A-Appendix

A-Workshop1A

```
%% A.1
% RLS without noise

clear all; close all; clc
load('data1.mat');

% Constants
Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

% RLS with inital params = 0

init_params = [1; 0; 0];
P_init_scale = 1;

% ---------------------------------- %
%     RLS: Different initial conditions.
% ---------------------------------- %

% RLS with initial params = 0

[output, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);

figure(1)
subplot(2,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (RLS no noise) b_2=b_3 = 0')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])


% RLS with inital params = 1

init_params = [1; 1; 1];
P_init_scale = 1;

[output, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);


subplot(2,1,2)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (RLS no noise) b_2=b_3 = 1')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

saveas(gcf, 'figures/q1a_params.png')
```

```
63
64  %%
65  % ---------------------------------- %
66  %     RLS: P scaling
67  % ---------------------------------- %
68  clear all
69  close all
70  clc
71  load('data1.mat');
72
73  Fs = 8192; %sampling freq
74  D1 = 1 * Fs; % delay 1
75  D2 = 2.5 * Fs; % delay 2
76  timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis
77
78
79  % RLS with P_init_scale = 1
80  init_params = [1; 0; 0];
81  P_init_scale = 1;
82
83  [output1, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);
84
85  figure(2)
86  subplot(3,1,1)
87  hold on
88  plot(timespan,theta_hat(1,:))
89  plot(timespan,theta_hat(2,:))
90  plot(timespan, theta_hat(3,:))
91  hold off
92
93  title('Parameter values (RLS no noise) Init P scale = 1')
94  xlabel('Seconds')
95  ylabel('Parameter value')
96  legend('b_1', 'b_2', 'b_3')
97  ylim([0 1.01])
98
99
100 % RLS with P_init_scale = 0.01
101 init_params = [1; 0; 0];
102 P_init_scale = 0.01;
103
104 [output001, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);
105
106 figure(2)
107 subplot(3,1,2)
108 hold on
109 plot(timespan,theta_hat(1,:))
110 plot(timespan,theta_hat(2,:))
111 plot(timespan, theta_hat(3,:))
112 hold off
113
114 title('Parameter values (RLS no noise) Init P scale = 0.01')
115 xlabel('Seconds')
116 ylabel('Parameter value')
117 legend('b_1', 'b_2', 'b_3')
118 ylim([0 1.01])
119
120 % RLS with P_init_scale = 100
121 init_params = [1; 0; 0];
122 P_init_scale = 100;
123
124 [output100, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);
125
126 figure(2)
127 subplot(3,1,3)
128 hold on
129 plot(timespan,theta_hat(1,:))
130 plot(timespan,theta_hat(2,:))
```

```
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (RLS no noise) Init P scale = 100')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

saveas(gcf, 'figures/q1a_Pinit.png')

figure(6)
subplot(3,1,1)
plot(timespan, transpose(loudspeaker) - output1)
title('Amplitude Difference (RLS without noise) Pinit = 1*I')
xlabel('Seconds')
ylabel('Amplitude')

subplot(3,1,2)
plot(timespan, transpose(loudspeaker) - output001)
title('Amplitude Difference (RLS without noise) Pinit = 0.01*I')
xlabel('Seconds')
ylabel('Amplitude')

subplot(3,1,3)
plot(timespan, transpose(loudspeaker) - output100)
title('Amplitude Difference (RLS without noise) Pinit = 100*I')
xlabel('Seconds')
ylabel('Amplitude')

saveas(gcf, 'figures/q1a_Pinit_output_diff.png')

%%
% ---------------------------------- %
%     RLS: With Noise
% ---------------------------------- %

clear all
close all
clc
load('data1.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

init_params = [1; 0; 0];
P_init_scale = 1;

[output, theta_hat] = RLS_function(loudspeaker, mike1, init_params, P_init_scale);
[noisy_output, noisy_theta_hat] = RLS_function(loudspeaker, noisymike1, init_params, ...
