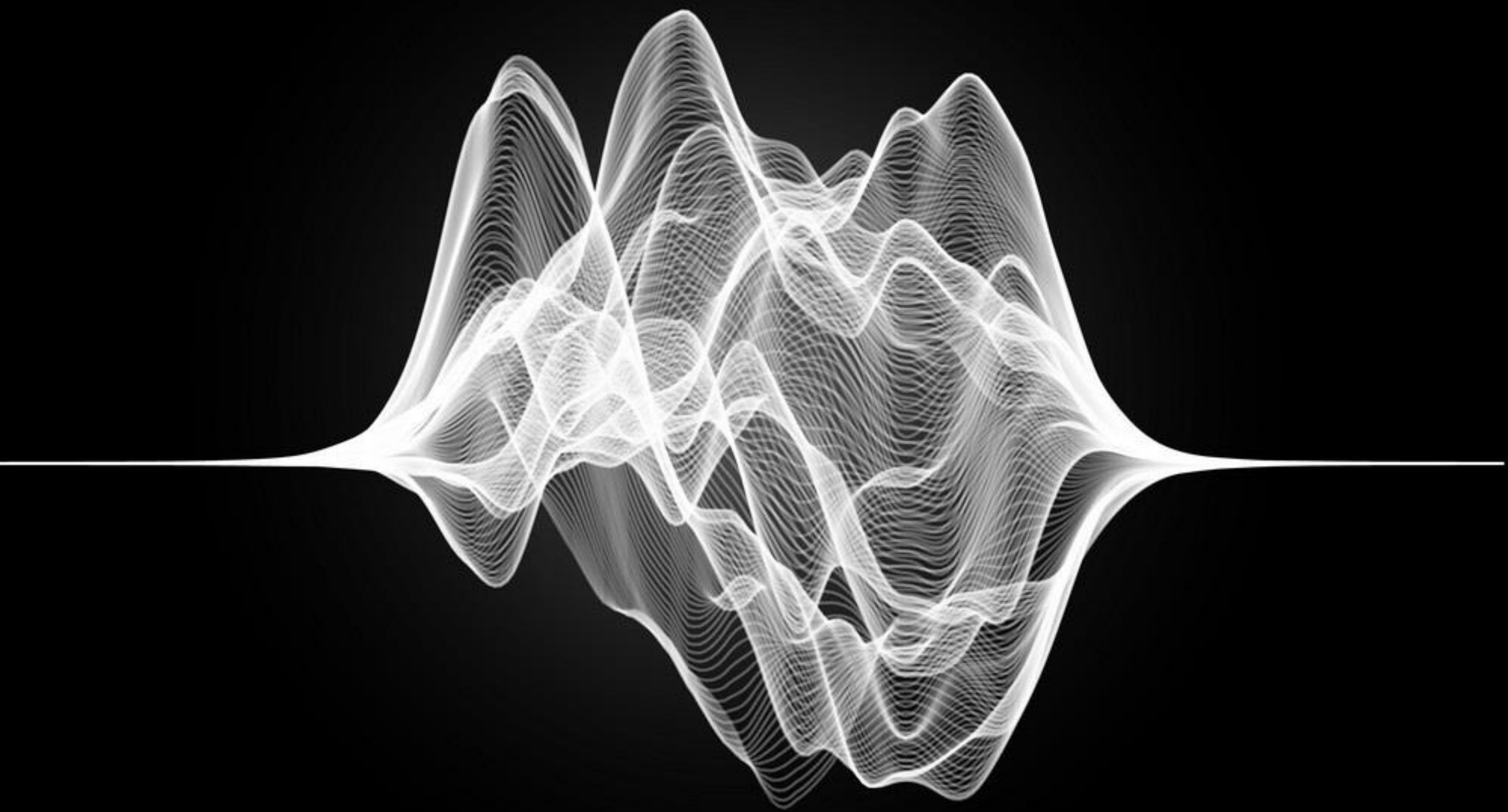


Signal Modulation-Demodulation

Using Matlab

Bangladesh University of Engineering & Technology

Department of Electrical & Electronic Engineering

Course Number: EEE-212

Course title: Numerical methods laboratory

Project number: 08

Submitted to:

Dr. Hafiz Imtiaz

Suzit Hasan Nayem

Submitted by:

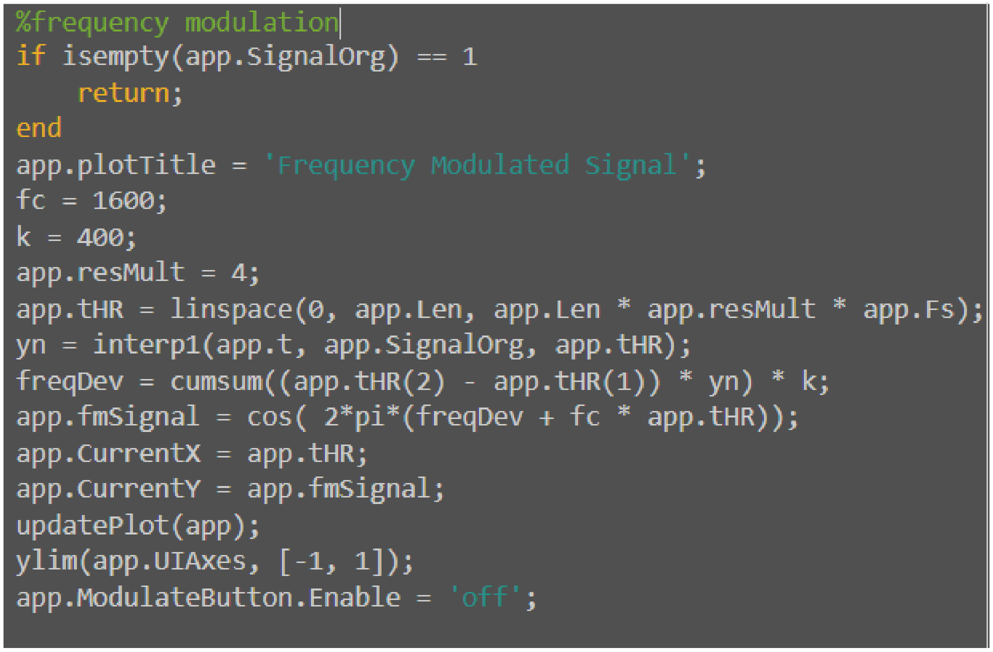
Name: Razi Mudassir

ID: 1806111

Objective: The objective of the project was to implement of analog modulation and demodulation scheme using matlab.

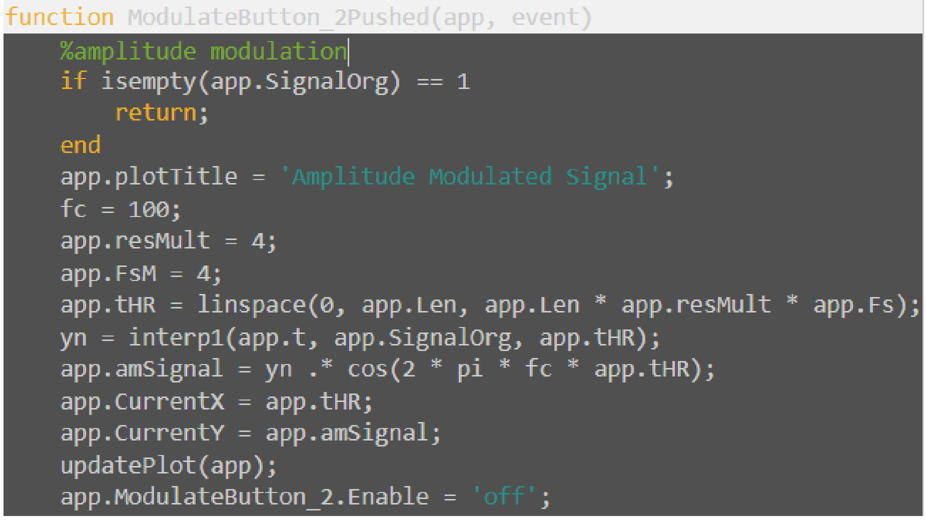
Code:

Frequency modulation uses the following code:



