

디지털신호처리설계

9조 (구해조)
안민혁(12181795)
강윤석(12163189)
최창환(12171174)

CONTENTS

- 000 팀원 업무 분담 소개
- 001 GUI INTRODUCTION
- 002 Grouping(성별, 주파수)
- 003 목소리 검출(시간영역, 주파수 영역)

004 필터 설계

005 데모

0. 팀원 업무 분담



안민혁 (12181795)

- 음원 분석
- 자동 분류
- 목소리 제거
- GUI 구현



최창환 (12171174)

- 음원 분석
- 팀원 목소리 찾기
- 발표자료



강윤석 (12163189)

- 목소리 제거

Part 1.

GUI Introduction



1.1 GUI INTRODUCTION

GUI 소개

[음원 추출]

- 추출 성공 시
성별, 주파수 별
자동 분류 기능

[성별 그룹핑 UI]

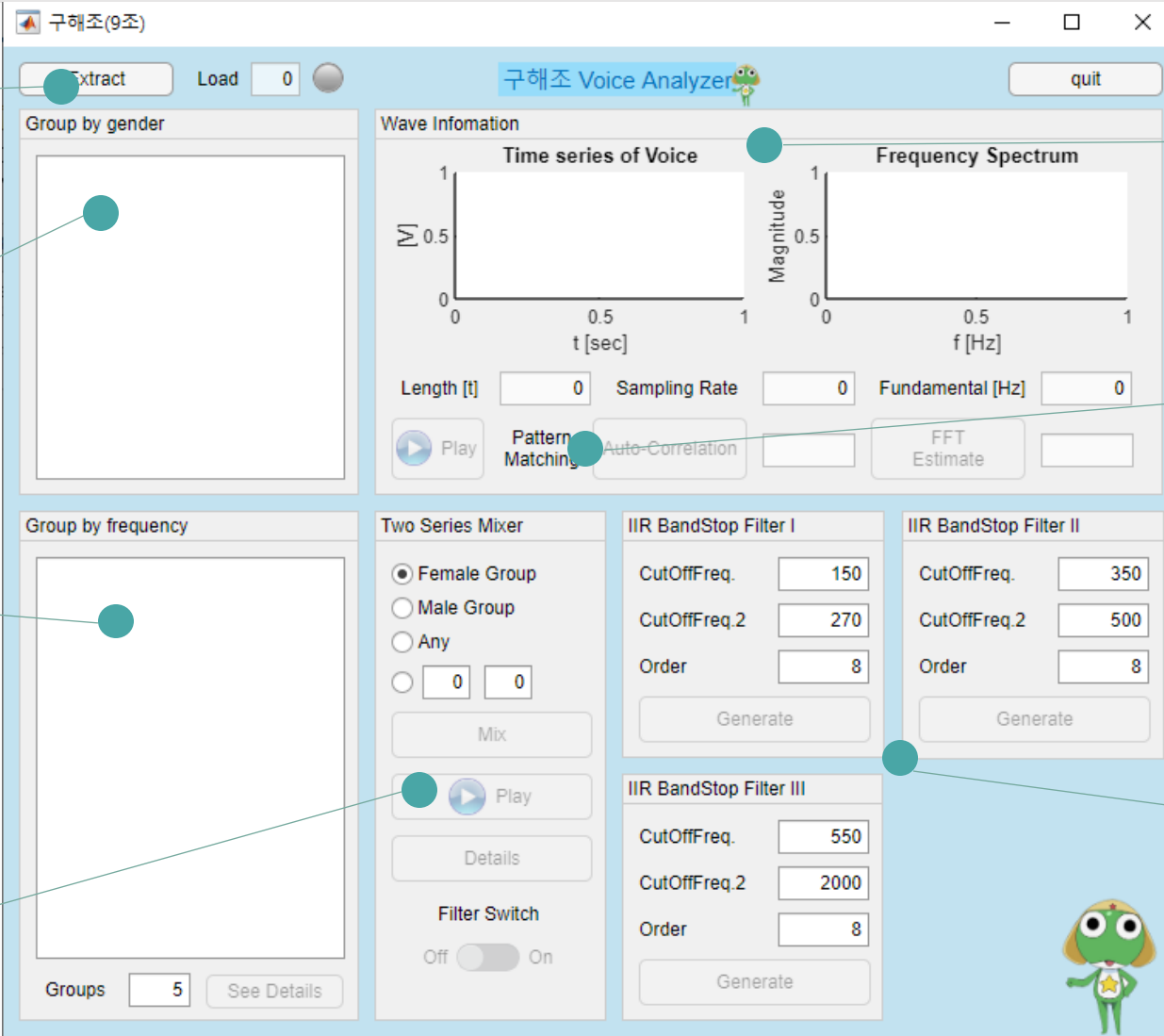
- 기본주파수를 찾아
여성과 남성을 자동 그룹핑

[주파수 별 그룹 UI]

- K-means 알고리즘을
이용해 기본 주파수 관측 값을
N개의 군집으로 분할 후
자동 그룹핑
- 그룹 개수 지정 기능
- 그룹과 그룹 중심 플로팅 기능

[음원 섞기 기능]

- 그룹 별 음원 섞기 기능
- 지정한 음원 끼리 섞기 기능
- 재생 기능
- 결과 플로팅 기능
- 필터 스위치 기능



[음원 정보 출력 기능]

- 음원 파형 플로팅 기능
- 주파수 스펙트럼 플로팅 기능
- 길이, Fs, 기본 주파수 출력
- 음원 재생 기능

[패턴 매칭 기능]

- 시간 영역에서 팀원 찾기
- 주파수 영역에서 팀원 찾기

[버터워스 대역저지 필터 기능]

- 최대 세 구간의 대역저지 기능
- Cut off 주파수 설정 기능
- 차수 설정 기능

1.2 GUI INTRODUCTION 고찰

GUI 소개

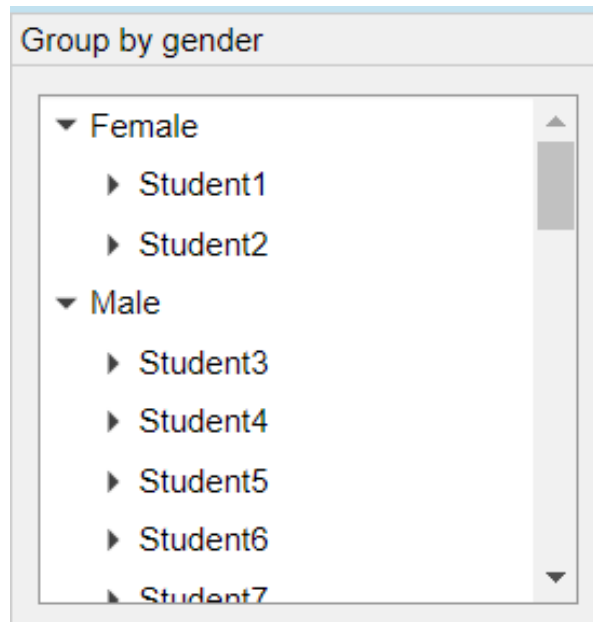
- uitreenode() 함수를 이용해서 그룹핑을 한 결과, 그룹핑 처리속도의 지연이 발생했지만 그룹핑 결과에 대한 직관성이 뛰어나게 증가한 효과를 보임
- 버튼 활성화/불활성화 코드를 통해서 직관성을 높이고, 중복 클릭으로 인한 오류/과부하 방지
- 변수 입력을 자유롭게 하는 것을 지향함
- 깔끔한 디자인 지향

Part 2.

