

Библиотека алгоритмов адаптивной фильтрации

Пояснительная записка

Листов 55

2004

АННОТАЦИЯ

Настоящий документ является описанием составной части прикладной библиотеки алгоритмов и программ ГУП НПЦ «ЭЛВИС» для интегральных микросхем (ИМС) МС12 и МС24 серии «МУЛЬТИКОР». Данный документ охватывает лишь алгоритмы адаптивной фильтрации и подробно описывает только алгоритмы, имеющие в настоящее время реализацию для ИМС МС12 и МС24 на языке программирования Assembler в формате с плавающей точкой (с одинарной точностью) и на языке С в формате с плавающей точкой (с одинарной и двойной точностями).

Кроме того, приводятся вычислительные процедуры нескольких других алгоритмов адаптивной фильтрации, которые в настоящее время еще не имеют указанной реализации. Эти алгоритмы (12 разновидностей) имеют прототипы на языке программирования MATLAB. Всего существует около 250 MATLAB прототипов различных алгоритмов адаптивной фильтрации. Их перечень приведен в данном документе.

Все алгоритмы получены в виде, предназначенном для использования в многоканальных адаптивных фильтрах с неодинаковым числом комплексных весовых коэффициентов в каналах. Алгоритмы для одноканальных фильтров или фильтров с действительными весовыми коэффициентами являются частными случаями данного общего решения.

Прототипы алгоритмов существуют в виде отдельных исполняемых программ. Кроме того, часть этих алгоритмов интегрирована в графический интерфейс пользователя (ГИП), реализованный на языке программирования MATLAB. Вариант ГИП с 12 алгоритмами, представленными в данном документе, является дополнением к настоящему документу. В документе приводится описание ГИП и основных приемов его работы.

Библиотека алгоритмов адаптивной фильтрации не является окончательно сформировавшимся продуктом. Алгоритмы, по мере их реализации на языках Assembler и С для ИМС МС12 и МС24 (а также для вновь разрабатываемых ИМС серии «МУЛЬТИКОР»), будут интегрироваться в ГИП в виде MATLAB прототипов. Также продолжается разработка новых алгоритмов адаптивной фильтрации и их MATLAB прототипов.

СОДЕРЖАНИЕ

1. Общие сведения	4
2. Перечень адаптивных алгоритмов прикладной библиотеки.....	7
3. Математическое описание алгоритмов адаптивной фильтрации	16
4. Краткая интерпретация алгоритмов	27
5. Графический интерфейс пользователя.....	31
6. Перечень функций графического интерфейса пользователя	52
7. Оценочные результаты реализации алгоритмов на языке Assembler.....	54
Заключение.....	55

1. ОБЩИЕ СВЕДЕНИЯ

Алгоритмы адаптивной фильтрации являются неотъемлемой составной частью многих изделий современной электроники, связи, радиолокации и ряда других областей. Примерами использования адаптивных фильтров являются адаптивные антенные решетки, компенсаторы сигналов электрического эхо в модемах проводных каналов связи, компенсаторы сигналов акустического эхо в устройствах голосовой связи, выравниватели каналов связи и ряд других устройств. В основе таких устройств находится адаптивный фильтр. В зависимости от решаемой задачи, он может быть одноканальным или многоканальным, число коэффициентов в каналах может быть одинаковым или неодинаковым, а сами весовые коэффициенты могут быть действительные или комплексные, что определяется типом обрабатываемых сигналов.

Обобщенная структура многоканального адаптивного фильтра с неодинаковым числом комплексных весовых коэффициентов в каналах показана на рис. 1.

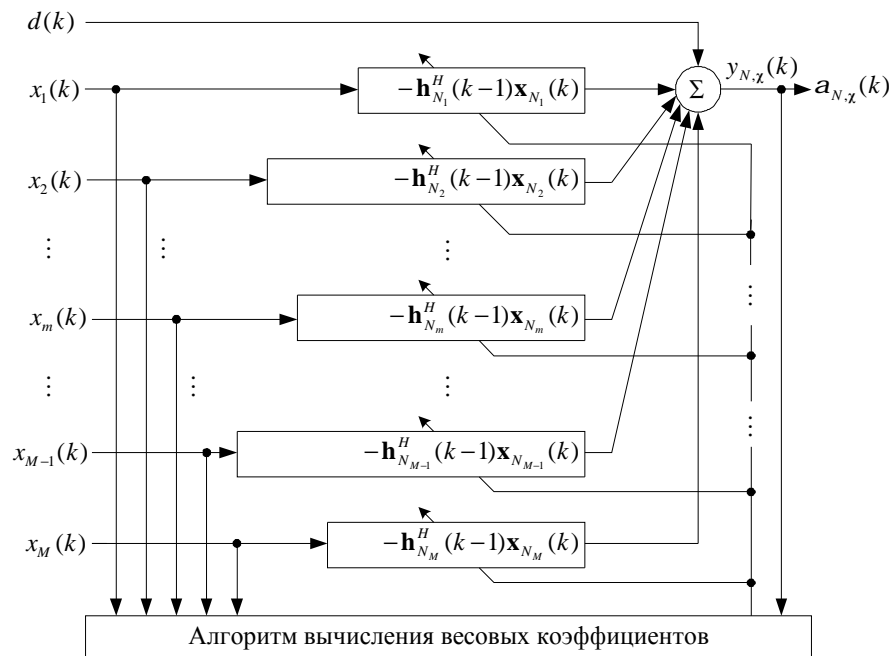


Рис. 1. Многоканальный адаптивный фильтр

Неодинаковое число весовых коэффициентов в каналах на рис. 1 условно обозначено в виде прямоугольников неодинаковой длины. Формулы внутри этих прямоугольников означают операцию вычисления одной итерации свертки для каждого из каналов.

Рассматриваемые далее алгоритмы представлены для случая комплексной арифметики. При использовании алгоритмов в адаптивных фильтрах с действительными весовыми коэффициентами, в математических обозначениях, представляющих описание алгоритмов, знак эрмитово сопряжения H (транспонирования матрицы или вектора и комплексное сопряжение элементов) следует заметить на знак транспонирования T , а операции комплексного сопряжения (*) просто опустить. Обратный переход от фильтров с действительными весовыми коэффициентами к фильтрам с комплексными весовыми коэффициентами не столь очевиден, так как некоторые скалярные переменные в фильтрах с комплексными весовыми коэффициентами принимают лишь действительные значения. Это является причиной того, почему базовые модели алгоритмов разработаны для случая комплексной арифметики.

В алгоритмах, все арифметические операции (умножение, сложение, вычитание и деление) выполняются в соответствии с законами используемой арифметики: действительная или комплексная.

Переменные, обозначенные на рис. 1, это $\mathbf{h}_N^H(k) = [\mathbf{h}_{N_1}^H(k), \mathbf{h}_{N_2}^H(k), \dots, \mathbf{h}_{N_m}^H(k), \dots, \mathbf{h}_{N_{M-1}}^H(k), \mathbf{h}_{N_M}^H(k)]$ – вектор весовых коэффициентов M -канального адаптивного фильтра; $\mathbf{h}_{N_m}(k) = [h_{0,m}, h_{1,m}, \dots, h_{N_m-2,m}, h_{N_m-1,m}]^T$ – вектор весовых коэффициентов в m -м канале фильтра; $\boldsymbol{\chi}_N^T(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_{M-1}}^T(k), \mathbf{x}_{N_M}^T(k)]$ – вектор сигналов в адаптивном фильтре; $\mathbf{x}_{N_m}(k) = [x_m(k), x_m(k-1), \dots, x_m(k-N_m+2), x_m(k-N_m+1)]^T$ – вектор сигналов в m -м канале фильтра; $N = \sum_{m=1}^M N_m$ – общее число весовых коэффициентов адаптивного фильтра; $d(k)$ – требуемый сигнал; $y_{N,\chi}(k)$ – выходной сигнал адаптивного фильтра; $a_{N,\chi}(k)$ – сигнал ошибки на выходе адаптивного фильтра; k – номер итерации алгоритма (дискретное время). Здесь и далее, матрицы и векторы обозначаются соответственно жирными прописными и строчными буквами, нижние индекс N и J используются для обозначения размерности векторов (с числом элементов N и J) и квадратных матриц (с числом элементов $N \times N$ и $J \times J$), а индексы NJ – для обозначения прямоугольных (не транспонированных) матриц (с числом элементов $N \times J$).

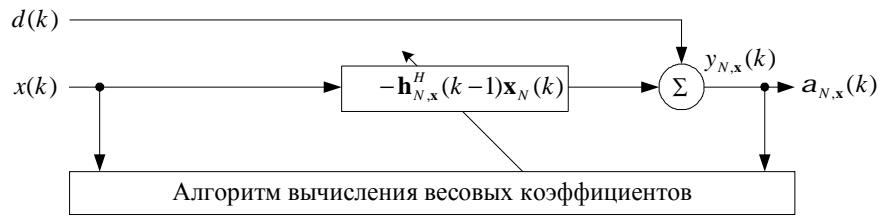


Рис. 2. Одноканальный адаптивный фильтр

Одноканальный адаптивный фильтр, представлен на рис. 2. В этом случае $\mathbf{h}_N^H(k) = [\mathbf{h}_{N_1}^H(k)]$ и $\boldsymbol{\chi}_N^T(k) = [\mathbf{x}_{N_1}^T(k)]$. Указанное соотношение между векторами сигналов и весовых коэффициентов одноканальных и многоканальных фильтров справедливо только для простых алгоритмов по критерию наименьшего среднеквадратичного отклонения (LMS, Least Means Squares), нормализованных LMS (NLMS, Normalized LMS) и рекурсивных алгоритмов по критерию наименьших квадратов (RLS, Recursive Least Squares). Для так называемых быстрых (вычислительно эффективных) RLS алгоритмов (в настоящем документе не представлены), это справедливо только в части вычислений, условно показанных на рис. 1 и рис. 2. Формирование входных сигналов, используемых для ряда других вычислений, в многоканальных быстрых RLS алгоритмах осуществляется несколько более сложным образом.

Качество работы адаптивного фильтра определяется используемым в нем адаптивным алгоритмом. В большинстве алгоритмов адаптивной фильтрации для вычисления весовых коэффициентов используются значения входных сигналов $x_m(k)$ и ошибки $a_{N,\chi}(k)$. В алгоритмах с линейными ограничениями, часть входных переменных задается аналитически. Перечень алгоритмов прикладной библиотеки, имеющих MATLAB прототипы, приведен в следующем разделе.

2. ПЕРЕЧЕНЬ АДАПТИВНЫХ АЛГОРИТМОВ ПРИКЛАДНОЙ БИБЛИОТЕКИ

В состав библиотеки адаптивных алгоритмов входят алгоритмы, многообразие которых определяется комбинацией следующих обозначений: LS – Least Squares (наименьших квадратов), RLS, QR GR (Givens Rotation) RLS – RLS на основе прямого QR разложения (с использованием вращений Гивенса), IQR GR RLS – RLS на основе обратного (Inverse) QR разложения (с использованием вращений Гивенса), IQR HT (Householder Transform) RLS – RLS на основе обратного QR разложения (с использованием преобразования Хаусхолдера), FK – Fast (быстрый) Kalman, FTF – Fast Transversal Filter, FAEST – Fast a Posteriori Error Sequential Technique, Stab. FAEST – Stabilized (стабилизированный) FAEST, IQR GR FRLS – быстрый RLS на основе обратного QR (с использованием вращений Гивенса), LS-2 – LS для реализации с помощью 2-х процессоров, RLS-2 – RLS для реализации с помощью 2-х процессоров, FRLS-2 – быстрый RLS для реализации с помощью 2-х процессоров, LS-4 – LS для реализации с помощью 4-х процессоров, RLS-4 – RLS для реализации с помощью 4-х процессоров, FRLS-4 – быстрый RLS для реализации с помощью 4-х процессоров, SR – Square Root (с операциями извлечения квадратного корня), SRF – Square Root Free (без операций извлечения квадратного корня), PW – бесконечное окно (с экспоненциальным взвешиванием, рис. 3), SW – скользящее окно (с экспоненциальным взвешиванием, рис. 4), RG – регуляризация обращения корреляционной матрицы, LC0 – линейные ограничения в LS, LC1–LC3 – три разновидности линейных ограничений в RLS и быстрых RLS.

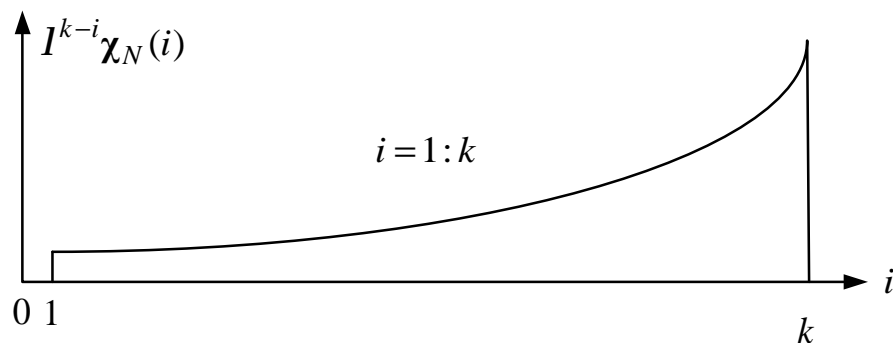


Рис. 3. Бесконечное окно с экспоненциальным взвешиванием

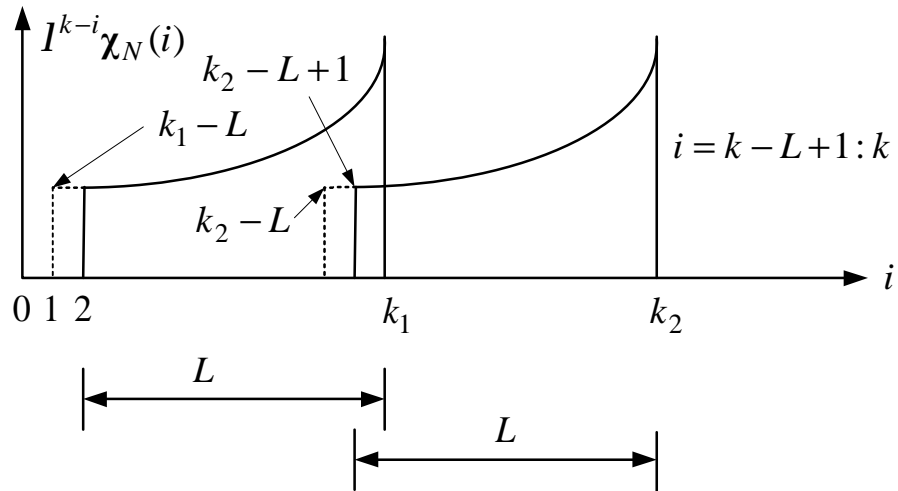


Рис. 4. Скользящее окно с экспоненциальным взвешиванием

В табл. 1 перечислено 206 RLS алгоритмов адаптивной фильтрации. Вместе с LMS, NLMS, LMS и NLMS с линейными ограничениями, LMS и NLMS в частотной области, аффинных проекций (AP, Affine Projection) и быстрыми AP (FAP, Fast AP) алгоритмами они составляют около 250 MATLAB прототипов алгоритмов адаптивной фильтрации.

