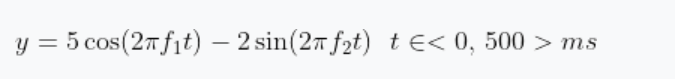
**Sample voltage signal y according to the definition:**



**fs = 200Hz, f1=2Hz, f2=8Hz**

**Properly generate and plot the signal, then calculate the following characteristics and show them in a table:**

**a) (mean), mean value**

**b) (rms), root mean square value**

**c) (var), variance**

**d) (std), standard deviation value**

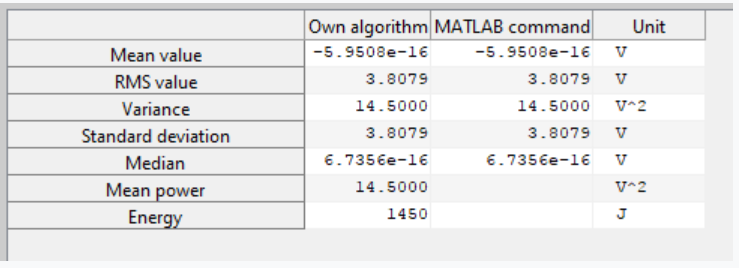
**e) (median), the median value**

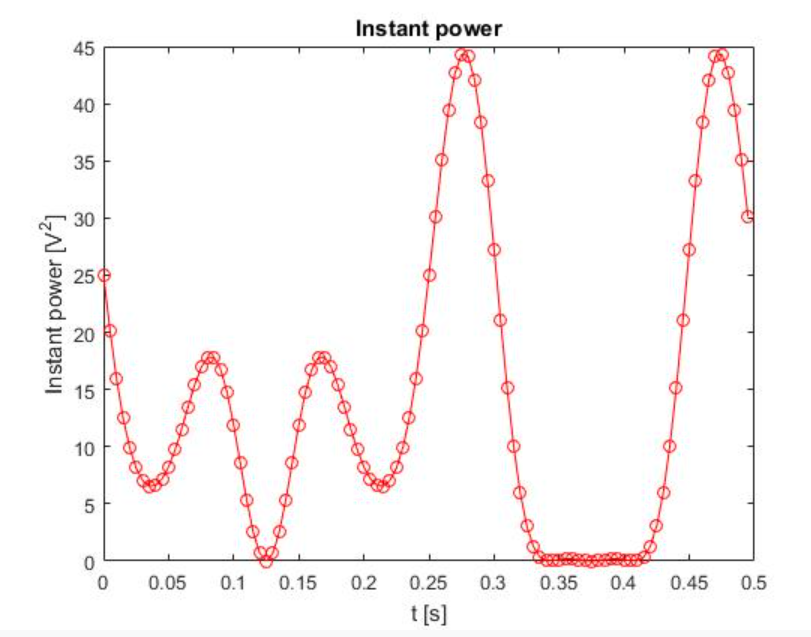
**f) immediate power**

**g) mean power**

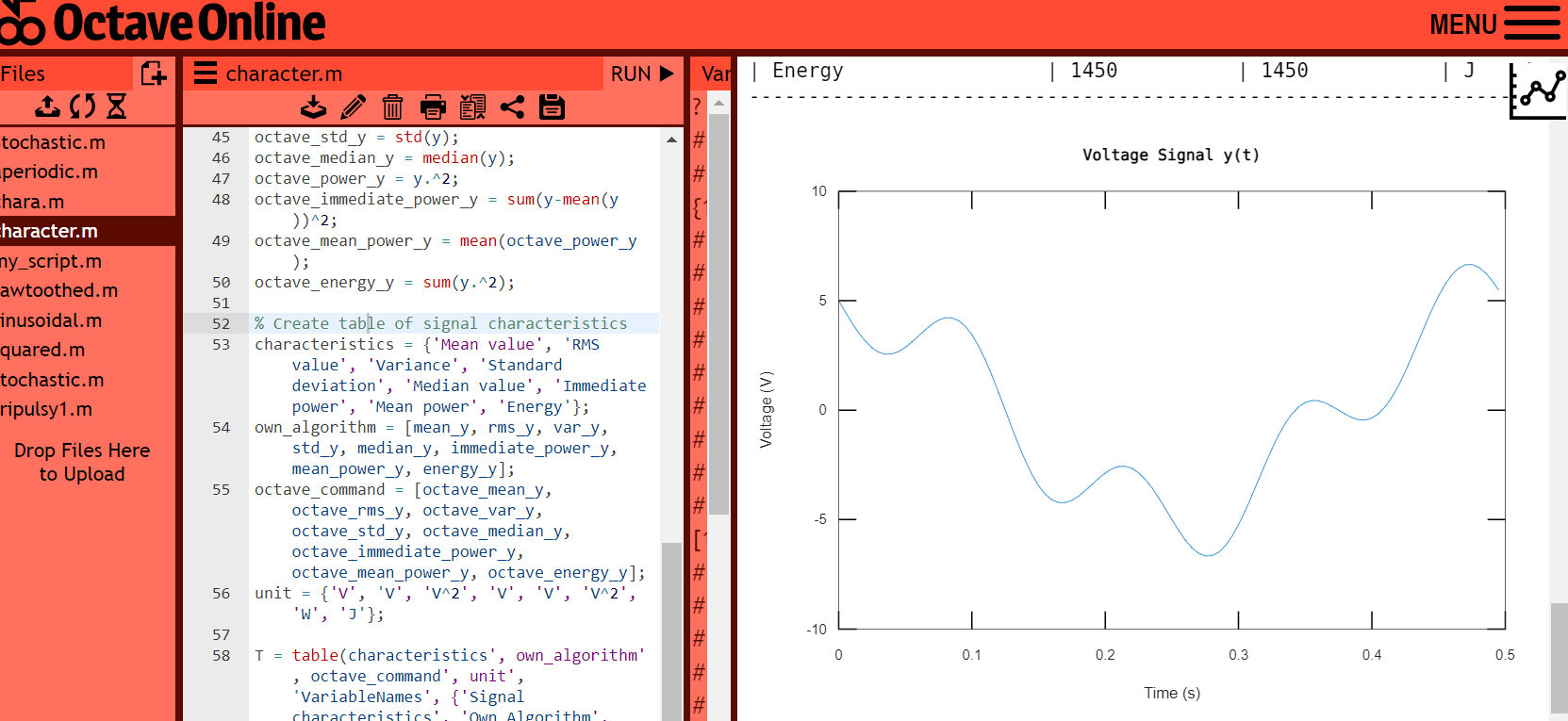