Grouping(성별, 주파수)



2.1 그룹핑 (성별)



남자와 여자의 목소리에 관련된 연구논문 및 자료조사

음성 스펙트럼

비성조 언어에서는 음성이 모음과 자음으로 구성되어 있다고 말할 수 있음.

모음 소리는 성대에 의해 생성되고 성대에 의해 필터링됨.

속삭임은 유성음이 없음

일반적으로 음높이 기본 주파수(f_0)는 남성의 경우 100-120Hz 범위에 있지만 이 범위를 벗어나는 변동이 발생할 수 있음. 여성의 f_0 은 약 한 옥타브 더 높음(200-240Hz). 어린이의 경우 f_0 약 300Hz.

자음은 목과 입, 특히 혀와 입술을 통해 공기가 통과하여 형성되는 공기 막힘 및 소음 소리에 의해 생성됨. 주파수 측면에서 자음은 500Hz 이상. (특히 2k~4kHz)

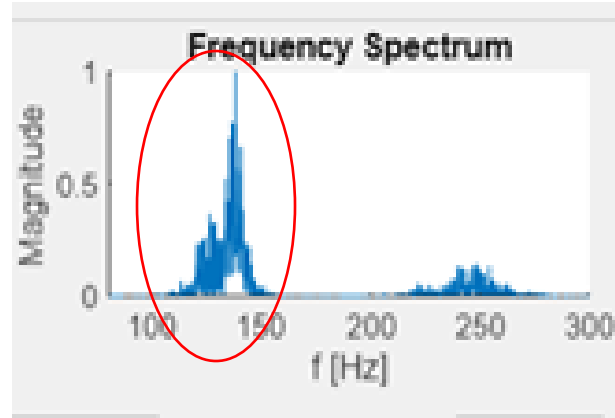
1. 성별에 따른 그룹핑을 위해 남녀의 주파수 범위를 자료조사를 통해 알아냈다.
2. 남성의 경우 기본 주파수가 100-120Hz 범위, 여성의 경우 기본 주파수가 200-240Hz였다.

2.1 그루핑 (성별)

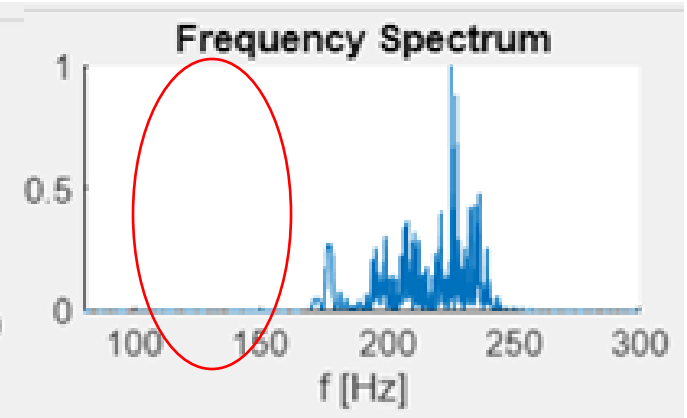
```
app.MaleRef = [80, 150];
```

```
[p, f] = periodogram(app.s, [], [], app.Fs); % 스펙트럼 구하기
p = abs(p/max(p)); %정규화
%% findpeaks %%
format short
[app.pks, app.locs] = findpeaks(p, 'MinPeakHeight', 0.02);
app.FreqIndex = f(app.locs);
%% Gender Detection

MaleCount = 0;
for i = 1:numel(app.FreqIndex)
    if(app.MaleRef(1)<=app.FreqIndex(i) && app.FreqIndex(i)<=app.MaleRef(2))
        MaleCount = MaleCount + 1;
    end
end
```



(남성 스펙트럼)



(여성 스펙트럼)

3. 자료조사를 바탕으로 MaleRef를 만들어 주었다.
4. Loop를 돌면서 모든 목소리를 주파수 영역에서 기준치 이상의 신호를 검출한다.
5. 검출된 신호들의 index가 남자 목소리의 주파수 범위에 있으면 MaleCount 변수를 증가시킨다.
6. MaleCount가 5보다 작으면 여성으로 5이상이면 남성으로 그룹핑을 한다.

2.1 그루핑 (성별) 코드

```
function OpenButtonPushed(app, event)
    app.OpenButton.Text = "Loading...";
    app.OpenButton.Enable = "off";
    pause(1);
    try
        app.dataset=load('allAudio.mat');
    catch
        app.VoiceAnalyzerLabel.Text = 'allAudio.mat Not found';
        app.Lamp.Color = [1.00 0.00 0.00]; % 적색 점멸
    end
    try
        app.TeamVoice=load('Team9.mat');
    catch
        app.VoiceAnalyzerLabel.Text = 'Team9.mat Not found';
        app.Lamp.Color = [1.00 0.00 0.00]; % 적색 점멸
    end
    count = 0;
    Female = uitreenode(app.Tree, 'Text', 'Female');
    Male = uitreenode(app.Tree, 'Text', 'Male');
    app.MaleRef = [80, 150];
    app.FemaleRef = [200 280];
    for Loop = 1:numel(app.dataset.Dataupload)
        app.LoadEditField.Value = Loop;
        %% extract and sound
        app.s = app.dataset.Dataupload{1,Loop}.signal;
        app.Fs = app.dataset.Dataupload{1,Loop}.Fs;
        if Loop>=16 && Loop <= 20
            app.Fs = 8000;
        end
        app.N = numel(app.s);
        app.t = (0:app.N-1)/app.Fs;
        %% periodogram %%
```

```
[p, f] = periodogram(app.s, [], [], app.Fs); % 스펙트럼 구하기
p = abs(p/max(p)); %정규화
%% findpeaks %%
format short
[app.pks, app.locs] = findpeaks(p, 'MinPeakHeight', 0.02);
app.FreqIndex = f(app.locs);
%% Gender Detection

MaleCount = 0;
for i = 1:numel(app.FreqIndex)
    if(app.MaleRef(1)<=app.FreqIndex(i) && app.FreqIndex(i)<=app.MaleRef(2))
        MaleCount = MaleCount + 1;
    end
end

if (MaleCount < 5) % Gender Tree
    app.GenderGroup(Loop) = 0;
    if(count==0)
        txt0 = string(1+round(Loop/5));
        txt = 'Student'+txt0;
        Person = uitreenode(Female, 'Text', txt);
    end
    txt0 = string(Loop);
    voice=uitreenode(Person, 'Text', txt0);

    if(count<4)
        count=count+1;
    else
        count =0;
    end
end
```

2.1 그루핑 (성별) 코드

```

else
    app.GenderGroup(Loop) = 1;
    if(count==0)
        txt0 = string(1+round(Loop/5));
        txt = 'student'+txt0;
        Person = uitreenode(Male, 'Text', txt);
    end
    txt0 = string(Loop);
    voice=uitreenode(Person, 'Text', txt0);

    if(count<4)
        count=count+1;
    else
        count =0;
    end
end
%% Frequency Peak
[pks2, locs2] = findpeaks(p, 'MinPeakHeight', 0.02, 'MinPeakDistance', 100);
FreqIndex2 = f(locs2);

if(app.GenderGroup(Loop) == 0) % if Female
    for i = 1:numel(FreqIndex2)
        if(FreqIndex2(i) < app.FemaleRef(1) || FreqIndex2(i) > app.FemaleRef(2))
            pks2(i) = 0;
        end
    end
else % if Male
    for i = 1:numel(FreqIndex2)
        if(FreqIndex2(i) < app.MaleRef(1) || FreqIndex2(i) > app.MaleRef(2))
            pks2(i) = 0;
        end
    end
end
end
end

