****

**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Matavimo sistemos tyrimas**

Laboratorinis darbas

|  |  |
| --- | --- |
|  |  |
| **Žygimantas Marma**  Studentas, EMEI-2 | (parašas) (data) |
|  |  |
| **Prof. Darius Gailius**  Dėstytojas | (parašas) (data) |
|  |  |

**Kaunas, 2023**

Turinys

[Lentelių sąrašas 3](#_Toc152020011)

[Paveikslų sąrašas 4](#_Toc152020012)

[Įvadas 5](#_Toc152020013)

[1. Signalų matavimas ir genravimas 6](#_Toc152020014)

[2. Impedanso matavimo sistema 10](#_Toc152020015)

[Išvados 14](#_Toc152020016)

[Priedai 15](#_Toc152020017)

Lentelių sąrašas

[1 lentelė. Impedanso matavimo rezultatai 12](#_Toc152020020)

[2 lentelė. Pirmosios laboratorinio darbo dalies programinis kodas 15](#_Toc152020021)

[3 lentelė. Antrosios laboratorinio darbo dalies programinis kodas 17](#_Toc152020022)

Paveikslų sąrašas

[1 pav. Laboratonio matavimo schema 6](#_Toc152020024)

[2 pav. Sugeneruotas sinusinis signalas 7](#_Toc152020025)

[3 pav. Signalo spektras 7](#_Toc152020026)

[4 pav. Stačiakampio signalo dalis 8](#_Toc152020027)

[5 pav. Stačiakampio signalo spektras 8](#_Toc152020028)

[6 pav. Stačiakampio signalo persidengimas 9](#_Toc152020029)

[7 pav. Fazių skirtumas tarp įtampų kai dažnis 20 Hz 11](#_Toc152020030)

[8 pav. Matavimas ties 40Hz, fazių skirtumas – 14,8 laipsnių 11](#_Toc152020031)

[9 pav. Matavimas ties 140Hz, fazių skirtumas – 5,2 laipsniai 12](#_Toc152020032)

[10 pav. Varžos dedamųjų dydžiai nagrinėjamoje schemoje 13](#_Toc152020033)

[11 pav. Kondensatoriaus talpos priklausomybė nuo dažnio 13](#_Toc152020034)

Įvadas

**Darbo tikslas**: susipažinti susignalų generatoriumi ir osciloskopu bei sukurti impedanso matavimo sistemą

**Darbo uždaviniai**:

Pirma dalis:

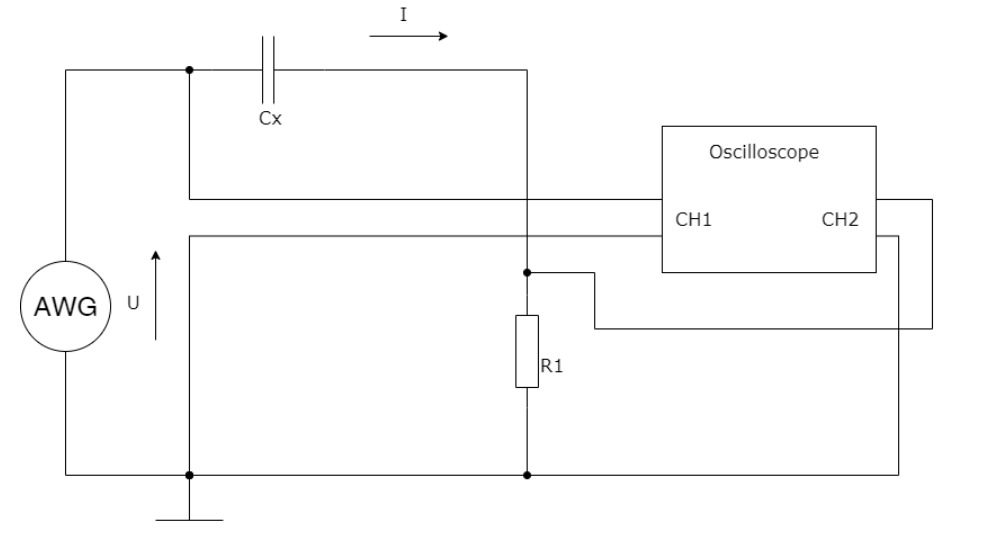
* Valdyti prijungtus įrenginius iš Matlab arba Python programavimo aplinkų naudojant VISA bibliotekas.
* Nuskaitykite duomenis iš osciloskopo sinusoidiniams ir bet kokiems nesinusiniams signalams ir apskaičiuokite jų spektrą naudodami FFT
* Sudaryti ataskaitą: trumpas aprašymas, sistemos sąranka, rezultatai, aptarimas ir išvados, naudota programa su komentarais.
* Būtų gerai, jei ataskaitoje būtų pateikti keli spektro persidengimo nutekėjimo pavyzdžiai su paaiškinimais, kaip sumažinti tokių efektų skaičių.

