

## ПРОГРАММА АДАПТИВНОГО LMS КОМПЕНСАТОРА ДЛЯ КОМПЛЕКСНЫХ ВЫБОРОК НАБЛЮДЕНИЙ

*Бартенев В.Г., Бартенев Г.В.*

### Введение

Одним из наиболее распространенных адаптивных алгоритмов является LMS (Least Mean Square) алгоритм [1], известный также как алгоритм Уидроу-Хоффа. Это наиболее простой и в то же время эффективный способ адаптивной обработки на основе метода наименьших квадратов (МНК). О распространенности этого алгоритма говорит хотя бы тот факт, что он используется почти в каждом сотовом телефоне для компенсации мешающих эхо-сигналов. Не трудно показать, что для комплексных выборок наблюдений LMS алгоритм можно представить в следующем виде

$$\dot{W}(n+1) = \dot{W}(n) + \mu \dot{p}(n) \left[ \dot{d}(n)^T - \dot{W}(n)^T \dot{p}(n)^* \right]$$

где  $\dot{d}(n)$  - входной обучающий вектор-сигнал;  $\dot{p}(n)$  - вектор-сигнал на входе адаптивного КИХ-фильтра;  $n$  - номер итерации;  $\dot{W}(n)$  - вектор весовых коэффициентов КИХ-фильтра, обновляемый на каждой  $n$ -ой итерации; точка сверху означает комплексное число,  $(\cdot)^T$  - операцию транспонирования, а  $(\cdot)^*$  - операцию комплексного сопряжения.

Именно это выражение и явилось основой для программирования LMS автокомпенсатора на языке MATLAB.

Эти программы приводятся далее в примерах моделирования двух типовых задач, для сигналов, представленных в комплексном виде.

### Адаптивная фильтрация помех

Появление в последнее время целых семейств мощных DSP, производимых фирмами Texas Instruments и Analog Devices, а также супер ПЛИС компаний Xilinx и Altera, открыло новые горизонты в создании программируемых систем адаптивной компенсации помех. Проекты, реализованные на DSP, проникают в такие области как адаптивные фазированные антенные решетки, программируемые адаптивные компенсаторы помех, действующие по боковым лепесткам ДНА. И в том и в другом случае полезные сигналы и помехи обычно представляют в комплексном виде, поэтому и в том и другом случае на практике может найти применение рассматриваемый модифицированный LMS алгоритм адаптивной компенсации помех.

При создании таких сложных программируемых устройств на DSP или ПЛИС одной из важнейших проблем является предварительная проверка программируемых алгоритмов. Прежде чем программировать тот или иной алгоритм на DSP, предназначенный для работы в реальном времени, целесообразно этот алгоритм проверить, моделируя его работу с помощью MATLAB. Чтобы

*Предложена программа адаптивного LMS алгоритма для комплексных выборок наблюдений. Рассмотрены два примера использования предложенной программы: для адаптивной фильтрации помех и для адаптивной идентификации параметров системы. Данные примеры подтверждают правильность работы программы. Статья имеет учебно-методическое значение.*

проиллюстрировать сказанное, рассмотрим практический пример программы компенсации помехи, действующей как в основном, так и в дополнительном канале адаптивного компенсатора с модифицированным LMS алгоритмом. Итак, на входе автокомпенсатора два сигнала: в основном канале искаженный помеховым фоном полезный сигнал, а во вспомогательном канале только помеха (рис. 1).

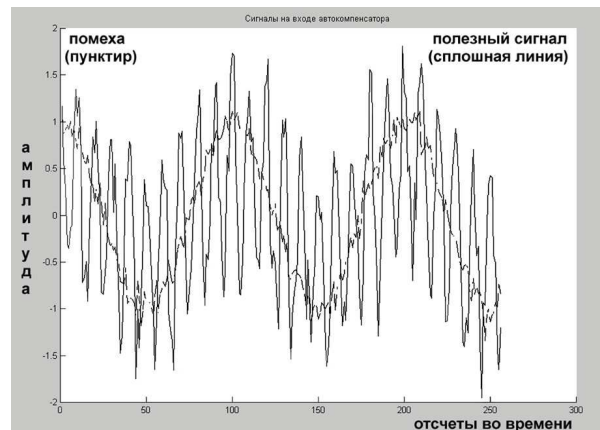


Рис.1 Временная диаграмма сигналов на входе автокомпенсатора

Спектры сигналов на входе автокомпенсатора представлены на рис.2.