```

```

for i = 1:numel(FreqIndex2)
    if(pks2(i) == max(pks2))
        app.HighFGroup(Loop) = FreqIndex2(i);
    end
end

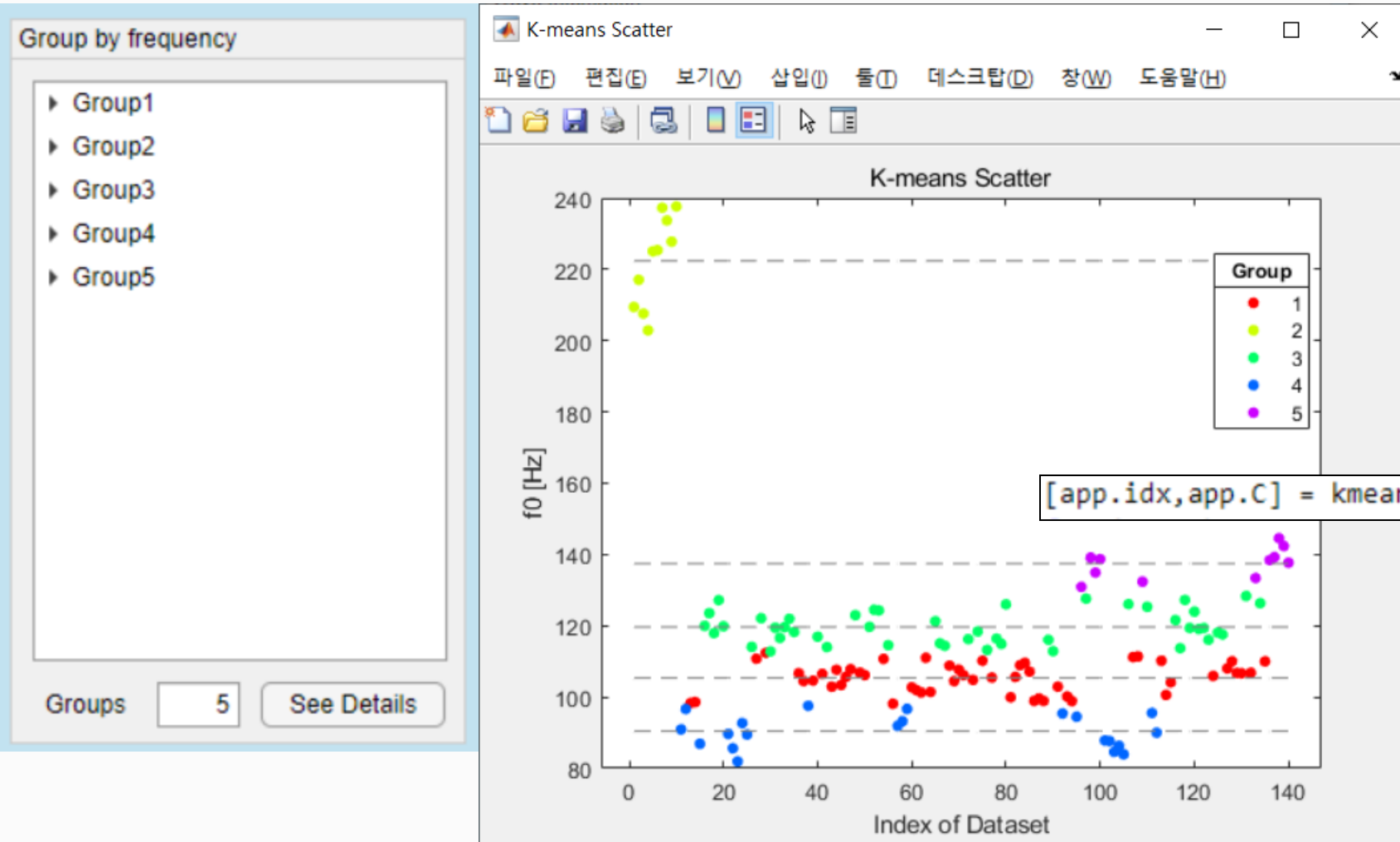
end %% Loop end
app.OpenButton.Enable = "off";
app.OpenButton.Text = "Extracted!";
app.MixButton.Enable = "on";
app.SeeDetailsButton.Enable = "on";
app.Lamp.Color = [0.00 1.00 0.00]; % 녹색 점멸

%% Frequency Grouping - kmeans
kgroups = app.GroupsEditField.Value;
[app.idx,app.C] = kmeans(app.HighFGroup',kgroups,'MaxIter',1000);
for j = 1:kgroups
    txt = 'Group' + string(j);
    nodes(j) = uitreenode(app.Tree2, 'Text', txt);
end
for i = 1:numel(app.dataset.Dataupload)
    txt0 = string(i);
    for n = 1:kgroups
        if (app.idx(i)==n)
            voice = uitreenode(nodes(n), 'Text', txt0);
        end
    end
end
end

```

2.2 주파수가 유사한 사람에 대해 그룹핑하기

Using K-means clustering algorithm



각 음원들의 기본 주파수 좌표 사이의 거리를 통해서 그룹의 중심을 1000번 추정 반복 계산하는 알고리즘으로 음원의 기본 주파수 데이터를 지정한 개수의 그룹으로 분할합니다.

음원 기본 주파수 분포와 그룹 상태를 색상 별로 확인할 수 있는 플로팅 기능도 구현하였습니다.

2.2 주파수가 유사한 사람에 대해 그룹핑하기 코드

Using K-means clustering algorithm

```
% Value changed function: GroupsEditField
function GroupsEditFieldValueChanged(app, event)
    %% Frequency Grouping - kmeans
    app.Tree2.Children.delete;
    try
        kgroups = app.GroupsEditField.Value;
        [app.idx, app.C] = kmeans(app.HighFGroup', kgroups, 'MaxIter', 1000);
        for j = 1:kgroups
            txt = 'Group' + string(j);
            nodes(j) = uitreenode(app.Tree2, 'Text', txt);
        end
        for i = 1:numel(app.dataset.Dataupload)
            txt0 = string(i);
            for n = 1:kgroups
                if (app.idx(i) == n)
                    voice = uitreenode(nodes(n), 'Text', txt0);
                end
            end
        end
        app.Lamp.Color = [0.00 1.00 0.00]; % 녹색 점멸
    catch
    end
end
```

2.3 그룹핑 고찰

- 그룹핑(성별)

성별에 따른 그룹핑의 경우 주파수 대역의 차이가 커서 자료 조사에 따른 기준 설정이 명확했다.

- 그룹핑(주파수)

주파수에 따른 그룹핑에 대한 기준 설정 어려웠다.