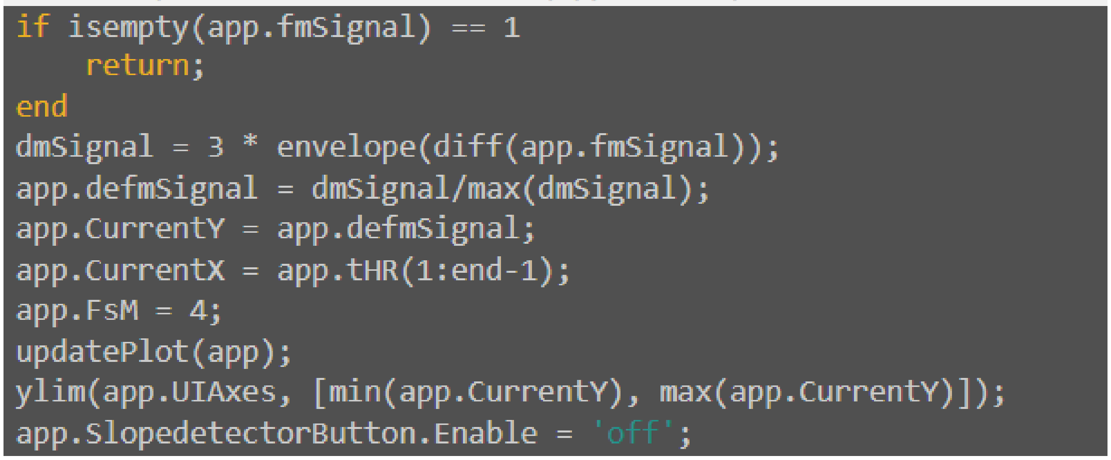
```
%frequency modulation
if isempty(app.SignalOrg) == 1
    return;
end
app.plotTitle = 'Frequency Modulated Signal';
fc = 1600;
k = 400;
app.resMult = 4;
app.tHR = linspace(0, app.Len, app.Len * app.resMult * app.Fs);
yn = interp1(app.t, app.SignalOrg, app.tHR);
freqDev = cumsum((app.tHR(2) - app.tHR(1)) * yn) * k;
app.fmSignal = cos( 2*pi*(freqDev + fc * app.tHR));
app.CurrentX = app.tHR;
app.CurrentY = app.fmSignal;
updatePlot(app);
ylim(app.UIAxes, [-1, 1]);
app.ModulateButton.Enable = 'off';
```

Amplitude modulation is done using the code below:



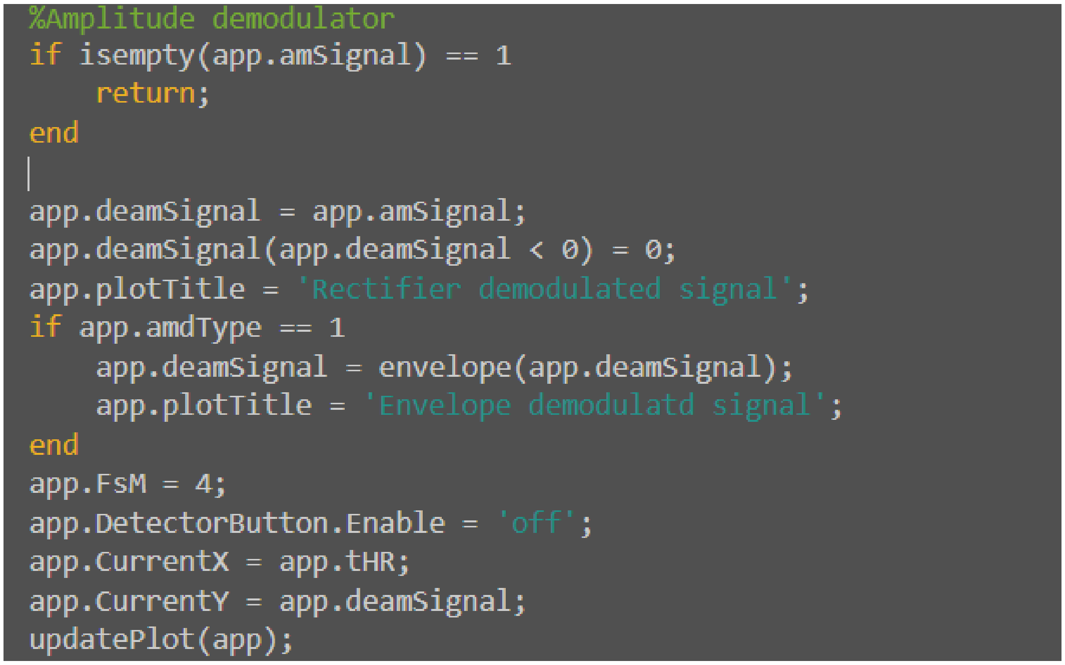
```
function ModulateButton_2Pushed(app, event)
    %amplitude modulation
    if isempty(app.SignalOrg) == 1
        return;
    end
    app.plotTitle = 'Amplitude Modulated Signal';
    fc = 100;
    app.resMult = 4;
    app.FsM = 4;
    app.tHR = linspace(0, app.Len, app.Len * app.resMult * app.Fs);
    yn = interp1(app.t, app.SignalOrg, app.tHR);
    app.amSignal = yn .* cos(2 * pi * fc * app.tHR);
    app.CurrentX = app.tHR;
    app.CurrentY = app.amSignal;
    updatePlot(app);
    app.ModulateButton_2.Enable = 'off';
```

Frequency demodulation using slop detector is done using the code below:



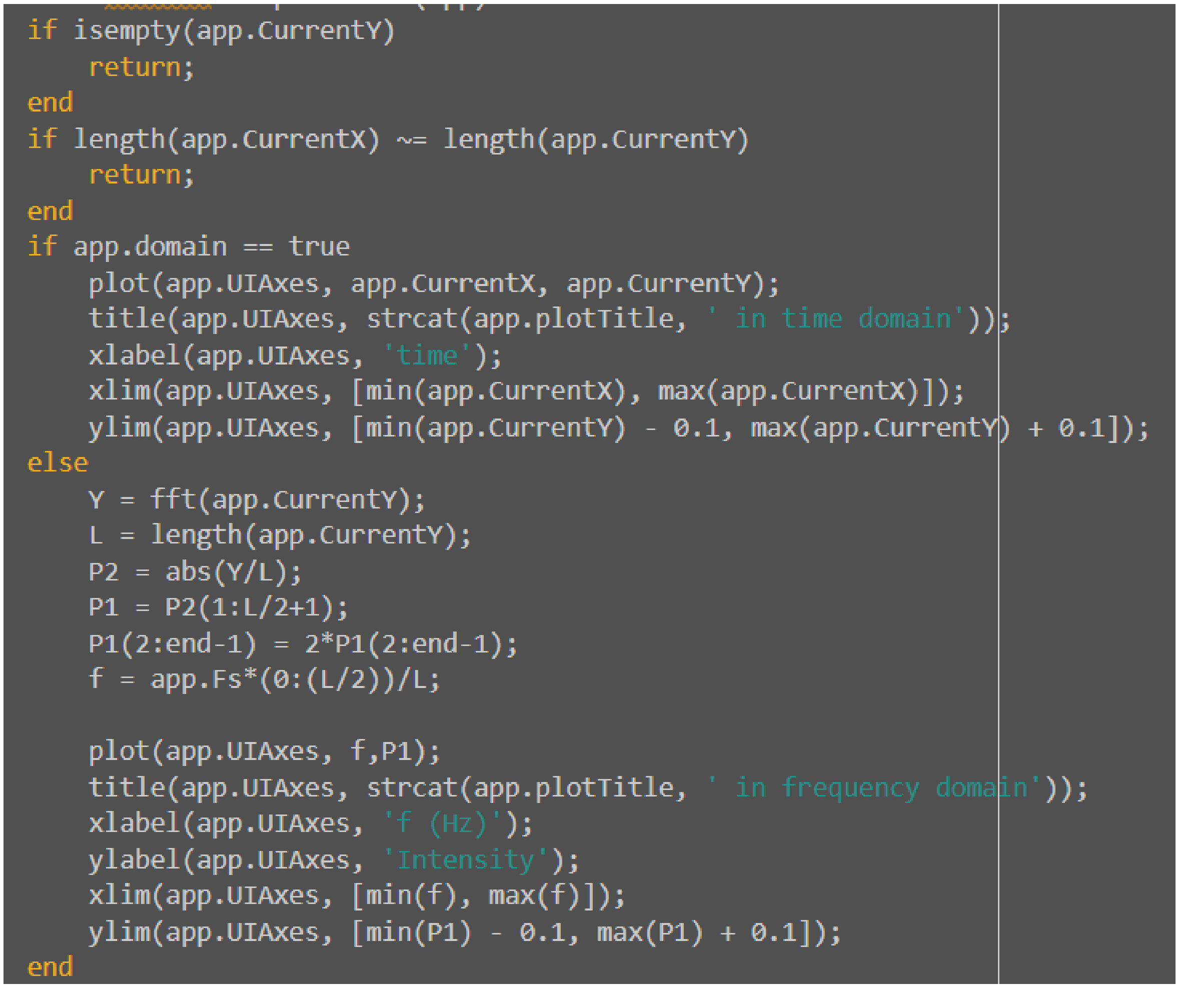
```
if isempty(app.fmSignal) == 1
    return;
end
dmSignal = 3 * envelope(diff(app.fmSignal));
app.defmSignal = dmSignal/max(dmSignal);
app.CurrentY = app.defmSignal;
app.CurrentX = app.tHR(1:end-1);
app.FsM = 4;
updatePlot(app);
ylim(app.UIAxes, [min(app.CurrentY), max(app.CurrentY)]);
app.SlopedetectorButton.Enable = 'off';
```

