****

**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Automatizuotas impedanso dažninės charakteristikos matavimas**

Laboratorinio darbo ataskaita Nr. 1

|  |  |
| --- | --- |
|  |  |
| **Dominykas Simpukas**  **Erikas Gintilas**  **Žygimantas Marma**  **Egidijus Akramas**  Studentai | (parašas) (data) |
|  |  |
| **Prof. dr. Linas Svilainis**  Dėstytojas | (parašas) (data) |
|  |  |

**Kaunas, 2023**

Turinys

[Paveikslų sąrašas 3](#_Toc154101403)

[1. Įvadas 4](#_Toc154101404)

[2. Praktinė dalis 5](#_Toc154101405)

[2.1. Grandinės schema 5](#_Toc154101406)

[2.2. Duomenys 5](#_Toc154101407)

[2.3. Duomenų apdirbimas 6](#_Toc154101408)

[2.3.1. DFT skaičiavimo metodas 6](#_Toc154101409)

[2.3.2. SWC skaičiavimo metodas 8](#_Toc154101410)

[2.3.3. SWC-OS skaičiavimo korekcija 8](#_Toc154101411)

[2.3.4. SWC-OSL skaičiavimo korekcija 10](#_Toc154101412)

[2.3.5. Teorinis skaičiavimas 11](#_Toc154101413)

[2.4. Kondensatoriaus impedanso rezultatai 12](#_Toc154101414)

[Išvados 15](#_Toc154101415)

[Priedai 16](#_Toc154101416)

[1 priedas. Visas MatLAB programos kodas 16](#_Toc154101417)

[2 priedas. Funkcija gauti kompleksines įtampos vertes 21](#_Toc154101418)

[3 priedas. Visas MatLAB programos kodas 22](#_Toc154101419)

Paveikslų sąrašas

[1 pav. Impedanso matavimo schema. *R1* = 220 Ω; *Cx* = 4,7 μF. 5](#_Toc154101393)

[2 pav. Kondensatoriaus impedanso menamosios dalies (reaktanto) priklausomybė nuo dažnio matuojant skirtingais metodais (kairėje logaritminė skalė, dešinėje – pusiau logaritminė). 12](#_Toc154101394)

[3 pav. Kondensatoriaus pilno impedanso priklausomybė nuo dažnio matuojant skirtingais metodais (kairėje logaritminė skalė, dešinėje – pusiau logaritminė). 13](#_Toc154101395)

[4 pav. Kondensatoriaus impedanso realios dalies priklausomybė nuo dažnio matuojant skirtingais metodais. 13](#_Toc154101396)

[5 pav. Lygiagrečiai su realiu kondensatoriaus prijungtos varžos kitimas nuo dažnio. 14](#_Toc154101397)

[6 pav. Kondensatoriaus talpos priklausomybė nuo dažnio. 14](#_Toc154101398)

# Įvadas

**Darbo uždaviniai**:

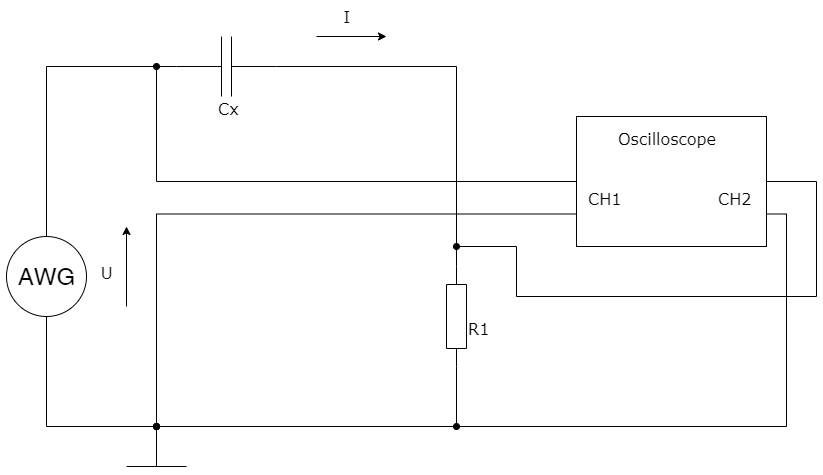
1. Susipažinti su eksperimento aplinka, reikalinga automatiniam matavimui.
2. Pateikti blokinę laboratorinio darbo schemą.
3. Išsaugoti bangų formas iš osciloskopo į failus.
4. Atlikti gautų duomenų apdorojimą.

**Darbo tikslas**: Atlikti automatizuotą impedanso dažninės charakteristikos matavimą nurodytai grandinei.

# Praktinė dalis

## Grandinės schema

Pagrindinė laboratorinio darbo grandinė pateikta žemiau:



1 pav. Impedanso matavimo schema. *R1* = 220 Ω; *Cx* = 4,7 μF.

Pagrindinės dalys šios grandinės yra:

* Signalų generatorius;
* 4,7 μF kondensatorius;
* 220 Ω rezistorius;
* Dviejų kanalų osciloskopas;

Rezistorius ir kondensatorius yra sujungti nuosekliai. Osciloskopo kanalai yra prijungti taip, jog pirmas kanalas rodytų įtampą tenkančią per abu pasyvius komponentus, o antras kanalas matuoja įtampą esantį rezistoriuje.