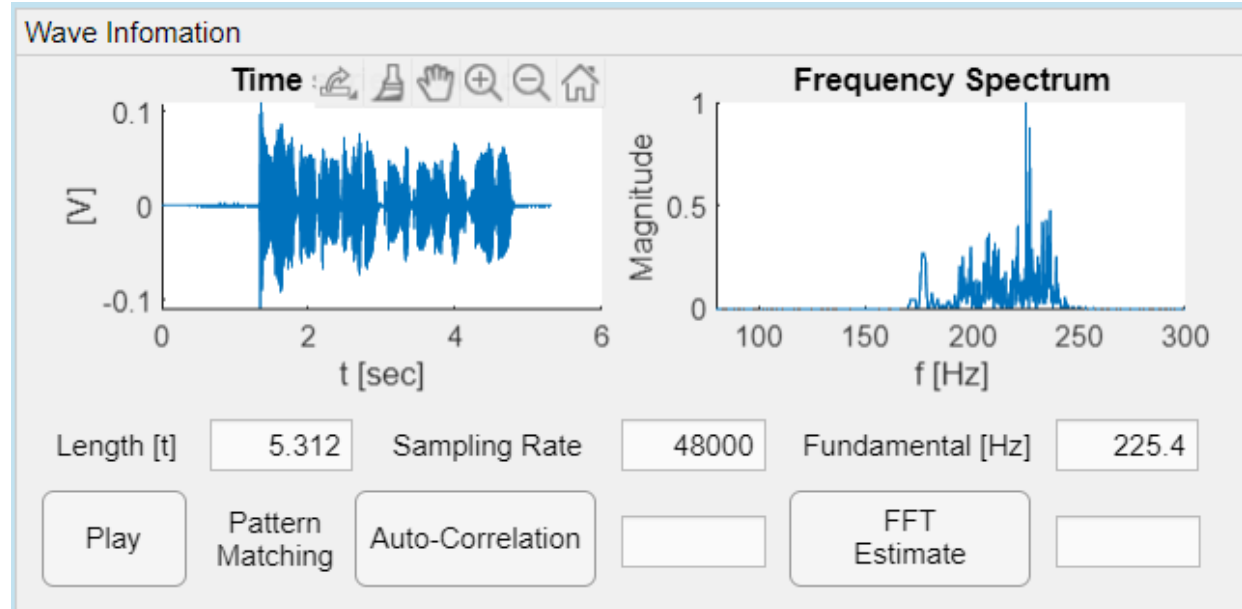
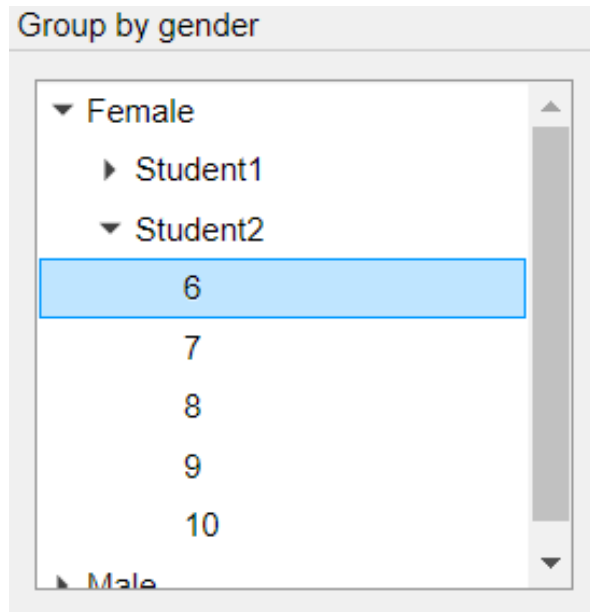
시행 착오 끝에 k-means 클러스터링 알고리즘으로 구현

Part 3.

목소리검출(시간영역, 주파수영역)



3.1 팀 멤버 목소리 찾기 (시간영역)

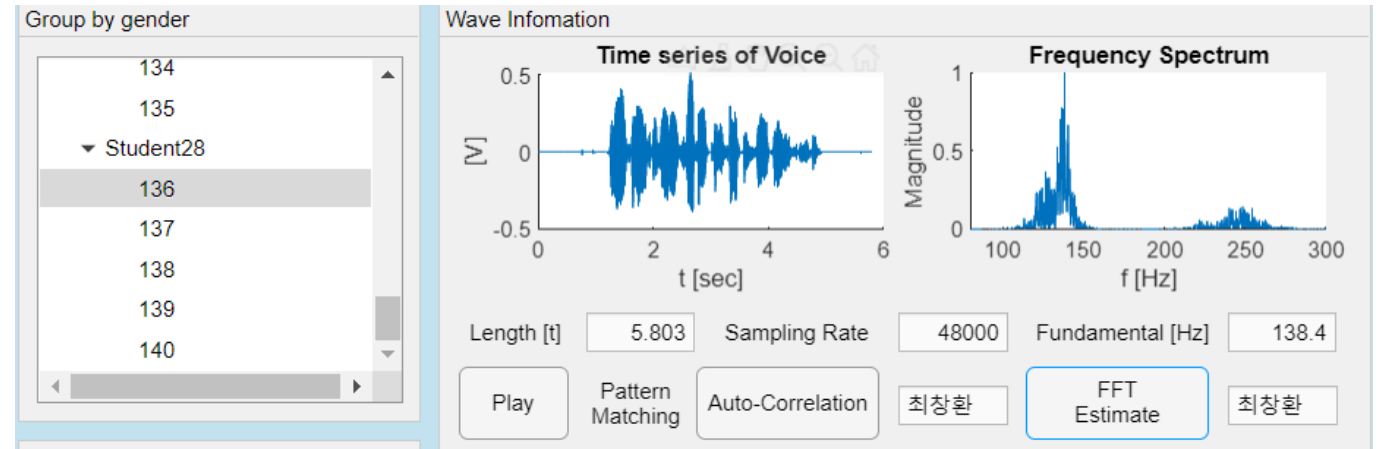


1. 9조의 팀 멤버의 목소리가 있는 파일을 하나 만든다.
2. GUI에서 Student를 고른 후 Auto-Correlation 버튼을 누르면 Student신호와 1.에서 만든 팀 멤버의 목소리를 xcorr한다.

3.1 팀 멤버 목소리 찾기 (시간영역)

% corr 성능 위해 P/Q 배로 resample 하기

```
if Fs9 >= app.Fs
    [P, Q] = rat(app.Fs/Fs9);
    s9 = resample(s9, P, Q);
    [C1, lag] = xcorr(app.s, s9);
else
    [P, Q] = rat(Fs9/app.Fs);
    ss = resample(app.s, P, Q);
    [C1, lag] = xcorr(ss, s9);
end
```



- 상호상관함수의 안정성을 위하여 두 신호중 작은 샘플링 레이트로 resampling해준다.
- xcorr 한 결과를 바탕으로 팀 멤버의 목소리와 일치하면 팀 멤버의 이름을 디스플레이 한다.

3.1 팀 멤버 목소리 찾기 (시간영역) 코드

```
% Button pushed function: AutoCorrelationButton
function AutoCorrelationButtonPushed(app, event)
    count = 0;
    count2 = 0;
    for Loop = 1:numel(app.TeamVoice.Team9)
        s9 = app.TeamVoice.Team9{1,Loop}.signal;
        Fs9 = app.TeamVoice.Team9{1,Loop}.Fs;
        % corr 성능 위해 P/Q 배로 resample 하기

        if Fs9 >= app.Fs
            [P, Q] = rat(app.Fs/Fs9);
            s9 = resample(s9, P, Q);
            [C1, lag] = xcorr(app.s, s9);
        else
            [P, Q] = rat(Fs9/app.Fs);
            ss = resample(app.s, P, Q);
            [C1, lag] = xcorr(ss, s9);
        end

        C1=abs(C1)/max(C1);
        [pks, locs] = findpeaks(C1, 'MinPeakHeight', 0.5, 'MinPeakDistance', 30000);
        if(numel(pks)==1)
            count = count + 1;
            if(C1(round(numel(C1)/2)) >= 0.9)
                if Loop<=5
                    name = '강윤석';
                elseif Loop<=10
                    name = '안민혁';
                elseif Loop<=15
                    name = '최창환';
                end
            end
        end
    end
end
```