Antra dalis:

* Sukurkite varžos matavimo sistemą naudodami AWG, osciloskopą ir prijungtą grandinę, sudarytą iš kondensatoriaus ir rezistoriaus
* Apskaičiuokite bandomojo prijungto įrenginio varžą (DUT yra kondensatorius parengtoje schemoje).
* Jei įmanoma, išmatuokite varžos pokyčius dažnių diapazone.

# Signalų matavimas ir genravimas

Darbe yra naudojami komponentai sujungti taip kaip pavaizduota 1 pav. Varžos vertė yra 200 omų, o kondenstoriaus talpa 4700 nano faradų.

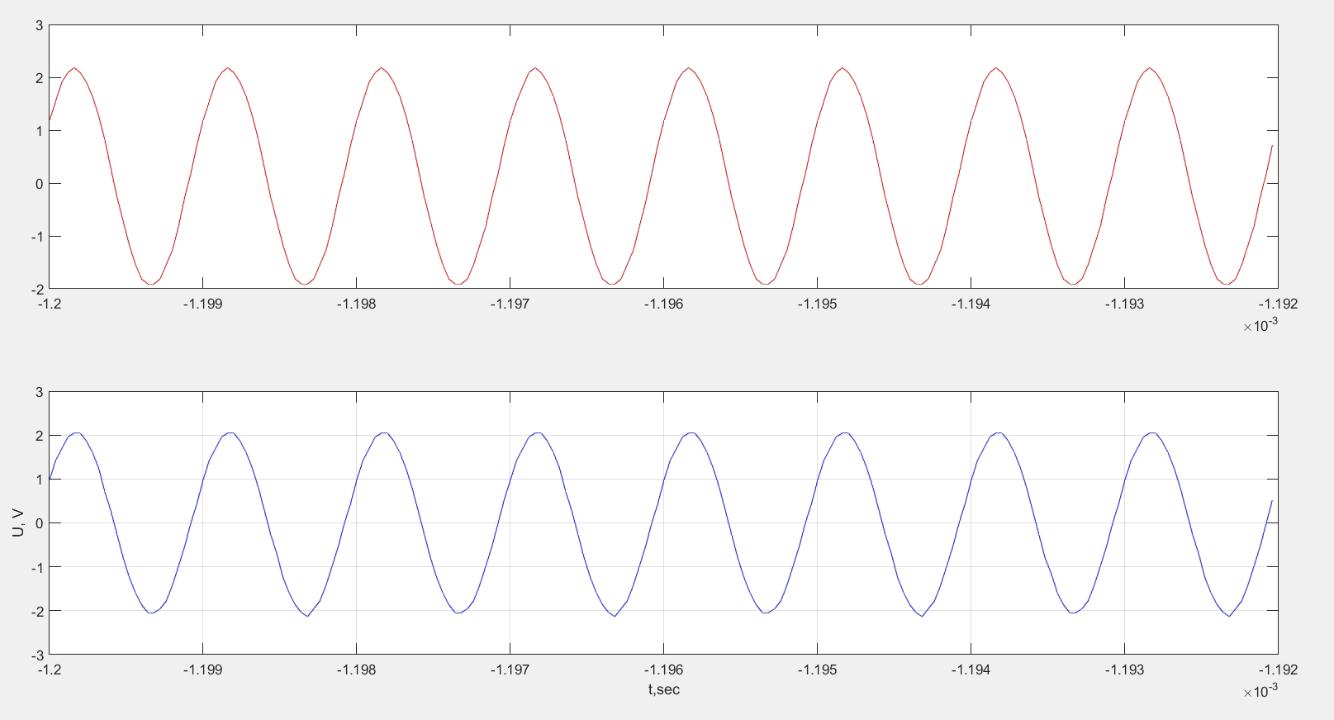


1 pav. Laboratonio matavimo schema

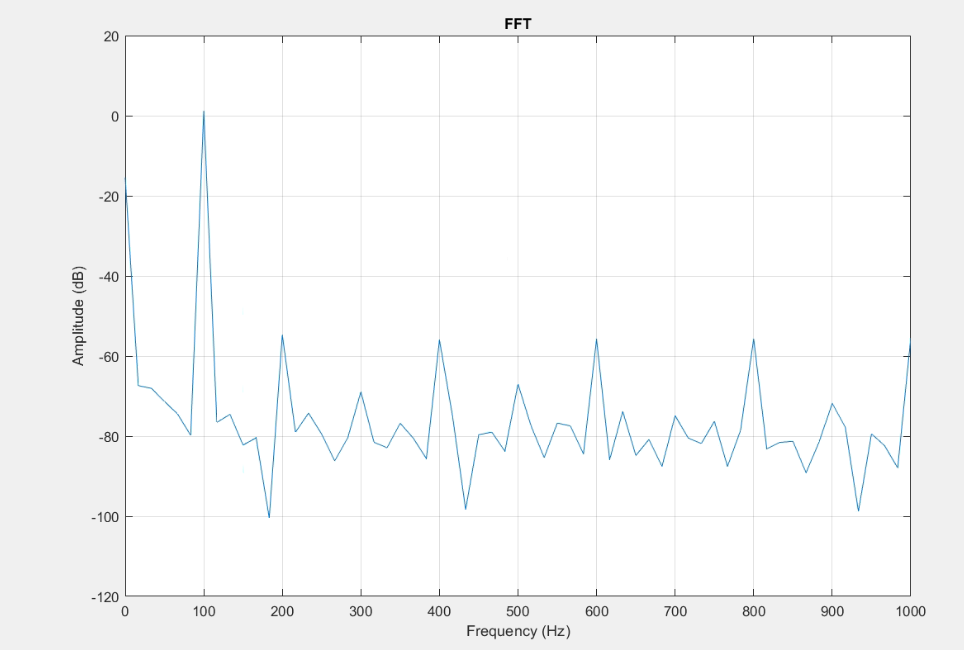
Taip pat nuotoliniam darnui sujungti naudojamas osciloskopas (Rigol DS1074Z) ir signalų generatorius (Rigol DG4062). Jie valdomi kompiuteriumi per USB sąsają.

## Sinusinis signalas

Iš nagrinėjus pavyzdinį MATLAB projekta buvo aišku kaip yra valdomas osciloskopas ir generatorius. Toliau buvo sugeneruotas sinusiis 100 Hz, 4V amplitudės signalas (kaip pavaizduota 2 pav.)