    P_init_scale);

figure(4)
subplot(2,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (RLS no noise)')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

```

```
subplot(2,1,2)
hold on
plot(timespan,noisy_theta_hat(1,:))
plot(timespan,noisy_theta_hat(2,:))
plot(timespan, noisy_theta_hat(3,:))
hold off

title('Parameter values (RLS with noise)')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

saveas(gcf, 'figures/q1a_noise_params.png')

% ---------------------------------- %
%     RLS: With noise, comparing output
% ---------------------------------- %

figure(5)
subplot(2,1,1)
plot(timespan, transpose(loudspeaker) - output)
title('Amplitude Difference between Loudspeaker and output signal(RLS without noise)')
xlabel('Seconds')
ylabel('Amplitude')
ylim([-0.1 0.15])
subplot(2,1,2)
plot(timespan, transpose(loudspeaker) - noisy_output)
title('Amplitude Difference between Loudspeaker and output signal(RLS with noise)')
xlabel('Seconds')
ylabel('Amplitude')
ylim([-0.1 0.15])

saveas(gcf, 'figures/q1a_noise_diff.png')



%%
% ---------------------------------- %
%     LMS: No Noise
% ---------------------------------- %

clear all
clc
close all
load('data1.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

% LMS no noise with mu = 3 b2=b3 = 0
init_params = [1; 0; 0];
mu = 0.3;
[output, theta_hat] = LMS_function(loudspeaker, mike1, init_params, mu);

% LMS no noise with mu = 3 b2=b3 = 1
init_params = [1; 1; 1];
[output2, theta_hat2] = LMS_function(loudspeaker, mike1, init_params, mu);


figure(2)
subplot(2,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
```

```
hold off

title('Parameter values (LMS no noise) mu = 0.3; b_2 = b_3 = 1')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

subplot(2,1,2)
hold on
plot(timespan,theta_hat2(1,:))
plot(timespan,theta_hat2(2,:))
plot(timespan, theta_hat2(3,:))
hold off

title('Parameter values (LMS no noise) mu = 0.3; b_2 = b_3 = 1')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

saveas(gcf,'figures/q1a_lms_params.png')


%%
% ---------------------------------- %
%     LMS:Changing Mu
% ---------------------------------- %

clear all
clc
close all
load('data1.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

% LMS no noise with mu = 3 b2=b3 = 0
init_params = [1; 0; 0];
mu = 0.3;
[output, theta_hat] = LMS_function(loudspeaker, mike1, init_params, mu);

% LMS no noise with mu = 0.03 b2=b3 = 0
mu = 0.03;
[output2, theta_hat2] = LMS_function(loudspeaker, mike1, init_params, mu);

% LMS no noise with mu = 3 b2=b3 = 0
mu = 3;
[output3, theta_hat3] = LMS_function(loudspeaker, mike1, init_params, mu);

figure(2)
subplot(3,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (LMS no noise) mu = 0.3')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

subplot(3,1,2)
hold on
```

```
plot(timespan,theta_hat2(1,:))
plot(timespan,theta_hat2(2,:))
plot(timespan, theta_hat2(3,:))
hold off

title('Parameter values (LMS no noise) mu = 0.03')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

subplot(3,1,3)
hold on
plot(timespan,theta_hat3(1,:))
plot(timespan,theta_hat3(2,:))
plot(timespan, theta_hat3(3,:))
hold off

title('Parameter values (LMS no noise) mu = 3')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

saveas(gcf,'figures/q1a_lms_mu.png')


%% A.2
% ---------------------------------- %
%     LMS with noise
% ---------------------------------- %

clear all
close all
clc
load('data1.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

init_params = [1; 0; 0];
mu = 0.3;

[output, theta_hat] = LMS_function(loudspeaker, mike1, init_params, mu);
[noisy_output, noisy_theta_hat] = LMS_function(loudspeaker, noisymike1, init_params, mu);

figure(4)
subplot(2,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Parameter values (LMS no noise)')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

subplot(2,1,2)
hold on
plot(timespan,noisy_theta_hat(1,:))
plot(timespan,noisy_theta_hat(2,:))
plot(timespan, noisy_theta_hat(3,:))
hold off
```

```
402
403 title('Parameter values (LMS with noise)')
404 xlabel('Seconds')
405 ylabel('Parameter value')
406 legend('b_1', 'b_2', 'b_3')
407 ylim([0 1])
408
409 saveas(gcf, 'figures/q1a_lms_noise_params.png')
410
411 % ---------------------------------- %
412 %     LMS: Noise, output graphs
413 % ---------------------------------- %
414
415 figure(5)
416 subplot(2,1,1)
