

Ahsanullah University of Science and Technology

Department of Electrical and Electronic Engineering

PROJECT REPORT

LAB GROUP-4

EEE 3110 [NUMERICAL TECHNIQUE LAB]

MUSICAL PITCH GENERATOR

SUBMITTED BY

TAMZID HASAN (20210105089)

HAMIM INTESAR (20210105094)

NAFIUR RAHMAN (20210105100)

JAWADUL ALAM TRIPTA (20210205188)

SUBMITTED TO:

SHAMEEM HASAN

TASIN INTISAR

Introduction:

We have developed an application called "Musical Pitch Generator" which generates musical pitches through the multiplication of two distinct sound signals. Each signal possesses varying frequencies and amplitudes. By combining these signals, a new pitch sound signal is produced.

Basic principle of the Application:

One of them transmits the signal (sine wave) from the other. It works with two modulating oscillators.

One of these two oscillators is a modulator and the other is a carrier.

- Oscillator: Device that produces waveforms.
- Signal 1 Frequency (fc): Frequency of signal-1
- Signal 2 Frequency (fm): Frequency of the emitted signal 2.

Basic input Signal equation: $e = A \sin(\alpha t + D \sin(\beta t))$

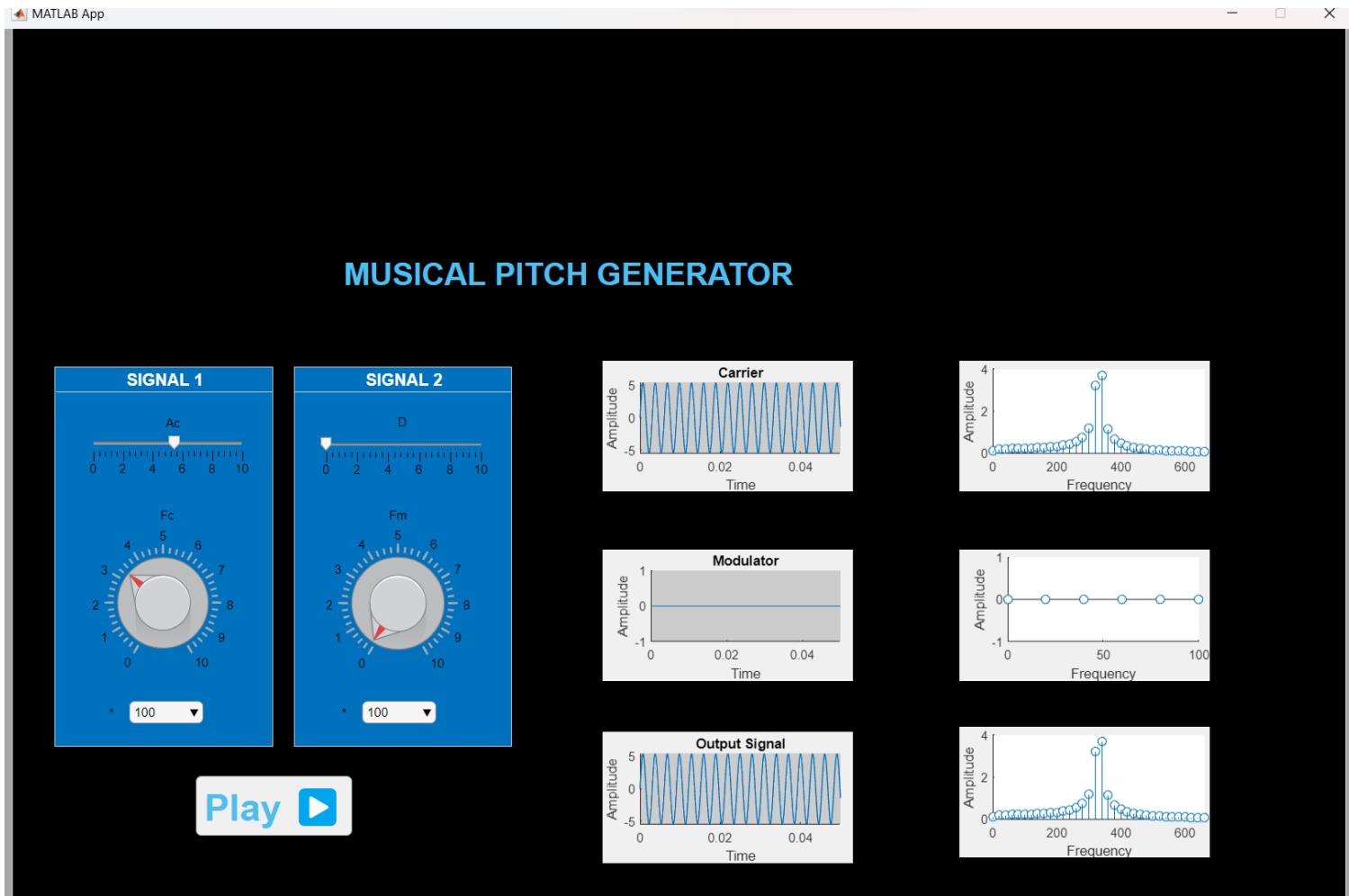
e is the instantaneous amplitude of the modulated carrier. A is the peak amplitude.