## Duomenys

Duomenys buvo renkami panaudojus MATLAB programą. Buvo matuojamos dvi įtampos: viena visos grandinės, o kita tik rezistoriaus. Signalų generatoriuje buvo sugeneruotas sinusinis signalas, kurio dažnis kiekvienu matavimu kito sekančiai:

* Nuo 10 Hz iki 100 Hz kas 10 Hz;
* Nuo 100 Hz iki 1000 Hz kas 100 Hz;
* Nuo 1000 Hz iki 10000 Hz kas 1000 Hz.

Iš viso buvo atlikti 28 matavimai.

## Duomenų apdirbimas

### DFT skaičiavimo metodas

Pirmiausia įtampos buvo paverstos į kompleksinius dydžius panaudojus funkciją *f\_DFT\_to\_complex()*. Ši funkcija veikia taip, jog pirmiausia išmatuotus įtampų signalus paskaičiuoja juos diskrečiąja Furjė transformacija (DFT) ir yra gaunamas signalo spektras. Tuomet atrandama didžiausia amplitudės vertė tokio signalo ir kokia yra fazė signalo tuo metu, kai yra didžiausia amplitudė. Turint amplitudę ir fazę galima pagal Eulerio formulę tos signalo įtampos realią ir menamą dalis. Ir galiausiai funkcija pateikia didžiausią įtampos vertę kompleksiniu skaičiumi, kuris bus naudojamas tolimesniuose skaičiavimuose. Taip buvo gautos kompleksinės didžiausios įtampos vertės visos grandinės ir varžos esant skirtingam dažniui.

Kadangi grandinė yra sudaryta iš nuosekliai sujungtos varžos bei kondensatoriaus ir žinoma kondensatoriaus įtampą, tuomet galima paskaičiuoti kompleksinę įtampą ties kondensatoriumi:

(1)

čia: *Uz* – kompleksinė kondensatoriaus įtampa, V;

*Uc* – kompleksinė grandinės įtampa, V;

*U1c* – kompleksinė varžos įtampa, V.

Kadangi grandinė yra nuosekli ir žinant kompleksinę varžos įtampą galima paskaičiuoti pagal Omo dėsnį kompleksinę srovę:

(2)

čia: *Iz* – kompleksinė grandinės srovė, A;

*U1c* – kompleksinė varžos įtampa, V;

*R* – rezistoriaus varža, Ω (šiuo atveju 220 Ω).

Turint grandinės kompleksinę įtampą ir kompleksinę srovę, galima pagal Omo dėsnį paskaičiuoti kondensatoriaus impedansą:

(3)

čia: *Zx* – kondensatoriaus impedansas, Ω;

*Uz* – kompleksinė kondensatoriaus įtampa, V;

*Iz* – kompleksinė grandinės srovė, A.

Kadangi realus kondensatorius yra sudarytas iš lygiagrečiai sujungto kondensatoriaus ir varžos, dėl lengvesnių skaičiavimų tokios grandinės buvo nuspręsta pakeisti kondensatoriaus impedansą į kompleksinį laidumą, nes realios ir menamos laidumo dalys atitinkamai nusako laidumą varžos ir kondensatoriaus. Žinant tai, tolimesni skaičiavimai vyksta taip:

(4)

čia: *Yx* – kompleksinis kondensatoriaus laidumas, S;

*Zx* – kondensatoriaus impedansas, Ω.

Turint kompleksinę varžą gali apskaičiuoti kondensatoriaus varžą ir reaktantą (5, 6 lygtys)

(5)

čia: *Rp* – kondensatoriaus varža, Ω;

*Re(Yx)* – realioji kondensatoriaus laidumo dalis, S.

(6)

čia: *Xp* – kondensatoriaus reaktantas, Ω;

*Im(Yx)* – realioji kondensatoriaus laidumo dalis, S.

Žinant kondensatoriaus reaktanto apskaičiavimą, galima apskaičiuoti kondensatoriaus talpą:

(7)

čia: *Cp* – kondensatoriaus talpa, F;

*f* – signalo dažnis, Hz;

*Xp* – kondensatoriaus reaktantas, Ω.

DFT metodu apskaičiuotas kondensatoriaus varža, talpa ir impedansas bus pažymėti atitinkamai *RDFTp, CDFTp* ir *ZDFT*.

1 lentelė. Algoritmo implementacija MATLAB programa.

|  |
| --- |
| for i = 1:28  file\_name = ['iteration\_' num2str(i) '\_data.mat'];  loaded\_data = load(file\_name);  t = loaded\_data.t;  U\_s = loaded\_data.U\_sign;  U\_1 = loaded\_data.U\_cap;  U\_c(i) = f\_DFT\_to\_complex(t, U\_s);  U1\_c(i) = f\_DFT\_to\_complex(t, U\_1);  end  Uz = U\_c - U1\_c;  Iz = U1\_c ./ 220;  Zx = Uz ./ Iz;  Yx = 1 ./ Zx;  Rp = 1 ./ (real(Yx));  Xp = 1 ./ (imag(Yx));  Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);  R\_DFTp = Rp;  C\_DFTp = Cp;  Z\_DFT = Zx; |

### SWC skaičiavimo metodas

Šioje dalyje vietoj pirmiausia panaudotos DFT funkcijos, buvo panaudota *SWCtruncated()* funkcija kaip pirmasis žingsnis apdirbti signalą. Turint po funkcijos gautas grandinės ir varžos įtampas, tolimesni veiksmai buvo vienodai atlikti, kaip ir 2.3.1 dalyje. Galiausiai kondensatoriaus varža, talpa ir impedansas buvo atitinkamai pažymėti *RSWCp, CSWCp* ir *ZSWC*.