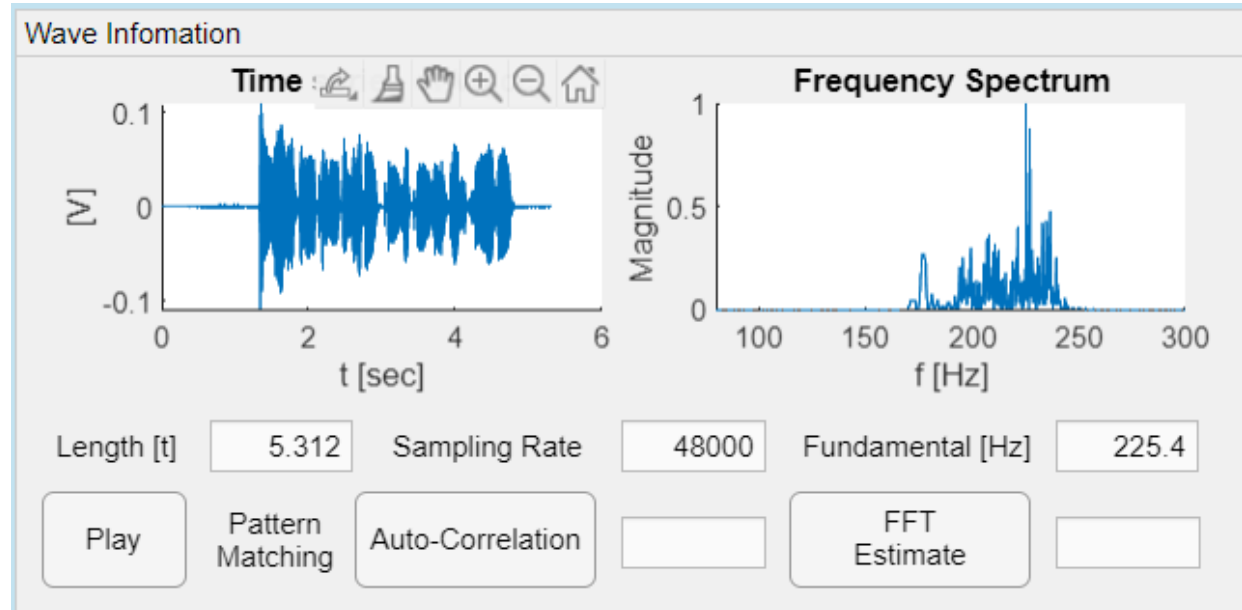
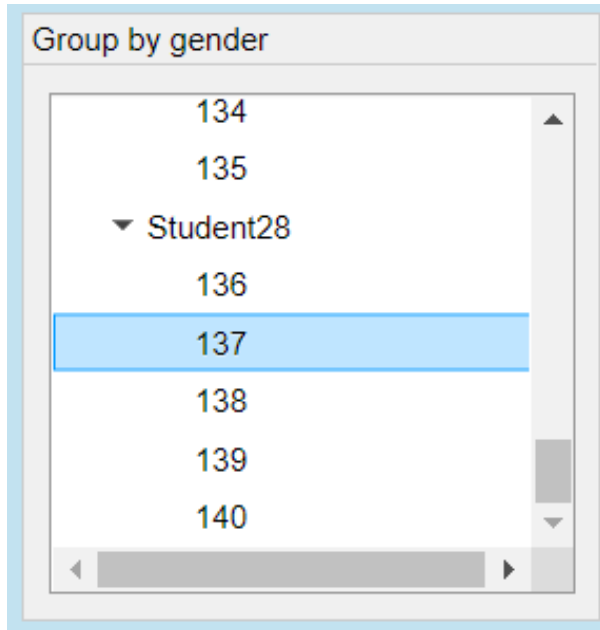
```
        count2 = count2 + 1;
        break;
    end
end

end

if count >= 1 && count2 >= 1
    app.WhoEditField.Value = name;
else
    app.WhoEditField.Value = 'Unidentified';
end

end
```

3.2 팀 멤버 목소리 찾기 (주파수 영역)

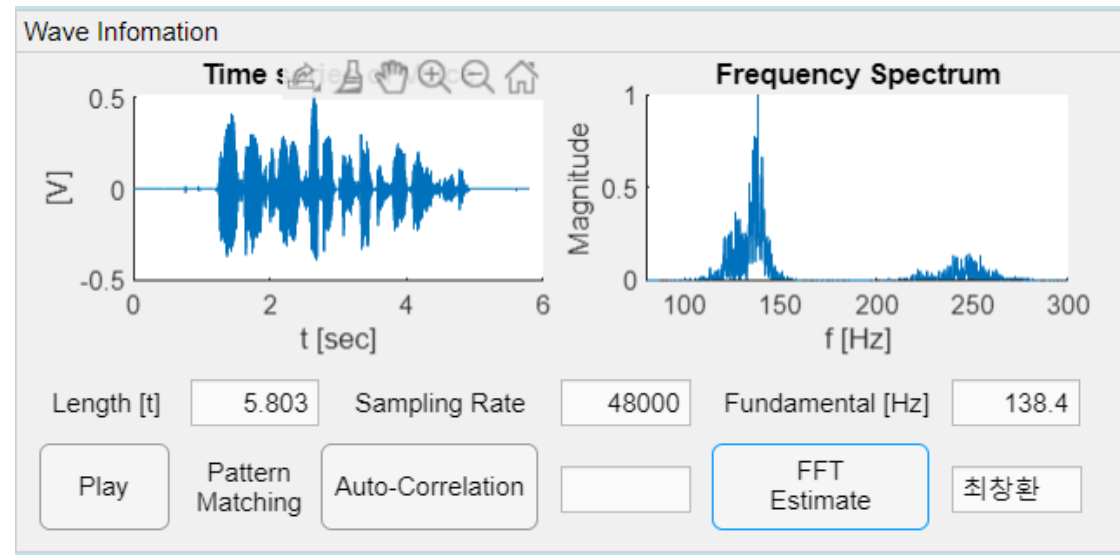


1. GUI에서 학생을 고른 후 FFT Estimate 버튼을 누르면 학생의 목소리와 팀 멤버의 목소리를 FFT한다.

3.2 팀 멤버 목소리 찾기 (주파수 영역)

% P/Q 배로 resample 하기

```
if Fs9 >= app.Fs
    [P, Q] = rat(app.Fs/Fs9);
    s9 = resample(s9, P, Q);
    s = fft(app.s);
else
    [P, Q] = rat(Fs9/app.Fs);
    ss = resample(app.s, P, Q);
    s = fft(ss);
end
```



2. 두 신호의 샘플링 레이트를 일치시키기 위해 두 신호 중 작은 샘플링 레이트로 resampling해준다.
3. 각각의 FFT에서 피크의 인덱스를 비교해 팀멤버의 목소리인지 판별한다.
4. 팀 멤버의 목소리와 일치하면 팀 멤버의 이름을 디스플레이 한다.

