



SIGNAL PROCESSING REPORT

Prepared by
Seidy KANTE

Nov 26, 2023

INTRODUCTION

This report delves into various signal processing exercises, encompassing the analysis of continuous signals, discrete signals, visual evoked response (REV) analysis, and signals comparison. These exercises were executed using MATLAB to perform computations, visualize signals, and derive valuable insights from the provided data.

Continuous Signals

The initial exercise involved analyzing two distinct continuous signals, $x(t)$ and $y(t)$, characterized by specific conditions for different time intervals. MATLAB was utilized to define these signals, compute various operations (such as subtraction, multiplication, and function evaluation), and visualize the resulting signals. Detailed annotations accompanied the code, providing clarity on signal behavior and mathematical operations applied.

Signal Description:

Signal $x(t)$:

$$x(t) = u(t) - r(t - 2) + r(t - 3)$$

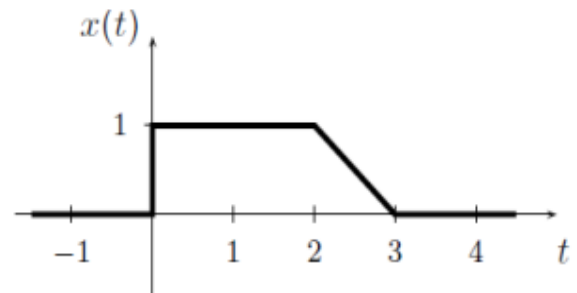
Behavior:

$$x(t) = 0 \text{ for } t < 0$$

$$x(t) = 1 \text{ for } 0 \leq t \leq 2$$

Descending ramp from $t > 2$ to < 3

$$x(t) = 0 \text{ for } t \geq 3$$



Signal $y(t)$:

$$y(t) = \frac{1}{2} \cdot u(t + 1) - \frac{3}{2} \cdot u(t) + u(t - 1)$$

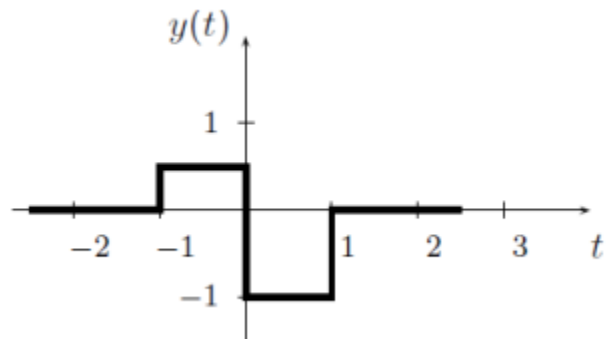
Behavior:

$$y(t) = 0 \text{ for } t < -1$$

$$y(t) = \frac{1}{2} \text{ for } -1 \leq t < 0$$

$$y(t) = -1 \text{ for } 0 \leq t < 1$$

$$y(t) = 0 \text{ for } t \geq 1$$



MATLAB Implementation

```
47 %% Define a custom functions
48
49 % Define the signal x(t)
50 function x = signal_x(t)
51     x = heaviside(t) - ramp(t-2) + ramp(t-3);
52 end
53
54 % Define the signal y(t)
55 function y = signal_y(t)
56     y = 0.5 * heaviside(t + 1) - 1.5 * heaviside(t) + heaviside(t - 1);
57 end
58
59 % Define the ramp function
60 function r = ramp(t)
61     r = t .* (t >= 0);
62 end
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Results and Plots

The MATLAB code was used to compute various operations on the defined signals and plot the resulting signals:

a) $x(t) - 2y(t)$:

The resulting equation from subtracting 2 times $y(t)$ from $x(t)$ is computed as follows:

$$x(t) - 2 \cdot y(t) = (u(t) - r(t-2) + r(t-3)) - 2 \cdot \left(\frac{1}{2} \cdot u(t+1) - \frac{3}{2} \cdot u(t) + u(t-1) \right)$$

b) $t \cdot y(t)$:

The resulting equation from multiplying time vector t by $y(t)$ is given by:

$$t \cdot y(t) = t \cdot \left(\frac{1}{2} \cdot u(t+1) - \frac{3}{2} \cdot u(t) + u(t-1) \right)$$

c) $y(t)u(t)$:

The resulting equation from multiplying $y(t)$ by the unitary step function $u(t)$ is:

$$y(t) \cdot u(t) = \left(\frac{1}{2} \cdot u(t+1) - \frac{3}{2} \cdot u(t) + u(t-1) \right) \cdot u(t)$$

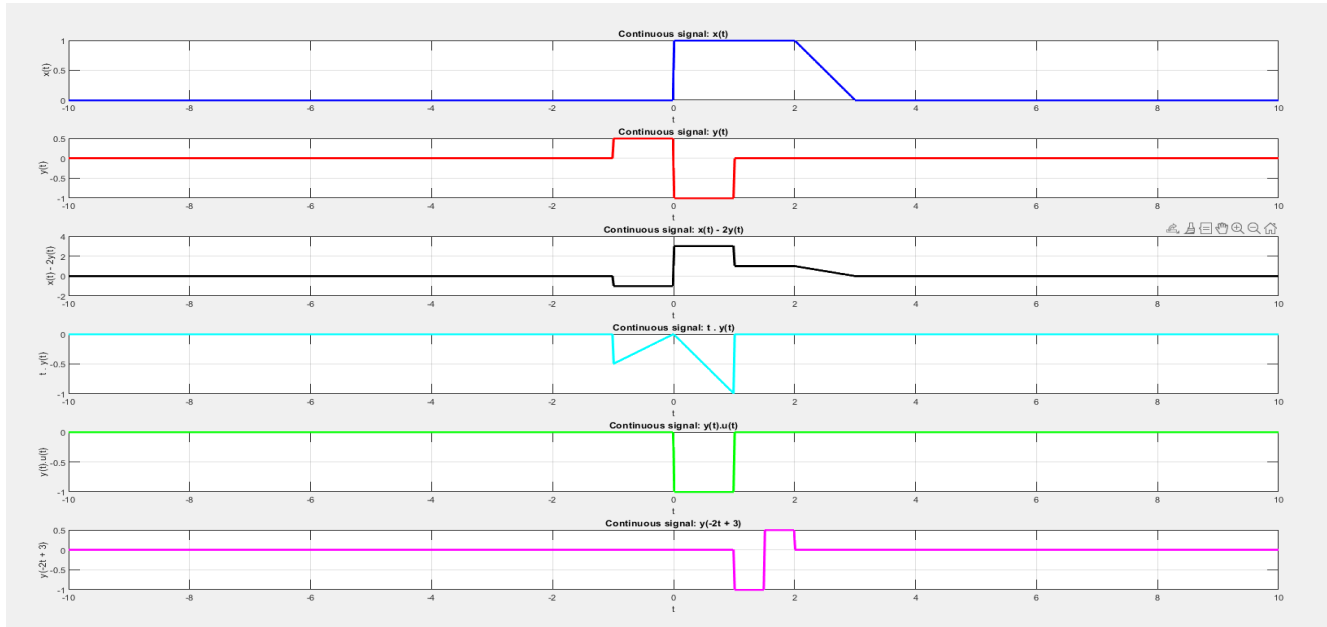