α and β are the respective signal 1 and signal frequencies (fc and fm).

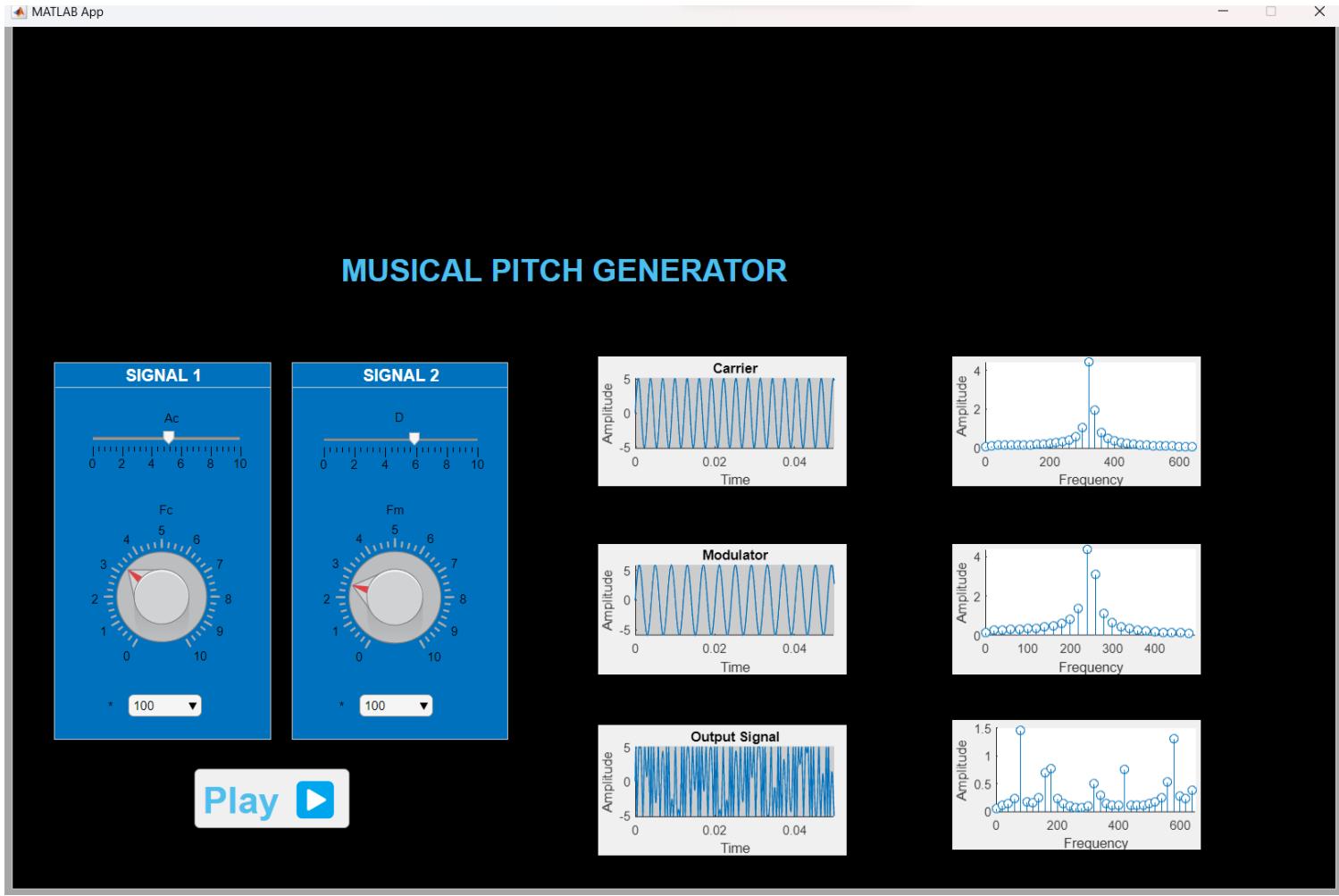
D is a scale of the amount of frequency deviation .