2 lentelė. SWC skaičiavimo metodo implementacija naudojant MATLAB.

|  |
| --- |
| for i = 1:28  file\_name = ['iteration\_' num2str(i) '\_data.mat'];  loaded\_data = load(file\_name);  t = loaded\_data.t;  U\_s = loaded\_data.U\_sign;  U\_1 = loaded\_data.U\_cap;  U\_c(i) = SWCtruncated(t, 2\*pi\*f\_signal(i), U\_s);  U1\_c(i) = SWCtruncated(t, 2\*pi\*f\_signal(i), U\_1);  end  Uz = U\_c - U1\_c;  Iz = U1\_c ./ 220;  Zx = Uz ./ Iz;  Yx = 1 ./ Zx;  Rp = 1 ./ (real(Yx));  Xp = 1 ./ (imag(Yx));  Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);  R\_SWCp = Rp;  C\_SWCp = Cp;  Z\_SWC = Zx; |

### SWC-OS skaičiavimo korekcija

Kadangi grandinė sujungta yra laidais, o jie turi savo impedansus, privaloma atlikti matavimus, kai grandinė yra atvira (*open*) ir užtrumpinta (*short*). Gauti impedansai būtų panaudoti patikslinti rezultatą.

Matavimas buvo atliktas, kaip minima 2.3.2 dalyje, tačiau nebuvo skaičiuojama kondensatoriaus talpa ir varža kaip grandinė yra atvira ir uždara. Buvo tik apskaičiuoti kondensatoriaus impedansai *Zopen* (atvira grandinė) ir *Zshort* (uždara grandinė).

Toliau buvo atliekama impedanso korekcija pagal 8 formulę:

(8)

čia: *ZOS* – *open-short* koreguotas kondensatoriaus impedansas, Ω;

*ZSWC* – SWC metodu apskaičiuotas pirminis kondensatoriaus impedansas, Ω;

*Zshort* – užtrumpintos grandinės kondensatoriaus impedansas, Ω;

*Zopen* – atviros grandinės kondensatoriaus impedansas, Ω;

Kondensatoriaus varža ir talpa buvo apskaičiuota analogiškai kaip ir 2.3.1 dalyje. Galiausiai kondensatoriaus varža, talpa ir impedansas buvo atitinkamai pažymėti *ROSp, COSp* ir *ZOS*.

3 lentelė. SWC-OS skaičiavimas

|  |
| --- |
| % Calculate open impedance  for i = 1:28  file\_name = ['open\_' num2str(i) '.mat'];  loaded\_data = load(file\_name);  open\_data = loaded\_data.M;  t\_open = open\_data(:, 1);  Us\_open = open\_data(:, 2);  Uref\_open = open\_data(:, 3);  U\_c(i) = SWCtruncated(t\_open, 2\*pi\*f\_signal(i), Us\_open);  U1\_c(i) = SWCtruncated(t\_open, 2\*pi\*f\_signal(i), Uref\_open);  end  Uz = U\_c - U1\_c;  Iz = U1\_c ./ 220;  Zx = Uz ./ Iz;  Z\_open = Zx;  % Calculate short impedance  for i = 1:28  file\_name = ['short\_' num2str(i) '.mat'];  loaded\_data = load(file\_name);  short\_data = loaded\_data.M;  t\_short = short\_data(:, 1);  Us\_short = short\_data(:, 2);  Uref\_short = short\_data(:, 3);  U\_c(i) = SWCtruncated(t\_short, 2\*pi\*f\_signal(i), Us\_short);  U1\_c(i) = SWCtruncated(t\_short, 2\*pi\*f\_signal(i), Uref\_short);  end  Uz = U\_c - U1\_c;  Iz = U1\_c ./ 220;  Zx = Uz ./ Iz;  Z\_short = Zx;  Z\_OS = (Z\_SWC - Z\_short) ./ (1 - (Z\_SWC - Z\_short) ./ Z\_open );  Yx = 1 ./ Z\_OS;  Rp = 1 ./ (real(Yx));  Xp = 1 ./ (imag(Yx));  Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);  R\_OSp = Rp;  C\_OSp = Cp; |

### SWC-OSL skaičiavimo korekcija

Kadangi 2.3.3 dalyje *open-short* metodas neatsižvelgia į sistematines paklaidas (osciloskopo kanalų paklaidos, osciloskopo laidų paklaidos, naudojamos 220 Ω varžos paklaida), reikia atlikti *open-short-load* (OSL) matavimą/korekciją.