**h) energy value**

**For their calculations use your own algorithm according to definitions and verify them with MATLAB commands. Put results in a well-arranged table.**





1. **Voltage signal**



**Source code for octave online**

% Define the sampling frequency and time axis

fs = 200; %sampling frequency (Hz)

t = 0:1/fs:0.5; %time vector (s)

% Define the signal

f1 = 2; %frequency of first component (Hz)

f2 = 8; %frequency of second component (Hz)

y = 5\*cos(2\*pi\*f1\*t) - 2\*sin(2\*pi\*f2\*t);

% Plot the signal

plot(t, y);

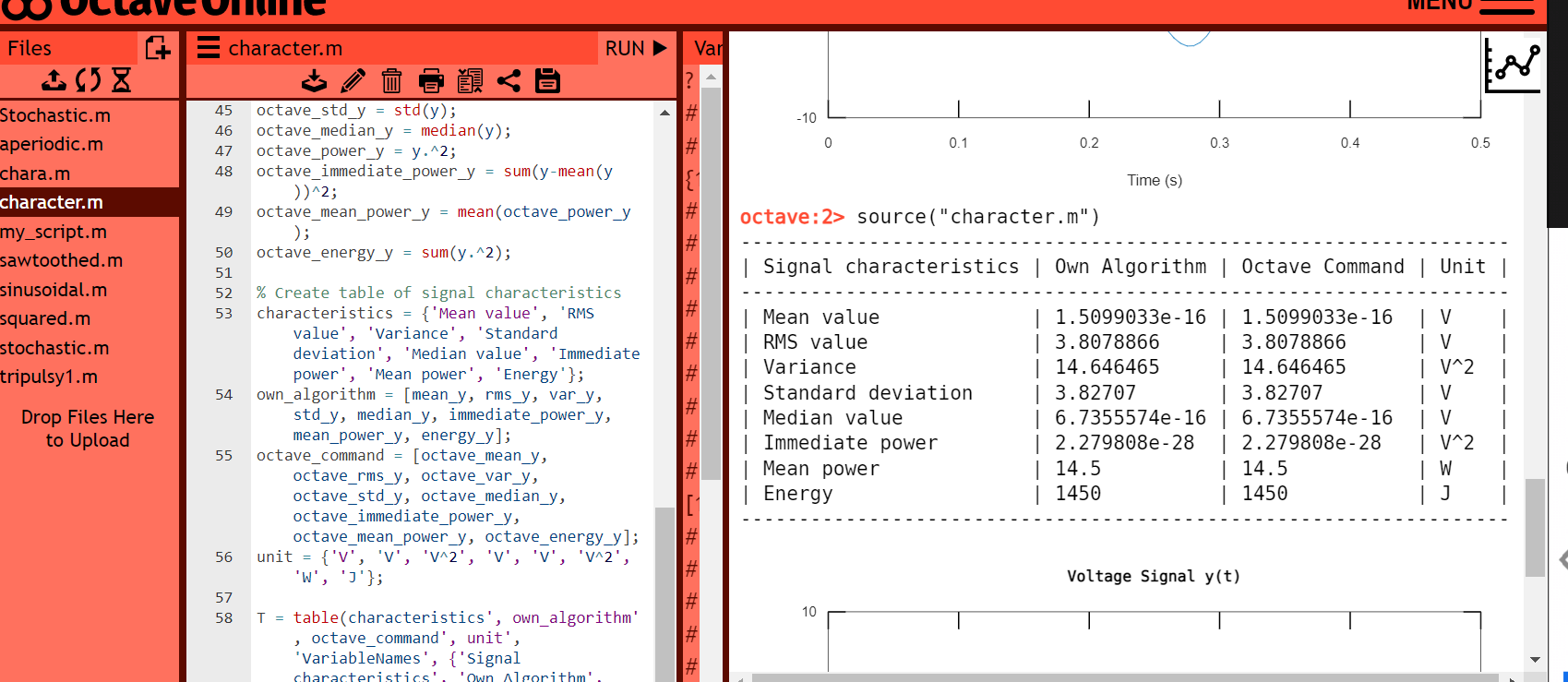
xlabel('Time (s)');