2 pav. Sugeneruotas sinusinis signalas



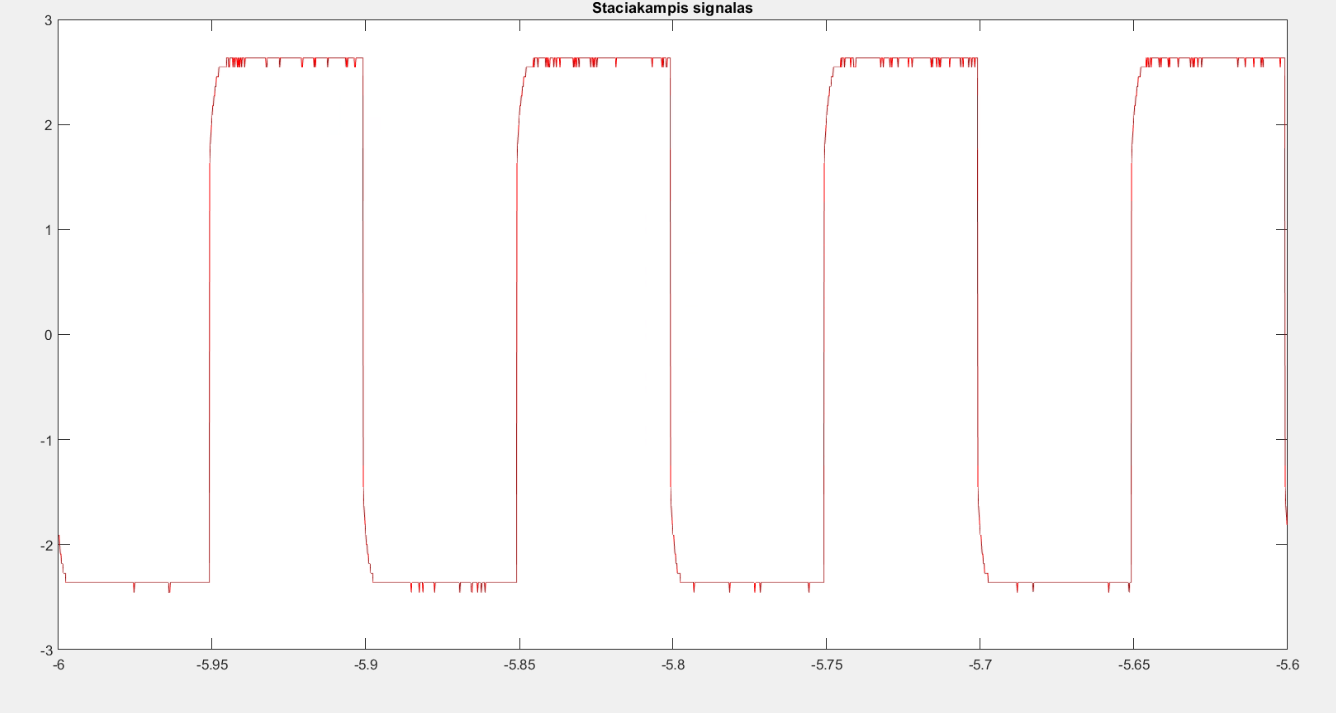
3 pav. Signalo spektras

## Stačiakampis signalas

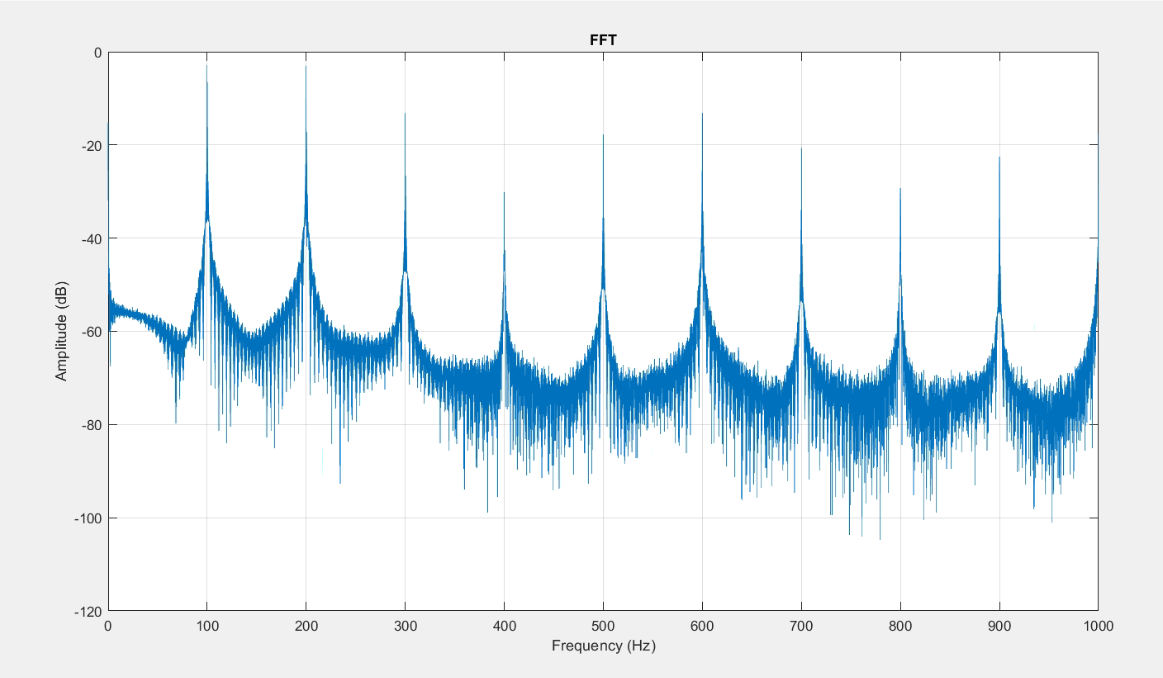
Norint generuoti stačiakampį signalą reikėjo pakeisti šiuos nustatymus

**fprintf(H\_AWG, ":SOUR1:FUNC SQUare")**

Šio sugeneruoto signalo dažnis yra 100 Hz, amplitudė 5V. Analizuojant signalo spektrą galima matyti pasikartojančias harmonikas kas 100Hz kas yra būdinga stačiakampiams signalams.



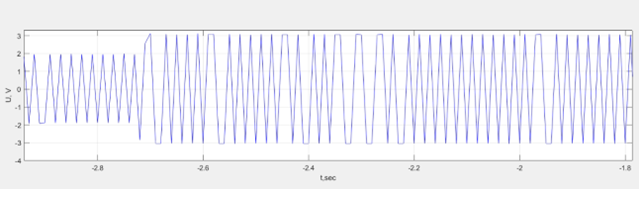
4 pav. Stačiakampio signalo dalis



5 pav. Stačiakampio signalo spektras

## Dažnių persidengimas

Testavimas buvo atliktas su staciakampiu signalu, nes sinusiniam signalui didinant dažnį virš Nyquist‘o jo fazė tiesiog persistumia ir gaunasi kitas sinusinis signalas.



6 pav. Stačiakampio signalo persidengimas

Norinit minimizuoti šiuos efektuos reikia didinti diskretizavimo dažnį, kad jis būtų bent dvigubai didesnis nei didžiausio dažnio dedamoji.