Pirmiausia buvo prijungtas žinomas 390 Ω impedansas, kuris yra žymimas *Zstdload*, prie grandinės ir išmatuotos įtampos. Atliekami vienodi žingsniai kaip ir 2.3.2 dalyje ir apskaičiuojamas kondensatoriaus impedansas *Zload*. Tuomet pagal 9 formulę galima apskaičiuoti kondensatoriaus impedansą su OSL korekcija:

(9)

čia: *ZOSL* – *open-short-load* koreguotas kondensatoriaus impedansas, Ω;

*ZSWC* – SWC metodu apskaičiuotas pirminis kondensatoriaus impedansas, Ω;

*Zshort* – užtrumpintos grandinės kondensatoriaus impedansas, Ω;

*Zopen* – atviros grandinės kondensatoriaus impedansas, Ω;

*Zstdload* – žinomas impedansas, šiuo atveju 390 Ω;

*Zload* –grandinės su žinoma apkrova kondensatoriaus impedansas, Ω.

Kondensatoriaus varža ir talpa buvo apskaičiuota analogiškai kaip ir 2.3.1 dalyje. Galiausiai kondensatoriaus varža, talpa ir impedansas buvo atitinkamai pažymėti *ROSLp, COSLp* ir *ZOSL*.

4 lentelė. Skaičiavimas SWC-OSL metodu

|  |
| --- |
| Z\_stdload = 390; % Ohm  % Calculate load impedance  for i = 1:28  file\_name = ['load\_' num2str(i) '.mat'];  loaded\_data = load(file\_name);  load\_data = loaded\_data.M;  t\_load = load\_data(:, 1);  Us\_load = load\_data(:, 2);  Uref\_load = load\_data(:, 3);  U\_c(i) = SWCtruncated(t\_load, 2\*pi\*f\_signal(i), Us\_load);  U1\_c(i) = SWCtruncated(t\_load, 2\*pi\*f\_signal(i), Uref\_load);  end  Uz = U\_c - U1\_c;  Iz = U1\_c ./ 220;  Zx = Uz ./ Iz;  Z\_load = Zx;  Z\_OSL = Z\_stdload .\* ((Z\_open - Z\_load) .\* (Z\_SWC - Z\_short)) ./ ...  ((Z\_load - Z\_short) .\* (Z\_open - Z\_SWC) );  Yx = 1 ./ Z\_OSL;  Rp = 1 ./ (real(Yx));  Xp = 1 ./ (imag(Yx));  Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);  R\_OSLp = Rp;  C\_OSLp = Cp; |

### Teorinis skaičiavimas

Norint palyginti kaip kiekvienas impedanso matavimo metodas gerai veikia reikia paskaičiuoti teorinį impedansą, žinant jog kondensatoriaus talpa yra 4,7 μF. Kadangi kiekvienas realus kondensatorius turi lygiagrečiai prijungtą varžą, kurios vertė yra didelė, tam kad toje varžoje (realiame kondensatoriuje) tekėtų maža srovė. Šiame bandyme buvo priimta, jog realaus kondensatoriaus teorinė lygiagreti varža yra 1MΩ. Tuomet reikia paskaičiuoti kondensatoriaus impedansą kaip šių komponentų lygiagretų jungimą. Pirmiausia, reikia suskaičiuoti kondensatoriaus reaktantą:

(10)

čia: *Cx* – žinoma kondensatoriaus talpa, šiuo atveju 4,7 μF;

*f* – signalo dažnis, Hz;

*Xexpected* – teorinis kondensatoriaus reaktantas, Ω.

Turint kondensatoriaus varžą ir reaktantą, galima apskaičiuoti lygiagretaus komponentų jungimo impedansą:

(11)

čia: *Zexpected* – teorinis kondensatoriaus impedansas, Ω;

*Rexpected* – teorinė kondensatoriaus realioji varža, Ω;

*Xexpected* – teorinė kondensatoriaus reaktyvioji varža, Ω.

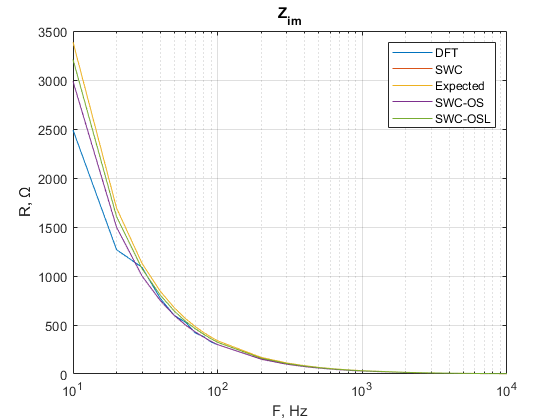
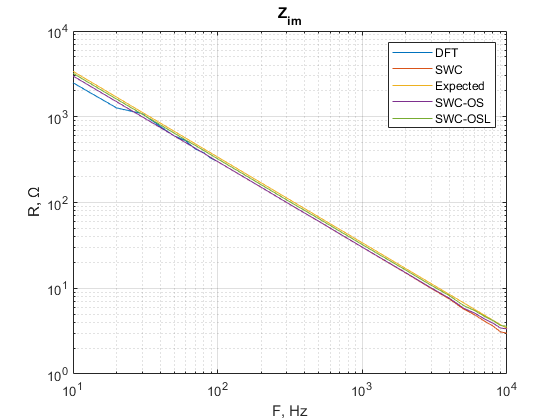
5 lentelė. Varžų apskaičiavimas

|  |
| --- |
| %% expected Zx (Z\_xe)  Cx = 4.7 \* 1E-6 + f\_signal\*0; % 4700 nF  Rx = 1E6 + f\_signal\*0; % Expected resistance 1 Mohm  R1 = 220; % ohm  R\_expexted = 1E6 + 0 .\* f\_signal;  X\_expexted = 1 ./ (1i .\* 2.\*pi.\*f\_signal .\* Cx);  Z\_expexted = 1 ./ ((1 ./ R\_expexted) + (1 ./ X\_expexted)); |