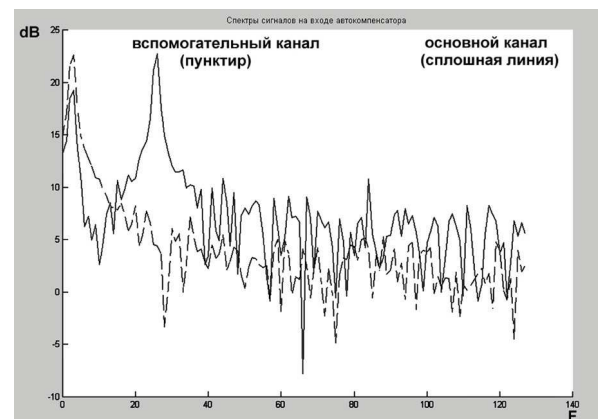


Рис.2 Спектральное представление сигналов на входе автокомпенсатора

Выберем порядок адаптивного КИХ фильтра равным 4 и произведем автокомпенсацию помехи. Сигнал ошибки на выходе автокомпенсатора (рис.3) после настройки весовых коэффициентов фактически и есть полезный сигнал. В подтверждение сказанного рассмотрим спектр сигнала ошибки на выходе (рис.4).

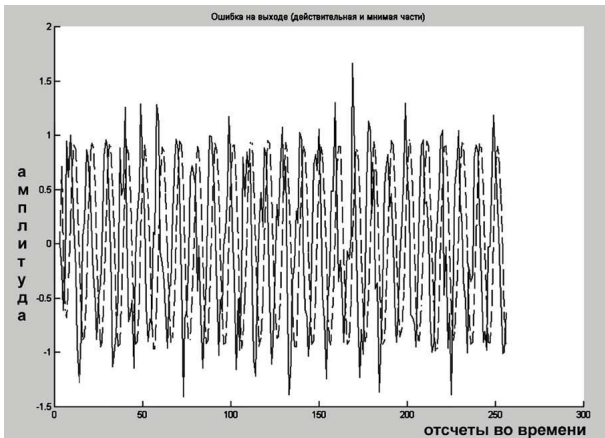


Рис.3 Сигнал ошибки на выходе автокомпенсатора.

Из рис.4 следует – спектральная составляющая помехи подавлена, а спектральная составляющая полезного сигнала присутствует.

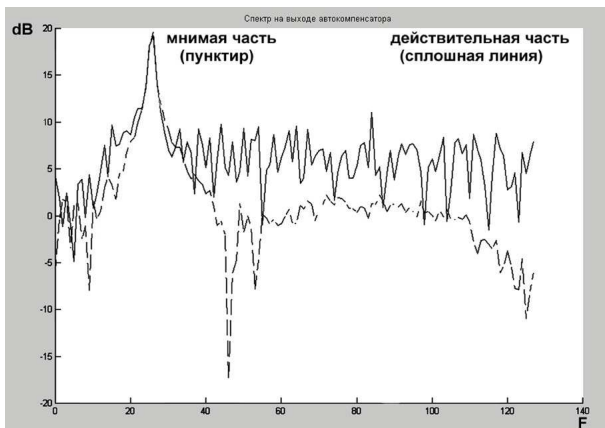


Рис.4 Спектры сигнала ошибки (для действительной и мнимой части).