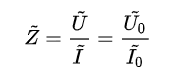
# Impedanso matavimo sistema

## Darbo aplinka

Išnagrinėjus pateiktą laboratorinio darbo schemą, toliau buvo atliktas impedanso matavimo tyrimas. Darbe nagrinėjamas signalų generatorius, geba generuoti skirtingų dažniu signalus, būtent pasinaudojant šia savybę ir bus apskaičiuojamas fazės skirtumas tarp įtampų 20-200Hz diapazone.

Norint atlikti šiuos skaičiavimui reikia žinoti bendro impedanso formulę:

Iš omo dėsnio žinoma kad kiekvienam komponetui skirta įtampą galima paskaičiuoti taip:

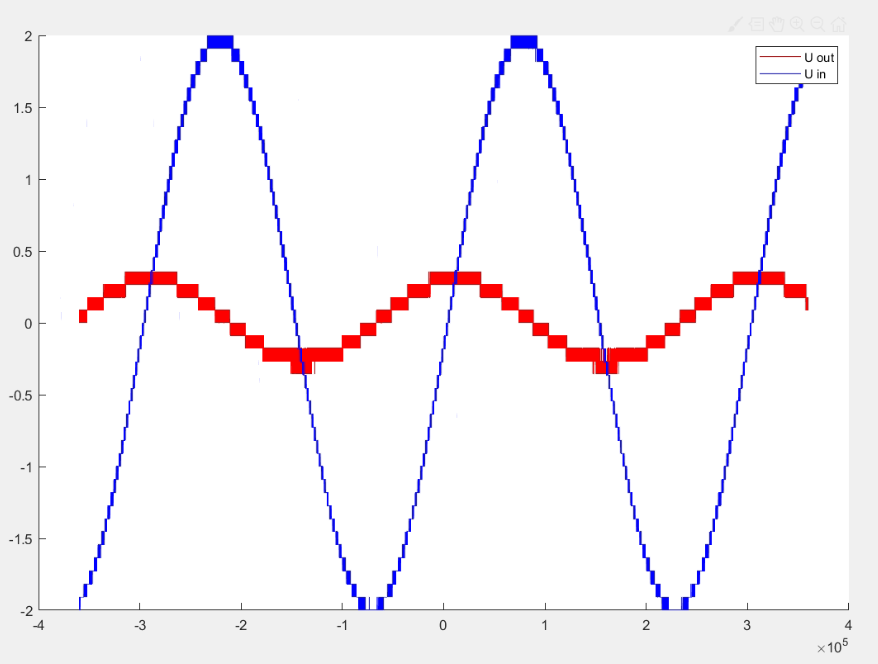


O galiausiai kondensatoriaus varžą tamtikrame dažnyje gausime taip:



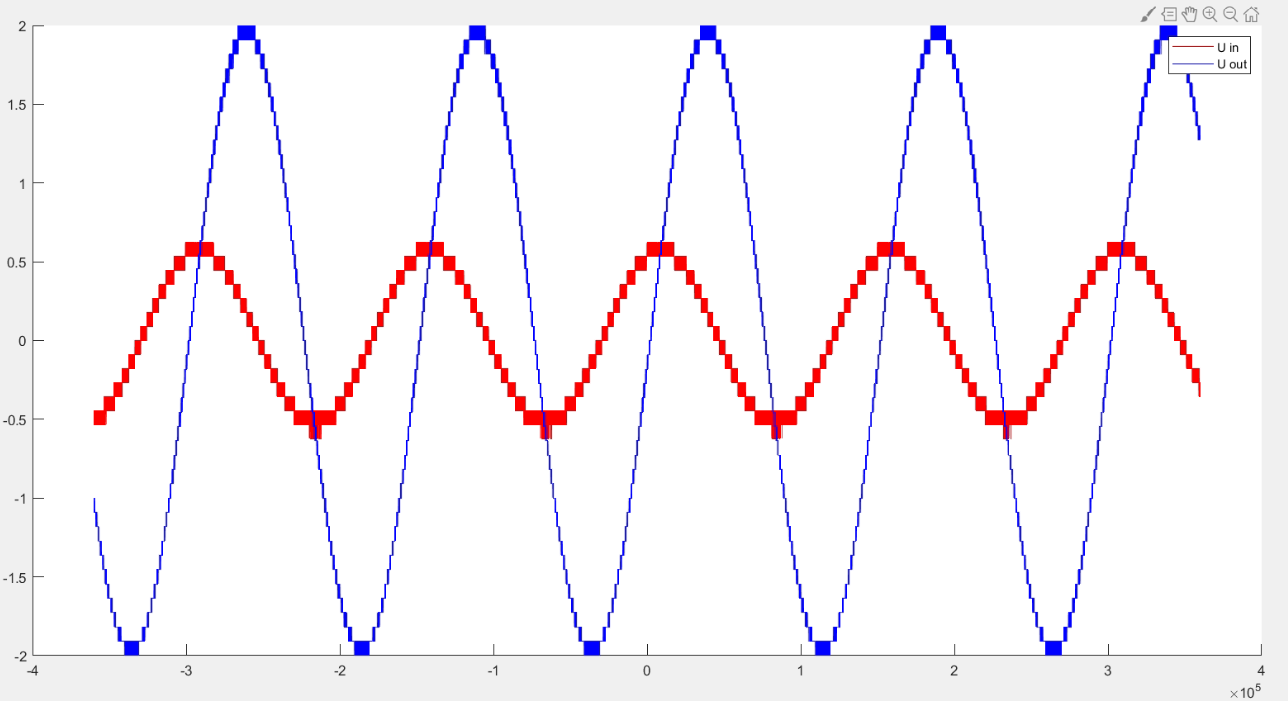
## Impedanso matavimas

Šiame tyrime dažnis yra keičiamas nuo 20 iki 200Hz kas 10 Hz išmatuojant įtampas ant abiejų elemtų ir apskaičiuojant varžas. Kaip pavyzdį galima paimti matavimą ties 20Hz dažniu, kur matomas 32,5 laipnių skirtumas tarp įtampų.

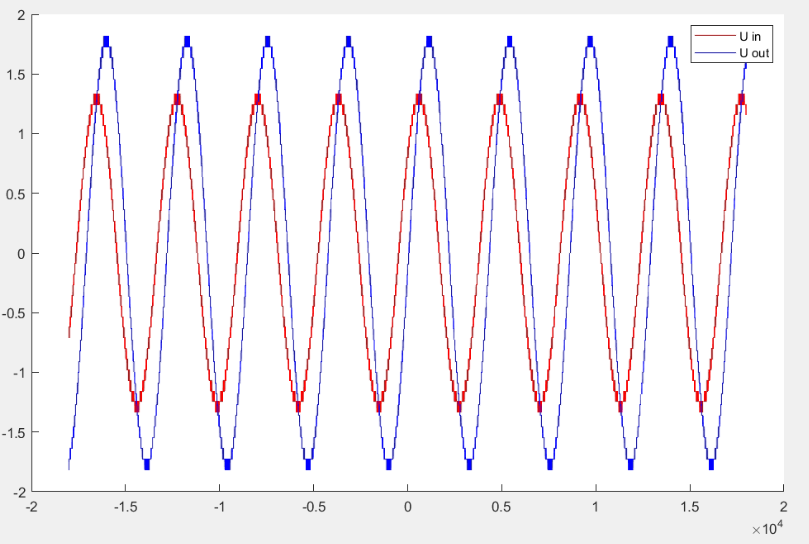


7 pav. Fazių skirtumas tarp įtampų kai dažnis 20 Hz

Didanant dažnį yra matomas fazių skirtumo ir amplitudžių mažėjimas.



8 pav. Matavimas ties 40Hz, fazių skirtumas – 14,8 laipsnių



9 pav. Matavimas ties 140Hz, fazių skirtumas – 5,2 laipsniai

## Matavimo rezultatai

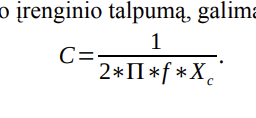
Gauti rezultatai pateikti lentelės ir grafiko forma.

1 lentelė. Impedanso matavimo rezultatai

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dažnis** | **10.0** | **20.0** | **30.0** | **40.0** | **50.0** | **60.0** | **70.0** | **80.0** | **90.0** | **100.0** |
| Rc | 839.3 | 822.0 | 706.4 | 463.1 | 351.1 | 304.4 | 235.4 | 200.0 | 167.2 | 138.8 |
| Xc | 2248.6 | 750.4 | 341.9 | 198.0 | 232.1 | 164.9 | 119.2 | 86.4 | 69.2 | 56.3 |
| Faziu skirtumas | -65.3 | -32.6 | -20.5 | -14.8 | -22.5 | -17.4 | -14.8 | -11.9 | -10.2 | -8.9 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Dažnis** | **110.0** | **120.0** | **130.0** | **140.0** | **150.0** | **160.0** | **170.0** | **180.0** | **190.0** | **200.0** |
| Rc | 140.1 | 99.0 | 99.6 | 78.6 | 78.8 | 69.4 | 60.4 | 60.6 | 38.5 | 46.7 |
| Xc | 44.5 | 38.9 | 31.4 | 27.3 | 24.2 | 20.4 | 17.9 | 15.8 | 14.4 | 12.8 |
| Faziu skirtumas | -7.4 | -6.8 | -5.8 | -5.2 | -4.7 | -4.1 | -3.8 | -3.4 | -3.1 | -2.9 |

10 pav. Varžos dedamųjų dydžiai nagrinėjamoje schemoje

Turint šiuos skaičius galima gauti kondensatoriaus talpą pasinaudojant šia formule:



11 pav. Kondensatoriaus talpos priklausomybė nuo dažnio

Kaip galima matyti kondensatoriaus talpa kinta ganėtinai per daug, kad galėtume ją nustatyti nano faradų tikslumu, tačiau galime teigti jog ji yra 200-500nF eilės.