Function of the Application :

Here we used two sin wave signals. Here using the knob we can set the frequency of both signals and also set the amplitude by using corresponding signal amplitude slider which we can see in the axes on right side.



we can also see the frequency modulation output in another graph axes layout. Where both signals modulate and the output is shown in graph. We also see the Fourier transform of the 3 signals just the right of the corresponding signal graph.



After setting the amplitude and frequency of both signals when we press the play button we can get the sound output of Fm of both signals.

Uses of Frequency Modulation: creating sounds that are difficult to achieve with subtractive synthesizers—sounds such as bell timbres, metallic tones, and the tine tones of electric pianos.

Code of the Application:

```
classdef fm_synthesis < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure           matlab.ui.Figure
        TabGroup           matlab.ui.container.TabGroup
        Tab_fm             matlab.ui.container.Tab
        UIAxes_FftResult  matlab.ui.control.UIAxes
        UIAxes_FftM        matlab.ui.control.UIAxes
        UIAxes_FftC        matlab.ui.control.UIAxes
        UIAxes_Result      matlab.ui.control.UIAxes
        UIAxes_TM          matlab.ui.control.UIAxes
        UIAxes_TC          matlab.ui.control.UIAxes
        Panel_carrier     matlab.ui.container.Panel
        AcSliderLabel      matlab.ui.control.Label
        Slider_carrier_amp matlab.ui.control.Slider
        FcKnobLabel        matlab.ui.control.Label
        Knob_carrier_freq  matlab.ui.control.Knob
        DropDown_2Label    matlab.ui.control.Label
        DropDown_carrier_mul matlab.ui.control.DropDown
        Panel_mod          matlab.ui.container.Panel
        Slider_mod_amp    matlab.ui.control.Slider
        DLabel              matlab.ui.control.Label
        FmKnob_2Label      matlab.ui.control.Label
        Knob_mod_freq       matlab.ui.control.Knob
        Label               matlab.ui.control.Label
        DropDown_mod_mul   matlab.ui.control.DropDown
        Button_play_fmtab  matlab.ui.control.Button
        Title               matlab.ui.control.Label
    end

    properties (Access = public)
        carrier_amp_prev;
        carrier_freq_prev;
        mod_amp_prev;
        mod_freq_prev;

        carrier_amp;           %A
        carrier_freq;          %C
    end
```

```
mod_amp; %D  
mod_freq;
```

```
end
```

```
methods (Access = public)
```

```
function plotFnc(app,A,C,D,M)
```

```
T=0.000001;
```

```
Fs=1/T;
```

```
t = 0:T:.05; %seconds
```

```
y=A*sin(2*pi*C*t+D*sin(2*pi*M*t));
```

```
c=A*sin(2*pi*C*t);
```

```
m=D*sin(2*pi*M*t);
```

```
plot(app.UIAxes_TC,t,c);
```

```
l=length(c);
```

```
Y=fft(c);
```

```
P2 = abs(Y/l);
```

```
P1 = P2(1:(l/2+1));
```

```
P1(2:end-1) = 2*P1(2:end-1);
```

```
f = Fs*(0:(l/2))/l;
```

```
stem(app.UIAxes_FftC,f,P1);
```

```
app.UIAxes_FftC.XLim = [0 max(2*C,100)];
```

```
plot(app.UIAxes_TM,t,m);
```

```
l=length(m);
```

```
Y=fft(m);
```

```
P2 = abs(Y/l);
```

```
P1 = P2(1:l/2+1);
```

```
P1(2:end-1) = 2*P1(2:end-1);
```

```
f = Fs*(0:(l/2))/l;
```

```
stem(app.UIAxes_FftM,f,P1);
```

```
app.UIAxes_FftM.XLim = [0 max(2*M,100)];
```

```
plot(app.UIAxes_Result,t,y);
```

```

l=length(y);
Y=fft(y);
P2 = abs(Y/l);
P1 = P2(1:l/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(l/2))/l;

stem(app.UIAxes_FftResult,f,P1);
app.UIAxes_FftResult.XLim = [0 max(max(2*C,2*M),100)];

end
end