Таблица 1. Разновидности RLS алгоритмов адаптивной фильтрации

№	Алгоритм	Окно	Регуляризация	Ограничения
1	LS	PW	-	-
2	RLS	PW	-	-
3	QR GR SR RLS	PW	-	-
4	IQR GR SR RLS	PW	-	-
5	IQR HT SR RLS	PW	-	-
6	QR GR SRF RLS	PW	-	-
7	IQR GR SRF RLS	PW	-	-
8	FK	PW	-	-
9	FTF	PW	-	-
10	FAEST	PW	-	-
11	Stab. FAEST	PW	-	-
12	IQR GR SR FRLS	PW	-	-
13	IQR GR SRF FRLS	PW	-	-
14	LS	PW	RG	-
15	RLS	PW	RG	-
16	QR GR SR RLS	PW	RG	-

17	IQR GR SR RLS	PW	RG	-
18	IQR HT SR RLS	PW	RG	-
19	QR GR SRF RLS	PW	RG	-
20	IQR GR SRF RLS	PW	RG	-
21	FK	PW	RG	-
22	FTF	PW	RG	-
23	FAEST	PW	RG	-
24	Stab. FAEST	PW	RG	-
25	IQR GR SR FRLS	PW	RG	-
26	IQR GR SRF FRLS	PW	RG	-
27	LS	PW	-	LC0
28	RLS	PW	-	LC1
29	RLS	PW	-	LC2
30	RLS	PW	-	LC3
31	IQR GR SR RLS	PW	-	LC1
32	IQR GR SR RLS	PW	-	LC2
33	IQR GR SR RLS	PW	-	LC3
34	IQR HT SR RLS	PW	-	LC1
35	IQR HT SR RLS	PW	-	LC2
36	IQR HT SR RLS	PW	-	LC3
37	IQR GR SRF RLS	PW	-	LC1
38	IQR GR SRF RLS	PW	-	LC2
39	IQR GR SRF RLS	PW	-	LC3
40	FK	PW	-	LC1
41	FK	PW	-	LC2
42	FK	PW	-	LC3
43	FTF	PW	-	LC1
44	FTF	PW	-	LC2
45	FTF	PW	-	LC3
46	FAEST	PW	-	LC1
47	FAEST	PW	-	LC2
48	FAEST	PW	-	LC3
49	Stab. FAEST	PW	-	LC1
50	Stab. FAEST	PW	-	LC2
51	Stab. FAEST	PW	-	LC3
52	IQR GR SR FRLS	PW	-	LC1
53	IQR GR SR FRLS	PW	-	LC2
54	IQR GR SR FRLS	PW	-	LC3
55	IQR GR SRF FRLS	PW	-	LC1

56	IQR GR SRF FRLS	PW	-	LC2
57	IQR GR SRF FRLS	PW	-	LC3
58	LS	PW	RG	LC0
59	RLS	PW	RG	LC1
60	RLS	PW	RG	LC2
61	RLS	PW	RG	LC3
62	IQR GR SR RLS	PW	RG	LC1
63	IQR GR SR RLS	PW	RG	LC2
64	IQR GR SR RLS	PW	RG	LC3
65	IQR HT SR RLS	PW	RG	LC1
66	IQR HT SR RLS	PW	RG	LC2
67	IQR HT SR RLS	PW	RG	LC3
68	IQR GR SRF RLS	PW	RG	LC1
69	IQR GR SRF RLS	PW	RG	LC2
70	IQR GR SRF RLS	PW	RG	LC3
71	FK	PW	RG	LC1
72	FK	PW	RG	LC2
73	FK	PW	RG	LC3
74	FTF	PW	RG	LC1
75	FTF	PW	RG	LC2
76	FTF	PW	RG	LC3
77	FAEST	PW	RG	LC1
78	FAEST	PW	RG	LC2
79	FAEST	PW	RG	LC3
80	Stab. FAEST	PW	RG	LC1
81	Stab. FAEST	PW	RG	LC2
82	Stab. FAEST	PW	RG	LC3
83	IQR GR SR FRLS	PW	RG	LC1
84	IQR GR SR FRLS	PW	RG	LC2
85	IQR GR SR FRLS	PW	RG	LC3
86	IQR GR SRF FRLS	PW	RG	LC1
87	IQR GR SRF FRLS	PW	RG	LC2
88	IQR GR SRF FRLS	PW	RG	LC3
89	LS-2	PW	RG	-
90	RLS-2	PW	RG	-
91	FRLS-2	PW	RG	-
92	LS-2	PW	RG	LC0
93	RLS-2	PW	RG	LC1
94	RLS-2	PW	RG	LC2

95	RLS-2	PW	RG	LC3
96	FRLS-2	PW	RG	LC1
97	FRLS-2	PW	RG	LC2
98	FRLS-2	PW	RG	LC3
99	LS	SW	-	-
100	RLS	SW	-	-
101	QR GR SR RLS	SW	-	-
102	IQR GR SR RLS	SW	-	-
103	IQR HT SR RLS	SW	-	-
104	QR GR SRF RLS	SW	-	-
105	IQR GR SRF RLS	SW	-	-
106	FK	SW	-	-
107	FTF	SW	-	-
108	FAEST	SW	-	-
109	Stab. FAEST	SW	-	-
110	IQR GR SR FRLS	SW	-	-
111	IQR GR SRF FRLS	SW	-	-
112	LS	SW	RG	-
113	RLS	SW	RG	-
114	QR GR SR RLS	SW	RG	-
115	IQR GR SR RLS	SW	RG	-
116	IQR HT SR RLS	SW	RG	-
117	QR GR SRF RLS	SW	RG	-
118	IQR GR SRF RLS	SW	RG	-
119	FK	SW	RG	-
120	FTF	SW	RG	-
121	FAEST	SW	RG	-
122	Stab. FAEST	SW	RG	-
123	IQR GR SR FRLS	SW	RG	-
124	IQR GR SRF FRLS	SW	RG	-
125	LS	SW	-	LC0
126	RLS	SW	-	LC1
127	RLS	SW	-	LC2
128	RLS	SW	-	LC3
129	IQR GR SR RLS	SW	-	LC1
130	IQR GR SR RLS	SW	-	LC2
131	IQR GR SR RLS	SW	-	LC3
132	IQR HT SR RLS	SW	-	LC1
133	IQR HT SR RLS	SW	-	LC2

134	IQR HT SR RLS	SW	-	LC3
135	IQR GR SRF RLS	SW	-	LC1
136	IQR GR SRF RLS	SW	-	LC2
137	IQR GR SRF RLS	SW	-	LC3
138	FK	SW	-	LC1
139	FK	SW	-	LC2
140	FK	SW	-	LC3
141	FTF	SW	-	LC1
142	FTF	SW	-	LC2
143	FTF	SW	-	LC3
144	FAEST	SW	-	LC1
145	FAEST	SW	-	LC2
146	FAEST	SW	-	LC3
147	Stab. FAEST	SW	-	LC1
148	Stab. FAEST	SW	-	LC2
149	Stab. FAEST	SW	-	LC3
150	IQR GR SR FRLS	SW	-	LC1
151	IQR GR SR FRLS	SW	-	LC2
152	IQR GR SR FRLS	SW	-	LC3
153	IQR GR SRF FRLS	SW	-	LC1
154	IQR GR SRF FRLS	SW	-	LC2
155	IQR GR SRF FRLS	SW	-	LC3
156	LS	SW	RG	LC0
157	RLS	SW	RG	LC1
158	RLS	SW	RG	LC2
159	RLS	SW	RG	LC3
160	IQR GR SR RLS	SW	RG	LC1
161	IQR GR SR RLS	SW	RG	LC2
162	IQR GR SR RLS	SW	RG	LC3
163	IQR HT SR RLS	SW	RG	LC1
164	IQR HT SR RLS	SW	RG	LC2
165	IQR HT SR RLS	SW	RG	LC3
166	IQR GR SRF RLS	SW	RG	LC1
167	IQR GR SRF RLS	SW	RG	LC2
168	IQR GR SRF RLS	SW	RG	LC3
169	FK	SW	RG	LC1
170	FK	SW	RG	LC2
171	FK	SW	RG	LC3
172	FTF	SW	RG	LC1

173	FTF	SW	RG	LC2
174	FTF	SW	RG	LC3
175	FAEST	SW	RG	LC1
176	FAEST	SW	RG	LC2
177	FAEST	SW	RG	LC3
178	Stab. FAEST	SW	RG	LC1
179	Stab. FAEST	SW	RG	LC2
180	Stab. FAEST	SW	RG	LC3
181	IQR GR SR FRLS	SW	RG	LC1
182	IQR GR SR FRLS	SW	RG	LC2
183	IQR GR SR FRLS	SW	RG	LC3
184	IQR GR SRF FRLS	SW	RG	LC1
185	IQR GR SRF FRLS	SW	RG	LC2
186	IQR GR SRF FRLS	SW	RG	LC3
187	LS-2	SW	RG	-
188	RLS-2	SW	RG	-
189	FRLS-2	SW	RG	-
190	LS-2	SW	RG	LC0
191	RLS-2	SW	RG	LC1
192	RLS-2	SW	RG	LC2
193	RLS-2	SW	RG	LC3
194	FRLS-2	SW	RG	LC1
195	FRLS-2	SW	RG	LC2
196	FRLS-2	SW	RG	LC3
197	LS-4	SW	RG	-
198	RLS-4	SW	RG	-
199	FRLS-4	SW	RG	-
200	LS-4	SW	RG	LC0
201	RLS-4	SW	RG	LC1
202	RLS-4	SW	RG	LC2
203	RLS-4	SW	RG	LC3
204	FRLS-4	SW	RG	LC1
205	FRLS-4	SW	RG	LC2
206	FRLS-4	SW	RG	LC3

Вместо экспоненциального взвешивания обрабатываемых сигналов может быть применено равномерное взвешивание (см. рис. 5 и рис. 6), что порождает еще 206 RLS и быстрых RLS алгоритмов. Такие алгоритмы имеют несколько меньшую вычислительную

сложность (по сравнению с алгоритмами с экспоненциальным взвешиванием) за счет отсутствия операций умножения на параметр экспоненциального взвешивания $0 << I < 1$. Данный параметр и скользящее окно являются «инструментами», с помощью которых обеспечивается возможность слежения за изменяющимися сигналами в случае обработки нестационарных сигналов.

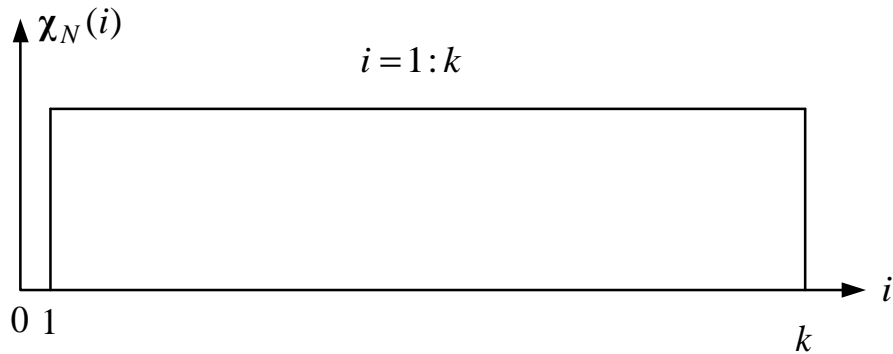


Рис. 5. Бесконечное окно с равномерным взвешиванием

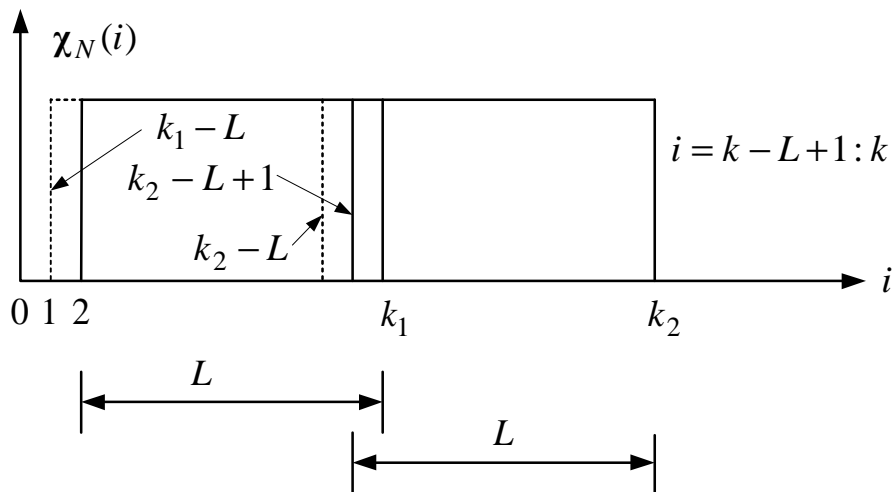


Рис. 6. Скользящее окно с равномерным взвешиванием

Экспоненциальный множитель используется для уменьшения вклада более ранних отсчетов обрабатываемых сигналов при оценке корреляционной матрицы адаптивного фильтра, в результате чего эти выборки как бы «забываются». Англоязычное название этого параметра - forgetting factor. Параметр I должен находиться в пределах значений $\max\{1 - 0,4/N_m\} \leq I \leq 1$, поскольку в противном случае адаптивный фильтр может становиться неустойчивым. Следовательно, регулирование следящих свойств RLS

алгоритмов с помощью этого параметра ограничивается числом весовых коэффициентов адаптивного фильтра.

В RLS алгоритмах со скользящим окном при оценке корреляционной матрицы используется всего L отсчетов. Более ранние отсчеты полностью исключаются. Это исключение осуществляется на каждой итерации. Скользящее окно обуславливает примерно удвоенную вычислительную сложность (число арифметических операций, необходимых для выполнения одной итерации алгоритма) по сравнению с одноименными RLS алгоритмами с бесконечным окном.

Регуляризация - это использование дополнительного искусственного сигнала при оценке корреляционной матрицы входных сигналов адаптивного фильтра, которая в явном или в неявном виде используется во всех RLS алгоритмах адаптивной фильтрации. Данный прием является одним из способов повышения устойчивости таких алгоритмов. Вычислительная сложность регуляризованных алгоритмов примерно в два раза больше одноименных нерегуляризованных RLS алгоритмов.

Дополнительная вычислительная сложность адаптивных RLS алгоритмов с экспоненциальным взвешиванием, скользящим окном и регуляризацией является той ценой, которую приходится платить за возможность обработки нестационарных сигналов с помощью таких алгоритмов.

Одноканальные алгоритмы, алгоритмы с действительными весовыми коэффициентами являются частными случаями алгоритмов табл. 1, т.е. возможны 4 разновидности любого из перечисленных алгоритмов. На основе алгоритмов табл. 1 могут быть созданы $K_{EiN} K_{A\ddot{e}\ddot{a}\ddot{i}\ddot{o}\ddot{i}\ddot{a}} K_{D\grave{a}\grave{c}\grave{i}\grave{t}\grave{a}\grave{e}\grave{a}\grave{i}\grave{i}} \grave{n}\grave{o}\grave{a}\grave{e} = 2 \times (250 + 200) \times 4 = 3600$ функций на языке Assembler ИМС МС12 и МС24. Такое же число функций порождается и при их реализации на языке С с одинарной и двойной точностью.

В настоящее время реализовать такое число функций не представляется возможным. Поэтому реализация данных функций на основе имеющихся алгоритмов табл. 1 и их прототипов на языке MATLAB осуществляется согласно внутреннему плану ГУП НПЦ «ЭЛВИС», который может корректироваться при поступлении заказов на интересующие адаптивные алгоритмы из табл. 1 или другие адаптивные алгоритмы, которые могут быть разработаны силами предприятия.

В следующем разделе приведено математическое описание 12 алгоритмов адаптивной фильтрации, поставляемых вместе с ГИП.

3. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ АЛГОРИТМОВ АДАПТИВНОЙ ФИЛЬТРАЦИИ

Обозначения основных переменных, используемых при описании следующих далее алгоритмов 1 - 12, представлены табл. 2. Переменные, не имеющие специального функционального назначения и не имеющие общепринятого названия в табл. 2 отсутствуют.