## Kondensatoriaus impedanso rezultatai

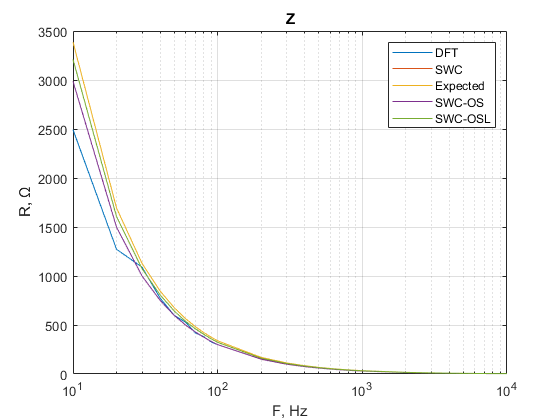
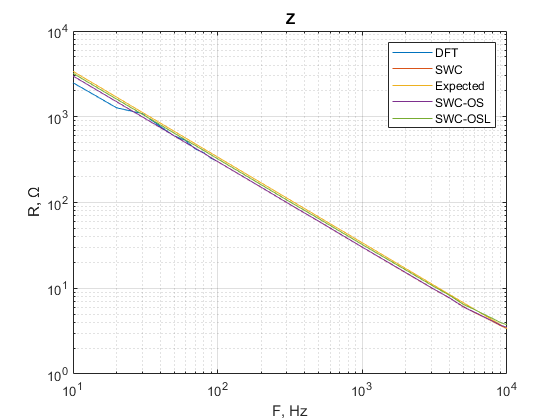
Šioje dalyje bus apžvelgiama, kaip kinta DFT, SWC, SWC-OS, SWC-OSL impedanso skaičiavimo būdai lyginant tarp jų pačių ir teorinės impedanso vertės.

Kondensatoriaus impedanso menamoji dalis yra ganėtinai arti teorinės vertės kitimui nuo 10 iki   
10000 Hz. Visos vertės mažėja, dėl to, nes didėjant dažniui (*f*) mažėja reaktyvioji varža (*Xc*) (10 formulė) ir būtent ji yra pavaizduota kaip impedanso menamoji dalis. Tačiau matoma, jog tarp 10 ir 200 Hz labiausiai yra nukrypęs DFT skaičiavimo metodas, o arčiausias tai SWC-OSL metodu matuojamas impedansas, nes OSL kompensuoja menamąją impedanso dalį. Taip pat, visi impedansai menamojoje dalyje supanašėja didėjant dažniui.



2 pav. Kondensatoriaus impedanso menamosios dalies (reaktyviosios varžos) priklausomybė nuo dažnio matuojant skirtingais metodais (kairėje logaritminė skalė, dešinėje – pusiau logaritminė).

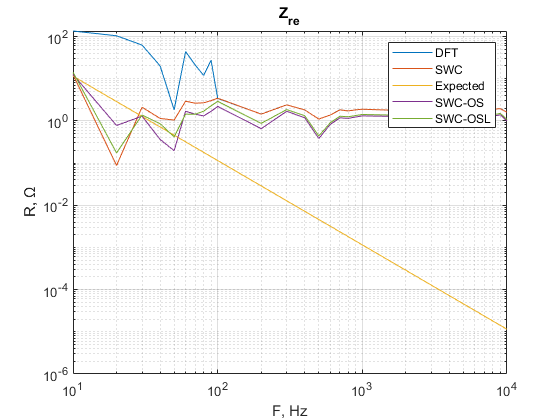
Matuojant bendrą kondensatoriaus impedansą taip pat pastebima, jog grafikai tarp 2 pav. ir 3 pav. mažai kas skiriasi nuo to, kas buvo minėta apie impedanso menamąją dalį.



3 pav. Kondensatoriaus pilno impedanso priklausomybė nuo dažnio matuojant skirtingais metodais (kairėje logaritminė skalė, dešinėje – pusiau logaritminė).

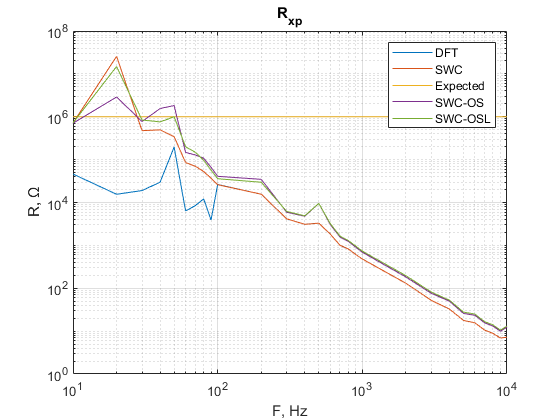
Impedanso matavimai gerokai nukrypo nuo teorinės kondensatoriaus realios dalies impedanso verčių. Nors visi metodai yra panašių verčių (tačiau labai skiriasi nuo kitų DFT metodas nuo 10 Hz iki   
100 Hz) tačiau nuo beveik 60 Hz signalas pradeda skirtis matuotos impedanso vertės lyginant su teorine verte. Arčiausiai teorinės vertės grafiko yra matuoto impedanso kitimas naudojant SWC-OS metodą. OS metodas leidžia kompensuoti realiąją impedanso dalį.