```

methods (Access = private)

```

% Code that executes after component creation
function startupFcn(app)
    %A=2;
    %C = 210;
    %D = 10;
    %M = 70;
%plotFnc(app,A,C,D,M);
    end

```

```

% Button pushed function: Button_play_fmtab
function Button_play_fmtabPushed(app, event)

```

```

T=0.00001;
Fs=1/T;
t = 0:T:1;
%Fs = 44100;
% T = 1/Fs;
% dur = 1.0;
%t = 0:T:dur;
A = app.Slider_carrier_amp.Value; %A
C=app.Knob_carrier_freq.Value*app.DropDown_carrier_mul.Value; %C
D=app.Slider_mod_amp.Value; %D
M= app.Knob_mod_freq.Value*app.DropDown_mod_mul.Value;

```

```

y=A*sin(2*pi*C*t+D*sin(2*pi*M*t));
sound(y,Fs);

end

% Value changed function: DropDown_carrier_mul,
% DropDown_mod_mul, Knob_carrier_freq, Knob_mod_freq,
% Slider_carrier_amp, Slider_mod_amp
function valueChanged(app, event)
    A = app.Slider_carrier_amp.Value; %A
    C=app.Knob_carrier_freq.Value*app.DropDown_carrier_mul.Value; %C
    D=app.Slider_mod_amp.Value; %D
    M= app.Knob_mod_freq.Value*app.DropDown_mod_mul.Value;

    plotFnc(app,A,C,D,M);
end
end

% App initialization and construction
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure
    app.UIFigure = uifigure;
    app.UIFigure.AutoResizeChildren = 'off';
    app.UIFigure.Color = [0.651 0.651 0.651];
    app.UIFigure.Position = [0 0 1299 995];
    app.UIFigure.Name = 'MATLAB App';
    app.UIFigure.Resize = 'off';

    % Create TabGroup
    app.TabGroup = uitabgroup(app.UIFigure);
    app.TabGroup.AutoResizeChildren = 'off';
    app.TabGroup.Position = [9 10 1283 978];

    % Create Tab_fm
    app.Tab_fm = uitab(app.TabGroup);

```

```

app.Tab_fm.AutoScaleChildren = 'off';
app.Tab_fm.Title = 'Frequency Modulation Synthesis';
app.Tab_fm.BackgroundColor = [0 0 0];
app.Tab_fm.ForegroundColor = [0 0.4471 0.7412];

% Create UIAxes_FftResult
app.UIAxes_FftResult = uiaxes(app.Tab_fm);
xlabel(app.UIAxes_FftResult, 'Frequency')
ylabel(app.UIAxes_FftResult, 'Amplitude')
zlabel(app.UIAxes_FftResult, 'Z')
app.UIAxes_FftResult.Position = [911 38 241 126];

% Create UIAxes_FftM
app.UIAxes_FftM = uiaxes(app.Tab_fm);
xlabel(app.UIAxes_FftM, 'Frequency')
ylabel(app.UIAxes_FftM, 'Amplitude')
zlabel(app.UIAxes_FftM, 'Z')
app.UIAxes_FftM.Position = [911 208 241 126];

% Create UIAxes_FftC
app.UIAxes_FftC = uiaxes(app.Tab_fm);
xlabel(app.UIAxes_FftC, 'Frequency')
ylabel(app.UIAxes_FftC, 'Amplitude')
zlabel(app.UIAxes_FftC, 'Z')
app.UIAxes_FftC.Position = [911 390 241 126];

% Create UIAxes_Result
app.UIAxes_Result = uiaxes(app.Tab_fm);
title(app.UIAxes_Result, ' FM output')
xlabel(app.UIAxes_Result, 'Time')
ylabel(app.UIAxes_Result, 'Amplitude')
zlabel(app.UIAxes_Result, 'Z')
app.UIAxes_Result.Color = [0.8 0.8 0.8];
app.UIAxes_Result.Position = [568 33 241 126];

% Create UIAxes_TM
app.UIAxes_TM = uiaxes(app.Tab_fm);
title(app.UIAxes_TM, 'Modulator')
xlabel(app.UIAxes_TM, 'Time')
ylabel(app.UIAxes_TM, 'Amplitude')
zlabel(app.UIAxes_TM, 'Z')
app.UIAxes_TM.Color = [0.8 0.8 0.8];
app.UIAxes_TM.Position = [568 208 241 126];