Таблица 2. Основные математические обозначения при описании LMS, NLMS алгоритмов и RLS алгоритмов с возрастающим окном

$\mathbf{R}_{N,\chi}$	Корреляционная матрица многоканального адаптивного фильтра.
\mathbf{C}_{NJ}	Матрица линейных ограничений.
$\mathbf{P}_{N,\chi}$	Проецирующая матрица многоканального адаптивного фильтра.
\mathbf{I}_N	Единичная матрица.
$\mathbf{\Lambda}_N$	Диагональная матрица.
$\mathbf{r}_{N,\chi}$	Вектор взаимной корреляции многоканального адаптивного фильтра
\mathbf{f}_J	Вектор значений линейных ограничений.
$\mathbf{h}_{N,\chi}$	Вектор весовых коэффициентов многоканального адаптивного фильтра.
$\mathbf{h}_{N_m,\mathbf{x}}$	Вектор весовых коэффициентов в m -м канале многоканального адаптивного фильтра.
χ_N	Вектор сигналов многоканального адаптивного фильтра.
\mathbf{x}_{N_m}	Вектор сигналов в m -м канале многоканального адаптивного фильтра.
$\mathbf{g}_{N,\chi}$	Вектор коэффициентов Калмана, для многоканального адаптивного фильтра.
$\mathbf{0}_N$	Нулевой вектор.
N	Число весовых коэффициентов адаптивного фильтра.
N_m	Число весовых коэффициентов в m -м канале многоканального адаптивного фильтра.
M	Число каналов многоканального адаптивного фильтра.
J	Число линейных ограничений.
k	Номер итерации работы адаптивного фильтра.
K	Число итераций работы адаптивного фильтра.
d	Требуемый (desired) сигнал адаптивного фильтра.
x_m	Входной сигнал m -го канала многоканального адаптивного фильтра.

$\mathbf{a}_{N,\chi}$	Сигнал ошибки многоканального адаптивного фильтра.
$y_{N,\chi}$	Выходной сигнал многоканального адаптивного фильтра.
l	Параметр экспоненциального взвешивания сигналов в RLS алгоритмах.
d	Параметр начальной регуляризации обращения корреляционной матрицы или регуляризации деления в определении динамического шага сходимости в NLMS алгоритмах.
m	Шаг сходимости в LMS алгоритмах или масштабирующий множитель шага сходимости в NLMS алгоритмах.
$()^*$	Знак комплексного сопряжения скалярной величины.
$()^T$	Знак транспонирования вектора или матрицы.
$()^H$	Знак эрмитово сопряжения вектора или матрицы (транспонирование и комплексное сопряжение элементов).

Следующие ниже 12 алгоритмов адаптивной фильтрации представлены в порядке, в котором они расположены в ГИП, рассматриваемом в разделе 5.

1. Многоканальный LMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{0}_N$

For $k = 1, 2, \dots, K$

1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$

2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$

3) $\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + m\boldsymbol{\chi}_N(k)\mathbf{a}_{N,\chi}^*(k)$

End for k

2. Многоканальный линейно-ограниченный LMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

- 0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{f}_J,$
- $\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{C}_{NJ}^H$
- For** $k = 1, 2, \dots, K$
- 1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
- $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1) \boldsymbol{\chi}_N(k)$
- 3) $\mathbf{h}_{N,\chi}(k) = \mathbf{P}_N [\mathbf{h}_{N,\chi}(k-1) + m \boldsymbol{\chi}_N(k) \mathbf{a}_{N,\chi}^*(k)] + \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{f}_J$
- End for** k

3. Модифицированный многоканальный линейно-ограниченный LMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

- 0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{Q}_{NJ} = \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1}, \mathbf{h}_{N,\chi}(0) = \mathbf{Q}_{NJ} \mathbf{f}_J$
- For** $k = 1, 2, \dots, K$
- 1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
- $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1) \boldsymbol{\chi}_N(k)$
- 3) $\mathbf{h}'_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + m \boldsymbol{\chi}_N(k) \mathbf{a}_{N,\chi}^*(k)$
- 4) $\mathbf{h}_{N,\chi}(k) = \mathbf{h}'_{N,\chi}(k) + \mathbf{Q}_{NJ} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N,\chi}(k)]$
- End for** k

4. Многоканальный NLMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

- 0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{0}_N$
- For** $k = 1, 2, \dots, K$

- 1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$
- 3) $\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \frac{m}{\boldsymbol{\chi}_N^H(k)\boldsymbol{\chi}_N(k) + d^2} \boldsymbol{\chi}_N(k) \mathbf{a}_{N,\chi}^*(k)$

End for k

5. Многоканальный линейно-ограниченный NLMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

- 0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{f}_J$,
 $\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{C}_{NJ}^H$

For $k = 1, 2, \dots, K$

- 1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$
- 3) $\mathbf{h}_{N,\chi}(k) = \mathbf{P}_N \left[\mathbf{h}_{N,\chi}(k-1) + \frac{m}{\boldsymbol{\chi}_N^H(k)\boldsymbol{\chi}_N(k) + d^2} \boldsymbol{\chi}_N(k) \mathbf{a}_{N,\chi}^*(k) \right] + \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} \mathbf{f}_J$

End for k

6. Модифицированный многоканальный линейно-ограниченный NLMS алгоритм адаптивной фильтрации с комплексными весовыми коэффициентами:

- 0) **Initialization** : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{Q}_{NJ} = \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1}, \mathbf{h}_{N,\chi}(0) = \mathbf{Q}_{NJ} \mathbf{f}_J$
- For** $k = 1, 2, \dots, K$

- 1) $\mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$

$$3) \mathbf{h}'_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \frac{m}{\boldsymbol{\chi}_N^H(k)\boldsymbol{\chi}_N(k) + d^2} \boldsymbol{\chi}_N(k) \mathbf{a}_{N,\chi}^*(k)$$

$$4) \mathbf{h}_{N,\chi}(k) = \mathbf{h}'_{N,\chi}(k) + \mathbf{Q}_{NJ} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N,\chi}(k)]$$

End for k

7. Многоканальный LS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами (алгоритм 1, табл. 1):

Initialization : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{0}_N$

$$0) \mathbf{R}_{N,\chi}(0) = d^2 \boldsymbol{\Lambda}_N, \boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$$

For $k = 1, 2, \dots, K$

$$1) \mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1 : M$$

$$\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$$

$$2) \mathbf{R}_{N,\chi}(k) = I \mathbf{R}_{N,\chi}(k-1) + \boldsymbol{\chi}_N(k) \boldsymbol{\chi}_N^H(k)$$

$$3) \mathbf{r}_{N,\chi}(k) = I \mathbf{r}_{N,\chi}(k-1) + \boldsymbol{\chi}_N(k) d^*(k)$$

$$4) \mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1) \boldsymbol{\chi}_N(k)$$

$$5) \mathbf{h}_{N,\chi}(k) = \mathbf{R}_{N,\chi}^{-1}(k) \mathbf{r}_{N,\chi}(k)$$

End for k

8. Многоканальный линейно-ограниченный LS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами (алгоритм 27, табл. 1):

Initialization : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{R}_{N,\chi}(0) = d^2 \boldsymbol{\Lambda}_N,$

$$0) \boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$$

$$\boldsymbol{\Gamma}_{NJ,\chi}(0) = \mathbf{R}_{N,\chi}^{-1}(0) \mathbf{C}_{NJ}, \boldsymbol{\Psi}_{J,\chi}^{-1}(0) = [\mathbf{C}_{NJ}^H \boldsymbol{\Gamma}_{NJ,\chi}(0)]^{-1}, \mathbf{h}_{N,\chi}(0) = \boldsymbol{\Gamma}_{NJ,\chi}(0) \boldsymbol{\Psi}_{J,\chi}^{-1}(0) \mathbf{f}_J$$

For $k = 1, 2, \dots, K$

- 1) $\mathbf{x}_{N_m}(k)|_{2,\dots,N_m} = \mathbf{x}_{N_m}(k-1)|_{1,\dots,N_m-1}, \mathbf{x}_{N_m}(k)|_1 = x_m(k), m = 1 : M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
- 2) $\mathbf{R}_{N,\chi}(k) = I\mathbf{R}_{N,\chi}(k-1) + \boldsymbol{\chi}_N(k)\boldsymbol{\chi}_N^H(k)$
- 3) $\mathbf{r}_{N,\chi}(k) = I\mathbf{r}_{N,\chi}(k-1) + \boldsymbol{\chi}_N(k)d^*(k)$
- 4) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$
- 5) $\mathbf{h}_{N,\chi}(k) = \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{r}_{N,\chi}(k) + \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{C}_{NJ}[\mathbf{C}_{NJ}^H\mathbf{R}_{N,\chi}^{-1}(k)\mathbf{C}_{NJ}]^{-1}[\mathbf{f}_J - \mathbf{C}_{NJ}^H\mathbf{R}_{N,\chi}^{-1}(k)\mathbf{r}_{N,\chi}(k)]$

9. Многоканальный RLS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами (алгоритм 1, табл. 2):

- Initialization :** $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{h}_{N,\chi}(0) = \mathbf{0}_N$
- 0) $\mathbf{R}_{N,\chi}^{-1}(0) = d^{-2}\boldsymbol{\Lambda}_N, \boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$
- For** $k = 1, 2, \dots, K$
- 1) $\mathbf{x}_{N_m}(k)|_{2,\dots,N_m} = \mathbf{x}_{N_m}(k-1)|_{1,\dots,N_m-1}, \mathbf{x}_{N_m}(k)|_1 = x_m(k), m = 1 : M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
 - 2) $\mathbf{g}_{N,\chi}(k) = \frac{\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}{I + \boldsymbol{\chi}_N^H(k)\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}$
 - 3) $\mathbf{R}_{N,\chi}^{-1}(k) = I^{-1}[\mathbf{R}_{N,\chi}^{-1}(k-1) - \mathbf{g}_{N,\chi}(k)\boldsymbol{\chi}_N^H(k)\mathbf{R}_{N,\chi}^{-1}(k-1)]$
 - 4) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$
 - 5) $\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \mathbf{g}_{N,\chi}(k)\mathbf{a}_{N,\chi}^*(k)$
- End for** k

10. Многоканальный линейно-ограниченный RLS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами №1 (алгоритм 28, табл. 1):

Initialization : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N$, $\mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(0) = d^{-2} \boldsymbol{\Lambda}_N$,

$$0) \boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$$

$$\boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) = \mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(0) \mathbf{C}_{NJ}, \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(0) = \left[\mathbf{C}_{NJ}^H \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) \right]^{-1}, \mathbf{h}_{N,\boldsymbol{\chi}}(0) = \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(0) \mathbf{f}_J$$

For $k = 1, 2, \dots, K$

$$1) \begin{aligned} & \mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M \\ & \boldsymbol{\chi}_N(k) = \left[\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \end{aligned}$$

$$2) \mathbf{g}_{N,\boldsymbol{\chi}}(k) = \frac{\mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(k-1) \boldsymbol{\chi}_N(k)}{I + \boldsymbol{\chi}_N^H(k) \mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(k-1) \boldsymbol{\chi}_N(k)}$$

$$3) \mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(k) = I^{-1} \left[\mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(k-1) - \mathbf{g}_{N,\boldsymbol{\chi}}(k) \boldsymbol{\chi}_N^H(k) \mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(k-1) \right]$$

$$4) \mathbf{l}_{J,\boldsymbol{\chi}}(k) = \frac{\boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(k-1) \mathbf{C}_{NJ}^H \mathbf{g}_{N,\boldsymbol{\chi}}(k)}{1 - \boldsymbol{\chi}_N^H(k) \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k-1) \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(k-1) \mathbf{C}_{NJ}^H \mathbf{g}_{N,\boldsymbol{\chi}}(k)}$$

$$5) \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(k) = I \left[\boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(k-1) + \mathbf{l}_{J,\boldsymbol{\chi}}(k) \boldsymbol{\chi}_N^H(k) \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k-1) \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(k-1) \right]$$

$$6) \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k) = I^{-1} \left[\boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k-1) - \mathbf{g}_{N,\boldsymbol{\chi}}(k) \boldsymbol{\chi}_N^H(k) \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k-1) \right]$$

$$8) \mathbf{a}_{N,\boldsymbol{\chi}}(k) = d(k) - \mathbf{h}_{N,\boldsymbol{\chi}}^H(k-1) \boldsymbol{\chi}_N(k)$$

$$9) \mathbf{h}_{N,\boldsymbol{\chi}}(k) = \mathbf{h}_{N,\boldsymbol{\chi}}(k-1) + \left[\mathbf{g}_{N,\boldsymbol{\chi}}(k) - I \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(k) \mathbf{l}_{J,\boldsymbol{\chi}}(k) \right] \mathbf{a}_{N,\boldsymbol{\chi}}^*(k)$$

End for k

11. Многоканальный линейно-ограниченный RLS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами №2 (алгоритм 29, табл. 1):

Initialization : $\boldsymbol{\chi}_N(0) = \mathbf{0}_N$, $\mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(0) = d^{-2} \boldsymbol{\Lambda}_N$,

$$0) \boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$$

$$\boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) = \mathbf{R}_{N,\boldsymbol{\chi}}^{-1}(0) \mathbf{C}_{NJ}, \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(0) = \left[\mathbf{C}_{NJ}^H \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) \right]^{-1}, \mathbf{h}_{N,\boldsymbol{\chi}}(0) = \boldsymbol{\Gamma}_{NJ,\boldsymbol{\chi}}(0) \boldsymbol{\Psi}_{J,\boldsymbol{\chi}}^{-1}(0) \mathbf{f}_J$$

For $k = 1, 2, \dots, K$

$$1) \begin{aligned} & \mathbf{x}_{N_m}(k) \Big|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1) \Big|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k) \Big|_1 = x_m(k), m = 1: M \\ & \boldsymbol{\chi}_N(k) = \left[\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \end{aligned}$$

- 2) $\mathbf{g}_{N,\chi}(k) = \frac{\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}{1 + \boldsymbol{\chi}_N^H(k)\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}$
 - 3) $\mathbf{R}_{N,\chi}^{-1}(k) = I^{-1}[\mathbf{R}_{N,\chi}^{-1}(k-1) - \mathbf{g}_{N,\chi}(k)\boldsymbol{\chi}_N^H(k)\mathbf{R}_{N,\chi}^{-1}(k-1)]$
 - 4) $\mathbf{l}_{J,\chi}(k) = \frac{\boldsymbol{\Psi}_{J,\chi}^{-1}(k-1)\mathbf{C}_{NJ}^H\mathbf{g}_{N,\chi}(k)}{1 - \boldsymbol{\chi}_N^H(k)\boldsymbol{\Gamma}_{NJ,\chi}(k-1)\boldsymbol{\Psi}_{J,\chi}^{-1}(k-1)\mathbf{C}_{NJ}^H\mathbf{g}_{N,\chi}(k)}$
 - 5) $\boldsymbol{\Psi}_{J,\chi}^{-1}(k) = I[\boldsymbol{\Psi}_{J,\chi}^{-1}(k-1) + \mathbf{l}_{J,\chi}(k)\boldsymbol{\chi}_N^H(k)\boldsymbol{\Gamma}_{NJ,\chi}(k-1)\boldsymbol{\Psi}_{J,\chi}^{-1}(k-1)]$
 - 6) $\boldsymbol{\Gamma}_{NJ,\chi}(k) = I^{-1}[\boldsymbol{\Gamma}_{NJ,\chi}(k-1) - \mathbf{g}_{N,\chi}(k)\boldsymbol{\chi}_N^H(k)\boldsymbol{\Gamma}_{NJ,\chi}(k-1)]$
 - 8) $\mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1)\boldsymbol{\chi}_N(k)$
 - 9) $\mathbf{P}_{N,\chi}(k) = \mathbf{I}_N - \boldsymbol{\Gamma}_{NJ,\chi}(k)\boldsymbol{\Psi}_{J,\chi}^{-1}(k)\mathbf{C}_{NJ}^H$
 - 10) $\mathbf{a}_{N,\chi}(k) = \boldsymbol{\Gamma}_{NJ,\chi}(k)\boldsymbol{\Psi}_{J,\chi}^{-1}(k)\mathbf{f}_J$
 - 11) $\mathbf{h}_{N,\chi}(k) = \mathbf{P}_{N,\chi}(k)[\mathbf{h}_{N,\chi}(k-1) + \mathbf{g}_{N,\chi}(k)\mathbf{a}_{N,\chi}^*(k)] + \mathbf{a}_{N,\chi}(k)$
- End for** k