Amplitude demodulation is done using two different methods. Rectifier detector & envelope detector. Their code is given below:



```
%Amplitude demodulator
if isempty(app.amSignal) == 1
    return;
end

app.deamSignal = app.amSignal;
app.deamSignal(app.deamSignal < 0) = 0;
app.plotTitle = 'Rectifier demodulated signal';
if app.amdType == 1
    app.deamSignal = envelope(app.deamSignal);
    app.plotTitle = 'Envelope demodulatd signal';
end
app.FsM = 4;
app.DetectorButton.Enable = 'off';
app.CurrentX = app.tHR;
app.CurrentY = app.deamSignal;
updatePlot(app);
```

Signals are converted from time domain to frequency domain and vise versa using the function below:

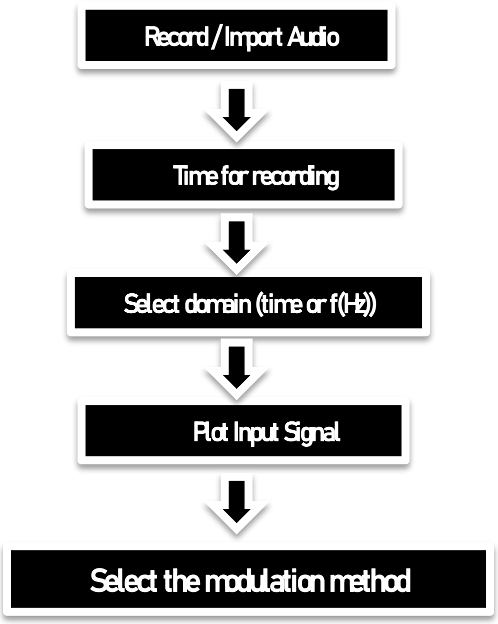


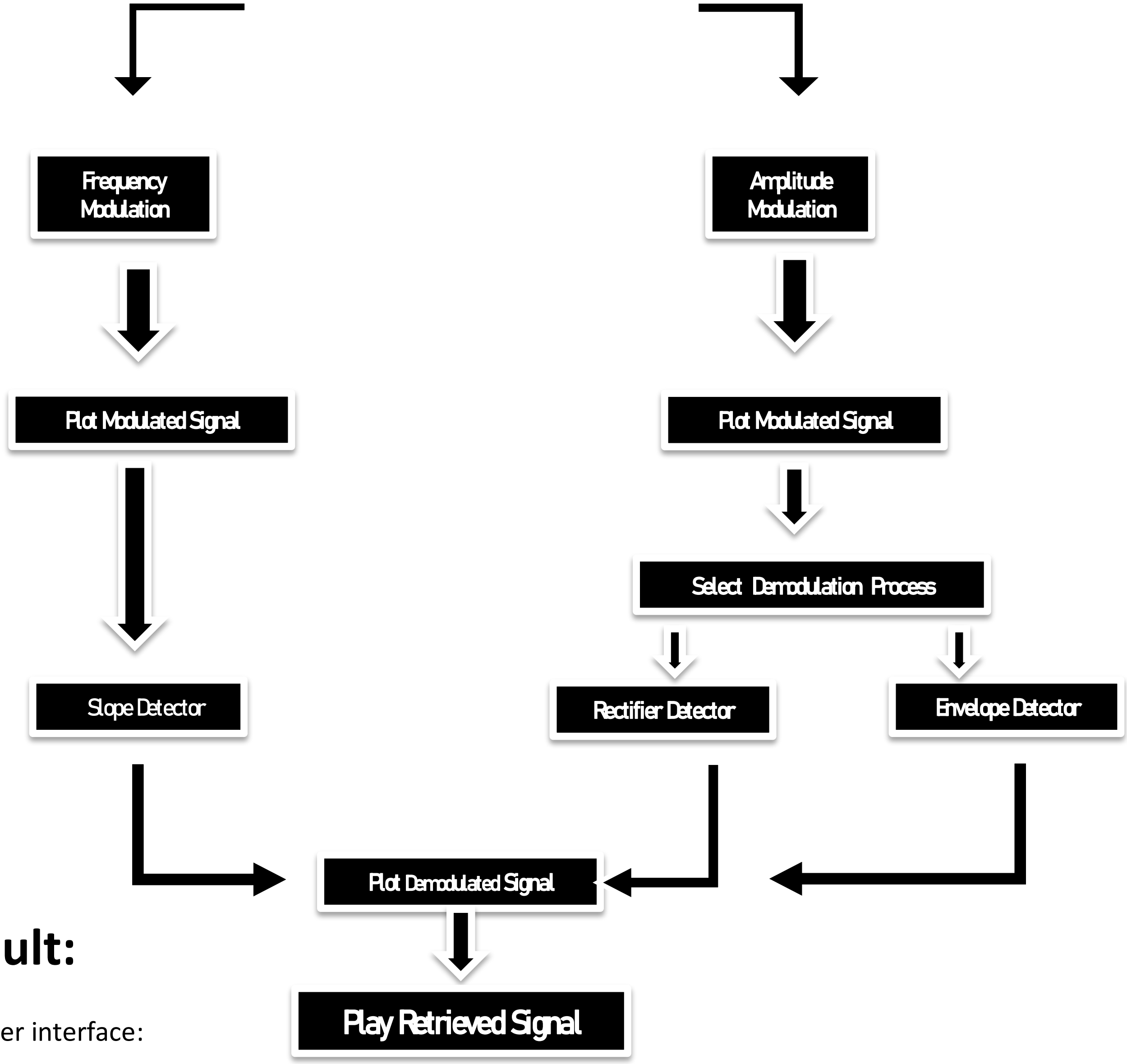
```
if isempty(app.CurrentY)
    return;
end
if length(app.CurrentX) ~= length(app.CurrentY)
    return;
end
if app.domain == true
    plot(app.UIAxes, app.CurrentX, app.CurrentY);
    title(app.UIAxes, strcat(app.plotTitle, ' in time domain'));
    xlabel(app.UIAxes, 'time');
    xlim(app.UIAxes, [min(app.CurrentX), max(app.CurrentX)]);
    ylim(app.UIAxes, [min(app.CurrentY) - 0.1, max(app.CurrentY) + 0.1]);
else
    Y = fft(app.CurrentY);
    L = length(app.CurrentY);
    P2 = abs(Y/L);
    P1 = P2(1:L/2+1);
    P1(2:end-1) = 2*P1(2:end-1);
    f = app.Fs*(0:(L/2))/L;

    plot(app.UIAxes, f,P1);
    title(app.UIAxes, strcat(app.plotTitle, ' in frequency domain'));
    xlabel(app.UIAxes, 'f (Hz)');
    ylabel(app.UIAxes, 'Intensity');
    xlim(app.UIAxes, [min(f), max(f)]);
    ylim(app.UIAxes, [min(P1) - 0.1, max(P1) + 0.1]);
end
```