```

```

% Create UIAxes_TC
app.UIAxes_TC = uiaxes(app.Tab_fm);
title(app.UIAxes_TC, 'Carrier')
xlabel(app.UIAxes_TC, 'Time')
ylabel(app.UIAxes_TC, 'Amplitude')
zlabel(app.UIAxes_TC, 'Z')
app.UIAxes_TC.Color = [0.8 0.8 0.8];
app.UIAxes_TC.Position = [568 390 241 126];

% Create Panel_carrier
app.Panel_carrier = uipanel(app.Tab_fm);
app.Panel_carrier.AutoResizeChildren = 'off';
app.Panel_carrier.ForegroundColor = [1 1 1];
app.Panel_carrier.TitlePosition = 'centertop';
app.Panel_carrier.Title = 'SIGNAL 1';
app.Panel_carrier.BackgroundColor = [0 0.4471 0.7412];
app.Panel_carrier.FontName = 'Bookman';
app.Panel_carrier.FontWeight = 'bold';
app.Panel_carrier.FontSize = 16;
app.Panel_carrier.Position = [41 145 210 365];

% Create AcSliderLabel
app.AcSliderLabel = uilabel(app.Panel_carrier);
app.AcSliderLabel.HorizontalAlignment = 'right';
app.AcSliderLabel.Position = [96 301 25 22];
app.AcSliderLabel.Text = 'Ac';

% Create Slider_carrier_amp
app.Slider_carrier_amp = uislider(app.Panel_carrier);
app.Slider_carrier_amp.Limits = [0 10];
app.Slider_carrier_amp.ValueChangedFcn      =      createCallbackFcn(app,
@valueChanged, true);
app.Slider_carrier_amp.Position = [37 290 143 3];

% Create FcKnobLabel
app.FcKnobLabel = uilabel(app.Panel_carrier);
app.FcKnobLabel.HorizontalAlignment = 'center';
app.FcKnobLabel.Position = [96 212 25 22];
app.FcKnobLabel.Text = 'Fc';

% Create Knob_carrier_freq
app.Knob_carrier_freq = uiknob(app.Panel_carrier, 'continuous');

```

```

app.Knob_carrier_freq.Limits = [0 10];
app.Knob_carrier_freq.ValueChangedFcn      =      createCallbackFcn(app,
@valueChanged, true);
app.Knob_carrier_freq.Position = [60 94 88 88];

% Create DropDown_2Label
app.DropDown_2Label = uilabel(app.Panel_carrier);
app.DropDown_2Label.HorizontalAlignment = 'right';
app.DropDown_2Label.FontName = 'Bookman';
app.DropDown_2Label.Position = [32 22 25 22];
app.DropDown_2Label.Text = '*';

% Create DropDown_carrier_mul
app.DropDown_carrier_mul = uidropdown(app.Panel_carrier);
app.DropDown_carrier_mul.Items = {'10', '100', '1000'};
app.DropDown_carrier_mul.ItemsData = [10 100 1000];
app.DropDown_carrier_mul.ValueChangedFcn    =    createCallbackFcn(app,
@valueChanged, true);
app.DropDown_carrier_mul.FontName = 'Bookman';
app.DropDown_carrier_mul.Position = [72 22 71 22];
app.DropDown_carrier_mul.Value = 10;

% Create Panel_mod
app.Panel_mod = uipanel(app.Tab_fm);
app.Panel_mod.AutoResizeChildren = 'off';
app.Panel_mod.ForegroundColor = [1 1 1];
app.Panel_mod.TitlePosition = 'centertop';
app.Panel_mod.Title = 'SIGNAL 2';
app.Panel_mod.BackgroundColor = [0 0.4471 0.7412];
app.Panel_mod.FontName = 'Bookman';
app.Panel_mod.FontWeight = 'bold';
app.Panel_mod.FontSize = 16;
app.Panel_mod.Position = [271 145 210 365];

% Create Slider_mod_amp
app.Slider_mod_amp = uislider(app.Panel_mod);
app.Slider_mod_amp.Limits = [0 10];
app.Slider_mod_amp.ValueChangedFcn      =      createCallbackFcn(app,
@valueChanged, true);
app.Slider_mod_amp.Position = [31 289 149 3];

% Create DLabel
app.DLabel = uilabel(app.Panel_mod);