3.2 팀 멤버 목소리 찾기 (주파수 영역) 코드

```
function FFTestimateButtonPushed(app, event)
    for Loop = 1:numel(app.TeamVoice.Team9)
        s9 = app.TeamVoice.Team9{1,Loop}.signal;
        Fs9 = app.TeamVoice.Team9{1,Loop}.Fs;
        % P/Q 배로 resample 하기

        if Fs9 >= app.Fs
            [P, Q] = rat(app.Fs/Fs9);
            s9 = resample(s9, P, Q);
            s = fft(app.s);
        else
            [P, Q] = rat(Fs9/app.Fs);
            ss = resample(app.s, P, Q);
            s = fft(ss);
        end
        s9 = fft(s9);
        s = real(s);
        s9 = real(s9);
        N1 = length(s);
        N2 = length(s9);
        s(round(N1/2) : N1) = 0;
        s9(round(N2/2) : N2) = 0;

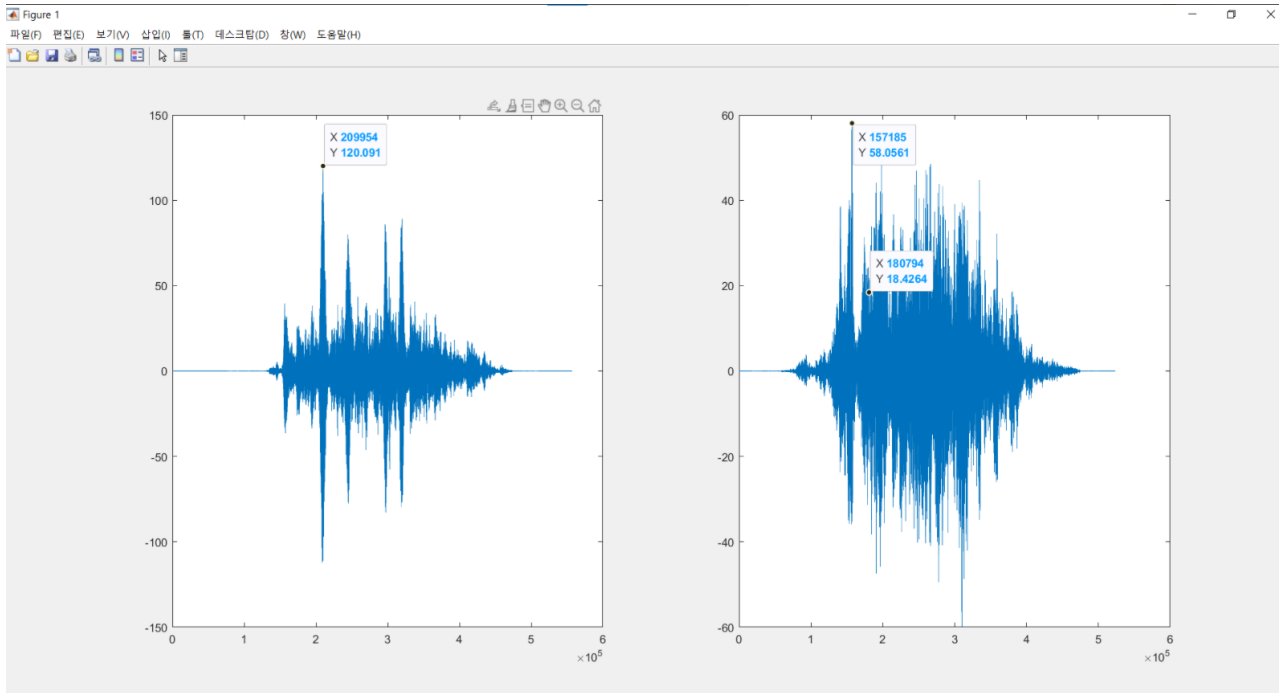
        [s_high_y, s_high_x] = max(s);
        [s9_high_y, s9_high_x] = max(s9);
    end
end
```

```
if(s9_high_y - 0.002 <= s_high_y && s_high_y <= s9_high_y + 0.002)
    if Loop<=5
        app.WhoEditField_2.Value = '강윤석';
    elseif Loop<=10
        app.WhoEditField_2.Value = '안민혁';
    elseif Loop<=15
        app.WhoEditField_2.Value = '최창환';
    end
    break;
else
    app.WhoEditField_2.Value = 'Unidentified';
end

end
```

3.3 팀 멤버 목소리 찾기 고찰

1. 시간영역 검출 고찰



-`xcorr()`를 이용하여 연관된 신호의 peak들 중 기준을 넘는 값의 수를 기준으로 목소리를 검출하려 시도함

-140개 음성파일 모두를 비교 검출해본 결과 특정인의 일부 목소리는 검출되지 않고 특정인이 아닌 사람은 검출되는 부정확함이 있었다.

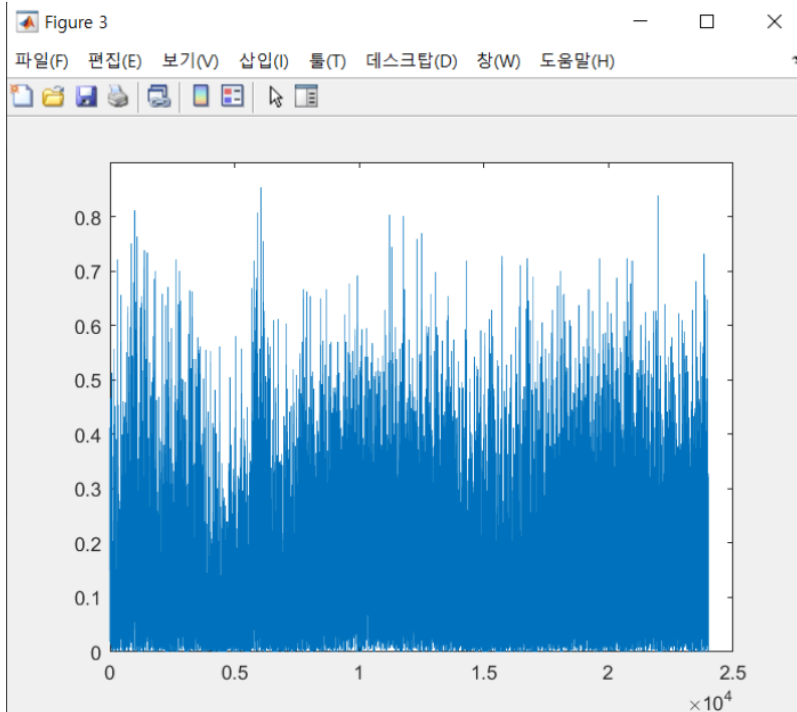
3.3 팀 멤버 목소리 찾기 고찰

1. 시간영역 검출 고찰

- 좋은 찾기 성능을 내는 적합한 척도를 찾는 것이 쉽지 않았다.
- 대신 정확히 일치하는 신호에 대해 자기상관의 결과가 나타나는 것을 포착하고 이를 이용해 음원의 목소리를 매칭시킬 수 있었음.