417 plot(timespan, transpose(loudspeaker) - output)
418 title('Amplitude Difference between Loudspeaker and output signal(LMS without noise)')
419 xlabel('Seconds')
420 ylabel('Amplitude')
421 ylim([-0.1 0.15])
422 subplot(2,1,2)
423 plot(timespan, transpose(loudspeaker) - noisy_output)
424 title('Amplitude Difference between Loudspeaker and output signal(LMS with noise)')
425 xlabel('Seconds')
426 ylabel('Amplitude')
427 ylim([-0.1 0.15])
428
429 saveas(gcf, 'figures/q1a_lms_noise_diff.png')
```

A-Workshop1B

```
1  %% B.1
2  % ---------------------------------- %
3  %     RLS: Time Varying. No noise
4  % ---------------------------------- %
5
6  clear all
7  close all
8  clc
9  load('data1.mat');
10 load('data2.mat');
11
12 Fs = 8192; %sampling freq
13 D1 = 1 * Fs; % delay 1
14 D2 = 2.5 * Fs; % delay 2
15 timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis
16
17
18 % RLS with inital params = 0
19
20 init_params = [1; 0; 0];
21 P_init_scale = 1;
22 lambda = 0.995;
23
24
25 [output, theta_hat] = RLS_function2(loudspeaker, mike2, init_params, P_init_scale, lambda);
26
27 figure(1)
28 hold on
29 plot(timespan,theta_hat(1,:))
30 plot(timespan,theta_hat(2,:))
31 plot(timespan, theta_hat(3,:))
32 hold off
33
34 title('Time varying Parameter values (RLS no noise)')
```

```
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

saveas(gcf, 'figures/q1b_params.png')

%%
% ---------------------------------- %
%     RLS: Time Varying.  Lambda changing
% ---------------------------------- %
clear all
close all
clc
load('data1.mat');
load('data2.mat')

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis


% RLS with P_init_scale = 1
init_params = [1; 0; 0];
P_init_scale = 1;
lambda = 0.999;

[output1, theta_hat] = RLS_function2(loudspeaker, mike2, init_params, P_init_scale, lambda);

figure(2)
subplot(3,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Time varying Parameter values (RLS no noise) Lambda = 0.999')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])


% RLS with lambda = 0.990;
lambda = 0.990;

[output001, theta_hat] = RLS_function2(loudspeaker, mike2, init_params, P_init_scale, lambda);

figure(2)
subplot(3,1,2)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Time varying Parameter values (RLS no noise) lambda = 0.990;')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

% RLS with lambda = 0.999;
init_params = [1; 0; 0];
P_init_scale = 100;
lambda = 0.995;
```

```
103
104 [output100, theta_hat] = RLS_function2(loudspeaker, mike2, init_params, P_init_scale, lambda);
105
106 figure(2)
107 subplot(3,1,3)
108 hold on
109 plot(timespan,theta_hat(1,:))
110 plot(timespan,theta_hat(2,:))
111 plot(timespan, theta_hat(3,:))
112 hold off
113
114 title('Time varying Parameter values (RLS no noise) lambda = 0.995')
115 xlabel('Seconds')
116 ylabel('Parameter value')
117 legend('b_1', 'b_2', 'b_3')
118 ylim([0 1.01])
119
120 saveas(gcf, 'figures/q1b_Pinit.png')
121
122 figure(6)
123 subplot(3,1,1)
124 plot(timespan, transpose(loudspeaker) - output1)
125 title('Amplitude Difference for Time Varying Echo Amplitude (RLS without noise) lambda = ...
        0.999;')
126 xlabel('Seconds')
127 ylabel('Amplitude')
128
129 subplot(3,1,2)
130 plot(timespan, transpose(loudspeaker) - output001)
131 title('Amplitude Difference for Time Varying Echo Amplitudes(RLS without noise) lambda = ...
        0.990;')
132 xlabel('Seconds')
133 ylabel('Amplitude')
134
135 subplot(3,1,3)
136 plot(timespan, transpose(loudspeaker) - output100)
137 title('Amplitude Difference for Time Varying Echo Amplitudes (RLS without noise) lambda = ...
        0.995;')
138 xlabel('Seconds')
139 ylabel('Amplitude')
140
141 saveas(gcf, 'figures/q1b_Pinit_output_diff.png')
142
143 %% B.1
144 % ---------------------------------- %
145 %     RLS: Time Varying with noise
146 % ---------------------------------- %
147
148 clear all
149 close all
150 clc
151 load('data1.mat');
152 load('data2.mat')
153
154 Fs = 8192; %sampling freq
155 D1 = 1 * Fs; % delay 1
156 D2 = 2.5 * Fs; % delay 2
157 timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis
158
159
160 init_params = [1; 0; 0];
161 P_init_scale = 1;
162 lambda = 0.995;
163
164 [output, theta_hat] = RLS_function2(loudspeaker, mike2, init_params, P_init_scale, lambda);
165 [noisy_output, noisy_theta_hat] = RLS_function2(loudspeaker, noisymike2, init_params, ...