Pagrindinė skirtumo nuo matuotos ir teorinės realios impedansų verčių priežastis yra, jog buvo priimta, kad lygiagrečiai su kondensatoriumi prijungta varža turi 1MΩ impedansą ir yra pastovus kintant dažniui. Tačiau kintant dažniui kinta ir impedansas.



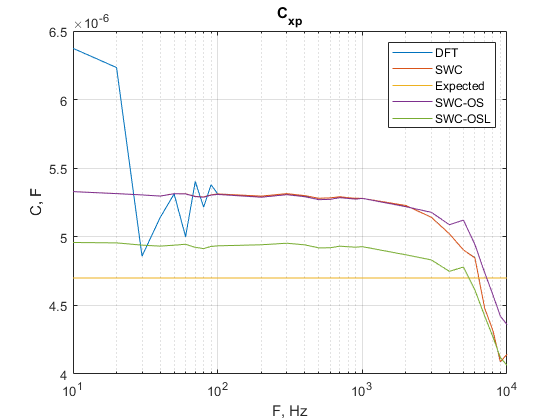
4 pav. Kondensatoriaus impedanso realios dalies priklausomybė nuo dažnio matuojant skirtingais metodais.

Kondensatoriaus lygiagrečiai prijungto varžos vertė gerokai skiriasi nuo išmatuotų nuo panašiai   
600 Hz. Taip yra todėl, nes buvo priimta, jog realaus kondensatoriaus varža nekinta nuo dažnio, nors realybėje varžos gali kisti nuo dažnio. Taip pat grafike pastebima, jog DFT metodas labai skiriasi savo vertėmis iki 100 Hz, o nuo to dažnio DFT metodu impedansas skaičiuojamas panašiai kaip ir kiti metodai.



5 pav. Lygiagrečiai su realiu kondensatoriaus prijungtos varžos kitimas nuo dažnio.

Kondensatoriaus talpa dažnių srityje palaiko savo vertę iki kažkur 1000-2000 Hz, nes po to apskaičiuota talpa yra labai maža. Taip pat grafike matome, kad DFT metodas yra nestabilus   
10-100 Hz ir vertės smarkiai šokinėja. Nors ir išmatuota varža gauta skirtinga, tačiau visi metodai pateikia ganėtinai stabilų kondensatoriaus talpą iki tam tikro dažnio.



6 pav. Kondensatoriaus talpos priklausomybė nuo dažnio.

Išvados

1. Apskaičiuotas impedansas skirtingais metodais. Arčiausiai teorinės vertės buvo SWC-OS metodas apskaičiuojant realiąją dalį, o SWC-OSL, kai buvo skaičiuojama menamoji dalis. Bendrą impedansą irgi arčiausiai teorinės vertės buvo apskaičiuota naudojant SWC-OSL metodą.
2. SWC-OS metodas kompensuoja realioje dalyje pasitaikiusias paklaidas, o SWC-OSL menamojoje dalyje pasitaikiusias paklaidas.
3. DFT metodas yra labiausiai nukrypęs nuo kitų skaičiavimo metodų ir nuo teorinių verčių, ypač žemose dažniuose.

Priedai

1. priedas. Visas MatLAB programos kodas

clear all;

dataFolder = fullfile(pwd, 'short');

addpath(dataFolder);

dataFolder = fullfile(pwd, 'open');

addpath(dataFolder);

dataFolder = fullfile(pwd, 'load');

addpath(dataFolder);

dataFolder = fullfile(pwd, 'Lab\_1');

addpath(dataFolder);

file\_name = ['frequency\_data.mat'];

loaded\_data = load(file\_name);

f\_signal = loaded\_data.f;

for i = 1:28

file\_name = ['iteration\_' num2str(i) '\_data.mat'];

loaded\_data = load(file\_name);

t = loaded\_data.t;

U\_s = loaded\_data.U\_sign;

U\_1 = loaded\_data.U\_cap;

U\_c(i) = f\_DFT\_to\_complex(t, U\_s);

U1\_c(i) = f\_DFT\_to\_complex(t, U\_1);

end

Uz = U\_c - U1\_c;

Iz = U1\_c ./ 220;

Zx = Uz ./ Iz;

Yx = 1 ./ Zx;

Rp = 1 ./ (real(Yx));

Xp = 1 ./ (imag(Yx));

Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);

R\_DFTp = Rp;

C\_DFTp = Cp;

Z\_DFT = Zx;

%% SWC-based Zx (Z\_xSWC)

for i = 1:28

file\_name = ['iteration\_' num2str(i) '\_data.mat'];

loaded\_data = load(file\_name);

t = loaded\_data.t;

U\_s = loaded\_data.U\_sign;

U\_1 = loaded\_data.U\_cap;

U\_c(i) = SWCtruncated(t, 2\*pi\*f\_signal(i), U\_s);

U1\_c(i) = SWCtruncated(t, 2\*pi\*f\_signal(i), U\_1);

end

Uz = U\_c - U1\_c;

Iz = U1\_c ./ 220;

Zx = Uz ./ Iz;

Yx = 1 ./ Zx;