3.3 팀 멤버 목소리 찾기 고찰

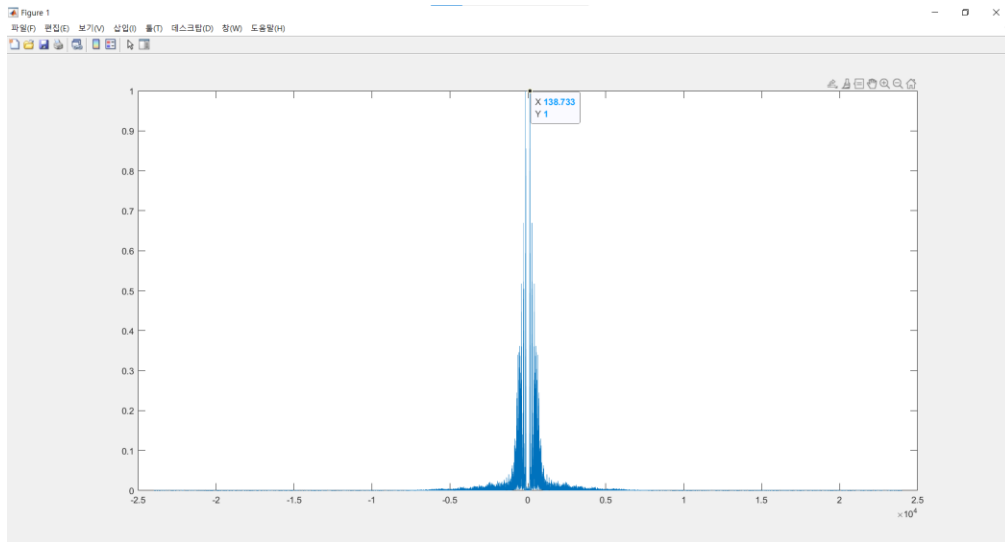
1. 주파수영역 검출 고찰



- 초기에는 mscohere()를 이용해 주파수 영역에서의 일관성이 기준 값 이상인 개수를 세어서 목소리를 검출하려 하였다.
- xcorr()와 마찬가지로 기준 설정에 대해 어려움이 있었다.

3.3 팀 멤버 목소리 찾기 고찰

1. 주파수영역 검출 고찰



- FFT 후 스펙트럼의 피크 값을 나타내는 주파수를 이용해 목소리를 검출하도록 하였다.

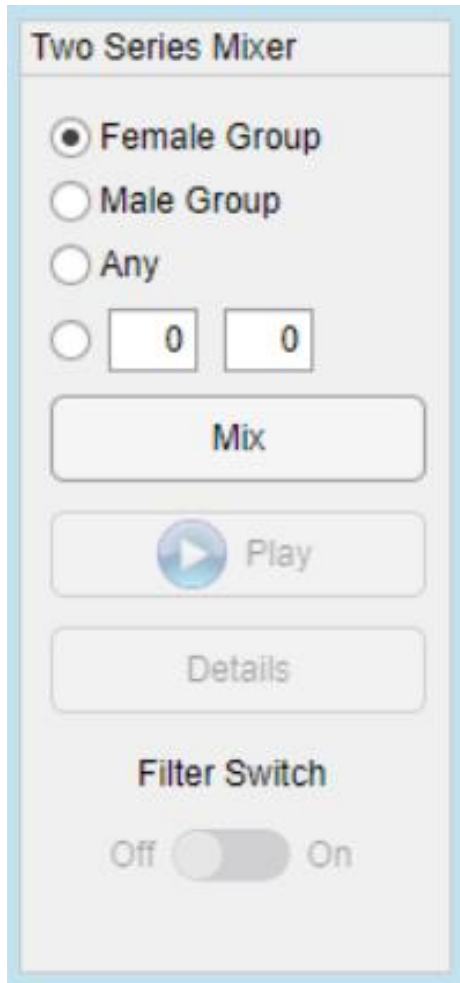
Part 4.

필터설계



4.1 음원 섞기 기능

2개의 음원을 섞습니다.



- 그룹 별 음원 섞기 기능
- 지정한 음원 끼리 섞기 기능
- 재생 기능
- 결과 플로팅 기능
- 필터 스위치 기능

두 신호의 샘플링 레이트가 서로 다른 경우,
큰 샘플링 레이트의 신호를 같은(낮은) 샘플링 레이트로
다운 샘플링(resample)합니다.

```
% Resampling
if Fs1 >= Fs2
    [P, Q] = rat(Fs2/Fs1);
    sig1 = resample(sig1, P, Q);
    Fs1 = Fs2;
else
    [P, Q] = rat(Fs1/Fs2);
    sig2 = resample(sig2, P, Q);
    Fs2 = Fs1;
end
```

```
n1 = 1:length(sig1);
n2 = 1:length(sig2);
n = min(min(n1), min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=sig1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=sig2;
app.MixSig= y1 + y2;
app.MixFs = Fs1;
app.MixN = n;
```

4.1 음원 섞기 기능 코드

2개의 음원을 섞습니다.

```
% Button pushed function: MixButton
function MixButtonPushed(app, event)
    if app.FemaleGroupButton.Value == 1
        RndIndex = randi([1 10],1,2);
        % RndIndex = [ 2 , 7];
        sig1 = app.dataset.Dataupload{1,RndIndex(1)}.signal;
        Fs1 = app.dataset.Dataupload{1,RndIndex(1)}.Fs;

        sig2 = app.dataset.Dataupload{1,RndIndex(2)}.signal;
        Fs2 = app.dataset.Dataupload{1,RndIndex(2)}.Fs;

        % Resampling downsampling
        if Fs1 >= Fs2
            [P, Q] = rat(Fs2/Fs1);
            sig1 = resample(sig1, P, Q);
            Fs1 = Fs2;
        else
            [P, Q] = rat(Fs1/Fs2);
            sig2 = resample(sig2, P, Q);
            Fs2 = Fs1;
        end
        n1 = 1:length(sig1);
        n2 = 1:length(sig2);
        n = min(min(n1), min(n2)):max(max(n1),max(n2));
        y1 = zeros(1,length(n));
        y2=y1;
        y1(find((n>=min(n1))&(n<=max(n1))==1))=sig1;
        y2(find((n>=min(n2))&(n<=max(n2))==1))=sig2;

        app.MixSig= y1 + y2;
        app.MixFs = Fs1;
        app.MixN = n;
    end
end
```

```
elseif app.MaleGroupButton.Value == 1
    RndIndex = randi([11 140],1,2);
    sig1 = app.dataset.Dataupload{1,RndIndex(1)}.signal;
    Fs1 = app.dataset.Dataupload{1,RndIndex(1)}.Fs;
    sig2 = app.dataset.Dataupload{1,RndIndex(2)}.signal;
    Fs2 = app.dataset.Dataupload{1,RndIndex(2)}.Fs;
    if 16<=RndIndex(1) && RndIndex(1)<20
        Fs1 = 8000;
    end
    if 16<=RndIndex(2) && RndIndex(2)<20
        Fs2 = 8000;
    end
    % Resampling
    if Fs1 >= Fs2
        [P, Q] = rat(Fs2/Fs1);
        sig1 = resample(sig1, P, Q);
        Fs1 = Fs2;
    else
        [P, Q] = rat(Fs1/Fs2);
        sig2 = resample(sig2, P, Q);
        Fs2 = Fs1;
    end
    n1 = 1:length(sig1);
    n2 = 1:length(sig2);
    n = min(min(n1), min(n2)):max(max(n1),max(n2));
    y1 = zeros(1,length(n));
    y2=y1;
    y1(find((n>=min(n1))&(n<=max(n1))==1))=sig1;
    y2(find((n>=min(n2))&(n<=max(n2))==1))=sig2;
    app.MixSig= y1 + y2;
    app.MixFs = Fs1;
    app.MixN = n;
end
```