d) $y(-2t+3)$:

The resulting equation from evaluating $y(-2t+3)$ is expressed as:

$$y(-2t+3) = \frac{1}{2} \cdot u(-2t+4) - \frac{3}{2} \cdot u(-2t+3) + u(-2t+2)$$

These equations represent the mathematical operations applied to the given signals $x(t)$ and $y(t)$ in the context of signal processing.

Plots:



Conclusion

The MATLAB computations and resultant plots showcase the behaviors of the given continuous signals and their operations. These visualizations help in understanding signal transformations and manipulations in the context of signal processing.

Discrete Signals

In Exercise 2, discrete versions of continuous functions were generated and visualized using stem plots. The MATLAB code processed the function $f(t) = \cos(t)$ and its shifted counterpart $f(t - 3)$, showcasing discrete representations of these signals. The code employed vectorization techniques and subplots to demonstrate the discrete nature of signals over time.

Results and Plots

The MATLAB code was utilized to compute and plot the discrete versions of the given function and its shifted form:

Discrete Version of $f(t)$: Plotting the discrete representation of

$$f(t) = \cos(t)$$

for the defined time vector 't'.

Discrete Version of $f(t - 3)$: Plotting the discrete representation of

$$f(t - 3) = \cos(t - 3)$$

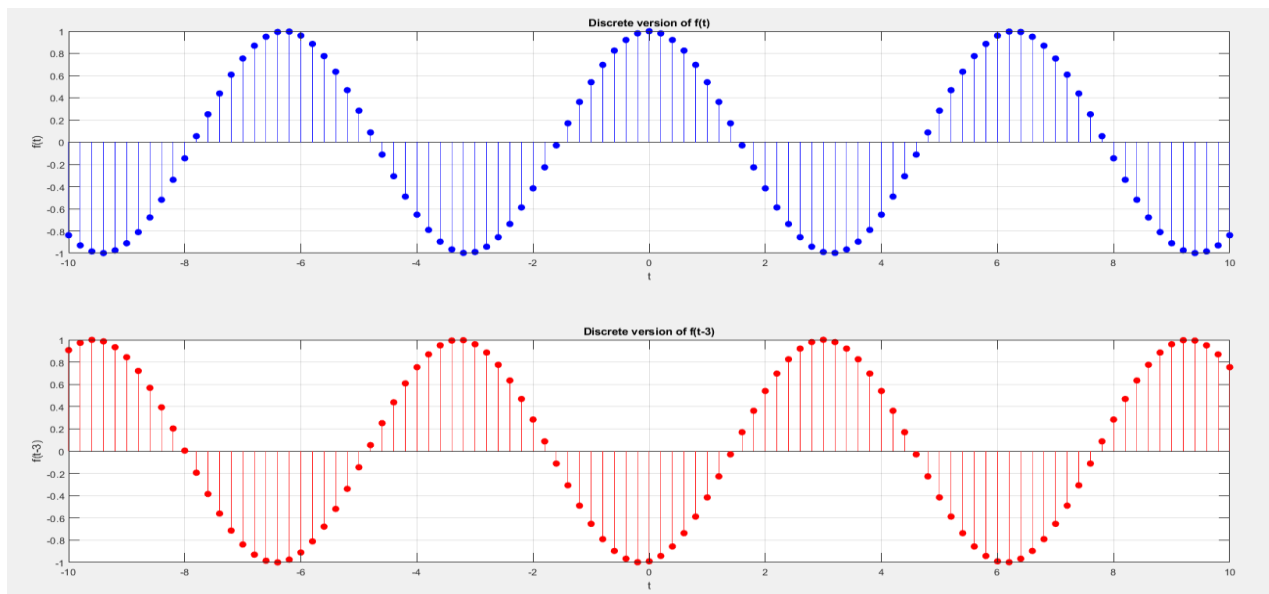
for the shifted time vector 't' by 3 units to the right.