12. Многоканальный линейно-ограниченный RLS алгоритм адаптивной фильтрации бесконечным окном и комплексными весовыми коэффициентами №3
(алгоритм 3, табл. 1):

- Initialization :** $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \mathbf{R}_{N,\chi}^{-1}(0) = d^{-2}\boldsymbol{\Lambda}_N,$
- $\boldsymbol{\Lambda}_N = \text{diag}(1, I, \dots, I^{N_1-1}, \dots, 1, I, \dots, I^{N_m-1}, \dots, 1, I, \dots, I^{N_M-1})$
- 0) $\boldsymbol{\Gamma}_{NJ,\chi}(0) = \mathbf{R}_{N,\chi}^{-1}(0)\mathbf{C}_{NJ}, \mathbf{Q}_{NJ,\chi}(0) = \boldsymbol{\Gamma}_{NJ,\chi}(0)[\mathbf{C}_{NJ}^H\boldsymbol{\Gamma}_{NJ,\chi}(0)]^{-1}$
 $\mathbf{h}_{N,\chi}(0) = \mathbf{Q}_{NJ,\chi}(0)\mathbf{f}_J$
- For** $k = 1, 2, \dots, K$
- 1) $\mathbf{x}_{N_m}(k)|_{2, \dots, N_m} = \mathbf{x}_{N_m}(k-1)|_{1, \dots, N_m-1}, \mathbf{x}_{N_m}(k)|_1 = x_m(k), m = 1: M$
 $\boldsymbol{\chi}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T$
 - 2) $\mathbf{g}_{N,\chi}(k) = \frac{\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}{1 + \boldsymbol{\chi}_N^H(k)\mathbf{R}_{N,\chi}^{-1}(k-1)\boldsymbol{\chi}_N(k)}$

$$3) \mathbf{R}_{N,\chi}^{-1}(k) = I^{-1} \left[\mathbf{R}_{N,\chi}^{-1}(k-1) - \mathbf{g}_{N,\chi}(k) \boldsymbol{\chi}_N^H(k) \mathbf{R}_{N,\chi}^{-1}(k-1) \right]$$

$$4) \mathbf{v}_{J,\chi}(k) = \mathbf{C}_{NJ}^H \mathbf{g}_{N,\chi}(k)$$

$$5) \mathbf{v}_{J,\chi}^H(k) = \boldsymbol{\chi}_N^H(k) \mathbf{Q}_{NJ,\chi}(k-1)$$

$$6) \mathbf{Q}'_{NJ,\chi}(k) = \left[\mathbf{Q}_{NJ,\chi}(k-1) - \mathbf{g}_{N,\chi}(k) \mathbf{v}_{J,\chi}^H(k) \right] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J,\chi}(k) \mathbf{v}_{J,\chi}^H(k)}{1 - \mathbf{v}_{J,\chi}^H(k) \mathbf{v}_{J,\chi}(k)} \right]$$

$$7) \mathbf{Q}_{NJ,\chi}(k) = \mathbf{Q}'_{NJ,\chi}(k) + \mathbf{C}_{NJ} \left(\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right)^{-1} \left[\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ,\chi}(k) \right]$$

$$8) \mathbf{a}_{N,\chi}(k) = d(k) - \mathbf{h}_{N,\chi}^H(k-1) \boldsymbol{\chi}_N(k)$$

$$9) \mathbf{h}'_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \mathbf{g}_{N,\chi}(k) \mathbf{a}_{N,\chi}^*(k)$$

$$10) \mathbf{h}_{N,\chi}(k) = \mathbf{h}'_{N,\chi}(k) + \mathbf{Q}_{NJ,\chi}(k) \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N,\chi}(k) \right]$$

End for k

При инициализации корреляционной матрицы или обратной корреляционной матрицы алгоритмов 7–12 вместо диагональной матрицы $\boldsymbol{\Lambda}_N$ может быть использована единичная матрица \mathbf{I}_N . Такая инициализация RLS алгоритмов обычно присутствует в публикациях, посвященных таким алгоритмам.

Инициализация по типу алгоритмов 7–12 позволяет сравнивать RLS и быстрые RLS алгоритмы. В быстрых RLS алгоритмах корреляционная матрица в явном виде отсутствует, в результате чего отсутствуют матричные операции с вычислительной сложностью $O(N^2)$ и алгоритмы становятся «быстрыми» (вычислительно эффективными). Поэтому, если инициализировать RLS алгоритмы так как это показано в вычислительных процедурах 7–12, то эти алгоритмы будут математически эквивалентными быстрым RLS алгоритмам (в настоящем документе не представлены). Эквивалентность означает обеспечение одинаковых показателей качества функционирования (например, сигнала ошибки) на каждой итерации сравниваемых алгоритмов при одинаковых параметрах адаптивных фильтров (например, одинаковом числе каналов, весовых коэффициентов в каналах и т.п.) и одинаковых входных сигналах.

Данный прием позволил разработать большое число алгоритмов адаптивной фильтрации (см. табл. 1), с более сложными математическим выражением, чем алгоритмы

7-12, и убедиться в работоспособности (правильности полученных вычислительных процедур) путем моделирования на языке программирования MATLAB в арифметике с плавающей точкой.

Вычислительная сложность рассмотренных алгоритмов адаптивной фильтрации (число арифметических операций: умножений, сложений и делений, выполняемых за одну итерацию алгоритма (интервал дискретизации сигналов) представлена в табл. 3. Номер в таблице соответствует номеру алгоритма. В названиях алгоритмов используются обозначения, представленные в разделе 2, Mod означает «Модифицированный» (см. названия LMS и NLMS алгоритмов).

Вычислительная сложность определяется как число арифметических операций, возникающих только при выполнении векторных и матричных операций. Т.е. вычислительная сложность является функцией от M , N и J .

Вычислительная сложность тестовых LS алгоритмов не определяется, поскольку они не предназначены для реализации в составе библиотеки, а служат для проверки правильности функционирования RLS алгоритмов.

Вычислительная сложность NLMS алгоритмов приводится в общем виде (без учета свойства инвариантности сигналов к сдвигу, см. следующий раздел). В случае если обрабатываемые сигналы обладают свойством инвариантности к сдвигу, вычислительная сложность NLMS алгоритмов уменьшается на N операций умножения и N операций сложения, т.е. совпадает с вычислительной сложностью соответствующих LMS алгоритмов.

Число операций деления (не зависящее от размерности используемых матриц и векторов) приводится в отдельной графе.

Фигурирующие в LMS и NLMS алгоритмах с линейными ограничениями матрицы \mathbf{P}_N , \mathbf{Q}_{NJ} и $\mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1}$ вычислительной сложностью «не обладают», поскольку являются постоянными параметрами, а, значит, вычисляются лишь один раз, в качестве начальных значений алгоритмов адаптации (в не реальном масштабе времени), и к вычислительной сложности рассматриваемого алгоритма отношения не имеют.

Символ * в полях вычислительной сложности LS алгоритмов означает, что вычислительная сложность не определена.

Таблица 3. Вычислительная сложность LMS, NLMS алгоритмов и RLS алгоритмов с возрастающим окном за одну итерацию

№ алго-ритма	Название алгоритма	Число Умножений	Число сложений	Число делений
1	LMS	$2N$	$2N$	-
2	LMS LC	$N^2 + 2N$	$3N$	-
3	LMS LC Mod	$2N + 2NJ$	$3N + 2NJ + J$	-
4	NLMS	$3N$	$3N$	1
5	NLMS LC	$N^2 + 3N$	$4N$	1
6	NLMS LC Mod	$3N + 2NJ$	$4N + 2NJ + J$	1
7	LS PW, тестовый для алгоритма 9	*	*	*
8	LS PW LC, тестовый для алгоритмов 10, 11, 12	*	*	*
9	RLS PW	$4N^2 + 4N$	$3N^2 + 3N$	1
10	RLS PW LC №1	$4N^2 + 4N + 5NJ + 4J^2 + 3J$	$3N^2 + 4N + 4NJ + 3J^2 + 2J$	2
11	RLS PW LC №2	$N^2J + 5N^2 + 4N + NJ^2 + 5NJ + 4J^2 + 2J$	$N^2J + 5N^2 + 4N + NJ^2 + 4NJ + 3J^2 + 2J$	2
12	RLS PW LC №3	$4N^2 + 4N + 3NJ^2 + 3NJ + 2J^2 + J$	$3N^2 + 4N + 3NJ^2 + 4NJ + 2J^2 + 2J$	2

4. КРАТКАЯ ИНТЕРПРЕТАЦИЯ АЛГОРИТМОВ

В данном разделе дается математическое введение в представленные в предыдущем разделе алгоритмы адаптивной фильтрации. Такая последовательность обусловлена тем, что данный документ не предназначен для изложения теории адаптивной фильтрации. Основная цель настоящего раздела - представить с единой точки зрения вычислительные процедуры алгоритмов адаптивной фильтрации, которые могут быть получены на основе известных приемов теории адаптивной обработки сигналов. Под единой точкой зрения понимается объяснения того факта, что большинство алгоритмов можно рассматривать как результат тождественных преобразований, в результате которых появляются данные алгоритмы. При этом единными являются большинство используемых обозначений, что позволяет легко ориентироваться в многообразии имеющихся алгоритмов.

Сначала будут рассмотрены некоторые отправные точки получения RLS алгоритмов, а затем будет показано, что NLMS и LMS алгоритмы можно рассматривать как частные случаи RLS алгоритмов.

Переход от RLS алгоритмов к NLMS и LMS алгоритмам осуществляется вследствие естественного стремления уменьшить вычислительную сложность. В литературе, NLMS и LMS алгоритмы получены на основе других отправных точек, а именно на основе рассмотрения процедур градиентного спуска.

RLS алгоритмы можно получить следующим образом. Целью адаптивной фильтрации по критерию наименьших квадратов является минимизация на интервале наблюдения энергии ошибок между сигналом основного входа $d(k)$ и выходным сигналом адаптивного фильтра:

$$E_N(k) = \sum_{i=1}^k I^{k-i} \left[d(i) - \mathbf{h}_{N,\chi}^H(k) \boldsymbol{\chi}_N(i) \right]^2. \quad (1)$$

В алгоритмах с линейными ограничениями также накладывается условие

$$\mathbf{C}_{NJ}^H \mathbf{h}_{N,\chi}(k) = \mathbf{f}_J. \quad (2)$$

Минимизация функционала (1) позволяет найти вектор оптимальных весовых коэффициентов адаптивного фильтра (Винеровское решение) как

$$\mathbf{h}_{N,\chi}(k) = \mathbf{R}_{N,\chi}^{-1}(k) \mathbf{r}_{N,\chi}(k) \quad (3)$$

в случае алгоритмов без ограничений, и как

$$\mathbf{h}_{N,\chi}(k) = \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{r}_{N,\chi}(k) + \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_{N,\chi}^{-1}(k)\mathbf{r}_{N,\chi}(k) \right] \quad (4)$$

в случае алгоритмов с линейными ограничениями. Из уравнения (4) следует, что LS алгоритм с линейными ограничениями можно условно разделить на две части. Первая часть является такой же, как и в алгоритме без ограничений, а вторая часть представляет дополнительные вычисления, обусловленные наличием ограничений.

Уравнение (3) реализует алгоритм 7 настоящего документа, а уравнение (4) – алгоритм 8. Данные алгоритмы не предназначены для реализации в составе прикладной библиотеки, поскольку их вычислительная сложность пропорциональна $O(N^3)$ арифметических операций, которые необходимо выполнить за одну итерацию работы алгоритма. Это оценка обусловлена наличием операций обращения матриц, вычислительная сложность которых пропорциональна $O(N^3)$.

Термин «Recursive Least Squares» означает рекурсивное вычисление обратной корреляционной матрицы в уравнениях (3) и (4). Такое вычисление возможно благодаря тому, что на каждой итерации корреляционная матрица может быть представлена суммой двух составляющих, и инвертирование такой матрицы может быть осуществлено рекурсивным способом на основе использования леммы об инверсии матрицы. Подобные рекурсивные вычисления могут выполняться и для обращения матриц во втором слагаемом в уравнении (4). Таким способом могут быть получены RLS алгоритмы с линейными ограничениями и без ограничений с вычислительной сложностью $O(N^2)$ арифметических операций. Данные алгоритмы в настоящем документе представлены алгоритмами 9–12.

В RLS алгоритмах без ограничений, вектор оптимальных коэффициентов на каждой итерации определяется как

$$\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \mathbf{g}_{N,\chi}(k)\mathbf{a}_{N,\chi}^*(k). \quad (5)$$

Здесь вектор коэффициентов Калмана $\mathbf{g}_{N,\chi}(k)$ определяется как

$$\mathbf{g}_{N,\chi}(k) = \mathbf{R}_{N,\chi}^{-1}(k)\boldsymbol{\chi}_N(k). \quad (6)$$

В RLS алгоритмах рекурсивное вычисление этого вектора и обратной корреляционной матрицы являются взаимосвязанными. С учетом (6), уравнение (5) может быть представлено как

$$\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \mathbf{R}_{N,\chi}^{-1}(k)\boldsymbol{\chi}_N(k)\mathbf{a}_{N,\chi}^*(k). \quad (7)$$

Если в этом уравнении корреляционную матрицу аппроксимировать сначала диагональной (т.е. оставить только диагональные элементы), а затем аппроксимировать просто суммой этих диагональных элементов (следом корреляционной матрицы), то уравнение для вычисления коэффициентов фильтра упрощается как

$$\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + \frac{1}{\chi_N^H(k)\chi_N(k)} \chi_N(k) a_{N,\chi}^*(k). \quad (8)$$

Уравнение (8) представляет собой NLMS алгоритм (без учета слагаемого, обусловленного необходимостью регуляризации деления в знаменателе, и масштабирующего множителя m). Переменная $\frac{1}{\chi_N^H(k)\chi_N(k)}$ является динамическим (зависящим от входных сигналов)

шагом сходимости NLMS алгоритма, а коэффициент m в данном случае является просто масштабирующим множителем этого шага сходимости (дополнительной степенью свободы, позволяющей регулировать скорость сходимости алгоритма). Шаг сходимости является меняющейся (динамической) скалярной действительной величиной. Если вычисление динамического шага сходимости заменить фиксированным шагом m , то будет получен LMS алгоритм

$$\mathbf{h}_{N,\chi}(k) = \mathbf{h}_{N,\chi}(k-1) + m\chi_N(k) a_{N,\chi}^*(k). \quad (9)$$

Из теории устойчивости таких алгоритмов следует, что допустимое значение шага сходимости должно удовлетворять условию

$$0 < m \leq \frac{2}{\chi_N^H(k)\chi_N(k)}. \quad (10)$$

LMS, NLMS и RLS алгоритмы с линейными ограничениями получают с помощью несколько разных подходов, но в их основе находится один и тот же метод, известный как метод множителей Лагранжа. С помощью этого метода решение задачи условной оптимизации (обуславливается наличием ограничений) сводится к эквивалентной безусловной задаче (без ограничений).