Ниже приведена моделирующая программа для этого примера.

```
% Моделирование LMS алгоритма
% адаптивная фильтрация комплексных %сигналов
clear all
close all
hold off
sysorder = 4 ;%порядок КИХ фильтра с %перемен-
ными весовыми коэффициентами
N=256;%Число отсчетов на входе
n = 1:N;
%формируем помеху
p=exp(.02*pi*n*sqrt(-1));
%добавляем шум
inp = p+.25*randn(N,1);
%формируем сигнал
s=exp(.2*pi*n*sqrt(-1));
z= randn(N,1); %добавляем шум
% аддитивная смесь
```

```
d=0.5*inp+0.25*z+s';
%на основном входе автокомпенсатора
figure(1)
hold on
plot(real(d(1:256)),'R')
plot(real(inp(1:256)),'B')
D = fft(d',256);
f = 256*(0:127)/256;
INP = fft(inp',256);
f = 256*(0:127)/256;
title('Сигналы на входе автокомпенсатора') ;
figure(2)
hold on
plot(f,10*log10(abs(D(1:128))), 'R')
plot(f,10*log10(abs(INP(1:128))), 'B')
title('Спектры сигналов на входе автокомпенсатора') ;
totallength=size(d,1);
%вводим параметр-объем выборки
%ВСЕ ГОТОВО ЧТОБЫ НАЧАТЬ РАБОТУ
%АВТОКОМПЕНСАТОРА
w = zeros ( sysorder , 1 ) ;
%начальные весовые коэффициенты
%зададим равными нулю
for n = sysorder : N
%основной цикл работы по отсчетам
u = inp(n-1:n-sysorder+1) ;
%отсчеты на входе КИХ фильтра
y(n)= (w).' * conj(u);
%сама фильтрация адаптивным фильтром
e(n) = d(n)' - y(n) ;
%вычисление ошибки
mu=0.15;%
%ИТЕРАЦИИ КОГДА ПРОИЗВОДИТСЯ
%РАСЧЕТ ВЕСОВЫХ КОЭФФИЦИЕНТОВ
w = w + mu * u * e(n) ;
figure(3)
hold on
plot(real(e),'K')
plot(imag(e),'G')
title('Ошибка на выходе') ;
end %Закончили работу
% автокомпенсатора
ER = fft(real(e),256);
f = 256*(0:127)/256;
EI = fft(imag(e),256);
f = 256*(0:127)/256;
figure(4)
hold on
plot(f,10*log10(abs(ER(1:128))), 'K')
plot(f,10*log10(abs(EI(1:128))), 'G')
title('Спектр на выходе автокомпенсатора');
Итак, задав полезный сигнал в виде комплексной синусоиды, мы его подвергли воздействию помехи, а затем подали на один из входов автокомпенсатора. На другой вход автокомпенсатора был подан сигнал помехи, также представленный в комплексном виде. В результате чего, уже через несколько итераций помеха отфильтрована автокомпенсатором, а на выходе остался только полезный сигнал и собственный шум, что подтверждается спектром сигналов на выходе автокомпенсатора.
```

## Адаптивная идентификация параметров

В данном примере рассматривается задача идентификации параметров системы. Рассматривается случай, когда на входе действует комплексная выборка наблюдений, а в качестве системы рассматривается трехкратная схема череспериодного вычитания (ЧПВ), блок схема которой изображена рис. 5. Задача состоит в адаптивной идентификации параметров ЧПВ или, проще говоря, в нахождении весовых коэффициентов такого КИХ фильтра (рис 5), который был бы эквивалентен трехкратной схеме ЧПВ.

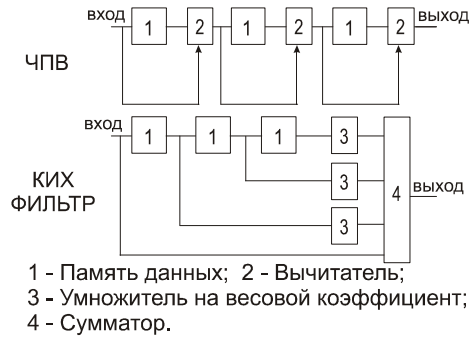


Рис.5 Блок-схема идентифицируемой системы ЧПВ и ее эквивалентный КИХ фильтр, весовые коэффициенты которого – искомые параметры.