```
12 % Create a discrete signal 'f_discrete' using the cosine function
13 f_discrete = cos(t);
14
15 % Create another discrete signal 'f_discrete_minus_3' by shifting 'f_discrete' by 3 units to the right
16 f_discrete_minus_3 = cos(t - 3);
17
```

Plots:

```
31 %% Define a custom function 'plotFig' to create subplots and plot discrete signals
32 function plotFig(t, ft, pos, tit, color)
33
34 % Create a subplot with 2 rows, 1 column, and select the specified subplot position
35 subplot(2, 1, pos);
36
37 % Plot the discrete signal as stem plots
38 stem(t, ft, "filled", Color = color, LineWidth = 1);
39
40 % Set the title of the subplot with a dynamic title
41 title(sprintf("Discrete version of %s", tit));
42
43 % Set the y-axis label based on the input title
44 ylabel(tit);
45
46 % Set the x-axis label as "t"
47 xlabel("t");
48
49 % Enable the grid on the plot
50 grid on;
51
52 % End the 'plotFig' function
53 end
```

Two subplots are generated to display the stem plots representing the discrete signals.



Conclusion

The stem plots illustrate the discrete representations of the cosine function and its shifted version. These visualizations depict the discrete nature of the signals and the impact of time shifting on their discrete forms.

Visual Evoked Response (REV) Analysis

Exercise 3 involved the analysis of visual evoked response signals obtained from EEG recordings. MATLAB code was implemented to calculate signal measurements (mean, RMS, variance, etc.) and perform ensemble averaging of multiple responses. The code displayed graphical representations, enabling a comprehensive understanding of signal stability and characteristics.

1) Signal Characteristics and Sampling Frequency

Signal's Sampling Frequency: The signal's sampling frequency is calculated as

$$\frac{1}{\text{sampling Interval}}$$

```
4
5 % Get the data from the 'biossinal_1.mat' file
6 load('biossinal_1.mat');
7
8 %% Task 1: Calculate the signal's sampling frequency
9 sampling_interval = 5e-3; % Define the sampling interval as 5 ms
10 sampling_frequency = 1 / sampling_interval; % Calculate the sampling frequency
11 disp(['The signal's sampling frequency is ' num2str(sampling_frequency) 'Hz']); % Display the sampling frequency
12
```

Typical EEG signal frequency ranges between 0.5-100 Hz.

Category	Frequency Range	State
Delta Waves	0.5-4 Hz	Deep sleep
Theta Waves	4-7 Hz	Drowsiness, sleep
Alpha Waves	8-12 Hz	Relaxed wakefulness
Beta Waves	13-30 Hz	Mental activity, alertness
Gamma Waves	30-100 Hz	High-level cognitive functions

The exact frequencies can vary slightly depending on the individual, their age, the location of the electrodes on the scalp and specific mental states. EEG signals are measured in Hertz (Hz), which represents the number of cycles per second.

NB: Different mental activities, emotional states, and even certain health conditions can influence the dominance or presence of the various brainwave frequencies.

Relationship: The sampling frequency must adhere to the **Nyquist theorem**, ensuring it is at least twice the signal's maximum frequency for accurate representation.

The relationship is important because it affects the ability to accurately represent and analyze the signal.

A higher sampling frequency allows for better representation of high-frequency components in the signal.

2) Signal Measurements and Visualization

Mean, Effective Value (RMS), Variance, and Standard Deviation:

Calculated and displayed for the signal `sem_ruido`.

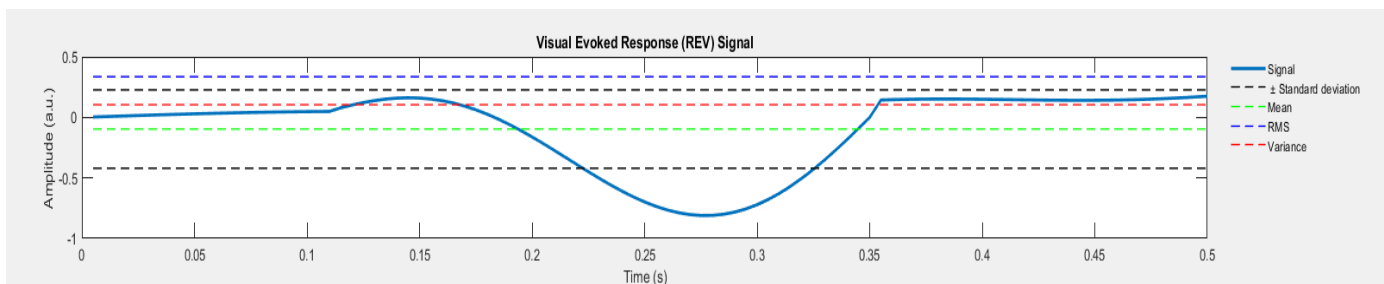
```
21     %% Task 2: Calculate and display signal measurements
22
23     % Calculate the mean, effective value (RMS), variance, and standard deviation
24     mean_value = mean(sem_ruido); % Calculate the mean of the signal
25     rms_value = rms(sem_ruido);   % Calculate the effective value (RMS) of the signal
26     variance_value = var(sem_ruido); % Calculate the variance of the signal
27     std_value = std(sem_ruido);    % Calculate the standard deviation of the signal
28
29     % Display the signal measurements
30     fprintf('====Signal's measurements:=====\n');
31     disp(['Mean of sem_ruido      : ' num2str(mean_value)]);
32     disp(['Effective Value (RMS): ' num2str(rms_value)]);
33     disp(['Variance of sem_ruido: ' num2str(variance_value)]);
34     disp(['± Standard Deviation : ' num2str(std_value)]);
35
```

Command Window