Вычислительная сложность LMS и NLMS алгоритмов пропорциональна $O(N)$ арифметических операций за одну итерацию работы, а именно $2N$ (без учета вычисления динамического шага сходимости). Вычисления динамического шага сходимости в NLMS алгоритмах также требует N арифметических операций. Однако при обработке сигналов, обладающих свойством инвариантности к сдвигу, это вычисление можно свести к вычислению, не зависящему от N . В этом случае знаменатель в выражении шага

сходимости адаптивного NLMS алгоритма можно определить с помощью рекуррентного соотношения как

$$P(k) = P(k-1) + \sum_{m=1}^M \{x_m^2(k) - x_m^2(k - N_m)\}. \quad (11)$$

Вычислительная сложность LMS и NLMS алгоритмов адаптивной фильтрации с линейными ограничениями пропорциональна $O(N^2)$ арифметических операций за одну итерацию. Это обусловлено вычислениями, связанными с проецирующей матрицей \mathbf{P}_N . С помощью тождественных преобразований эти алгоритмы могут быть получены в модифицированном виде с вычислительной сложностью $O(N)$, см. алгоритмы 3 и 6. Допустимое значение шага сходимости LMS алгоритма с линейными ограничениями определяется не уравнением (10), а как

$$0 < \mu \leq \frac{2}{3[\boldsymbol{\chi}_N^H(k)\boldsymbol{\chi}_N(k)]}. \quad (12)$$

Таким образом, NLMS алгоритм адаптивной фильтрации можно рассматривать как частный случай RLS алгоритма (при длине окна, равной одной выборке обрабатываемых сигналов), а LMS алгоритм как частный случай NLMS алгоритма. Такое упрощение объясняет то, что LMS и NLMS алгоритмы обладают худшими показателями качества, чем RLS алгоритмы, зависящими от шага сходимости, который отсутствует в RLS алгоритмах. Данное утверждение в одинаковой мере справедливо в отношении алгоритмов без ограничений и с линейными ограничениями.

5. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

MATLAB прототипы алгоритмов адаптивной фильтрации реализованы в виде исполняемых моделей. Большинство алгоритмов табл. 1 также интегрировано в графический интерфейс пользователя (ГИП). В качестве приложения к настоящему документу прилагается набор закрытых функций, реализующих ГИП и интегрированные алгоритмы адаптивной фильтрации, которые представлены в разделе 3. Перечень функций приведен в следующем разделе.

ГИП является виртуальным прибором. Этот прибор позволяет демонстрировать работоспособность алгоритмов адаптивной фильтрации при работе с источниками внутренних тестовых сигналов или исследовать поведение этих алгоритмов при решении прикладных задач. В последнем случае используется режим работы ГИП с файлами внешних источников сигналов. Число элементов управления ГИП определяется алгоритмами, табл. 1.

При работе с источниками внутренних тестовых сигналов, адаптивные алгоритмы решают задачу идентификации неизвестного импульсного отклика. Решение именно этой задачи позволяет выделить очевидные критерии, демонстрирующие работоспособность алгоритмов. Возможность сохранения входных-выходных сигналов в эксперименте позволяет использовать эти сигналы в качестве тестовых векторов при проверке алгоритмов, перенесенных на другие языки программирования.

В ГИП невозможно ввести неправильные параметры или параметры, не соответствующих выбранному алгоритму. Для этого вводимые параметры проверяются на соответствие допустимым значениям. Определенную информацию о допустимых значениях параметров дают их исходные значения, отображаемых ГИП при его включении. Несвойственные выбранному алгоритму параметры или органы управления являются недоступными, что облегчает пользование ГИП.

Настоящая версия ГИП разработана и протестирована в среде программирования MATLAB-6.5 версия 13 под управлением операционной системы Windows 2000 и Windows XP. Полная работоспособность ГИП в более ранних и более поздних версиях данного языка программирования и операционной системы Windows не гарантируется.

Для работы с ГИП необходимо директорию с функциями (см. следующий раздел) назначить в качестве текущей директории (Current Directory) оболочки MATLAB.

Затем, в основном окне оболочки языка программирования MATLAB нужно набрать и выполнить команду `adaptive_filters_gui`, являющуюся основной исполняемой функцией ГИП. После некоторого времени, определяемого ресурсами ПК, появится окно, рис. 7, являющееся исходным окном ГИП. Все последующие изменения в отображении ГИП происходят в пределах данного окна, за исключением случая вывода графиков стандартного размера.

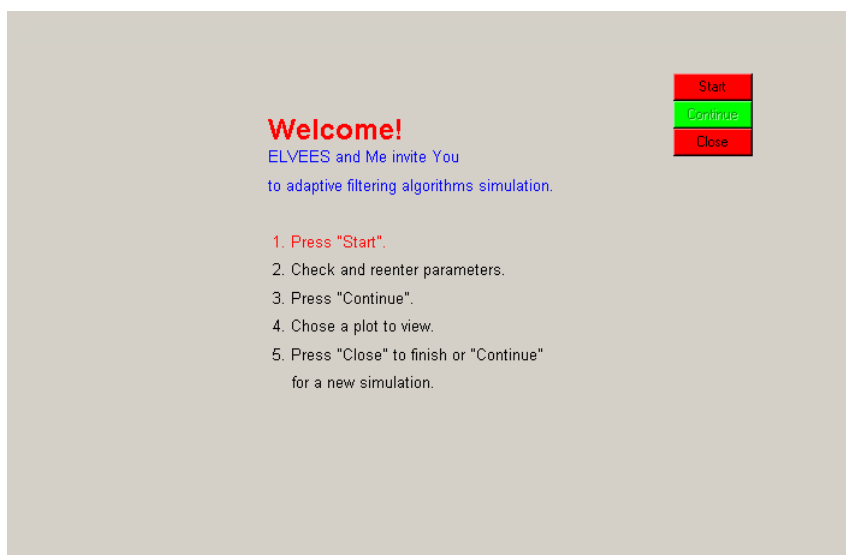


Рис. 7. Вид ГИП после включения

Рис. 7 представляет начальное состояние ГИП после включения. В этом состоянии в ГИП отображается приглашение: «Welcome! ELVEES and Me invite You to adaptive filtering algorithms simulation», что означает «Добро пожаловать. ЭЛВИС и я (ГИП) приглашают вас к моделированию алгоритмов адаптивной фильтрации».

Ниже этого сообщения приводится перечень последовательности шагов работы с ГИП. Среди этих шагов первый шаг выделен красным цветом. В данном случае это шаг 1. Press «Start», означает «Нажмите кнопку Start (Начало)». В данном состоянии отображаются только 3 управляющие кнопки ГИП, из которых только 2 являются доступными: уже упомянутая кнопка «Start» и кнопка «Close», означающая «Закреть». Нажатие этой кнопки или значка «x» (крестик, в верхнем правом углу ГИП) приводит к выключению ГИП. Кнопка «Continue» («Продолжить») в данном состоянии ГИП не доступна. Надписи на недоступных органах управления выделяются серым цветом. Кнопка «Continue» становится доступной только после нажатия кнопки «Start». Изображение ГИП не может

быть увеличено в размере. Это изображение может быть только минимизировано (свернуто) с помощью символа «_», расположенного в правом верхнем углу ГИП.

После нажатия кнопки «Start», внешний вид ГИП несколько меняется, см. рис. 8. В этом положении кнопка «Start» является недоступной, а кнопка «Continue» доступной. Кроме того, становятся видимыми и доступными органы управления, с помощью которых может быть выбран интересующий алгоритм адаптивной фильтрации и соответствующие этому алгоритму параметры. Эти органы расположены в левой и нижней частях ГИП. Параметры, не свойственные выбранному алгоритму, являются недоступными для изменения. Значения этих параметров отображаются серым цветом, а надписи, обозначающие эти параметры, отображаются синим цветом. Значения доступных параметров (т.е. тех, которые можно изменять) отображаются черным цветом, а надписи над этими параметрами - красным цветом.

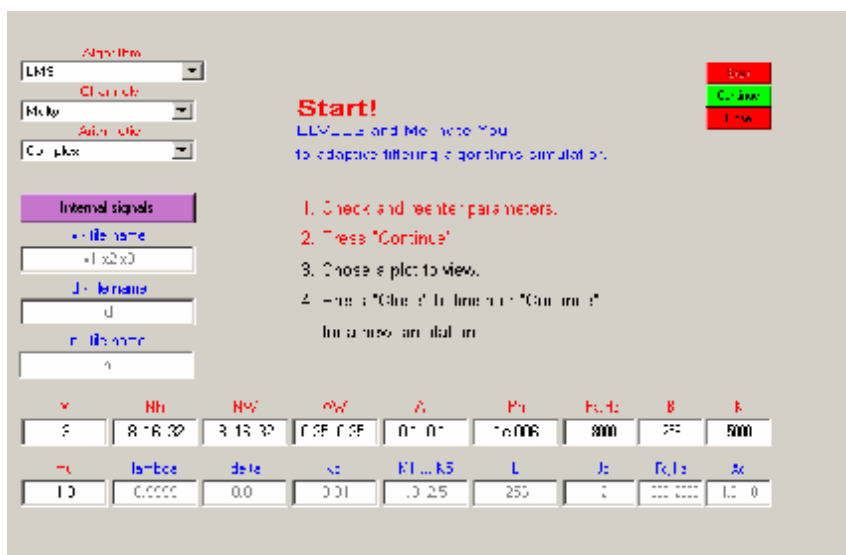


Рис. 8. ГИП после нажатия кнопки «Start»

При вводе некорректного значения любого из доступных параметров, группа зависимых параметров отображается желтым цветом, а в верхней части ГИП появляется предупреждающее сообщение на красном фоне, которое исчезает по истечении нескольких секунд, см. рис. 9. В этом состоянии кнопка «Continue» является недоступной. После изменения неправильно введенного параметра на правильное значение, или соответствующего изменения всей группы выделенных желтым цветом параметров на согласованное, желтый цвет исчезает, а кнопка «Continue» становится доступной. Это

означает, что в таком состоянии работа с ГИП может быть продолжена, о чем также информируют помеченные красным цветом сообщения в центральной части ГИП: «1. Check and reenter parameters. 2. Press «Continue», означающие «1. Проверьте и измените параметры. 2. Нажмите кнопку «Continue».

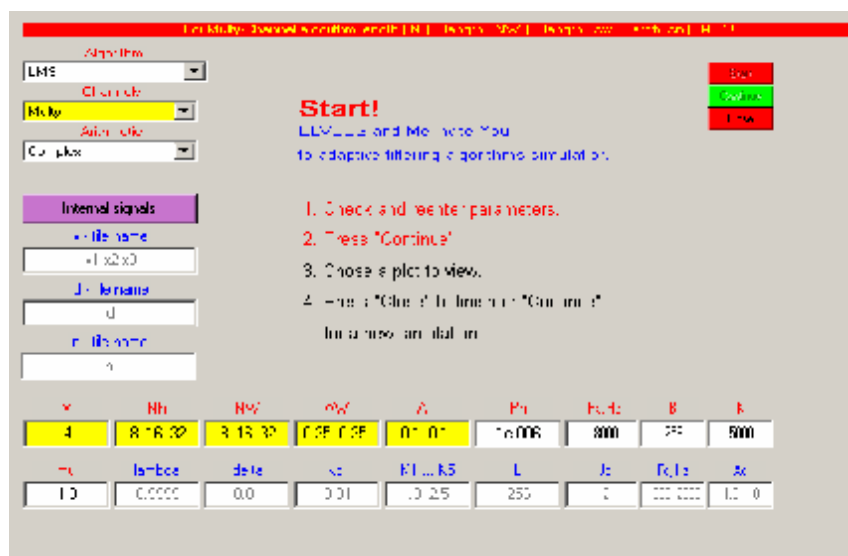


Рис. 9. Состояние ГИП после нажатия кнопки «Start» и ввода неправильного значения одного из параметров

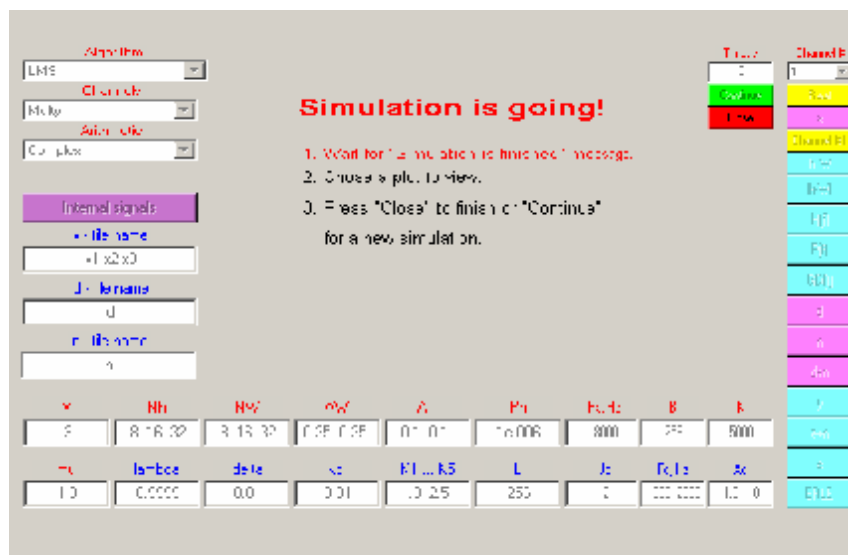


Рис. 10. ГИП после нажатия кнопки «Continue»

После нажатия кнопки «Continue», все органы управления ГИП, кроме кнопки «Close», становятся недоступными, а центральной части ГИП появляется новое сообщение, рис. 10. Надпись «Simulation is going!» означает «Идет моделирование!», а надпись «Wait for

«Simulation is finished» message» означает «Ждите появления сообщения «Моделирование закончилось»».

На этой стадии работы ГИИ исчезает кнопка «Start», и на ее месте появляется окно с надписью «Time, s» . Это окно сохраняется все время при моделировании других алгоритмов. После окончания моделирования в данном окне появляется время в секундах, означающее время моделирования заданного числа итераций выбранного алгоритма на данном персональном компьютере. Это позволит ориентироваться в ожидаемом времени последующего моделирования этого же алгоритма с таким же числом весовых коэффициентов адаптивного фильтра, но с другим числом итераций алгоритма. Кроме того, в правой части ГИП появляется ряд кнопок, которые в процессе моделирования недоступны. Это кнопки выбора графиков, отображающих результаты моделирования. После окончания моделирования, данные кнопки становятся доступными, см. рис. 11.

Окончание моделирования сопровождается сообщением в центральной части ГИП: «Simulation is finished!» - «Моделирование закончилось!» Ниже красным цветом выделяется сообщение «Chose a plot to view» - «Выберите график для просмотра». В окошке на месте кнопки «Start» отображается время моделирования алгоритма. В верхней части ГИП синим цветом отображается полное название алгоритма, который моделировался.

После нажатия любой из кнопок вывода графиков, в центральной части ГИП появляется окно с соответствующим графиком, а ниже кнопки «Close» - дополнительный набор органов управления, рис. 12.