Išvados

1. Atlikus laboratorinį darbą buvo įgytos žinios kaip valdyti ir vykdyti pagrindines funkcijas osciloskope bei funkciniame generatoriuje, naudojant asmeninį kompiuterį ir MATLAB programavimo kalbą.
2. Buvo atlikta signalo analizė, apsakaičiuota FFT bei parodytas spektro persidengimo efektas su stačiakampiu signalu.
3. Darbe buvo išmokta naudotis automatine impedanso matavimo sistema ir sukurtas algoritmas išmatuoti nežinomos talpos kondensatoriaus talpą keičiant signalų dažnį.

Priedai

2 lentelė. Pirmosios laboratorinio darbo dalies programinis kodas

|  |
| --- |
| % demo program V2.1  % for impedance measurement using Zx+R1 circuit  % equipment:  % oscilloscope Rigol DS2072A  % generator Rigol DG4162  clc  close all  clear all  delete( instrfind)  vh\_winfo = instrhwinfo('visa','ni');  % Find a VISA-USB objects  INST\_ALL=vh\_winfo.ObjectConstructorName;  %CONNECTION SETTINGS BASED ON DEVICE INFO  % use the object that was found.  for I=1:length(INST\_ALL)  if strfind(INST\_ALL{I},'::DG')>0  H\_AWG = eval(INST\_ALL{I}); % AWG handle  %sprintf('AWG found')  end  if strfind(INST\_ALL{I},'::DS')>0  H\_scope = eval(INST\_ALL{I}); % scope handle  %sprintf('Scope found')  end  end  H\_scope.InputBufferSize = 1400000\*3;  H\_scope.Timeout = 3;  H\_AWG.Timeout = 1;  PointsN=6000\*10; % transfer 60k points/channel  fopen(H\_scope);  fopen(H\_AWG);  %% Set AWG parameters  fprintf(H\_AWG, '\*IDN?'); id\_AWG=fscanf(H\_AWG)  fprintf(H\_AWG, ":SOUR1:FUNC SQUare");  %fprintf(H\_AWG,":SOUR1:FUNC SINusoid");  fprintf(H\_AWG,":OUTP1 ON");  Fg=200; % output signal frequency  fprintf(H\_AWG,":SOUR1:FREQ %f", Fg);  Ag=5.0; % output signal amplitude  fprintf(H\_AWG,":SOUR1:VOLT %f", Ag);  pause(0.2);  %% transfer data from the scope  fprintf(H\_scope, '\*IDN?');  id\_scope=fscanf(H\_scope)  %fprintf(ScopeHandler, ':STOP');  fprintf(H\_scope, ':RUN'); % to change MDEPTH the osciloscope must be in RUN mode  fprintf(H\_scope, ':ACQ:MDEP %d', PointsN\*1);  %fprintf(ScopeHandler, ':ACQ:MDEP 1400000'); pause(p1);  fprintf(H\_scope, ':ACQ:MDEP?');  md=str2num(fscanf(H\_scope));  pause(0.5);  %%  fprintf(H\_scope, ':TIMebase:MAIN:SCALe %d',1); %we wrote here.  pause(5);  fprintf(H\_scope, ':STOP'); % to acquire all data  pause(0.5);  [X,Y1]=getRigolDSOData(H\_scope,'CHAN1',-1);  [X1,Y2]=getRigolDSOData(H\_scope,'CHAN2',-1);  pause(0.2);  fprintf(H\_scope, ':RUN');  Ts=X(2)-X(1); Fs=1/Ts;  %% plot data  figure  clf; hold on;  subplot(2,1,1);  plot(X, Y1, 'r');  subplot(2,1,2);  plot(X1, Y2, 'b');  grid on; xlabel('t,sec'); ylabel('U, V');    %% Freq Spectrum    %Signal Y1  SigLen = length(Y1);  %Window=hanning(SigLen);  Window=flattopwin(SigLen);  Ww=mean(Window)  Y1w=Y1;  %SigFFT = fft(Y1w,2^17);  SigFFT = fft(Y1w,2^20);  sigfftLen = length(SigFFT);  %SigFFT = fft(Y1w);  %SigLen = length(Y1);  %SigFFT = fft(Y1);  Sigabs = abs(SigFFT/ sigfftLen);  SigSpectrum1 = Sigabs(1: sigfftLen/2+1);  SigSpectrum1(2:end-1) = 2\*SigSpectrum1(2:end-1);  f1 = Fs\*(0:(sigfftLen/2))/sigfftLen  SigLen = length(Y2);  %Window=hanning(SigLen);  Window=flattopwin(SigLen);  Ww=mean(Window)  Y2w=Y2; %.\*Window/Ww;  %SigFFT = fft(Y2w,2^17);  SigFFT = fft(Y2w, 2^20);  sigfftLen = length(SigFFT);  %SigLen = length(Y2);  %SigFFT = fft(Y2);  Sigabs = abs(SigFFT/sigfftLen);  SigSpectrum2 = Sigabs(1:sigfftLen/2+1);  SigSpectrum2(2:end-1) = 2\*SigSpectrum2(2:end-1);  f2 = Fs\*(0:(sigfftLen/2))/sigfftLen;    figure  ax1=subplot(2,1,1);  plot(f1,SigSpectrum1);  xlabel("freq in Hz")  xlim([0 Fg\*1.5]);  ylabel("Amplitude in V")  title(['Y1 with zeropadding and flattopwin for Fs= ', num2str(Fs)])  ax2=subplot(2,1,2);  plot(f2,SigSpectrum2);  xlabel("freq in Hz")  xlim([0 Fg\*1.5]);  ylabel("Amplitude in V")  title(['Y2 with zeropadding and flattopwin for Fs= ', num2str(Fs)])    %% FFT  % Calculate the sampling frequency  Fs = 1/mean(diff(X));  % Perform FFT  Y\_fft = fft(Y1);  n = length(Y\_fft); % Length of signal  % Calculate amplitude  P2 = abs(Y\_fft/n);  % P1 = P2(1:floor(n/2+1));  % P1(2:end-1) = 2\*P1(2:end-1);  % P1\_dB = 20\*log10(P1);  P2\_dB = 20\*log10(P2);  f = Fs\*(0:n-1)/n; % Frequency axis  idx\_max = find(f <= 1000);  % Plot the amplitude and phase spectrum  figure(3);  plot(f(idx\_max), P2\_dB(idx\_max));  title('FFT');  xlabel('Frequency (Hz)'); ylabel('Amplitude (dB)');  grid on;    %%  fclose(H\_scope); delete(H\_scope);  fclose(H\_AWG); delete(H\_AWG);  save('data1.mat','Fs','X1','Y1','Y2') |