Rp = 1 ./ (real(Yx));

Xp = 1 ./ (imag(Yx));

Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);

R\_SWCp = Rp;

C\_SWCp = Cp;

Z\_SWC = Zx;

%% OS-corrected SWC Zx (Z\_xOS)

% Calculate open impedance

for i = 1:28

file\_name = ['open\_' num2str(i) '.mat'];

loaded\_data = load(file\_name);

open\_data = loaded\_data.M;

t\_open = open\_data(:, 1);

Us\_open = open\_data(:, 2);

Uref\_open = open\_data(:, 3);

U\_c(i) = SWCtruncated(t\_open, 2\*pi\*f\_signal(i), Us\_open);

U1\_c(i) = SWCtruncated(t\_open, 2\*pi\*f\_signal(i), Uref\_open);

end

Uz = U\_c - U1\_c;

Iz = U1\_c ./ 220;

Zx = Uz ./ Iz;

Z\_open = Zx;

% Calculate short impedance

for i = 1:28

file\_name = ['short\_' num2str(i) '.mat'];

loaded\_data = load(file\_name);

short\_data = loaded\_data.M;

t\_short = short\_data(:, 1);

Us\_short = short\_data(:, 2);

Uref\_short = short\_data(:, 3);

U\_c(i) = SWCtruncated(t\_short, 2\*pi\*f\_signal(i), Us\_short);

U1\_c(i) = SWCtruncated(t\_short, 2\*pi\*f\_signal(i), Uref\_short);

end

Uz = U\_c - U1\_c;

Iz = U1\_c ./ 220;

Zx = Uz ./ Iz;

Z\_short = Zx;

Z\_OS = (Z\_SWC - Z\_short) ./ (1 - (Z\_SWC - Z\_short) ./ Z\_open );

Yx = 1 ./ Z\_OS;

Rp = 1 ./ (real(Yx));

Xp = 1 ./ (imag(Yx));

Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);

R\_OSp = Rp;

C\_OSp = Cp;

%% OSL-corrected SWC Zx (Z\_xOSL)

Z\_stdload = 390; % Ohm

% Calculate load impedance

for i = 1:28

file\_name = ['load\_' num2str(i) '.mat'];

loaded\_data = load(file\_name);

load\_data = loaded\_data.M;

t\_load = load\_data(:, 1);

Us\_load = load\_data(:, 2);

Uref\_load = load\_data(:, 3);

U\_c(i) = SWCtruncated(t\_load, 2\*pi\*f\_signal(i), Us\_load);

U1\_c(i) = SWCtruncated(t\_load, 2\*pi\*f\_signal(i), Uref\_load);

end

Uz = U\_c - U1\_c;

Iz = U1\_c ./ 220;

Zx = Uz ./ Iz;

Z\_load = Zx;

Z\_OSL = Z\_stdload .\* ((Z\_open - Z\_load) .\* (Z\_SWC - Z\_short)) ./ ...

((Z\_load - Z\_short) .\* (Z\_open - Z\_SWC) );

Yx = 1 ./ Z\_OSL;

Rp = 1 ./ (real(Yx));

Xp = 1 ./ (imag(Yx));

Cp = 1 ./ (2 .\* pi .\* f\_signal .\* Xp);

R\_OSLp = Rp;

C\_OSLp = Cp;

%% expected Zx (Z\_xe)

Cx = 4.7 \* 1E-6 + f\_signal\*0; % 4700 nF

Rx = 1E6 + f\_signal\*0; % Expected resistance 1 Mohm

R1 = 220; % ohm

R\_expexted = 1E6 + 0 .\* f\_signal;

X\_expexted = 1 ./ (1i .\* 2.\*pi.\*f\_signal .\* Cx);

Z\_expexted = 1 ./ ((1 ./ R\_expexted) + (1 ./ X\_expexted));

%% Plot Z\_re, Z\_im, Z

figure(1)

loglog(f\_signal, abs(real(Z\_DFT)), f\_signal, abs(real(Z\_SWC)), f\_signal, abs(real(Z\_expexted)), f\_signal, abs(real(Z\_OS)), f\_signal, abs(real(Z\_OSL)))

title('Z\_r\_e');

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

figure(2)

loglog(f\_signal, abs(imag(Z\_DFT)), f\_signal, abs(imag(Z\_SWC)), f\_signal, abs(imag(Z\_expexted)), f\_signal, abs(imag(Z\_OS)), f\_signal, abs(imag(Z\_OSL)))

title('Z\_i\_m')

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

figure(3)

loglog(f\_signal, abs(Z\_DFT), f\_signal, abs(Z\_SWC), f\_signal, abs(Z\_expexted), f\_signal, abs(Z\_OS), f\_signal, abs(Z\_OSL))

title('Z')

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

figure(4)

semilogx(f\_signal, abs(imag(Z\_DFT)), f\_signal, abs(imag(Z\_SWC)), f\_signal, abs(imag(Z\_expexted)), f\_signal, abs(imag(Z\_OS)), f\_signal, abs(imag(Z\_OSL)))

title('Z\_i\_m')

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

figure(5)

semilogx(f\_signal, abs(Z\_DFT), f\_signal, abs(Z\_SWC), f\_signal, abs(Z\_expexted), f\_signal, abs(Z\_OS), f\_signal, abs(Z\_OSL))

title('Z')