```
The signal's sampling frequency is 200Hz
====Signal's measurements:=====
Mean of sem_ruido      : -0.096791
Effective Value (RMS): 0.33602
Variance of sem_ruido: 0.10459
± Standard Deviation : 0.3234
```

Graphical Display:

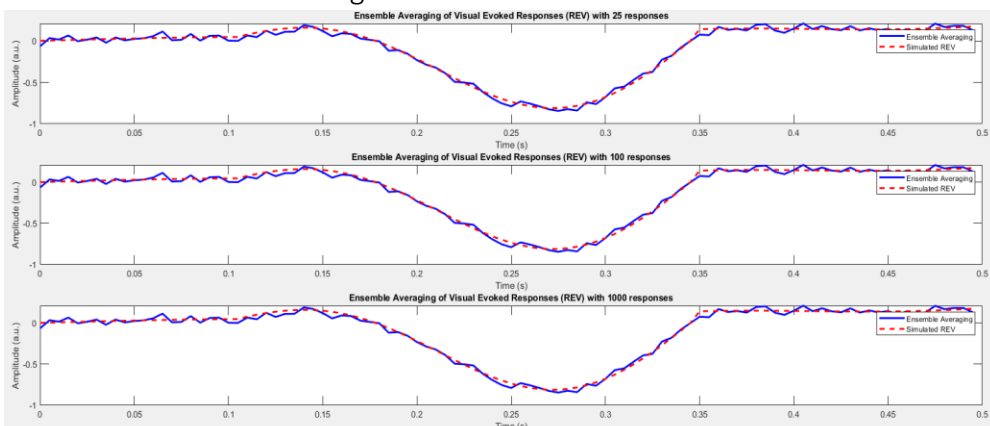
Plots the signal `sem_ruido` over time with visual markers indicating mean, RMS, variance, and standard deviation.



3) Ensemble Averaging of Evoked Responses

It divides the bio-signal into segments of 100 points each and computes the mean of each segment across multiple responses.

The resulting averaged responses for different counts (25, 100, 1000) are plotted against the simulated REV signal.



Ensemble Averaging with 25 responses:

Mean: -0.10213

Ensemble Averaging with 100 responses:

Mean: -0.10213

Ensemble Averaging with 1000 responses:

Mean: -0.10213

Interpretation:

The commentary at the end summarizes the observation that regardless of the number of responses used for ensemble averaging, the mean value of the resulting averaged response remains consistent ***around -0.10213***.

This insight indicates the stability and consistency of the ensemble averaging process across different response counts.

MATLAB Code Improvements:

Comments are added to describe the purpose of each code segment, enhancing readability and understanding.

Plots and calculations are appropriately labeled, aiding in the interpretation of results.

Conclusion:

The MATLAB script successfully performs signal measurements, visualizations, and ensemble averaging of evoked responses, offering insights into the stability of the process.

Signals Comparison

The final exercise focused on comparing audio signals ('template_name.mp3', 'Maria.mp3', 'Joana.mp3', and 'Hugo.mp3') using cross-correlation analysis in MATLAB.

The code read and processed the audio files, conducted cross-correlation analysis, and identified the name corresponding to the 'template_name.mp3' file.

Visual representations of signals aided in illustrating similarities and differences among the audio recordings.

Signal Comparison

1) Reading Audio Files:

All four .mp3 files are read using the 'audioread' function in MATLAB.