ylabel('Voltage (V)');

title('Voltage Signal y(t)');

Comment: This code generates and plots a voltage signal y according to the given equation, where y is defined as the difference between a cosine wave with frequency f1 and an amplitude of 5 volts, and a sine wave with frequency f2 and an amplitude of 2 volts. The signal is defined over a time range of 0 to 0.5 seconds with a sampling frequency of 200 Hz.

1. **Signal Characteristics**



**Source code for octave online**

% Turn off warning for shadowed functions

warning('off', 'Octave:shadowed-function')

% Load the tablicious package

pkg load tablicious

% Define the sampling frequency and time axis

fs = 200; %sampling frequency (Hz)

Ts = 1/fs;

t = 0:Ts:0.5 - Ts; %time vector (s)

% Define the signal

f1 = 2; %frequency of first component (Hz)

f2 = 8; %frequency of second component (Hz)

y = 5\*cos(2\*pi\*f1\*t) - 2\*sin(2\*pi\*f2\*t);

% Plot the signal

plot(t, y);

xlabel('Time (s)');

ylabel('Voltage (V)');

title('Voltage Signal y(t)');

% Calculate signal characteristics

mean\_y = (1/length(y))\*(sum(y)); % mean value

rms\_y = sqrt((1/length(y))\*sum(y.^2)); % root mean square value

var\_y = 1/(length(y)-1) \* sum((y-mean(y)).^2); % variance

std\_y = sqrt(var\_y); % standard deviation value

sorted\_y = sort(y);

midpoint = floor(length(sorted\_y)/2);

if rem(length(sorted\_y),2)==0

median\_y = (sorted\_y(midpoint)+sorted\_y(midpoint+1))/2; % median value

else

median\_y = sorted\_y(midpoint+1); % median value

end

power\_y = y.^2;

immediate\_power\_y = sum(y-mean(y))^2; % immediate power

mean\_power\_y = (1/length(y))\*sum(y.^2); % mean power

energy\_y = sum(y.^2); % energy value

% Verify calculations with Octave Online commands

octave\_mean\_y = mean(y);

octave\_rms\_y = rms(y);

octave\_var\_y = var(y);

octave\_std\_y = std(y);

octave\_median\_y = median(y);

octave\_power\_y = y.^2;

octave\_immediate\_power\_y = sum(y-mean(y))^2;

octave\_mean\_power\_y = mean(octave\_power\_y);

octave\_energy\_y = sum(y.^2);

% Create table of signal characteristics

characteristics = {'Mean value', 'RMS value', 'Variance', 'Standard deviation', 'Median value', 'Immediate power', 'Mean power', 'Energy'};

own\_algorithm = [mean\_y, rms\_y, var\_y, std\_y, median\_y, immediate\_power\_y, mean\_power\_y, energy\_y];

octave\_command = [octave\_mean\_y, octave\_rms\_y, octave\_var\_y, octave\_std\_y, octave\_median\_y, octave\_immediate\_power\_y, octave\_mean\_power\_y, octave\_energy\_y];

unit = {'V', 'V', 'V^2', 'V', 'V', 'V^2', 'W', 'J'};

T = table(characteristics', own\_algorithm', octave\_command', unit', 'VariableNames', {'Signal characteristics', 'Own Algorithm', 'Octave Command', 'Unit'});

prettyprint(T);

**Comment**: The code begins by defining the sampling frequency and time axis, and then generates a voltage signal with two components of different frequencies. The signal is then plotted using the plot function. Next, several signal characteristics including mean value, root mean square value, variance, standard deviation, median value, immediate power, mean power, and energy are calculated using custom algorithms and verified using Octave commands. Finally, a table of the signal characteristics is created and displayed using the tablicious package. The table shows the values calculated by the custom algorithm and the Octave commands, along with their corresponding units. Overall, the code provides a useful example of how to generate and analyze a voltage signal in Octave.