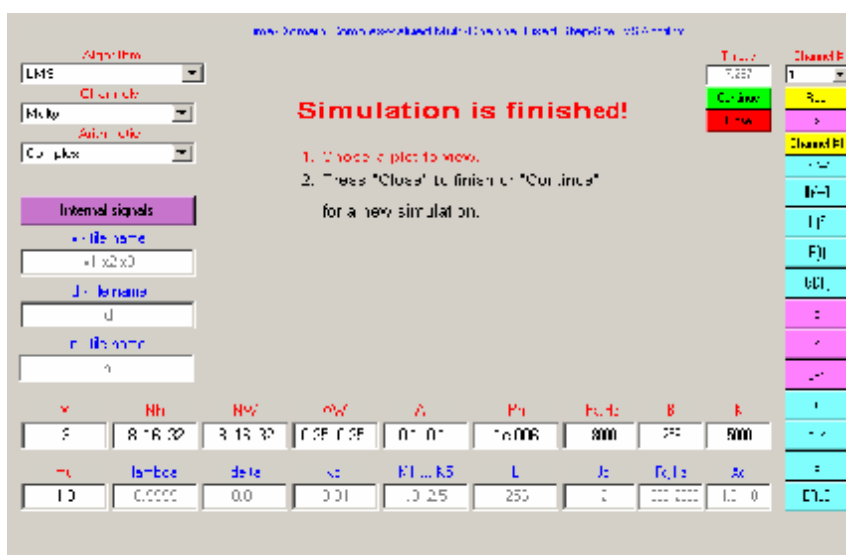


Рис. 11. ГИП после окончания моделирования

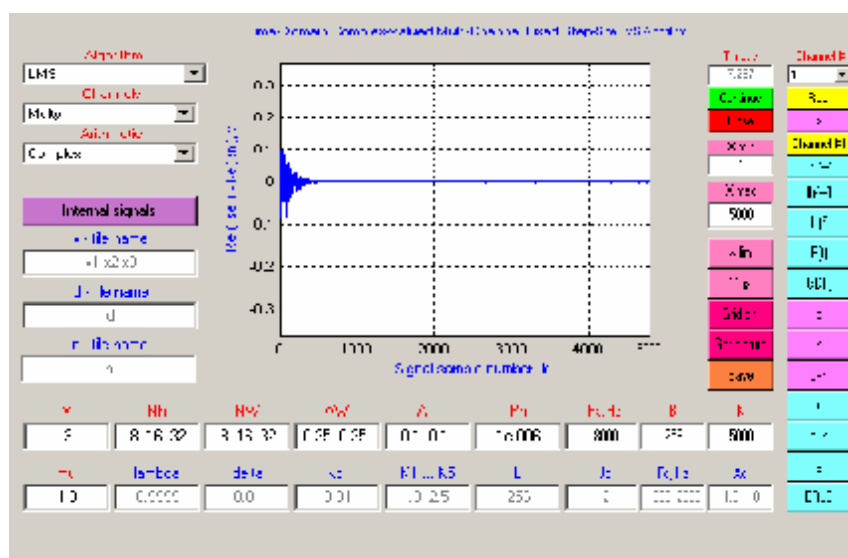


Рис. 12. ГИП после нажатия одной из кнопок вывода графиков результатов моделирования

Рис. 12 отображает все возможные органы управления разработанного ГИП, за исключением кнопки «Start». Описание органов управления ГИП приводится далее.

Пока лишь остановимся на кнопке «Get picture». Нажатие этой кнопки отображает текущий график, расположенный в центральной части ГИП, в отдельном окне как стандартный график, выводимый средствами языка MATLAB, рис. 13.

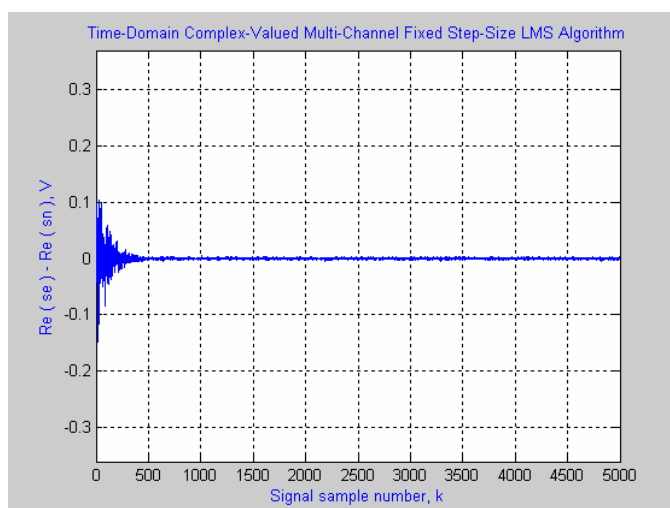


Рис. 13. График - копия графика в центральной части ГИП

Не показанные на рисунке меню и панель инструментов этого графического объекта позволяют копировать, сохранять, модифицировать график с целью дальнейшего

документирования или детального просмотра. Данный график не может быть выключен путем нажатия на символ «х» в правом верхнем углу окна. Он исчезает только при нажатии любой из кнопок вывода графиков, нажатии кнопки «Continue» или при выключении ГИП с помощью кнопки «Close».

После окончания моделирования можно поменять параметры алгоритма или выбрать иной алгоритм адаптивной фильтрации и провести следующее моделирование, нажав кнопку «Continue».

Выключать ГИП можно либо с помощью символа «х» в правом верхнем углу его окна или с помощью кнопки «Close». Процесс выключения сопровождается исчезновением всех управляющих органов и графиков ГИП. В центре ГИП на несколько секунд возникает сообщение, рис. 14, «See you next time!», «Hope, it was useful for you!» - «До следующей встречи», «Надеюсь, это было полезным». После этого график, рис. 13, если такой был активизирован, и ГИП выключаются.

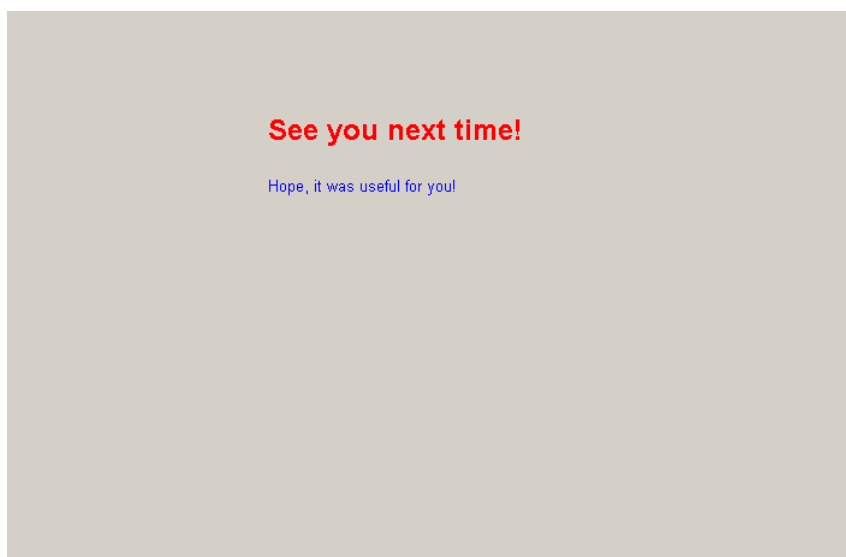


Рис. 14. Выключение ГИП.

На рис. 15 приведен внешний вид ГИП. Далее следует описание его элементов управления.

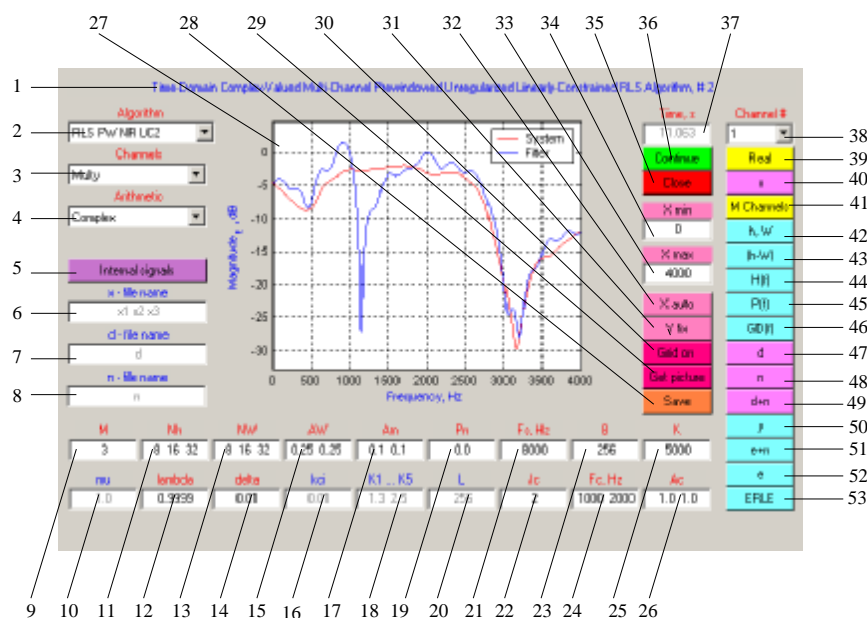


Рис. 15. Элементы ГИП

1. **Имя алгоритма адаптации.** Это имя выводится после окончания моделирования. До начала моделирования, при вводе неправильных значений параметров алгоритма или несогласованности этих параметров с зависимыми параметрами, на этом месте желтым цветом на красном фоне в течение нескольких секунд появляется предупреждающее сообщение, соответствующее ситуации.

2. **Выпадающее меню выбора алгоритма адаптивной фильтрации.** Исходное состояние (по умолчанию) определяет LMS алгоритм.

3. **Выпадающее меню выбора числа каналов алгоритма адаптивной фильтрации.** В исходном состоянии выбран многоканальный вариант адаптивного фильтра «Multy». Вторая возможность выбора «Single» определяет одноканальный адаптивный фильтр.

4. **Выпадающее меню выбора арифметики алгоритма адаптивной фильтрации.** В исходном состоянии выбрана комплексная арифметика «Complex». Вторая возможность выбора «Real» определяет адаптивный фильтр с действительными весовыми коэффициентами.

5. **Кнопка выбора типа тестовых сигналов.** В исходном состоянии на этой кнопке отображается надпись «Internal signals», означающая, что при моделировании в качестве входных сигналов адаптивного фильтра используются шумоподобные сигналы, вырабатываемые в ГИП. При нажатии на эту кнопку, на ней возникает надпись «External signals», означающая использование внешних сигналов для моделирования. Источниками

внешних сигналов являются файлы, имена которых вводятся в окнах, расположенных ниже этой кнопки.

6. Поле ввода имен файлов входных сигналов $x_m(k)$ адаптивного фильтра. Имена файлов формата wav вводятся без расширения (как показано на рисунке). Эти файлы должны быть помещены в рабочую директорию. Число файлов должно соответствовать числу каналов M адаптивного фильтра. В случае адаптивных фильтров с комплексными весовыми коэффициентами, действительная часть сигнала записывается в левый канал, а мнимая - в правый канал wav-файла формата стерео сигналов.

7. Поле ввода имени файла входного сигнала $d(k)$ адаптивного фильтра. Имя файла формата wav вводятся без расширения (как показано на рисунке). Этот файлы должны быть помещены в рабочую директорию. В случае адаптивных фильтров с комплексными весовыми коэффициентами, действительная часть сигнала записывается в левый канал, а мнимая - в правый канал wav-файла формата стерео сигналов.

8. Поле ввода имени файла шума $z(k)$ на входе $d(k)$ адаптивного фильтра. Имена файлов формата wav вводятся без расширения (как показано на рисунке). Эти файлы должны быть помещены в рабочую директорию. Число файлов должно соответствовать числу каналов M адаптивного фильтра. В случае адаптивных фильтров с комплексными весовыми коэффициентами, действительная часть сигнала записывается в левый канал, а мнимая - в правый канал wav-файла формата стерео сигналов.

9. Поле ввода числа каналов M адаптивного фильтра.

10. Поле ввода шага сходимости μ адаптивного фильтра. Поле активно в LMS, NLSM, AP и FAP алгоритмах.

11. Поле ввода числа коэффициентов N_m в каналах адаптивного фильтра. Это поле должно содержать M значений N_m , разделенных пробелами.

12. Поле ввода параметра I адаптивного фильтра. Поле активно в LS, RLS и быстрых RLS алгоритмах. Параметр I определяет эффективную память алгоритма.

13. Поле ввода числа коэффициентов N_m^W идентифицируемого импульсного отклика. Это поле должно содержать M значений числа коэффициентов в каналах идентифицируемого импульсного отклика, разделенных пробелами. Допустимые значения: $N_m^W \leq N_m$. Максимальное значение модуля коэффициента импульсного отклика равно единице. При работе с источниками внешних сигналов данное поле не активно.

14. **Поле ввода параметра d^2 адаптивного фильтра.** Поле активно при выборе LS, RLS и быстрых RLS алгоритмов. Параметр d^2 является параметром начальной регуляризации корреляционной матрицы.

15. **Поле ввода относительных амплитуд идентифицируемого импульсного отклика.** Это поле должно содержать M значений относительных амплитуд, разделенных пробелами. Данные амплитуды позволяют увеличивать или уменьшать коэффициенты идентифицируемого импульсного отклика. При работе с источниками внешних сигналов это поле не активно.

16. **Поле ввода параметра x^2 адаптивного фильтра.** Поле активно в LS, RLS и быстрых RLS алгоритмах с динамической регуляризацией. Параметр x^2 является параметром динамической регуляризации обращения корреляционной матрицы.

17. **Поле ввода амплитуд входных сигналов $x_m(k)$ адаптивного фильтра.** Это поле должно содержать M значений амплитуд, разделенных пробелами. Это амплитуды внутренних источников сигналов. При работе с источниками внешних сигналов данное поле не активно.

18. **Поле ввода коэффициентов стабилизации K_1, \dots, K_5 адаптивного фильтра.** Поле активно в быстрых RLS алгоритмах со стабилизацией.

19. **Поле ввода дисперсии S_z^2 шума на входе $d(k)$ адаптивного фильтра.** При работе с источниками внешних сигналов данное поле не активно, но в нем отображается значение дисперсии шума, оцененное на основе данных wav-файла с этим шумом.

20. **Поле ввода длины скользящего окна L адаптивного фильтра.** Поле активно в LS, RLS и быстрых RLS алгоритмах со скользящим окном, или в AP и FAP алгоритмах. В случае последних двух алгоритмов данный параметр является размером проекции.

21. **Поле ввода частоты дискретизации F_s обрабатываемых сигналов.** При работе с источниками внешних сигналов данное поле не активно, но в нем отображается значение частоты дискретизации, считанное из параметров wav-файлов. Все сигналы должны быть записаны в файлы с одной и той же частотой дискретизации. Частота дискретизации вводится или отображается в Гц.

22. **Поле ввода числа J линейных ограничений.** Поля активно при выборе алгоритмов с линейными ограничениями.

23. **Поле ввода длины B скользящего окна для вычисления $ERLE(k)$** (параметр Echo Return Loss Enhancement).

24. **Поле ввода частот F_j , $j = 1: J$, на которых задаются линейные ограничения.** Поля активно при выборе алгоритмов с линейными ограничениями.

25. **Поле ввода числа итераций K моделирования алгоритма адаптивной фильтрации.**

26. **Поле ввода ограниченных значений АЧХ f_j , $j = 1: J$.** Поля активно при выборе алгоритмов с линейными ограничениями.

27. **Поле графика с результатом моделирования.** Появляется по окончании моделирования после нажатия любой из кнопок 40, 42-53. При последующих нажатиях этих кнопок изображения в данном поле сменяют друг друга.

28. **Кнопка «Save» (Сохранить).** Данная кнопка появляется по окончании моделирования после нажатия любой из кнопок 40, 42-53. Нажатие на кнопку «Save» приводит к сохранению результатов моделирования в виде wav-файлов. Файлы сохраняются в рабочей директории. Примеры форматов файлов даны ниже. Файл x1_17_Dec_2003_14_57.wav это файл сигнала $x_1(k)$, образованный 17 декабря 2003 года в 14:57. Для M -канального адаптивного фильтра число таких файлов равно M . Файлы других сигналов отличаются только в самом начале имени. Так d_ означает файл сигнала $d(k)$, dn_ означает файл сигнала $d(k) + z(k)$, e_ означает файл сигнала ошибки на выходе адаптивного фильтра $e(k)$, en_ означает файл сигнала $e(k) + z(k)$, n_ означает файл шума $z(k)$ на входе $d(k)$, y_ файл выходного сигнала $y(k)$ адаптивного фильтра.

29. **Кнопка «Get picture» (Получить рисунок).** Нажатие на эту кнопку приводит к выводу в отдельном окне (вне ГИП) того же графика, что и в ГИП, см. рис. 13.

30. **Кнопка «Grid off» (Выключить сетку).** Нажатие на эту кнопку приводит к выключению или включению сетки на графике в центре ГИП. В исходном состоянии сетка включенна. При каждом из нажатий на кнопку, происходит последовательное включение или выключение сетки, а на самой кнопке последовательно меняется надпись «Grid on», «Grid off» (Сетку включить, Сетку выключить).

