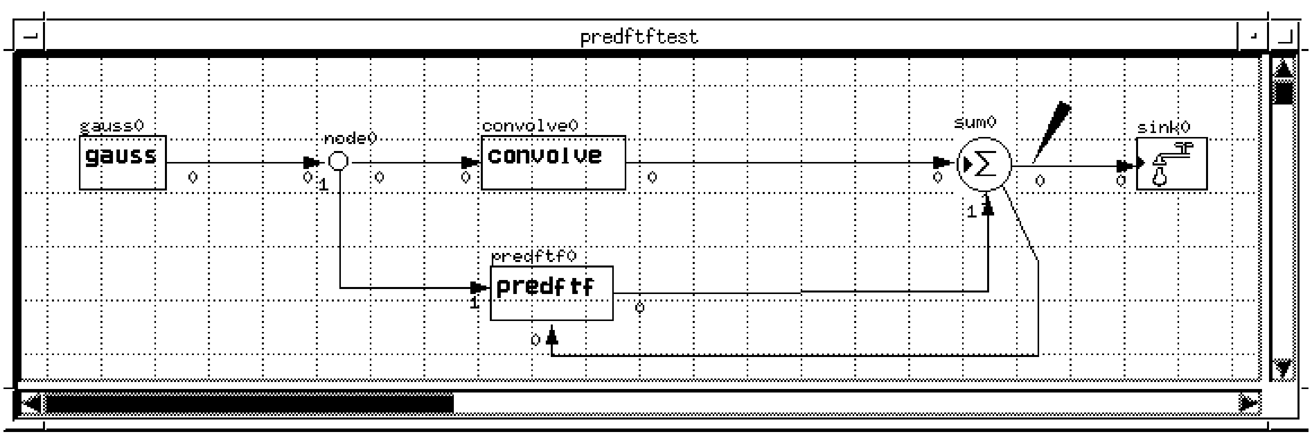

Figure 2. Topology for Fast Transversal Filter

text files of both topologies are included in the following pages. These files contain information about parameters. The default parameters should be okay but some parameters such as the number of points in the *gauss* star must be change. So refer to this information.

The *predftf* star requires a file called *prfile* which contains information about the number of input channels, outputs and the initial values of the filters. This file is included for your reference.

Construct both topologies and test them. The probe in both figures is a *plot* probe. You should observe very rapid convergence of the FTF algorithm.

Part I.

(a) In this part we will investigate the LMS algorithm. Connect a *stats* probe to the input to the *lms* star so we can measure the input power. Increase the number of points generated by the gauss star. We are interested in the convergence of the LMS algorithm with different gains. Try the following gains:

μ = 1.0, 0.5, 0.1. For each case, plot the learning curve in dB's. Calculate the rate of convergence (dB's per 100 iterations) and compare with the simulation result.

(b) Insert an *addnoise* star between the adaptive filters (desired response) and the *convolve* star output for both systems. In Figure 1 insert it between convolve0 and node1. In Figure 2, insert it between convolve0 and sum0. We are interested in investigating excess noise due to the LMS and FTF algorithms. Set the noise variance in the *addnoise* star to 0.2. Obtain the steady state error for the LMS algorithm for μ=1.0, 0.5, 0.1 and λ=1.0, 0.99, and 0.9 for the FTF algorithm. Calculate the mean square excess error theoretically and compare with the simulation results.

Part II

In this part we will investigate the effect of signal correlation on the performance of the adaptive filters. Replace the *gauss* star with the *arprocess* star. This star produces an autoregressive process. By changing the AR coefficient we can increase signal correlation. Start out with a first order process. Let the AR coefficient be 0.99 initially. Observe the convergence of the LMS algorithm. You may want to insert a *gain* star between the *arprocess* star and the *convolve* and *lms* star (between the *arprocess* star and node0) in order to keep the variance at 1.0. Let the LMS gain be 1.0. Can you explain the slow convergence of the LMS algorithm compared to the FTF algorithm?

Predict the convergence rate of the LMS algorithm. Compare it to simulation results.

```
1
2
3
4
5
4
3
2
1
0
0
```

Contents of *imp.dat* file

```
1
1
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Contents of *prfile*

```
# topology file:  lmstest.t

#———————————————————————————
# Title:
# Author:
# Date:
# Description:
#———————————————————————————

inform title
inform author
inform date
inform descrip

arg -1 (none)

param int 500
param float 1
param int 3759
param float 1
param int 128
star gauss0 gauss

star node0 node

param file imp.dat
param int 8
star convolve0 convolve

star node1 node

param int 10
param float 0.1
param int 0
star lms0 lms

param array 2  1  -1
star sum0 sum

param int 500
param int 0
param file lmserror
param file X
param file Y
param int 1
param int 1
star plot0 plot

star sink0 sink

connect gauss0 0 node0 0
connect node0 0 convolve0 0
connect node0 1 lms0 0
connect convolve0 0 node1 0
connect node1 0 lms0 1
connect node1 1 sum0 0
connect lms0 0 sum0 1
connect sum0 0 plot0 0
connect plot0 0 sink0 0
```

```
# topology file:  predftftest.t

#———————————————————————————
# Title:
# Author:
# Date:
# Description:
#———————————————————————————

inform title
inform author
inform date
inform descrip

arg -1 (none)

param int 1000
param float 1
param int 3759
param float 1
param int 128
star gauss0 gauss

star node0 node

param file imp.dat
param int 8
star convolve0 convolve

param array 2  1  -1
star sum0 sum

param file prfile
param file prfileo
param float 1
param float 0.0001
param int 0
param int -1
star predftf0 predftf

param int 256
param int 0
param file PLOT
param file X
param file Y
param int 1
param int 1
star plot0 plot

star sink0 sink

connect gauss0 0 node0 0
connect node0 0 convolve0 0
connect node0 1 predftf0 1
connect convolve0 0 sum0 0
connect sum0 0 plot0 0
connect sum0 1 predftf0 0
connect predftf0 0 sum0 1
connect plot0 0 sink0 0
```