4.1 음원 섞기 기능 코드

2개의 음원을 섞습니다.

```
elseif app.AnyButton.Value == 1
    RndIndex = randi([1 140],1,2);
    sig1 = app.dataset.Dataupload{1,RndIndex(1)}.signal;
    Fs1 = app.dataset.Dataupload{1,RndIndex(1)}.Fs;
    sig2 = app.dataset.Dataupload{1,RndIndex(2)}.signal;
    Fs2 = app.dataset.Dataupload{1,RndIndex(2)}.Fs;
    if 16<=RndIndex(1) && RndIndex(1)<20
        Fs1 = 8000;
    end
    if 16<=RndIndex(2) && RndIndex(2)<20
        Fs2 = 8000;
    end
    % Resampling
    if Fs1 >= Fs2
        [P, Q] = rat(Fs2/Fs1);
        sig1 = resample(sig1, P, Q);
        Fs1 = Fs2;
    else
        [P, Q] = rat(Fs1/Fs2);
        sig2 = resample(sig2, P, Q);
        Fs2 = Fs1;
    end
    n1 = 1:length(sig1);
    n2 = 1:length(sig2);
    n = min(min(n1), min(n2)):max(max(n1),max(n2));
    y1 = zeros(1,length(n));
    y2=y1;
    y1(find((n>=min(n1))&(n<=max(n1))==1))=sig1;
    y2(find((n>=min(n2))&(n<=max(n2))==1))=sig2;
    app.MixSig= y1 + y2;
    app.MixFs = Fs1;
    app.MixN = n;
```

```
elseif app.Manual.Value == 1

    try
        RndIndex(1) = app.signal1.Value;
        RndIndex(2) = app.signal2.Value;
        sig1 = app.dataset.Dataupload{1,RndIndex(1)}.signal;
        Fs1 = app.dataset.Dataupload{1,RndIndex(1)}.Fs;
        sig2 = app.dataset.Dataupload{1,RndIndex(2)}.signal;
        Fs2 = app.dataset.Dataupload{1,RndIndex(2)}.Fs;

        catch
            RndIndex(1) = 1;
            RndIndex(2) = 1;
            app.signal1.Value = 1;
            app.signal2.Value = 1;
            sig1 = app.dataset.Dataupload{1,RndIndex(1)}.signal;
            Fs1 = app.dataset.Dataupload{1,RndIndex(1)}.Fs;
            sig2 = app.dataset.Dataupload{1,RndIndex(2)}.signal;
            Fs2 = app.dataset.Dataupload{1,RndIndex(2)}.Fs;
        end

        if 16<=RndIndex(1) && RndIndex(1)<20
            Fs1 = 8000;
        end
        if 16<=RndIndex(2) && RndIndex(2)<20
            Fs2 = 8000;
        end
        % Resampling
        if Fs1 >= Fs2
            [P, Q] = rat(Fs2/Fs1);
            sig1 = resample(sig1, P, Q);
            Fs1 = Fs2;
```

4.1 음원 섞기 기능 코드

2개의 음원을 섞습니다.

```
else
    [P, Q] = rat(Fs1/Fs2);
    sig2 = resample(sig2, P, Q);
    Fs2 = Fs1;
end
n1 = 1:length(sig1);
n2 = 1:length(sig2);
n = min(min(n1), min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=sig1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=sig2;
app.MixSig= y1 + y2;
app.MixFs = Fs1;
app.MixN = n;
end
app.signal1.Value = RndIndex(1);
app.signal2.Value = RndIndex(2);
app.PlayButton_2.Enable = "on";
app.DetailsButton.Enable = "on";
app.GenerateButton.Enable = "on";
end
```

4.2 버터워스 BandStop 필터 기능

Filter 3번 적용 가능

IIR BandStop Filter I

CutOffFreq.

150

CutOffFreq.2

270

Order

8

Generate

IIR BandStop Filter III

CutOffFreq.

550

CutOffFreq.2

2000

Order

8

Generate

여러 구간의 이득을 효과적으로 줄이기 위해
3번의 필터를 사용할 수 있도록 구현했습니다.
각각의 필터는 BandStop Filter의 차단 주파수와
차수를 직접 정할 수 있습니다.

Mix된 음원에 적용하여 원하는 효과(특정 목소리 제거)를 얻고자 했습니다.

IIR BandStop Filter II

CutOffFreq.

350

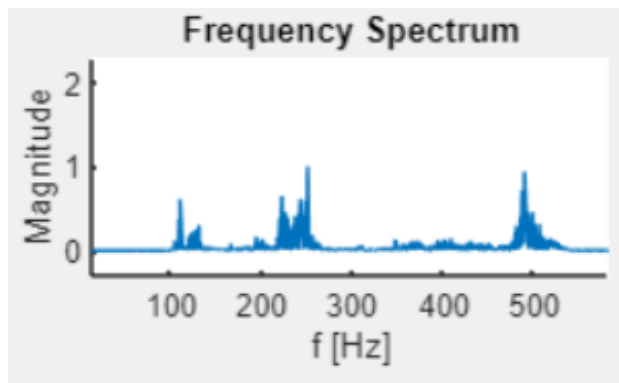
CutOffFreq.2

500

Order

8

Generate



```
CF1 = app.CutOffFreqEditField.Value;
CF2 = app.CutOffFreq2EditField.Value;
Order = app.OrderEditField.Value;

Filt = designfilt('bandstopiir','FilterOrder',Order, ...
    'HalfPowerFrequency1',CF1,'HalfPowerFrequency2',CF2, ...
    'SampleRate',app.MixFs,'DesignMethod','butter');

app.Filtered = filter(Filt,app.MixSig);
```

4.2 버터워스 BandStop 필터 기능 코드

Filter 3번 적용 가능

```
% Button pushed function: GenerateButton
function GenerateButtonPushed(app, event)
    CF1 = app.CutoffFreqEditField.Value;
    CF2 = app.CutoffFreq2EditField.Value;
    Order = app.OrderEditField.Value;

    Filt = designfilt('bandstopiir','FilterOrder',Order, ...
        'HalfPowerFrequency1',CF1,'HalfPowerFrequency2',CF2, ...
        'SampleRate',app.MixFs);

    app.Filtered = filter(Filt,app.MixSig);
    fvtool(app.Filtered);
    app.Switch.Enable = "on";
end

% Button pushed function: GenerateButton_2
function GenerateButton_2Pushed(app, event)
    CF1 = app.CutoffFreqEditField_2.Value;
    CF2 = app.CutoffFreq2EditField_2.Value;
    Order = app.OrderEditField_2.Value;

    Filt = designfilt('bandstopiir','FilterOrder',Order, ...
        'HalfPowerFrequency1',CF1,'HalfPowerFrequency2',CF2, ...
        'SampleRate',app.MixFs);

    app.Filtered = filter(Filt,app.Filtered);
    fvtool(app.Filtered);
    app.Switch_2.Enable = "on";
end
```

```
% Button pushed function: GenerateButton_3
function GenerateButton_3Pushed(app, event)
    CF1 = app.CutoffFreqEditField_3.Value;
    CF2 = app.CutoffFreq2EditField_3.Value;
    Order = app.OrderEditField_3.Value;

    Filt = designfilt('bandstopiir','FilterOrder',Order, ...
        'HalfPowerFrequency1',CF1,'HalfPowerFrequency2',CF2, ...
        'SampleRate',app.MixFs);

    app.Filtered = filter(Filt,app.Filtered);
    fvtool(app.Filtered);
    app.Switch_3.Enable = "on";
end
```


4.3 버터워스 BandStop 필터 기능 고찰

Filter 3번 적용 가능

•필터링

- 처음에는 FIR 필터로 구현하고자 했으나 처리시간 및 메모리 용량 관점에서 최적화하고자 IIR 필터로 구현
- designfilt() 함수를 통해 IIR 필터 구현
- 필터는 간편하게 구현했지만 GUI에 필터 플로팅 어려움을 겪었다.
- fvtool을 이용해 플로팅 했다.

Part 5.

데모



감사합니다

9조 구해조 일동