31. **Кнопка «Y fix» (Изменение масштаба по оси Y).** Нажатие на эту кнопку приводит к последовательному изменению масштаба по оси Y от фиксированного до автоматического. В исходном состоянии задан фиксированный масштаб. Фиксированный

масштаб это увеличение границ по оси Y на 10% относительно максимального и минимального значений переменной, выводимой на график. При выводе однородных переменных минимальные и максимальные значения выбираются по всем переменным. Однородными переменными являются все входные сигналы адаптивного фильтра; импульсные отклики всех каналов адаптивного фильтра; АЧХ всех каналов; ФЧХ всех каналов; групповое время задержки всех каналов; а также сигналы $d(k), d(k) + z(k), y(k), e(k), z(k), e(k) + z(k)$. Отображение однородных переменных в едином масштабе позволяет визуально судить об их относительных значениях. Более детальное рассмотрение переменной можно осуществить при переходе к автоматическому масштабу. При каждом из нажатий на кнопку, происходит последовательный переход от фиксированного к автоматическому масштабу, сопровождаемый соответствующей деформацией текущего графика и изменением надписи на самой кнопке: « Y fix» и « Y auto» (фиксированный и автоматический масштабы по оси Y).

32. Кнопка « X auto» (Изменение масштаба по оси X). Нажатие на эту кнопку приводит к изменению масштаба по оси X от фиксированного, ограниченного значениями, вводимыми в окнах 33 и 35, к автоматическому (все значения аргумента). Фиксированный масштаб позволяет рассмотреть графики более детально. Исходное состояние этой кнопки соответствует автоматическому значению масштаба. Тип масштаба отображается в виде надписи на данной кнопке как « X lim» и « X auto» (фиксированный и автоматический масштабы по оси X).

33. Поле ввода максимального значения координатной оси X . При выводе графика после нажатия кнопок 40, 42-53 в этом поле отображается максимальное значение координаты X , определяемое длиной вектора выводимых данных. Затем это значение может быть изменено на любое другое в пределах максимально допустимого значения. Изменение масштаба по оси X происходит сразу же после изменения в данном поле. При этом на кнопке 32 появляется надпись « X lim». Если введенное значение превышает допустимое значение, поле засвечивается желтым цветом и в нем красным цветом высвечивается предупреждающее сообщение. Затем значение данного поля устанавливается равным максимально допустимому (определяется отображаемым графиком).

34. **Поле ввода минимального значения координатной оси X.** Действия в этом поле аналогичны действиям в поле 33. Отличия заключаются лишь в том, что данное поле относится к минимальному значению координаты X.

35. **Кнопка «Close» (заккрыть).** Нажатие этой кнопки приводит к выключению ГИП.

36. **Кнопка «Continue» (продолжить).** Нажатие этой кнопки приводит к началу процесса моделирования алгоритма адаптивной фильтрации.

37. **Кнопка «Start» (начало).** Нажатие этой кнопки приводит к активизации ГИП. После первого же нажатия кнопки «Continue», на данном месте возникает поле, в котором по окончании моделирования отображается время, затраченное на моделирование (в секундах).

38. **Выпадающее меню выбора номера канала адаптивного фильтра.** Данное меню служит для выбора номера канала многоканального адаптивного фильтра при отображении входного сигнала, импульсного отклика, амплитудно-частотной характеристики (АЧХ), фазо-частотной характеристики (ФЧХ) или группового времени задержки.

39. **Кнопка «Real» определяет действительную или комплексную части отображаемого графика.** Исходное состояние этой кнопки соответствует выводу действительно части комплексных переменных. Каждое нажатие этой кнопки приводит к последовательному изменению ее состояния, сопровождаемого изменением надписи на самой кнопке как «Real», «Imag» (действительная, мнимая). В соответствии с состоянием кнопки, при нажатии кнопок 40, 42, 47-52 будет выводиться график действительной или мнимой части комплексной переменной.

40. **Кнопка «x».** Нажатие этой кнопки приводит к выводу графика входного сигнала адаптивного фильтра $x_m(k)$, рис. 16.

41. **Кнопка «Channel # 1».** На этой кнопке отображается номер канала адаптивного фильтра, выбранного с помощью поп-меню 38. Нажатие на эту кнопку ведет к появлению на ней надписи «Channel # M», а последующее нажатие - к появлению исходной надписи «Channel # 1». Надпись «Channel # M означает, что при нажатии кнопок 42-46 будут выводиться графики, соответствующие не отдельным каналам многоканального фильтра, а соответствующие многоканальному фильтру.

42. **Кнопка «h, W».** Нажатие этой кнопки приводит к выводу графика импульсного отклика адаптивного фильтра $\mathbf{h}_{N_m}(k)$ или $\mathbf{h}_N(k)$ (синим цветом), а также графика идентифицируемого импульсного отклика (красным цветом), рис. 17.

43. **Кнопка «|h - W|».** Нажатие этой кнопки приводит к выводу графика рассогласования $mismatch(k)$, рис. 18.

44. **Кнопка «H(f)».** Нажатие этой кнопки приводит к выводу графика АЧХ, рис. 19.

45. **Кнопка «P(f)».** Нажатие этой кнопки приводит к выводу графика ФЧХ, рис. 20.

46. **Кнопка «GD(f)».** Нажатие этой кнопки приводит к выводу графика группового времени задержки, рис. 21.

47. **Кнопка «d».** Нажатие этой кнопки приводит к выводу графика сигнала $d(k)$, рис. 22.

48. **Кнопка «n».** Нажатие этой кнопки приводит к выводу графика сигнала шума $z(k)$ на входе $d(k)$, рис. 23

49. **Кнопка «d+n».** Нажатие этой кнопки приводит к выводу графика сигнала $d(k) + z(k)$ (зашумленного сигнала на входе $d(k)$), рис. 24.

50. **Кнопка «y».** Нажатие этой кнопки приводит к выводу графика сигнала выходного сигнала $y(k)$ адаптивного фильтра, рис. 25.

51. **Кнопка «e+n».** Нажатие этой кнопки приводит к выводу графика зашумленного сигнала $e(k) + z(k)$ ошибки адаптивного фильтра, рис. 26.

52. **Кнопка «e».** Нажатие этой кнопки приводит к выводу графика сигнала $e(k)$ не зашумленного сигнала ошибки адаптивного фильтра, рис. 27.

53. **Кнопка «ERLE».** Нажатие этой кнопки приводит к выводу графика сигнала $ERLE(k)$, рис. 28.

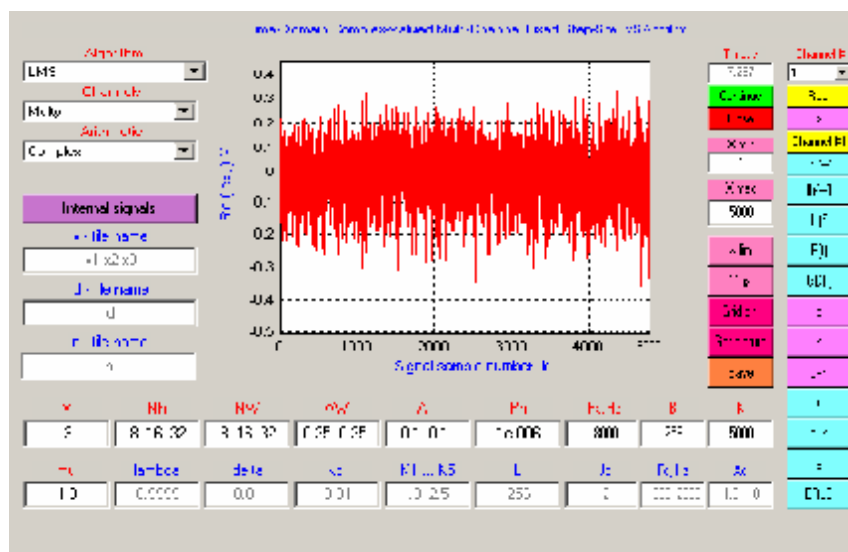


Рис. 16. ГИП после нажатия кнопки «x»

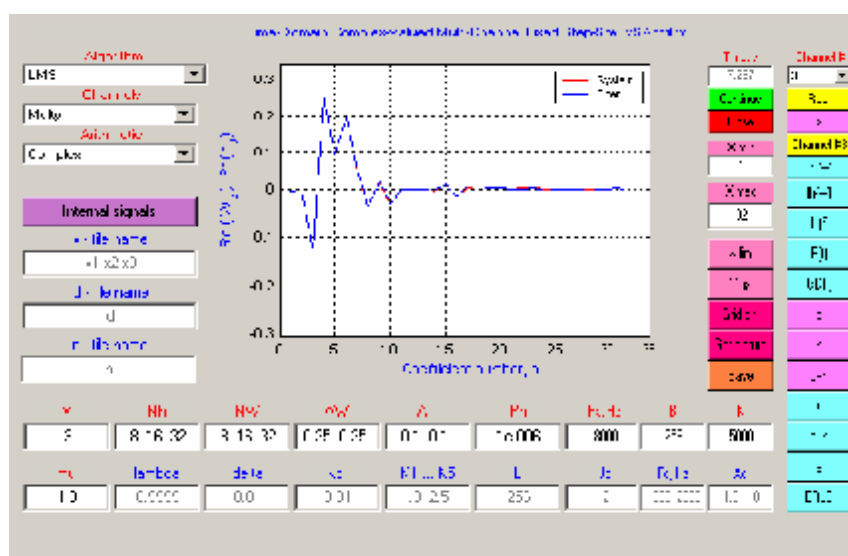


Рис. 17. ГИП после нажатия кнопки «h, W»

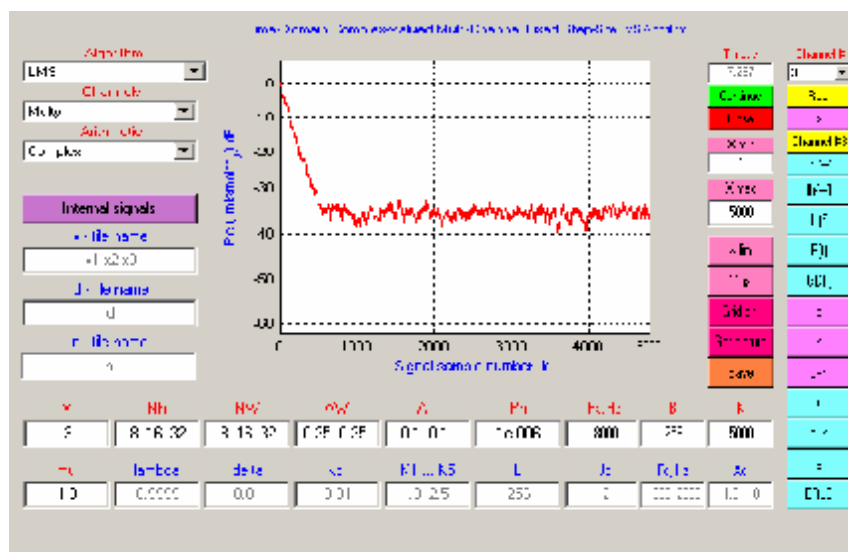


Рис. 18. ГИП после нажатия кнопки «| h - W |»

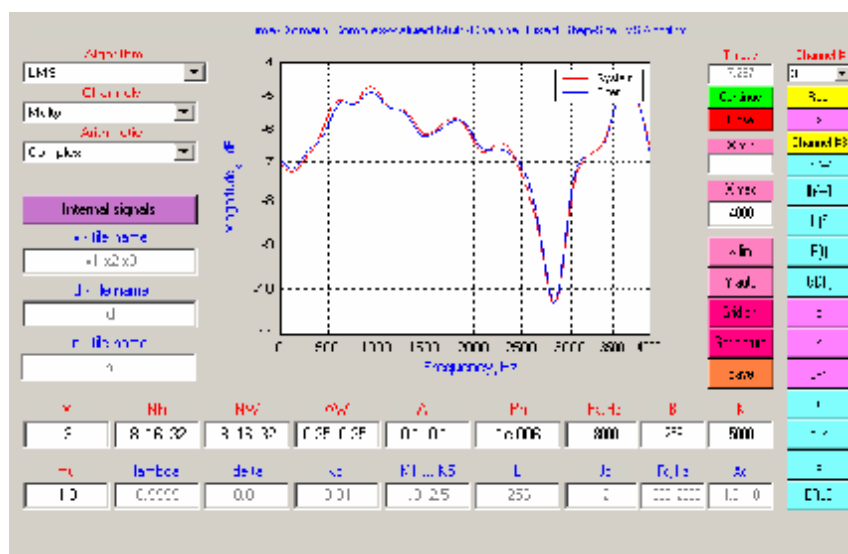


Рис. 19. ГИП после нажатия кнопки «H(f)»

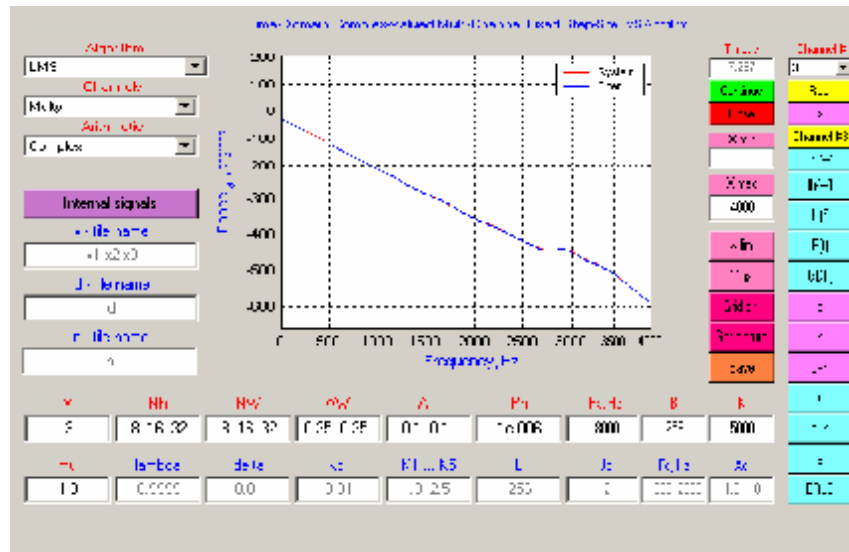


Рис. 20. ГИП после нажатия кнопки «P(f)»

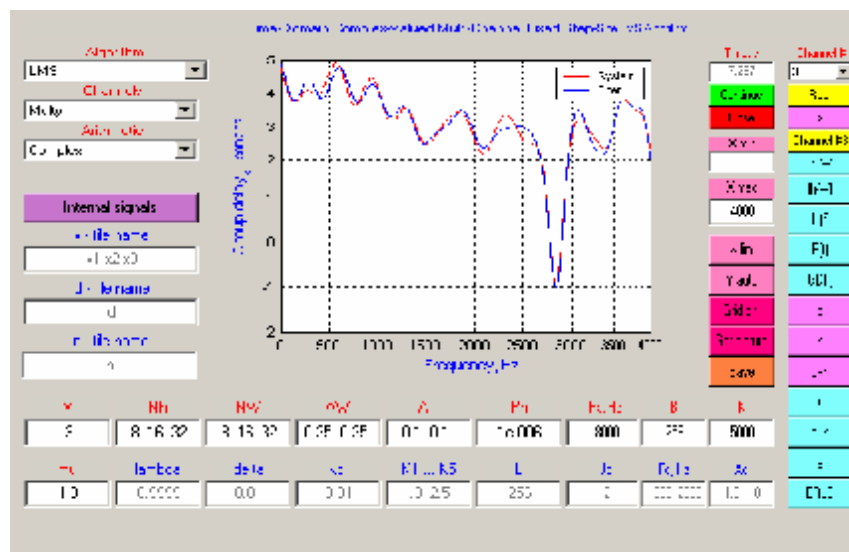


Рис. 21. ГИП после нажатия кнопки «GD(f)»