Code used by recorder app:

```
        % Button pushed function: StartButton
        function StartButtonPushed(app, event)
            recobj = audiorecorder;
            recordblocking(recobj, app.duration);
            app.recording = getaudiodata(recobj);
            t = linspace(0, app.duration, app.duration * 8000);
            plot(app.UIAxes, t, app.recording);
        end


        % Value changed function: DurationEditField
        function DurationEditFieldValueChanged(app, event)
            value = app.DurationEditField.Value;
            app.duration = value;
        end


        % Button pushed function: DoneButton
        function DoneButtonPushed(app, event)
            if isempty(app.recording) == 0
                app.callerApp.updateMainSignal(app.recording);

                %audiowrite('record1.wav', app.recording, 8000);
            end

            close(app.UIFigure);
        end


        % Button pushed function: RetryButton
        function RetryButtonPushed(app, event)
            app.recording = zeros(1, length(linspace(0, app.duration, 8000)));
            plot(app.UIAxes, app.recording);
        end


        % Button pushed function: PlayButton
function PlayButtonPushed(app, event)
sound(app.recording, 8000);
end
end
```

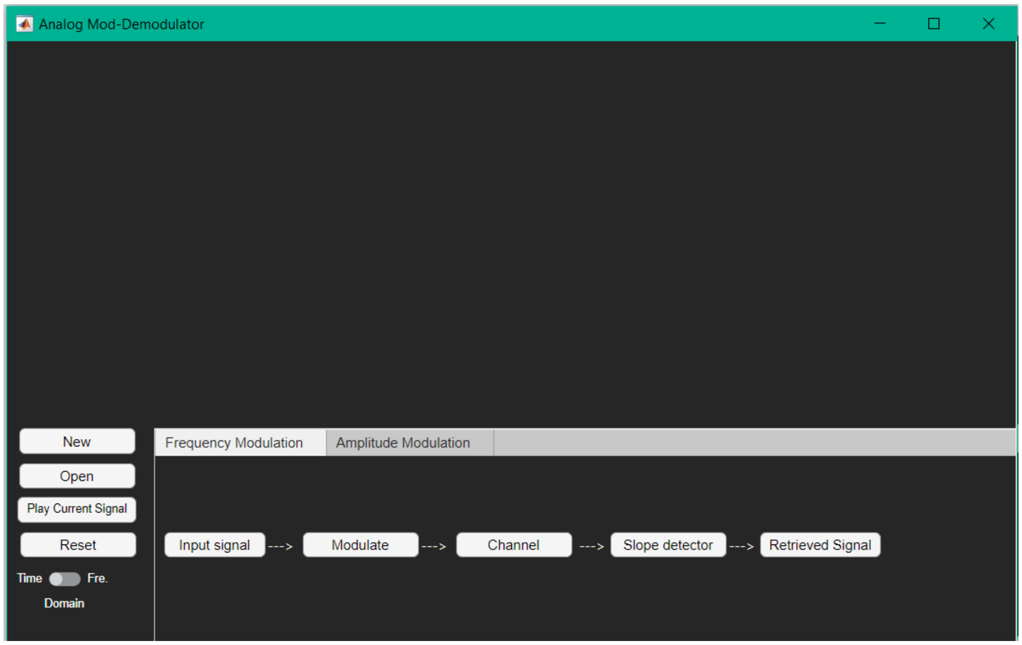
Flow Chart of Signal modulation and demodulation Project



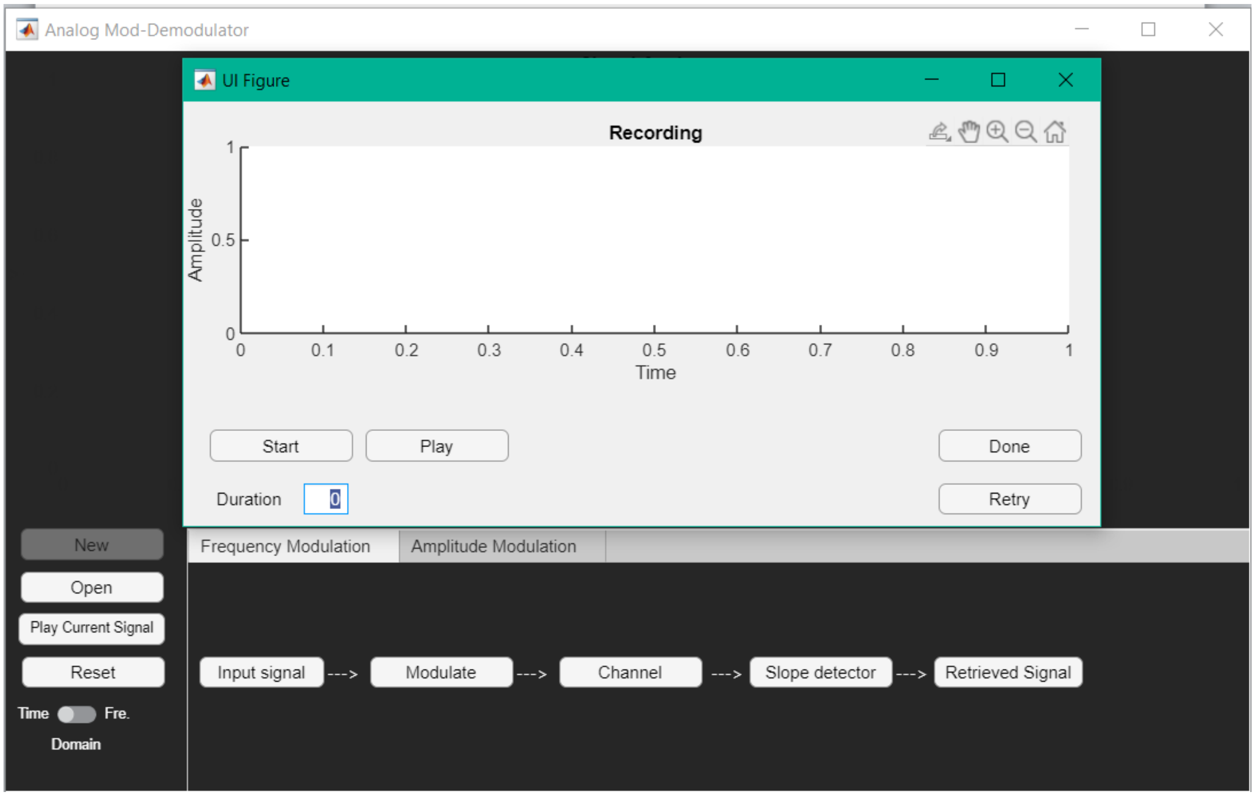


Result:

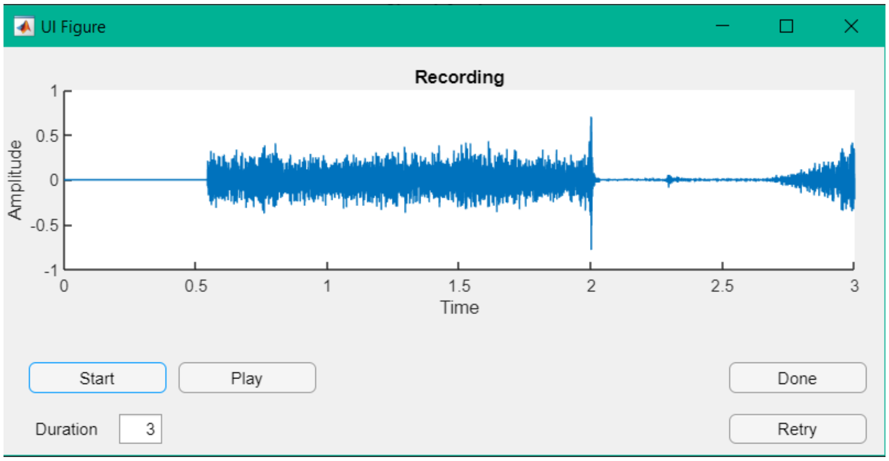
The user interface:



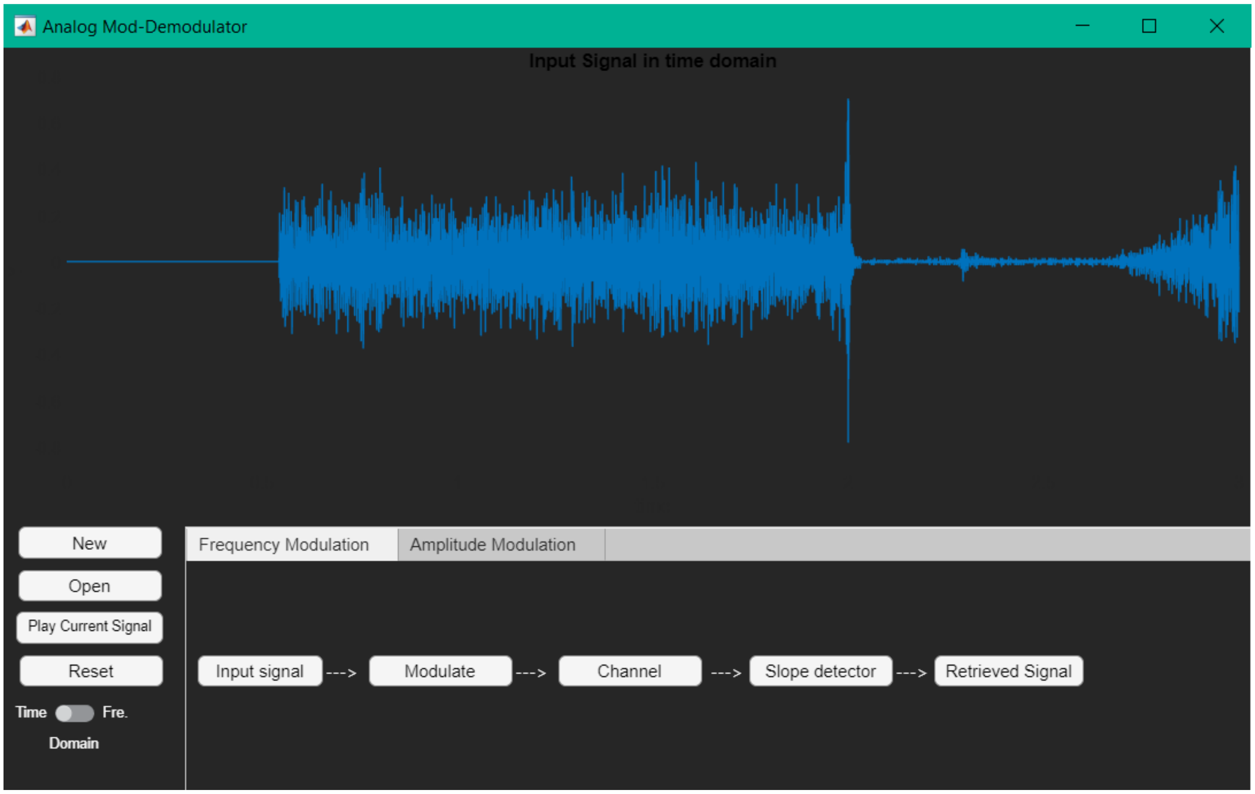
Record new audio:



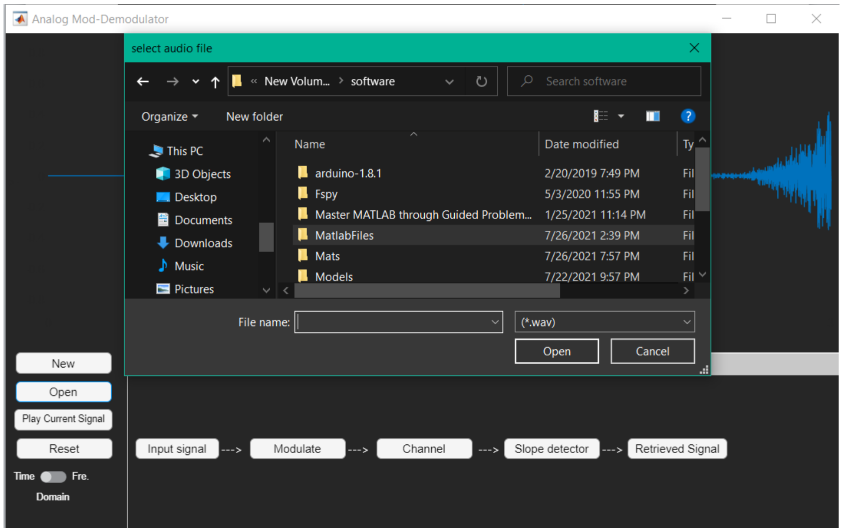
Input new sound:



Adding audio recording to Main application:



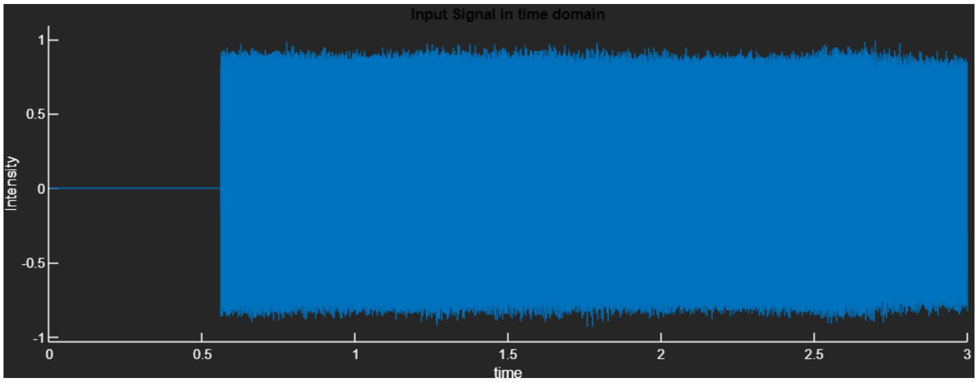
Or, Input signal from existing audio:



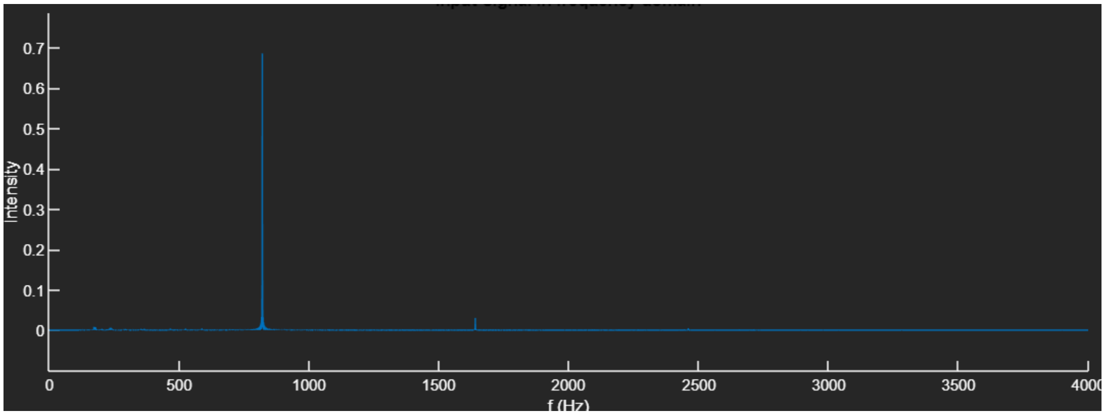
Name	Date modified	Ty
arduino-1.8.1	2/20/2019 7:49 PM	Fil
Fspy	5/3/2020 11:55 PM	Fil
Master MATLAB through Guided Problem...	1/25/2021 11:14 PM	Fil
MatlabFiles	7/26/2021 2:39 PM	Fil
Mats	7/26/2021 7:57 PM	Fil
Models	7/22/2021 9:57 PM	Fil

Input signal:

An 830Hz audio in time domain:



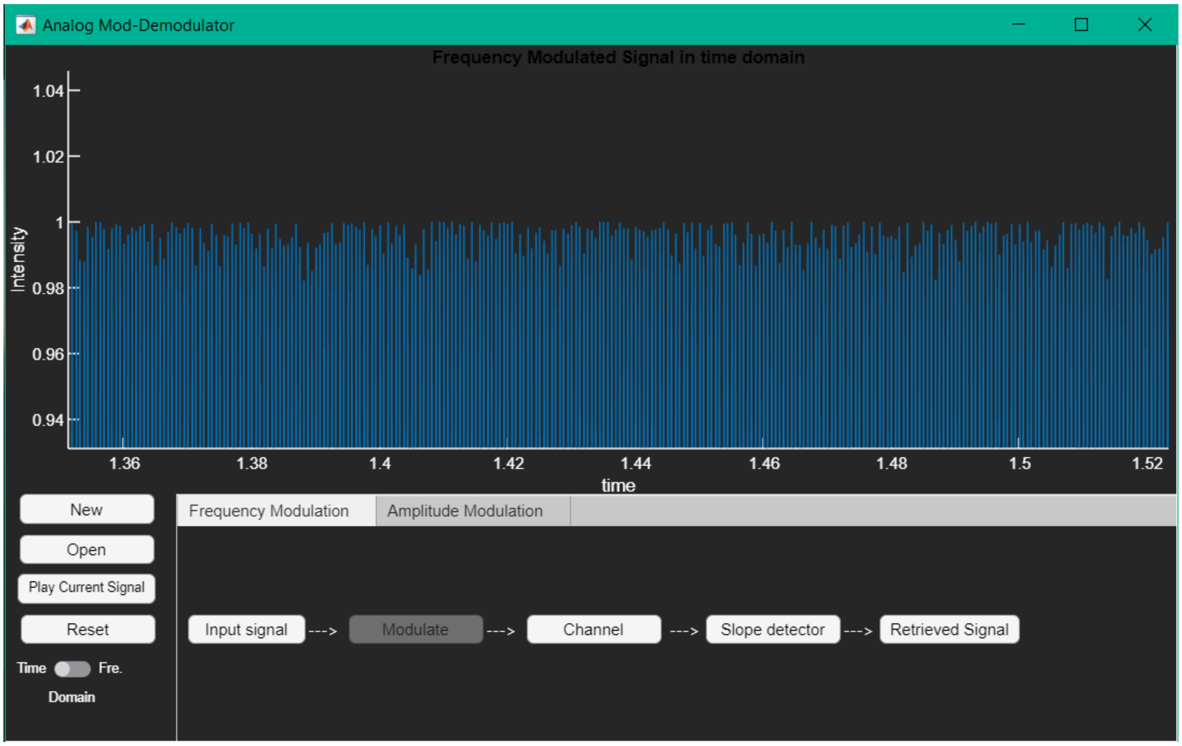
In frequency domain:



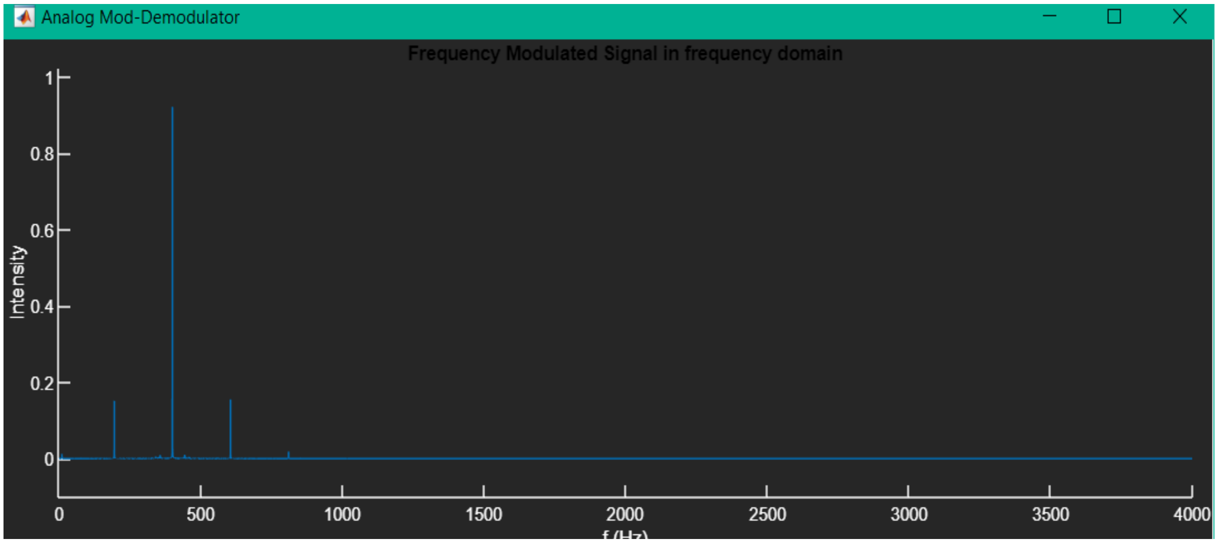
Frequency Modulation-demodulation

Frequency Modulation

Frequency modulated signal:

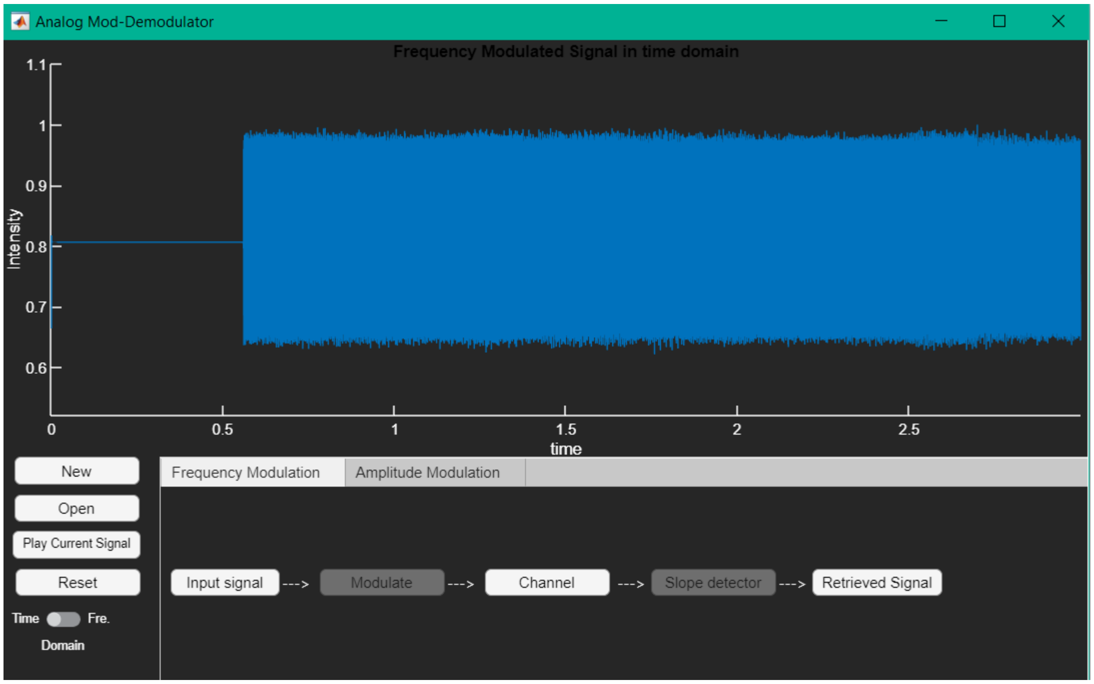


In frequency domain:

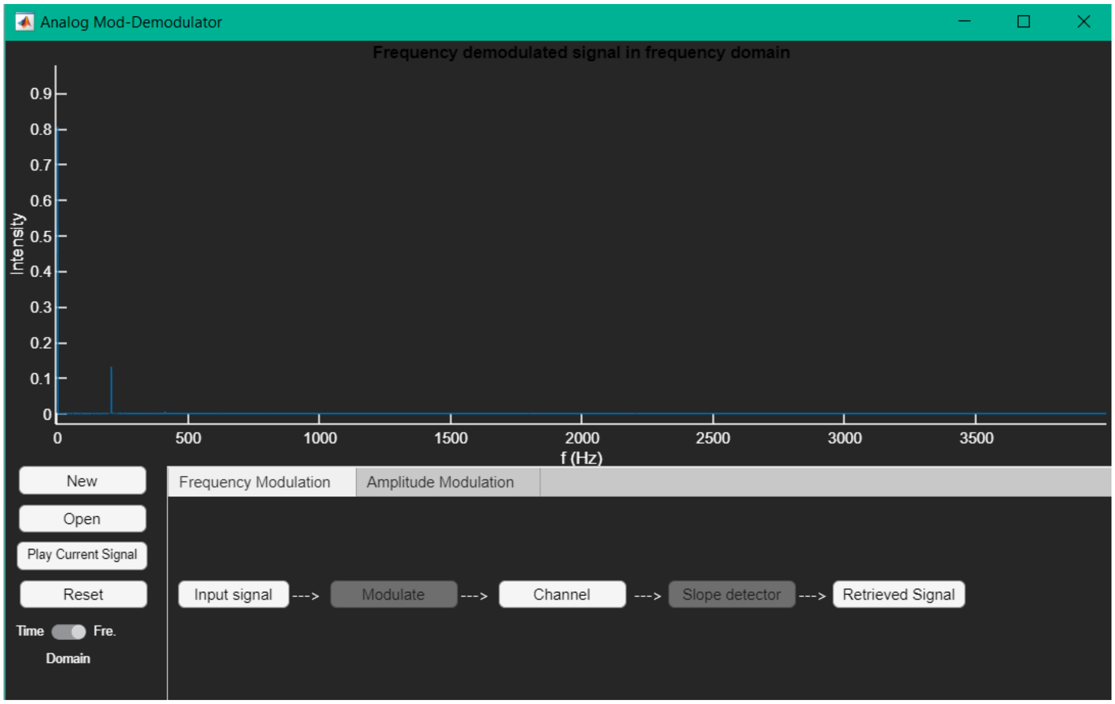


Frequency demodulation

Demodulated using slope detector:



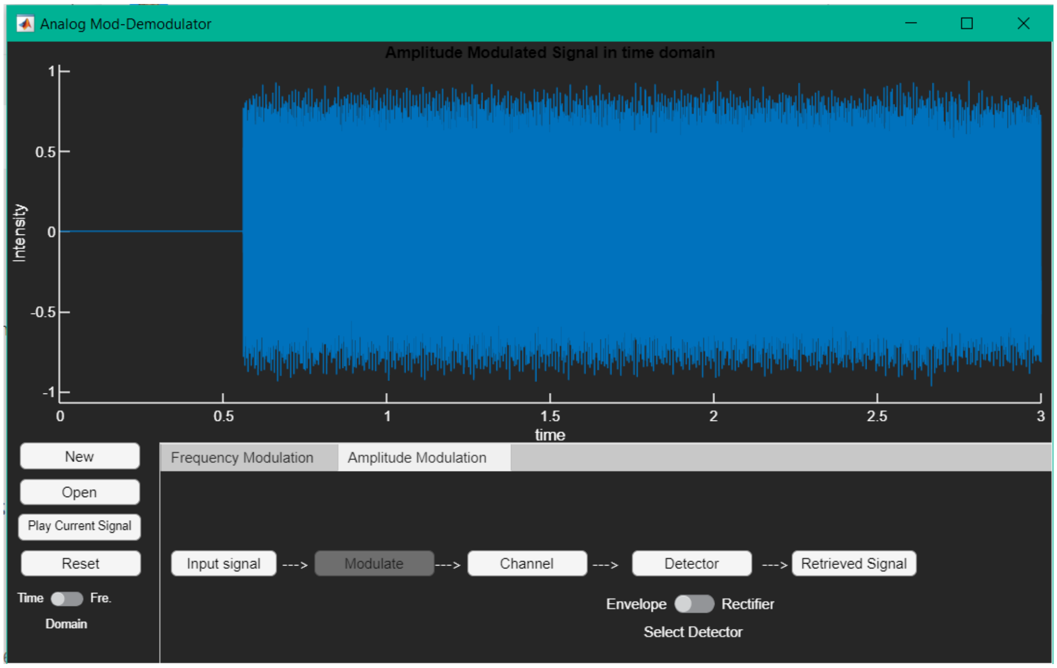
Frequency demodulated signal in frequency domain:



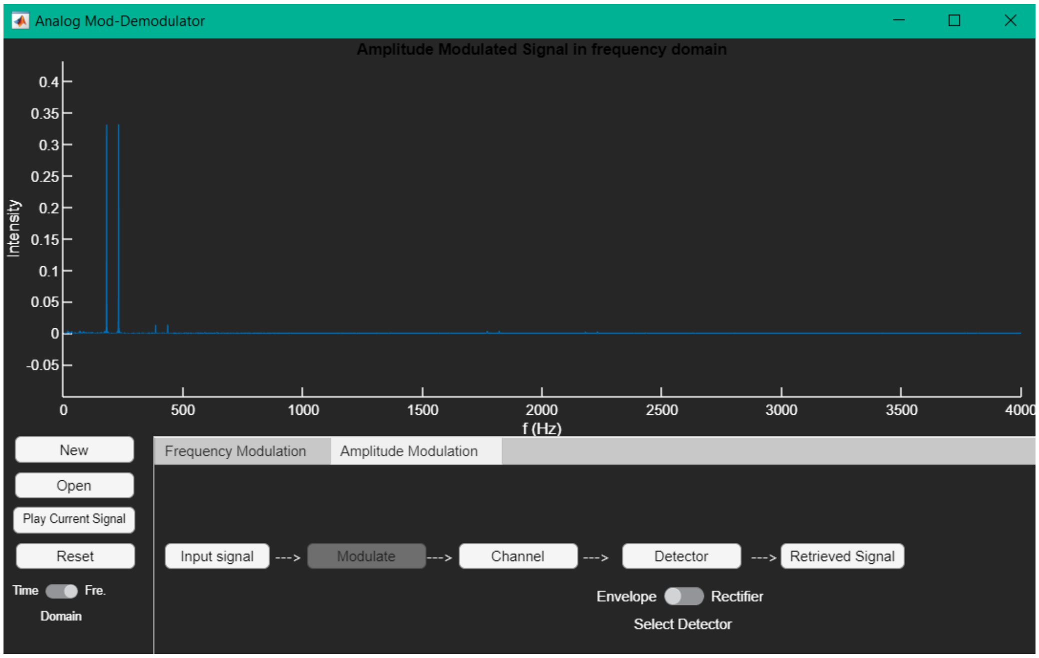
Amplitude modulation-demodulation:

Amplitude modulation

Amplitude modulated signal:

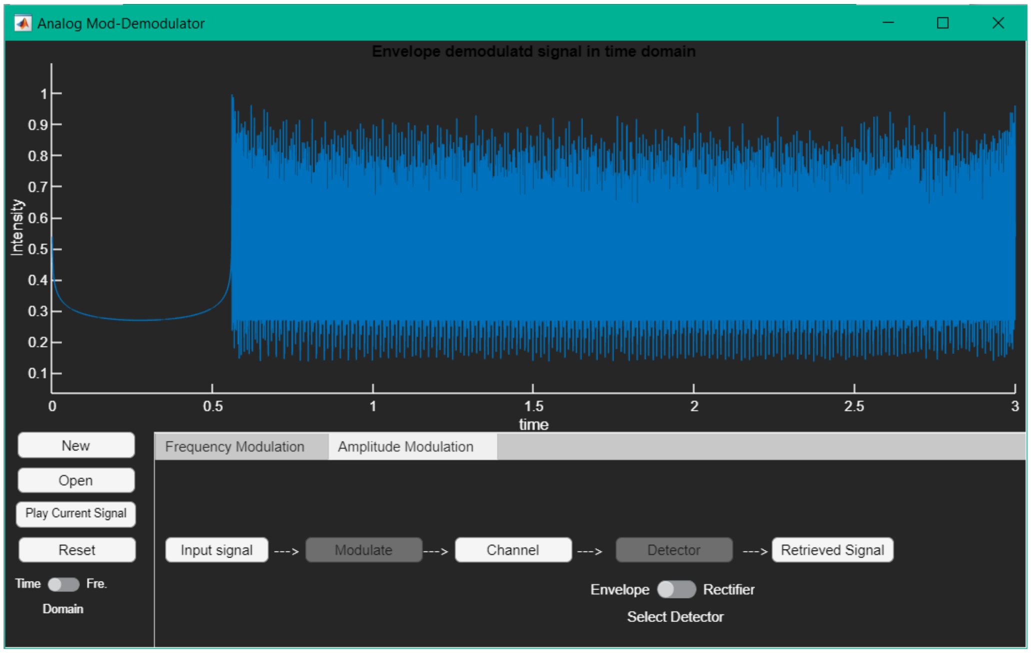


In frequency domain:

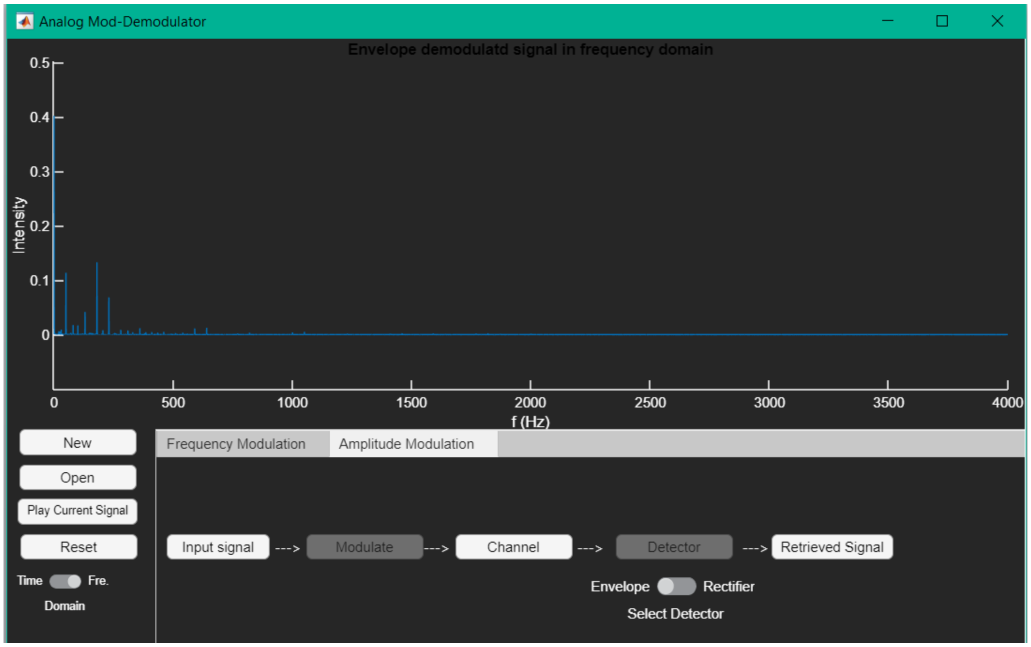


Amplitude demodulation

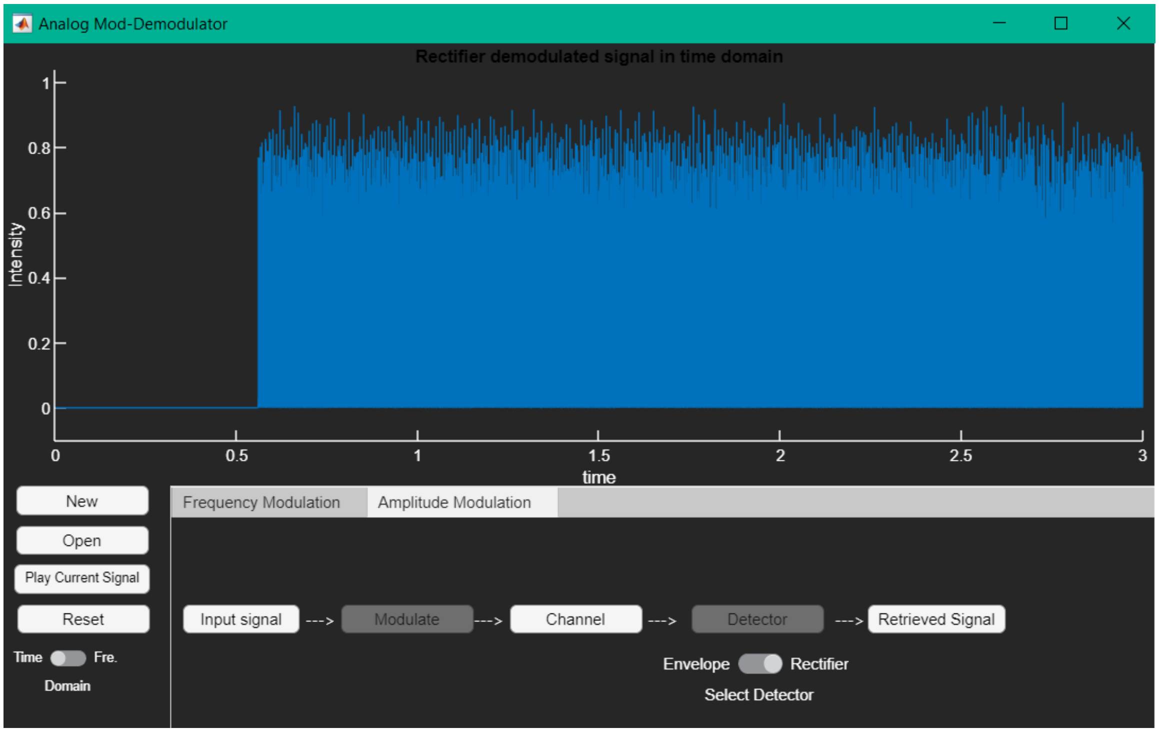
Demodulating using envelope detector:



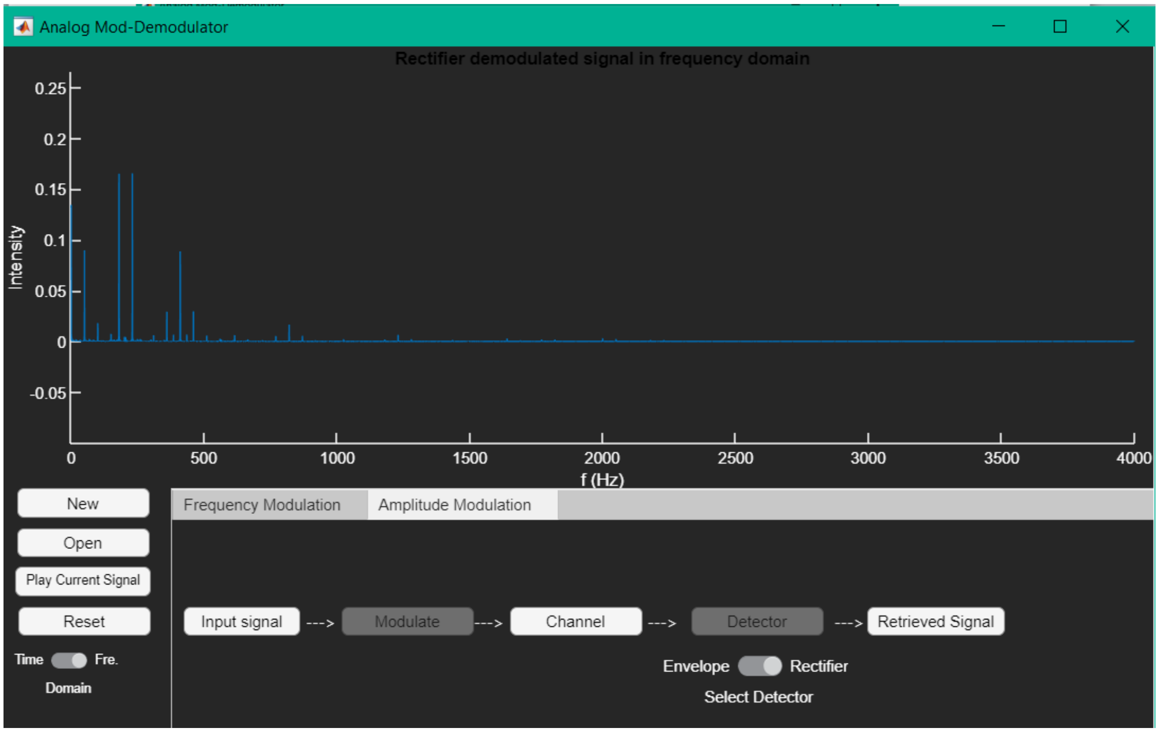
Envelope detector in frequency domain:



Demodulating using rectifier detector:



Rectifier detector in frequency domain:



Discussion:

The application uses input from user either from a existing file or a new recording. It can plot signals in both time and frequency domain. Two major types of modulation methods are applied here. They are – amplitude & frequency modulation. The user can select any method to modulate and demodulate the audio signal.

Signal modulation is widely used in communication systems. Here, we used two major types of modulation- frequency & amplitude modulation. From the results, we can conclude that frequency modulation is superior than amplitude modulation. The reason is clear. Out input signal was a pure 820Hz audio(which can be observed in frequency domain view of input). But the output signal from amplitude demodulation in frequency domain contains signal of other frequencies. That is, there is added noise in amplitude modulation. On the other hand, frequency modulated signal is clean and much more simillar to the original signal. So, we can conclude that, Frequency modulation results in better signal transmission. Also, there are other benefits of FM. Such as- it takes a small antenna in order to transmit signals. The receivers are much more cheaper in FM systems than that of AM. So, we can say that, frequency modulation is much more effective and efficient than AM.

Future prospect:

Signals are everywhere. Especially, audio signals are recorder and manipulated every single day. Radio stations use FM signals all the time. RADAR, wireless communication all require some kind of signal modulation and demodulation method so carry information. That’s why I think that, this application if further developed will be able to take part in industry standard work. It will be able to help analyze signals and improve on the system.