        P_init_scale, lambda);
166
```

```
% sound(noisy_output)


figure(4)
subplot(2,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Time Varying Parameter values (RLS no noise)')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

subplot(2,1,2)
hold on
plot(timespan,noisy_theta_hat(1,:))
plot(timespan,noisy_theta_hat(2,:))
plot(timespan, noisy_theta_hat(3,:))
hold off

title('Time varying Parameter values (RLS with noise)')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1])

figure(5)
subplot(2,1,1)
plot(timespan, transpose(loudspeaker) - output)
title('Amplitude Difference between Loudspeaker and output signal for Time Varying Echo ...
    Amplitudes(RLS without noise)')
xlabel('Seconds')
ylabel('Amplitude')
ylim([-0.1 0.15])
subplot(2,1,2)
plot(timespan, transpose(loudspeaker) - noisy_output)
title('Amplitude Difference between Loudspeaker and output signalfor Time Varying Echo ...
    Amplitudes(RLS with noise)')
xlabel('Seconds')
ylabel('Amplitude')
ylim([-0.1 0.15])


%%
% ---------------------------------- %
%     LMS: Time Varying
% ---------------------------------- %

% Changing mu

clear all
close all
clc
load('data1.mat');
load('data2.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis

init_params = [1; 0; 0];

% LMS no noise with mu = 3 b2=b3 = 0
```

```
mu = 0.3;
[output, theta_hat] = LMS_function(loudspeaker, mike2, init_params, mu);

% LMS no noise with mu = 0.03 b2=b3 = 0
mu = 0.03;
[output2, theta_hat2] = LMS_function(loudspeaker, mike2, init_params, mu);

% LMS no noise with mu = 3 b2=b3 = 0
mu = 3;
[output3, theta_hat3] = LMS_function(loudspeaker, mike2, init_params, mu);

figure(2)
subplot(3,1,1)
hold on
plot(timespan,theta_hat(1,:))
plot(timespan,theta_hat(2,:))
plot(timespan, theta_hat(3,:))
hold off

title('Time Varying Parameter values (LMS no noise) \mu = 0.3')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

subplot(3,1,2)
hold on
plot(timespan,theta_hat2(1,:))
plot(timespan,theta_hat2(2,:))
plot(timespan, theta_hat2(3,:))
hold off

title('Time Varying Parameter values (LMS no noise) \mu = 0.03')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

subplot(3,1,3)
hold on
plot(timespan,theta_hat3(1,:))
plot(timespan,theta_hat3(2,:))
plot(timespan, theta_hat3(3,:))
hold off

title('Time Varying Parameter values (LMS no noise) \mu = 3')
xlabel('Seconds')
ylabel('Parameter value')
legend('b_1', 'b_2', 'b_3')
ylim([0 1.01])

saveas(gcf,'figures/q1b_lms_mu.png')

%%
% ---------------------------------- %
%     LMS: Time Varying and Noise
% ---------------------------------- %

clear all
close all
clc
load('data1.mat');
load('data2.mat');

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2
timespan = [0: 1/Fs: length(loudspeaker)/Fs - 1/Fs]; % Timespan for time-axis
```

```
301
302 init_params = [1; 0; 0];
303 mu = 0.3;
304
305 [output, theta_hat] = LMS_function(loudspeaker, mike2, init_params, mu);
306 [noisy_output, noisy_theta_hat] = LMS_function(loudspeaker, noisymike2, init_params, mu);
307
308 figure(4)
309 subplot(2,1,1)
310 hold on
311 plot(timespan,theta_hat(1,:))
312 plot(timespan,theta_hat(2,:))
313 plot(timespan, theta_hat(3,:))
314 hold off
315
316 title('Time Varying Parameter values (LMS no noise)')
317 xlabel('Seconds')
318 ylabel('Parameter value')
319 legend('b_1', 'b_2', 'b_3')
320 ylim([0 1])
321
322 subplot(2,1,2)
323 hold on
324 plot(timespan,noisy_theta_hat(1,:))
325 plot(timespan,noisy_theta_hat(2,:))
326 plot(timespan, noisy_theta_hat(3,:))
327 hold off
328
329 title('Time Varying Parameter values (LMS with noise)')
330 xlabel('Seconds')
331 ylabel('Parameter value')
332 legend('b_1', 'b_2', 'b_3')
333 ylim([0 1])
334
335 saveas(gcf, 'figures/q1b_lms_noise_params.png')
336
337 figure(5)
338 subplot(2,1,1)
339 plot(timespan, transpose(loudspeaker) - output)
340 title('Time Varying Params: Amplitude Difference between Loudspeaker and output signal(LMS ...