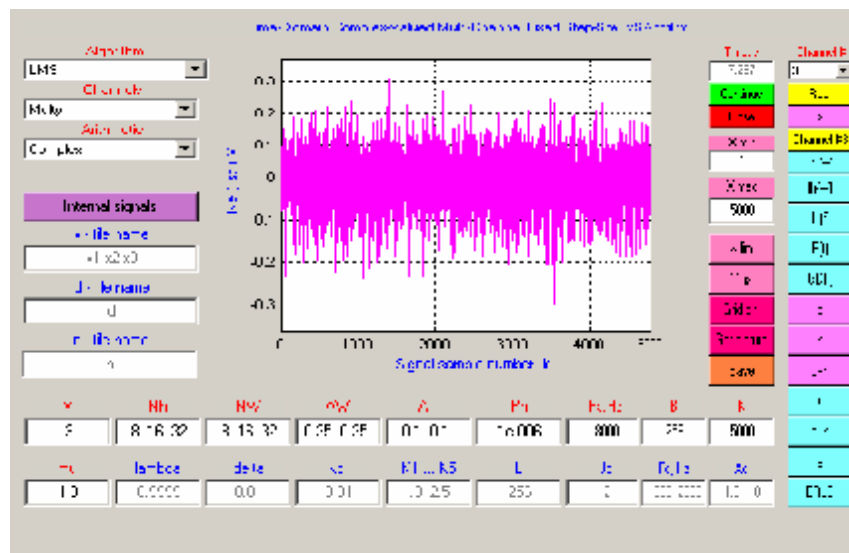


Рис. 22. ГИП после нажатия кнопки «d»

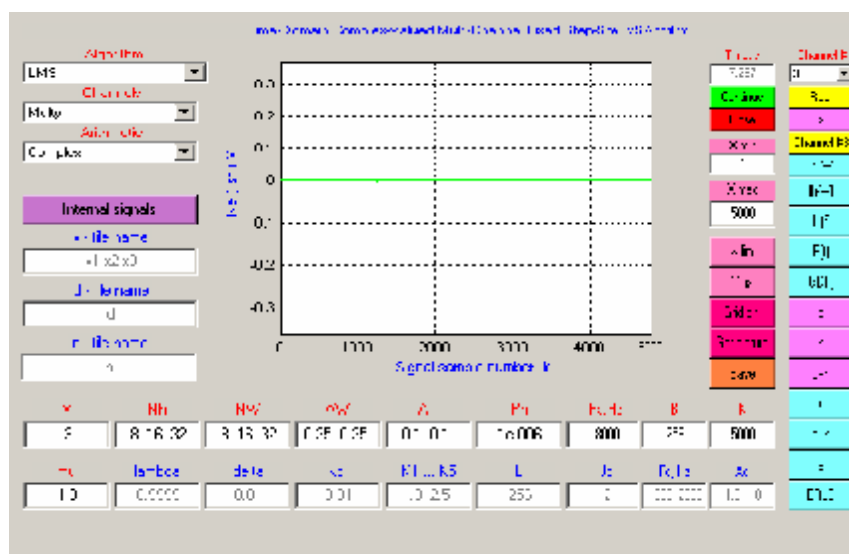


Рис. 23. ГИП после нажатия кнопки «n».

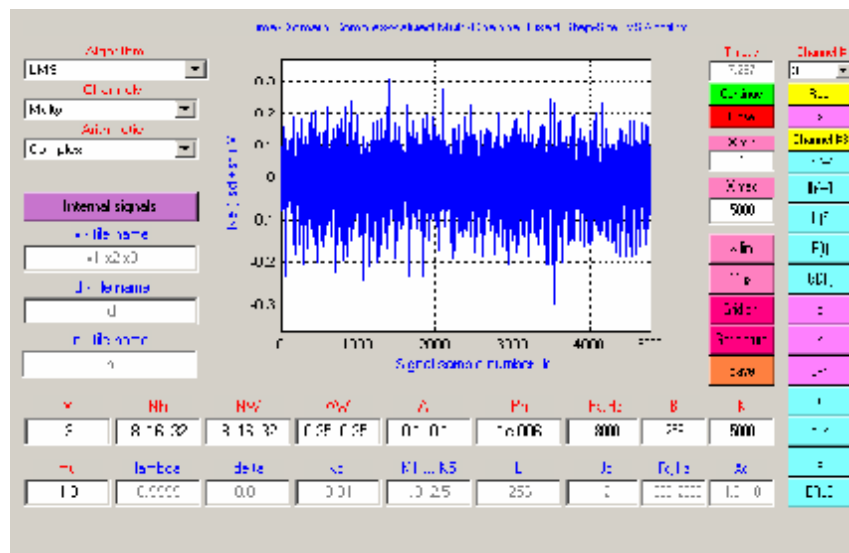


Рис. 24. ГИП после нажатия кнопки «d+n»

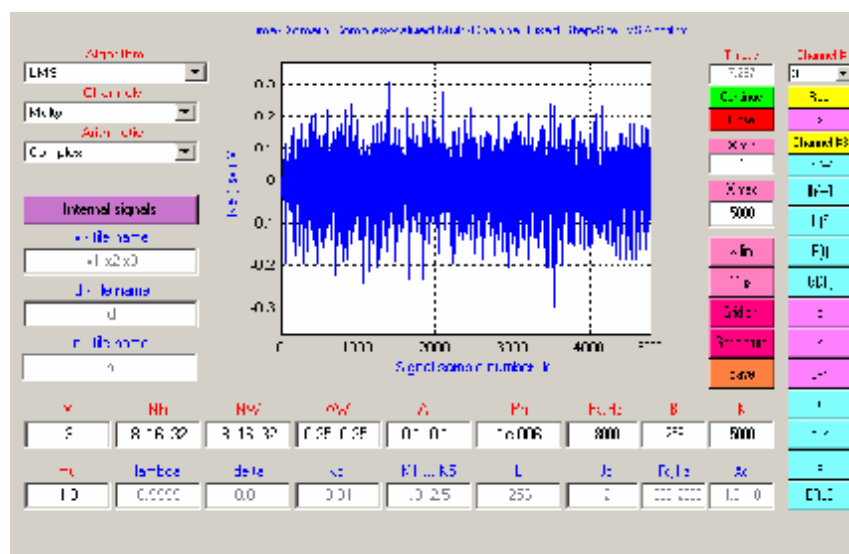


Рис. 25. ГИП после нажатия кнопки «у»

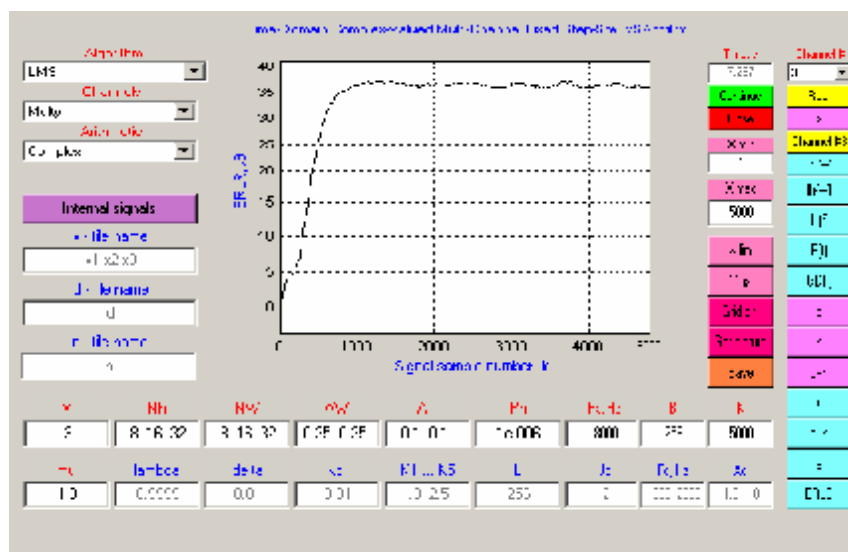


Рис. 28. ГИП после нажатия кнопки «ERLE».

6. ПЕРЕЧЕНЬ ФУНКЦИЙ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Поставляемые с данным документом программные модули ГИП расположены в папке GUI_P_M65. Это модули: adaptive_filters_gui.p (основная исполняемая программа), adaptive_filters_gui.fig (изображение ГИП), algorithm_name_pm.p, algorithm_titles.p, algorithms.p, Am_valuer.p, arithmetic_name_pm.p, AW_valuer.p, B_valuer.p, button_channel.p, button_d.p, button_dn.p, button_e.p, button_e_n.p, button_erle.p, button_fr.p, button_fs.p, button_GD.p, button_grid.p, button_inp_signal.p, button_ir.p, button_mm.p, button_n.p, button_np.p, button_pr.p, button_rc.p, button_sig_type.p, button_x_axes.p, button_y.p, button_y_axes.p, channel_num_pm.p, closer.p, constr_calc.p, continuer.p, data_generations.p, delta_valuer.p, dwav_valuer.p, erle_init.p, erle_mismatch_calc.p, Fc_valuer.p, Fs_valuer.p, impulses.p, Jc_valuer.p, K_valuer.p, K15_valuer.p, kci_valuer.p, L_valuer.p, lambda_valuer.p, lms_t_c_m_fss.p, lms_t_c_m_fss_LC1.p, lms_t_c_m_fss_LC2.p, ls_t_c_m_pw_ew_nr.p, ls_t_c_m_pw_ew_nr_LC0.p, M_valuer.p, mu_valuer.p, N_valuer.p, nlms_t_c_m_fss.p, nlms_t_c_m_fss_LC1.p, nlms_t_c_m_fss_LC2.p, NW_valuer.p, nwav_valuer.p, pm_channel_n_number.p, Pn_valuer.p, rls_t_c_m_pw_ew_nr.p, rls_t_c_m_pw_ew_nr_LC0.p, rls_t_c_m_pw_ew_nr_LC1.p, rls_t_c_m_pw_ew_nr_LC2.p, starter.p, string_div.p, string_valuer.p, wav_valuer.p, welcome.p, x_max_valuer.p, x_min_valuer.p, xwav_valuer.p.

С указанными файлами также поставляется ряд звуковых файлов формата wav, записанных с частотой дискретизации 8 кГц. Шумоподобные (комплексные) сигналы длиной в 500000 отсчетов каждый размещены в папке wav_50k_noise_8k, речевые (действительные) сигналы длиной в 500000 отсчетов каждый размещены в папке wav_50k_speech_8k, шумоподобные (комплексные) сигналы длиной в 5000 отсчетов каждый размещены в папке wav_5k_noise_8k. В каждой из папок находятся файлы сигналов d.wav - $d(k)$; n0.wav - $z(k)$ при $S_z = 0$; n1.wav - $z(k)$ при $S_z = 10^{-1}$; n2.wav - $z(k)$ при $S_z = 10^{-2}$; n3.wav - $z(k)$ при $S_z = 10^{-3}$, n4.wav - $z(k)$ при $S_z = 10^{-4}$; n5.wav - $z(k)$ при $S_z = 10^{-5}$; n6.wav - $z(k)$ при $S_z = 10^{-6}$; x1.wav - $x_1(k)$; x2.wav - $x_2(k)$, x3.wav - $x_3(k)$.

Для использования этих сигналов, файлы из соответствующей папки необходимо скопировать в рабочую директорию, а поле ввода имени файла шума $z(k)$ на входе $d(k)$ адаптивного фильтра «n-file name» скорректировать имя шумового файла как n0, n1, ..., n6, в зависимости от желаемого значения S_z .

В случае использования других звуковых файлов, их имена могут быть произвольными. Если эти файлы получены путем измерения реальных процессов, то сигнал $d(k)$ в таком случае обычно является зашумленным. Однако, для работы ГИП необходим «n-file», поэтому нужно создать такой wav файл, содержащий нули. Примером такого файла служит файл n0.wav.

Поставляемые файлы d.wav получены путем сигналов свертки трех импульсных откликов $N_m^W = [8,16,32]$, вырабатываемых ГИП, с сигналами $x_1(k)$, $x_2(k)$ и $x_3(k)$. Поэтому при использовании этих сигналов (в качестве демонстрации работы ГИП с внешними сигналами), необходимо использовать $M = 3$ и $N_m = [8,16,32]$. В остальных же случаях, M и N_m должны устанавливаться в соответствие с физической природой решаемой задачи.

При использовании файлов из папки wav_50k_speech_8k для моделирования необходимо выбрать действительную арифметику (в поле «Arithmetic»)

Примечание: в функциях ГИП, обрабатывающих ввод числовых параметров, не предусмотрена полная проверка и защита от некорректного ввода значений (в нижней части ГИП). Поэтому для правильной работы ГИП, эти параметры необходимо вводить или изменять в форматах, представление о которых содержится в параметрах, установленных по умолчанию.

7. ОЦЕНОЧНЫЕ РЕЗУЛЬТАТЫ РЕАЛИЗАЦИИ АЛГОРИТМОВ НА ЯЗЫКЕ ASSEMBLER

Некоторые из алгоритмов адаптивной фильтрации в настоящее время реализованы на языке Assembler ИМС MC12 и MC24. На рис. 29 представлена зависимость требуемых вычислительных ресурсов (clk - числа тактов) ИМС MC12 и MC24 для выполнения одной итерации четырех разновидностей LMS алгоритмов от числа весовых коэффициентов N адаптивного фильтра. Примерно такое же число тактов требуется и для NLMS алгоритмов. Вычислительная сложность этих простейших алгоритмов примерно равна $2N$ арифметических операций (сложений + умножений, действительных в фильтрах с действительными весовыми коэффициентами или комплексных в фильтрах с комплексными весовыми коэффициентами) при выполнении одной итерации. При заданной тактовой частоте MC12 и MC24 с помощью рис. 29 можно определить максимально достижимую тактовую частоту обработки сигналов с помощью LMS адаптивного фильтра.

Рис. 29 позволяет также оценить требуемые вычислительные ресурсы при реализации других алгоритмов адаптивной фильтрации.

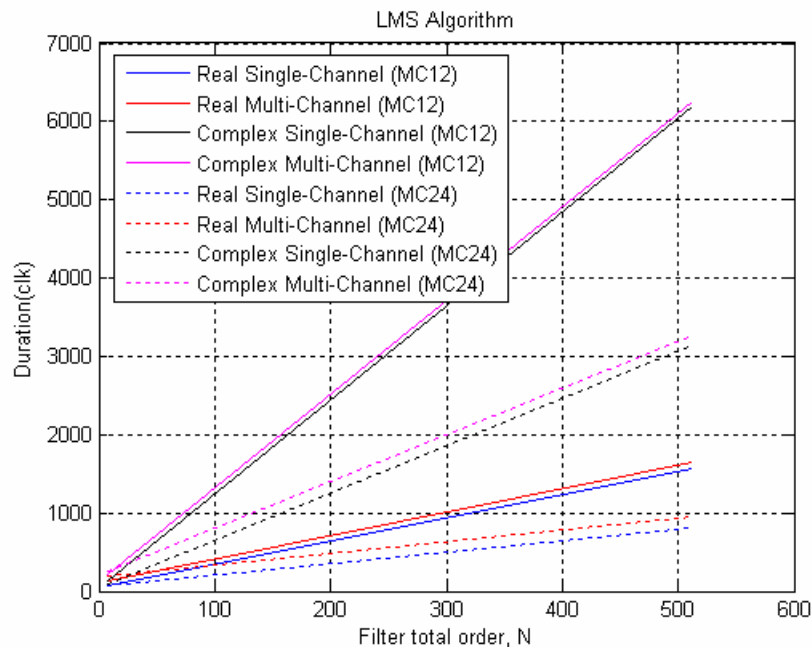


Рис. 29. Требуемые вычислительные ресурсы для реализации LMS алгоритмов адаптивной фильтрации на ИМС MC12 и MC24.

ЗАКЛЮЧЕНИЕ

Таким образом, в настоящем документе представлены различные аспекты прикладной библиотеки алгоритмов и программ ГУП НПЦ «ЭЛВИС» для ИМС МС12 и МС24 в части библиотеки алгоритмов адаптивной фильтрации. Данная библиотека постоянно развивается за счет разработки новых алгоритмов адаптивной фильтрации, а также перевода этих алгоритмов на языки *Assembler* и *C* ИМС МС12 и МС24.