3 lentelė. Antrosios laboratorinio darbo dalies programinis kodas

|  |
| --- |
| clear; clc;  close all;  format compact;  %% 10 - 40 Hz  points = 6000E3;  time\_scale = 4000E-6; % us  for i = 1:4  Fg = i \* 10;  disp("current freq " + Fg)  [CHN1\_X, CHN1\_Y, CHN2\_X, CHN2\_Y] = f\_get\_osciloscope\_data(Fg, points, time\_scale);  if isequal(CHN1\_X, CHN2\_X)  t = CHN1\_X;  end  U\_t = CHN1\_Y; U1\_t = CHN2\_Y;  [R, X] = f\_calc\_impedance(t, U\_t, U1\_t)  Rc(i) = R;  Xc(i) = X;  F(i) = Fg;  end  %% 50 - 200 Hz  points = 600E3;  time\_scale = 4000E-6; % us  for i = 5:20  Fg = i \* 10;  [CHN1\_X, CHN1\_Y, CHN2\_X, CHN2\_Y] = f\_get\_osciloscope\_data(Fg, points, time\_scale);  if isequal(CHN1\_X, CHN2\_X)  t = CHN1\_X;  end  U\_t = CHN1\_Y; U1\_t = CHN2\_Y;  [R, X] = f\_calc\_impedance(t, U\_t, U1\_t);  Rc(i) = R;  Xc(i) = X;  F(i) = Fg;  end  %% Plot linear scale  figure(1)  plot(F, Rc, F, abs(Xc));  legend('Rc', 'Xc');  xlim([10 200]);  grid on; xlabel('f, Hz'); ylabel('Z, ohm');  %% Plot logarithmic scale  figure(2)  semilogx(F, Rc, F, abs(Xc));  xlim([100, 10000]);  grid on; xlabel('f, Hz'); ylabel('ohm');  %% Plot data  figure(3);  clf; hold on;  plot(t\*points, U1\_t, 'r');  plot(t\*points, U\_t, 'b');  legend('U1', 'U');  grid on; xlabel('t,us'); ylabel('U, V'); |