Далее приводится текст программы с подробными комментариями.

```
%Моделирование LMS алгоритма для
%адаптивной идентификации параметров %трех-
кратной ЧПВ
clear all
close all
hold off
sysorder = 4 ;%порядок КИХ фильтра с %перемен-
ными весовыми коэффициентами
N=128;%Число отсчетов на входе
s = exp(sqrt(-1))*randn(N,1);
%на входе комплексный сигнал в виде %гауссова
шума
ns = randn(N,1); %собственный шум
%первая ступень вычитания
z1 = filter([1 -1],1,s);
%вторая ступень вычитания
z2 = filter([1 -1],1,z1);
%третья ступень вычитания
z3 = filter([1 -1],1,z2);
nn = ns/10;
%добавляем собственный шум в 10 раз
%ослабив
d = z3 + nn;% аддитивная смесь
figure(1)
hold on
plot(real(d(1:128)), 'R')
plot(real(s(1:128)), 'B')
D = fft(d,256);
f = 256*(0:127)/256;
S = fft(s,256);
f = 256*(0:127)/256;
title('Сигналы на входе автокомпенсатора');
figure(2)
```

```
hold on
plot(f,abs(D(1:128)), 'R')
plot(f,abs(S(1:128)), 'B')
title('Спектры сигналов на входе автокомпенсатора');
totallength=size(d,1);%вводим параметр-объем вы-
борки
w = zeros ( sysorder , 1 ) ;
%начальные весовые коэффициенты
%зададим равными нулю
for n = sysorder : N %основной цикл работы по отсчетам
u = s(n:-1:n-sysorder+1) ;
%отсчеты на входе фильтра
y(n)= (w).' *conj( u);
%сама фильтрация адаптивным фильтром
e(n) = d(n)' - y(n) ;
%вычисление ошибки
mu=0.15;%
w = w + mu * u * e(n) ;
%ЭТО ИТЕРАЦИИ КОГДА
%ПРОИЗВОДИТСЯ РАСЧЕТ ВЕСОВЫХ %
КОЭФИЦИЕНТОВ КИХ ФИЛЬТРА
figure(3)
hold on
title('Ошибка на выходе');
plot(real(e), 'K')
plot(imag(e), 'G')
hold on
end
%проверим как настроился
%своими весовыми коэффициентами
%автокомпенсатор
figure(4)
plot([1,-3,3,-1], 'rd')
hold on
plot(real(w), 'k*')
hold on
plot(imag(w), 'rx')
legend('истинные веса','оцененные веса')
title('Сравнение весовых коэффициентов фильтров');
grid
axis([0 4 -4 4])
```

Итак, задав помеху в виде белого шума, которая по-дана на один их входов автокомпенсатора, на другой его вход была подана та же помеха, но пропущенная через идентифицируемую систему в виде трехкратной ЧПВ. На рис.6 изображены спектры помехи до и после про-пускания ее через трехкратную схему ЧПВ, т.е. спектры сигналов на входе автокомпенсатора.

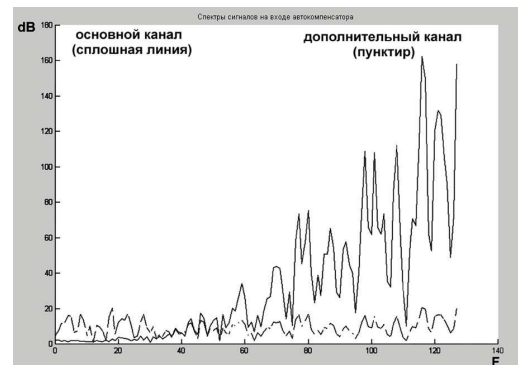


Рис.6 Спектр помехи на входе автокомпенсатора.

В результате уже через несколько итераций проишла настройка автокомпенсатора. На рис. 7 сигнал ошибки на выходе автокомпенсатора это подтверждает.