        without noise)')
341 xlabel('Seconds')
342 ylabel('Amplitude')
343 ylim([-0.1 0.15])
344 subplot(2,1,2)
345 plot(timespan, transpose(loudspeaker) - noisy_output)
346 title('Time Varying Params: Amplitude Difference between Loudspeaker and output signal(LMS ...
        with noise)')
347 xlabel('Seconds')
348 ylabel('Amplitude')
349 ylim([-0.1 0.15])
350
351 saveas(gcf, 'figures/q1b_lms_noise_diff.png')
```

A-RLS Function 1

```
1 function [ output, theta_hat ] = RLS_function(loudspeaker, mike1, init_params, P_init_scale)
2
3 Fs = 8192; %sampling freq
4 D1 = 1 * Fs; % delay 1
5 D2 = 2.5 * Fs; % delay 2
6
7 P = P_init_scale * eye(3);
8
9 theta_hat = zeros(3,length(loudspeaker));
```

```
loudspeakerDelay1 = 0;
loudspeakerDelay2 = 0;

output  = zeros(1,length(loudspeaker));


for i=1:length(loudspeaker)

    if (i > D1)
        loudspeakerDelay1 = loudspeaker(i - D1);
    end

    if (i > D2)
        loudspeakerDelay2 = loudspeaker(i - D2);
    end

    phi = [loudspeaker(i); loudspeakerDelay1;  loudspeakerDelay2];

    P = P - (P*phi*transpose(phi)*P)/(1 + transpose(phi)*P*phi);

    G = P*phi;

    if ( i > 1)
        theta_hat(:,i) = theta_hat(:,i-1) + G*(mike1(i) - transpose(phi)*theta_hat(:,i-1));
    else
        theta_hat(:,i) = init_params + G*(mike1(i) - transpose(phi)*init_params);
    end

    output(i) = mike1(i) - theta_hat(2,i)*loudspeakerDelay1 - ...
        theta_hat(3,i)*loudspeakerDelay2;
end


end
```

A-RLS Function with Lambda

```
function [ output, theta_hat ] = RLS_function2(loudspeaker, mike1, init_params, ...
    P_init_scale, lambda)

% ---------------------------------- %
%     RLS function with lambda value
% ---------------------------------- %

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2

P = P_init_scale * eye(3);

theta_hat = zeros(3,length(loudspeaker));
loudspeakerDelay1 = 0;
loudspeakerDelay2 = 0;

output  = zeros(1,length(loudspeaker));


for i=1:length(loudspeaker)

    if (i > D1)
        loudspeakerDelay1 = loudspeaker(i - D1);
    end

    if (i > D2)
        loudspeakerDelay2 = loudspeaker(i - D2);
```

```
    end

    phi = [loudspeaker(i); loudspeakerDelay1;  loudspeakerDelay2];

    P = (1/lambda)*(P - (P*phi*transpose(phi)*P)/(lambda + transpose(phi)*P*phi));

    G = P*phi;

    if ( i > 1)
        theta_hat(:,i) = theta_hat(:,i-1) + G*(mike1(i) - transpose(phi)*theta_hat(:,i-1));
    else
        theta_hat(:,i) = init_params + G*(mike1(i) - transpose(phi)*init_params);
    end

    output(i) = mike1(i) - theta_hat(2,i)*loudspeakerDelay1 - ...