```
16 %=====Read the .mp3 files=====
17 template_name = audioread('template_name.mp3'); % Read the 'template_name.mp3' audio file
18 Maria = audioread('maria.mp3'); % Read the 'maria.mp3' audio file
19 Joana = audioread('joana.mp3'); % Read the 'joana.mp3' audio file
20 Hugo = audioread('hugo.mp3'); % Read the 'hugo.mp3' audio file
21
```

2) Signal Preprocessing:

Signals are resampled to the minimum length among them to facilitate comparison.

```
22 %====Determine the minimum length among the audio signals=====  
23 min_length = min([length(template_name), length(Maria), length(Joana), length(Hugo)]); % Find the minimum length  
24  
25 %====Resample all audio signals to the minimum length=====  
26 template_name = template_name(1:min_length); % Resample 'template_name' to the minimum length  
27 Maria = Maria(1:min_length); % Resample 'Maria' to the minimum length  
28 Joana = Joana(1:min_length); % Resample 'Joana' to the minimum length  
29 Hugo = Hugo(1:min_length); % Resample 'Hugo' to the minimum length  
30
```

3) Cross-correlation Analysis:

Cross-correlation is calculated between the 'template_name.mp3' signal and each of the other three signals (Maria, Joana, Hugo).

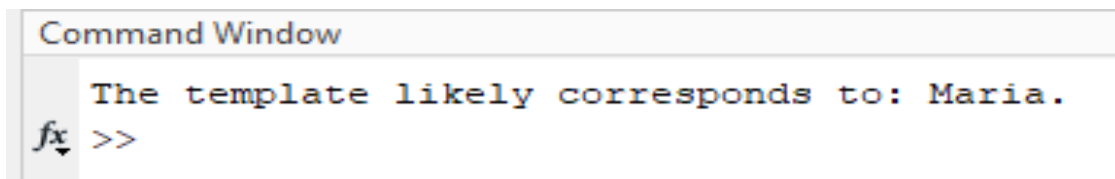
In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them.