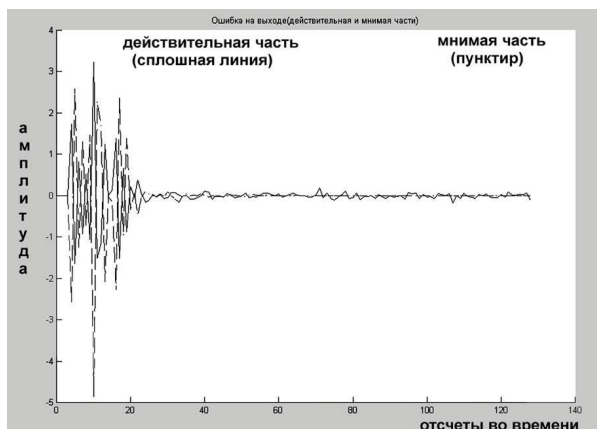


Рис.7 Сигнал ошибки автокомпенсатора (действительная и мнимая составляющие).

При этом, автокомпенсатор настроился так, что весовые коэффициенты его адаптивного КИХ фильтра практически соответствуют весовым коэффициентам идентифицируемого фильтра, т.е. 1, -3, 3, -1.

### Заключение

Предложенная программа LMS алгоритма, для обработки сигналов, представленных в комплексном виде, с помощью моделирования в MATLAB прошла всестороннюю проверку. Результаты адаптивного подавления помехи действующей как в основном, так и в дополнительном канале, а также адаптивная идентификация параметров трехкратной череспериодной схемы вычитания подтверждают ее правильность.

### Литература

1. Уидроу Б., Стирнз С. Адаптивная обработка сигналов.— М.: Радио и Связь, 1989.— 440с.

## Пакет Matrox Inspector – мощный инструмент анализа и обработки изображений

Matrox Inspector - это пакет интерактивного программного обеспечения и библиотека прототипов от признанного лидера, канадской компании Matrox Imaging. Нацеленный как на разработчиков, так и на конечных пользователей (научные и промышленные приложения), пакет Matrox Inspector представляет собой интерактивное приложение под Windows® 2000/XP для захвата, обработки, анализа, отображения и архивирования изображений.

Приложение Matrox Inspector основано на программном обеспечении для разработки Matrox Imaging Library (MIL) и предлагает простой и удобный интерактивный интерфейс для доступа к широкому спектру оптимизированных функций для обработки изображений, вывода закономерностей, анализа пятен, выделения и анализа контуров, измерения, распознавания символов, считывания одно- и двумерных кодов, калибровки и компрессии изображений.

Пользователи библиотек MIL применяют Matrox Inspector в качестве вспомогательного инструмента, позволяющего облегчить и упростить разработку приложений для машинного зрения, анализа изображений и медицинской визуализации. Конечные пользователи, такие как научные работники, технический персонал и инженеры, могут использовать пакет Matrox Inspector для выполнения и автоматизации работ по измерению и улучшению качества изображений.

#### Ключевые особенности и возможности пакета Matrox Inspector:

- прямое получение изображений от различных источников видео с помощью фрейм- грабберов Matrox;
- загрузка и сохранение изображений, последовательностей изображений в различных форматах (включая AutoCAD® DXF);
- высокая точность и широкий спектр средств для обработки и анализа;
- калибровка изображений для исправления визуальных искажений и выполнение измерений в реальных единицах;
- обработка изображений "вживую" или из архива, включая последовательности;
- статистика слежения, включая скорость в результатах анализа;
- конфигурирование средств анализа для использования с приложениями Matrox Imaging Library (MIL);
- компенсирование изображений текстом и графикой;
- управление и обмен данными с приложениями Microsoft® Windows®; - создание процедур совместимых с Microsoft® VBA или "С"- совместимых.

Текущая версия Matrox Inspector 8.0 включает в себя два новых модуля: String Reader - для ориентированного на топологию оптического распознавания символов и Edge Finder - для выделения признаков. Среди прочих нововведений реализована поддержка циклической обработки изображений без необходимости создания сценариев (для пакетного тестирования алгоритмов на тестовых последовательностях изображений), как предварительно сохраненных изображений, так и напрямую с камеры, с поддержкой ввода с нескольких камер или систем.

Приложение Matrox Inspector полностью совместимо со всеми аппаратными компонентами Matrox Imaging, включая Matrox Morphis, Solios и Helios. Приложение Matrox Inspector также доступно в версии, поддерживающей семейство Odyssey.