```

```

app.DLabel.HorizontalAlignment = 'right';
app.DLabel.Position = [84 302 25 22];
app.DLabel.Text = 'D';

% Create FmKnob_2Label
app.FmKnob_2Label = uilabel(app.Panel_mod);
app.FmKnob_2Label.HorizontalAlignment = 'center';
app.FmKnob_2Label.Position = [88 212 25 22];
app.FmKnob_2Label.Text = 'Fm';

% Create Knob_mod_freq
app.Knob_mod_freq = uiknob(app.Panel_mod, 'continuous');
app.Knob_mod_freq.Limits = [0 10];
app.Knob_mod_freq.ValueChangedFcn      =      createCallbackFcn(app,
@valueChanged, true);
app.Knob_mod_freq.Position = [55 93 90 90];

% Create Label
app.Label = uilabel(app.Panel_mod);
app.Label.HorizontalAlignment = 'right';
app.Label.FontName = 'Bookman';
app.Label.Position = [26 22 25 22];
app.Label.Text = '*';

% Create DropDown_mod_mul
app.DropDown_mod_mul = uidropdown(app.Panel_mod);
app.DropDown_mod_mul.Items = {'10', '100', '1000'};
app.DropDown_mod_mul.ItemsData = [10 100 1000];
app.DropDown_mod_mul.ValueChangedFcn      =      createCallbackFcn(app,
@valueChanged, true);
app.DropDown_mod_mul.FontName = 'Bookman';
app.DropDown_mod_mul.Position = [66 22 71 22];
app.DropDown_mod_mul.Value = 10;

% Create Button_play_fmtab
app.Button_play_fmtab = uibutton(app.Tab_fm, 'push');
app.Button_play_fmtab.ButtonPushedFcn      =      createCallbackFcn(app,
@Button_play_fmtabPushed, true);
app.Button_play_fmtab.BackgroundColor = [0.9412 0.9412 0.9412];
app.Button_play_fmtab.FontName = 'Bookman';
app.Button_play_fmtab.FontSize = 36;
app.Button_play_fmtab.FontWeight = 'bold';
app.Button_play_fmtab.FontColor = [0.302 0.749 0.9294];

```

```
app.Button_play_fmtab.Position = [177 59 150 58];
app.Button_play_fmtab.Text = 'Play ▶';

% Create Title
app.Title = uilabel(app.Tab_fm);
app.Title.HorizontalAlignment = 'center';
app.Title.FontName = 'Bookman';
app.Title.FontSize = 30;
app.Title.FontWeight = 'bold';
app.Title.FontColor = [0.302 0.749 0.9294];
app.Title.Position = [280 564 511 71];
app.Title.Text = 'MUSICAL PITCH GENERATOR';
end
end
```

methods (Access = public)

```
% Construct app
function app = fm_synthesis

% Create and configure components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)
```

```
if nargout == 0
    clear app
end
end
```

```
% Code that executes before app deletion
function delete(app)
```

```
    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```

Conclusion:

FM synthesis, popularized by the DX7, manipulates sound by "jiggling" the pitch of a steady tone with another wobbly signal. This interaction generates rich and evolving timbres like bells, electric pianos, and pads, making it a versatile tool for electronic music production. By adjusting key parameters and integrating with other methods, FM synthesis continues to offer exciting sonic possibilities for musicians and sound designers.