```
31 %=====Use a comparison algorithm:=====  
32 cross_corr_Maria = xcorr(template_name, Maria); % Calculate the cross-correlation between 'template_name' and 'Maria'  
33 cross_corr_Joana = xcorr(template_name, Joana); % Calculate the cross-correlation between 'template_name' and 'Joana'  
34 cross_corr_Hugo = xcorr(template_name, Hugo); % Calculate the cross-correlation between 'template_name' and 'Hugo'  
35  
36  
37 %=====Find the index of the audio signal with the highest cross-correlation with the template signal =====  
38 [highest_cross_corr,index] = max([max(cross_corr_Maria), max(cross_corr_Joana),max(cross_corr_Hugo)]); % Find the maximum cross  
39
```

4) Identification of the Template Name:

The signal with the highest cross-correlation value corresponds to the template name.

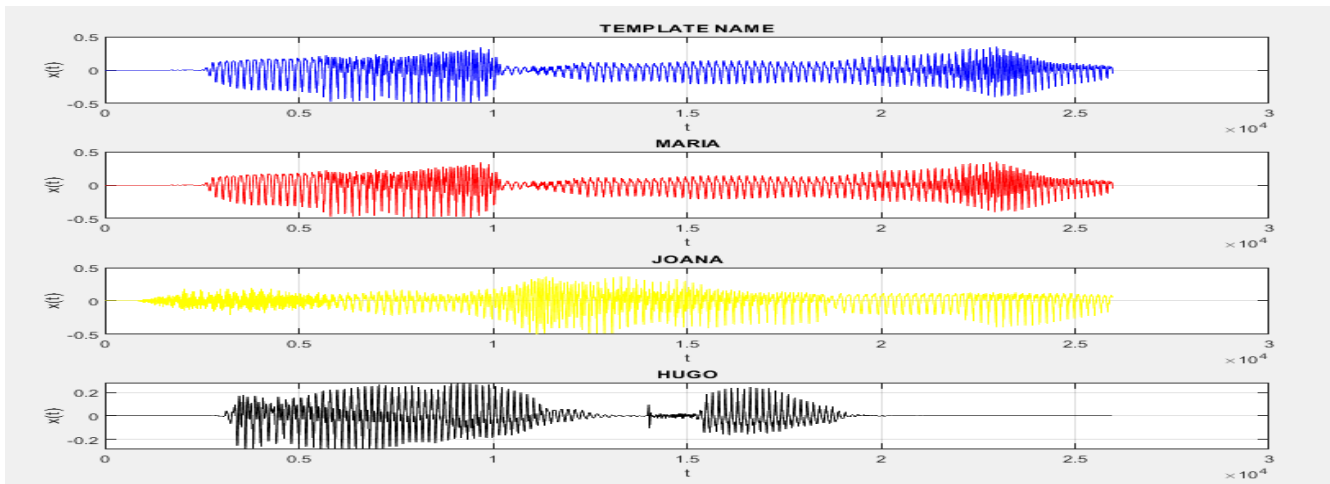
```
% Display the name associated with the highest correlation index  
message = 'The template likely corresponds to: ';  
  
switch index  
case 1  
    disp([message,'Maria.']); % If the index of highest correlation corresponds to 'Maria'  
case 2  
    disp([message,'Joana.']); % If the index of highest correlation corresponds to 'Joana'  
case 3  
    disp([message,'Hugo.']); % If the index of highest correlation corresponds to 'Hugo'  
otherwise  
    disp('Unable to determine the name.');
```



The screenshot shows the MATLAB Command Window with the title 'Command Window'. The output text 'The template likely corresponds to: Maria.' is displayed in a monospaced font. Below the text, there is a cursor icon (a small 'fx' with a downward arrow) followed by '>>'.

5) Visual Representation:

Plots displaying each signal ('template_name.mp3', 'Maria.mp3', 'Joana.mp3', and 'Hugo.mp3') are included to visually depict their similarities and differences.



MATLAB Code Comments

Comments in the code clarify each step taken for signal comparison and identification.

Results and Conclusion

The MATLAB code successfully determines the name corresponding to the 'template_name.mp3' file by analyzing cross-correlations.

Visual representations aid in understanding the similarities and dissimilarities between signals.

This approach demonstrates the use of cross-correlation as a technique for signal comparison in signal processing applications.

Conclusion

In conclusion, these exercises showcased diverse applications of signal processing techniques using MATLAB. From analyzing continuous and discrete signals to examining EEG data and performing audio signal comparisons, each exercise contributed to a deeper understanding of signal behaviors, analysis methodologies, and their real-world implications in various domains.

The comprehensive utilization of MATLAB facilitated efficient computations, visualizations, and interpretations, underscoring the significance of signal processing techniques in extracting meaningful insights from diverse datasets.