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

%% Plot R, C

figure(6)

loglog(f\_signal, abs(R\_DFTp), f\_signal, abs(R\_SWCp), f\_signal, abs(Rx), f\_signal, abs(R\_OSp), f\_signal, abs(R\_OSLp))

title('R\_x\_p');

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('R, Ω');

grid on;

figure(7)

semilogx(f\_signal, abs(C\_DFTp), f\_signal, abs(C\_SWCp), f\_signal, abs(Cx),f\_signal, abs(C\_OSp), f\_signal, abs(C\_OSLp))

title('C\_x\_p')

legend('DFT', 'SWC', 'Expected', 'SWC-OS', 'SWC-OSL');

xlabel('F, Hz'); ylabel('C, F');

grid on;

1. priedas. Funkcija gauti kompleksines įtampos vertes

function [U\_complex] = f\_DFT\_to\_complex(t, U)

Ts=t(2)-t(1);

Fs=1/Ts;

U\_fft = fft(U);

n = length(U\_fft); % Length of signal

% Calculate amplitude

P2 = abs(U\_fft / n);

P1 = P2(1:floor(n/2+1));

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

% Compute the phase spectrum (in degrees)

phase = angle(U\_fft) \* (180/pi);

% Plot the amplitude and phase spectrum

[amplitude, index] = max(P1);

phase = phase(index);

[U\_re, U\_im] = pol2cart(deg2rad(phase), amplitude);

U\_complex = U\_re + 1\*i \* U\_im;

end

1. priedas. Visas MatLAB programos kodas

function [U] = SWCtruncated(t,Wtrue,Signal);

%function [U] = SWCtruncated(t,Wtrue,Signal);

%Used to get the harmonic component of the Signal at frequency Wtrue

%Can be used as sinusoid amplitude and phase estimator

%or as a narrowband filter extracting particular frequency component

%Uses sine wave correlation (SWC)

%More details in:

%L.Svilainis, et al. The Automated Complex Impedance Measurement System.

% Electronics and Electrical Engineering. 2007. Nr. 4(76),pp. 59-62.

%Available at:

%http://eejournal.ktu.lt/index.php/elt/article/view/10720

%

% Syntax: [U] = SWCtruncated(t,Wtrue,Signal)

%

% Inputs:

% Signal - (1,N) array of sampled signal values

% t - time axis,(1,N) array, must be same length as Signal

% Wtrue - angular frequency for estimation (2\*pi\*F)

% Outputs:

% U - measured complex amplitude of the sinusoid

%Notes:

%The more periods you record the better is noise or other frequencies

%rejection, the better estimation accuracy;

%Recond length is truncated to aproximately fit the integer number of

%frequency Wtrue periods;

%Time axis t must be sampled equally;

%

% Example:

% %TestSWC

% clear all

% fsampl=100e6;%sampling frequency is 100MHz

% f0=1.3012e6;%sinusoid frequency is 1.3012MHz

% phi=30/180\*pi;%sinusoid phase is 30deg

% Um=0.141;%sinusoid amplitude is 0.141V

% N=100;%record length is 1000 samples

% %make time axis:

% t=(0:(N-1))/fsampl;

% %generate test signal:

% MySignal=Um\*cos(2\*pi\*f0\*t+phi);

% %add some noise:

% MySignal=MySignal+0.01\*randn(1,N);

% %estimate signal amplitude and phase:

% Uest = SWCtruncated(t,2\*pi\*f0,MySignal);

% Umag=abs(Uest)

% Uph\_rad=angle(Uest)

% Uph\_deg=Uph\_rad/pi\*180

% %plot the raw signal and estimated harmonic component:

% figure(1)

% plot(t\*1e6,MySignal,'r',t\*1e6,Umag\*cos(2\*pi\*f0\*t+Uph\_rad),'k')

% xlabel('Time, us')

% ylabel('Amplitude, V')

% grid on

% axis tight

%

% Other m-files required: none

% Subfunctions: none

% MAT-files required: none

%

% Authors: Linas Svilainis

% Kaunas University of Technology, Dept. of Electronics Engineering,

% Kaunas, Lithuania

% email address: linas.svilainis@ktu.lt

% December 2006; Last revision: 10-Jul-2018

% Copyright: Linas Svilainis

%------------- BEGIN CODE --------------

%estimate the sampling period from time vector:

dt=t(2)-t(1);

%get the record length:

Trec=max(t)-min(t);

%convert angular frequency to linear frequency

F=Wtrue/(2\*pi);

perNo=floor(Trec\*F);

SignalLength=perNo/(dt\*F);

SuppliedSignalLength=length(Signal);

if (floor(SignalLength)<SuppliedSignalLength)&(perNo>0),%truncate only long signals, longer than 1 period

SignalLength=floor(SignalLength);

else

SignalLength=SuppliedSignalLength;

end;

%fit using SWC

ss=sum(sin(t(1:SignalLength)\*Wtrue).\*Signal(1:SignalLength));

Scor=ss/SignalLength;

ss=sum(cos(t(1:SignalLength)\*Wtrue).\*Signal(1:SignalLength));

Ccor=ss/SignalLength;

U=2\*(Ccor-j\*Scor);

% figure(5)

% plot(Signal(1:SignalLength),'LineWidth',3)

end

%------------- END OF CODE --------------