        theta_hat(3,i)*loudspeakerDelay2;
end


end
```

A-LMS Function

```
function [ output, theta_hat ] = LMS_function(loudspeaker, mike1, init_params, mu)

% ---------------------------------- %
%     LMS: function
% ---------------------------------- %

Fs = 8192; %sampling freq
D1 = 1 * Fs; % delay 1
D2 = 2.5 * Fs; % delay 2

theta_hat = zeros(3,length(loudspeaker));
loudspeakerDelay1 = 0;
loudspeakerDelay2 = 0;

output  = zeros(1,length(loudspeaker));


for i=1:length(loudspeaker)

    if (i > D1)
        loudspeakerDelay1 = loudspeaker(i - D1);
    end

    if (i > D2)
        loudspeakerDelay2 = loudspeaker(i - D2);
    end

    phi = [loudspeaker(i); loudspeakerDelay1;  loudspeakerDelay2];

    G = mu*phi;

    if ( i > 1)
        theta_hat(:,i) = theta_hat(:,i-1) + G*(mike1(i) - transpose(phi)*theta_hat(:,i-1));
    else
        theta_hat(:,i) = init_params + G*(mike1(i) - transpose(phi)*init_params);
    end

    output(i) = mike1(i) - theta_hat(2,i)*loudspeakerDelay1 - ...
        theta_hat(3,i)*loudspeakerDelay2;
end


```

```
end
```

B- Gain Estimate

```
#include "SP2WS1.h"


// input signal history
float insignal1[100], insignal2[100];



// function prototypes
void gainestimateLMS(float,float,float,float[2]);
void gainestimateRLS(float,float,float,float[2]);


void gainestimate(float in1, float in2, float out, float gain[2])
{
    // record input signal history for checking signal magnitude using plot facility
    for (int i = 99; i > 0; i--)
    {
        insignal1[i] = insignal1[i-1];
        insignal2[i] = insignal2[i-1];
    }
    insignal1[0] = in1;
    insignal2[0] = in2;


    // estimate gain
    gain[0] = 0;
    gain[1] = 1;
//    gainestimateLMS(in1, in2, out, gain);
     gainestimateRLS(in1, in2, out, gain);
}

void gainestimateLMS(float in1, float in2, float out, float gain[2])
{
    // TODO: Implement gain estimation using LMS algorithm
    static float theta_hat1 = 0, theta_hat2 = 1.0;

    float mu = 1E-20;

    float err = out - (in1 * theta_hat1 + in2 * theta_hat2);

    theta_hat1 = theta_hat1 + (mu * in1) * err;
    theta_hat2 = theta_hat2 + (mu * in2) * err;

    gain[0] = theta_hat1;
    gain[1] = theta_hat2;

}

void gainestimateRLS(float in1, float in2, float out, float gain[2])
{
    // TODO: Implement gain estimation using RLS algorithm
    static float theta_hat1 = 0, theta_hat2 = 1.0, p11 = 1E-25, p12 = 0, p21 = 0, p22 = ...
        1E-25;  // p11 = p22 = 1 for identity

    float lambda = 0.9999;

    float err = out - (in1 * theta_hat1 + in2 * theta_hat2);


    float p11_in1 = p11 * in1;
```

```
61      float p11_in2 = p11 * in2;
62      float p12_in1 = p12 * in1;
63      float p12_in2 = p12 * in2;
64      float p21_in1 = p21 * in1;
65      float p21_in2 = p21 * in2;
66      float p22_in1 = p22 * in1;
67      float p22_in2 = p22 * in2;
68
69      float inv_lambda = 1.0 / lambda;
70
71      float denom = (lambda + in1 * (p11_in1 + p21_in2) + in2 * (p12_in1 + p22_in2));
72
73      float p11_new = inv_lambda * (p11 - (p11_in1 * (p11_in1 + p12_in2) + p21_in2 * ...
            (p11_in1 + p12_in2)) / denom);
74      float p12_new = inv_lambda * (p12 - (p12_in1 * (p11_in1 + p12_in2) + p22_in2 * ...
            (p11_in1 + p12_in2)) / denom);
75      float p21_new = inv_lambda * (p21 - (p11_in1 * (p21_in1 + p22_in2) + p21_in2 * ...
            (p21_in1 + p22_in2)) / denom);
76      float p22_new = inv_lambda * (p22 - (p12_in1 * (p21_in1 + p22_in2) + p22_in2 * ...
            (p21_in1 + p22_in2)) / denom);
77
78
79      p11 = p11_new;
80      p12 = p12_new;
81      p21 = p21_new;
82      p22 = p22_new;
83
84      theta_hat1 = theta_hat1 + ((p11 * in1) + (p12 * in2)) * err;
85      theta_hat2 = theta_hat2 + ((p21 * in1) + (p22 * in2)) * err;
86
87      gain[0] = theta_hat1;
88      gain[1] = theta_hat2;
89      // then save p values for next round
90
91  }
```