# Testovacie scenáre

Pri používaní spektrálneho analyzátora môže dôjsť k rôznym situáciam. Tomuto sa venuje kapitola testovacie scenáre, ktorá popisuje rôzne prípady, aké môžu nastať počas merania na spektrálnom alanyzátore pri použití SVOSA. Pri spustení programu sa najskôr inicializujú triedy (vtvoria sa inštancie objektov) Agilent a RLCparams aby sa s nimi dalo ďalej pracovať, nachádzajúce sa v samostatných jednotkách (agilent.py, respektíve rlcparams.py). Nasleduje popis tried a priebeh testovania v rámci behu kódu:

**rlcparams**

Triede RLCparams sa veľmi do podrobna venovať nebudem. Súvisí s hlbšou matematikou a fyzikou. Nachádzasa tu konštruktor triedy kde si inicializujeme atribúty triedy na nulové(prázdne hodnoty). Atribúty frequency, impedance, admitance, vyrez, vyref, vyrezimag, vyref\_fit, yre\_fit, yim\_fit sú štrukturované ako polia (lists) ďalej tu sú atribúty: R1, C0, C1, L1, fResonance, fResonanceFit, gnuplot, gnuplot1 , gnuplot2 (objekty Gnuplot.Gnuplot()) a timeStart. Trieda má metódy : data(self, frequency, impedance) - nastavenie dát; fit(self) - výpočet rezonancie pomocou fitovacej funkcie; metóda plot(self) - metóda na grafické zobrazenie celej krivky a výrezu a metódy plotParams(self, dataFile); plotAll(self, dataFile); plot\_re(self) ; plot\_im(self); saveParams(self, fileName = '') a load(filename). Táto trieda je plne funkčná a preto ju nebude treba špeciálne testovať.

**agilent**

V triede agilent má 3 základné atribúty = gnuplot = None a 2 atribúty typu slovník: data = {} a frequency = {}. Ako prvú vec si pri inicializácii vytvoríme objekt agilent\_scpi, ktorý inicializujeme. Táto trieda sa nachádza v jednotke(unite) agilent\_scpi.py a nach=adzaj=u sa v nej metódy určené priamo pre komunikáciu so SA : metóda na odosielanie príkazov(povelov): send\_cmd, metóda na čítanie prichádzajúcich dát zo SA - read\_data a metóda na zber chybových hlášok(errorov) - read\_err .Trieda má 3 atribúty: \_host\_ip - tu sa nastavuje a ukladá ip adresa prístroja ('192.168.1.102'), atribút \_scpi\_port - kde sa nastavuje komunikačný port (5025) a atribút \_connection: kde sa pomocou knižnice socket nastaví vhodný protokol, kde sa zadá ip adresa a port prístoja. Pri inicializovaní triedy si zadefinujeme v slovníku data pojmy: ['frequency'], ['Z'], ['impedance'], ['admitance'] a ['phase']. Ďalej ideme nastavovať SA prvým príkazom pomocou send\_cmd a tým je prýkaz ':syst:pres' (preset) ktorý nastavý SA na pôvodné továrenské nastavenie kde je nastavený aj východiskový kanál, ako kanál 1. Ďalej nastavujeme formát dát z viacerývh možných budeme v testoch používať formát typu Smythovho diagramu (R+jX) pomocou commandu ':calc1:form smit'. Potom nastavujeme východiskový režim na nepretržitý pomocou volania príkazu ':init1:cont on' a následne sa zavolá abort pomocou príkazu ':abor' , ktorý zastaví merania a zmení spúšťacie poradie pre všetky kanály do kľudového stavu. Ďalej nastavíme počet meracích bodov (hustota zobrazovania nameraných údajov - sweeping point) v ukážkovom teste na maximálnu možnú hodnotu pre náš typ SA, pre väčšiu presnosť merania a tou je 1601 bodov (minimum sú 2) príkazom ':sens1:swe:poin 1601'. V ďaľšiom kroku nastavíme formát pre prenos dát na ASCii formát vďaka príkazu ':form:data asc' a povieme analyzátoru aby počkal na vykonanie všetkých príkazov, ktoré boli doposiaľ analyzátoru zaslané. Nasleduje opýtanie sa na počiatočnú hodnotu sweep rangu - krivky(grafu)zobrazujúcej výsledky merania (pýtame sa na frekvenciu, čiže výslednú hodnotu dostávame v Hertzoch) pomocou commandu ':sens1:freq:star?' a načítame túto odpoveď do slovníka frequency ako jeho zložku ['start'] ako reálne číslo typu float pomocou funkcie read\_data(). Takisto sa opýtame na koncovú hodnotu frekvencie príkazom ':sens1:freq:stop?' a toto si zapíšeme do slovníka frequency a jeho zložky ['stop'] podobným spôsobom ako pri start. Ďalej nasleduje samotná inicializácia Gnuplotu (triedy pre vizuálne zobrazenie pomocou grafov - Tejto triede sa budem venovať neskôr).

metóda set\_frequency(self, start='', stop='', center='', span=''):

Najprv nastavíme SA vstupné parametre vďaka funkcii send\_cmd a to tak, že pre start použijeme príkaz ':sens1:freq:star ' + start - (teda k nemu rovno "prilepíme" samotnú číselnú hodnotu pre štartovaciu rekvenciu zo vstupnej premennej start). Nastavenie končiacej frekvencie: ':sens1:freq:stop ' + stop , nastavenie stredovej hodnoty pre sweep range (meranie) pomocou príkazu ':sens1:freq:cent ' + center , a nakoniec nastavenia rozpätia pre sweep range vďaka príkazu ':sens1:freq:span ' + span . Nasleduje príkaz '\*wai' , teda povieme analyzátoruaby počkal až kým sa nevykonajú všetky zadané príkazy a vypýtame si od SA nové štartovacie a koncové hodnoty frekvencií a uložíme si ich do slovníka data :

self.send\_cmd(':sens1:freq:start?') , self.frequency['start'] = float(self.read\_data())

self.send\_cmd(':sens1:freq:stop?'), self.frequency['stop'] = float(self.read\_data())

metóda measure(self):

Metóda pre samotné meranie. V prvom kroku si vyprázdnime celý slovník data (vyprázdnime jeho položky ['frequency'],['Z'],['impedance'],['admitance'] a ['phase']). V ďalšiom kroku nastavýme režim SA na hold, teda vypnutie kontinuálneho režimu príkazom: ':init1:cont off' . Počkáme kým sa tak udeje ( príkaz \*wai) a pre aktívnu stopu kanála 1 prečítame formátované dátové polia vo formáte, aký bol predtým nastavený, teda v našom prípade to je ASCI, pomocou príkazu ':calc1:data:fdat?' a tieto dáta ukladáme do slovníka data a budeme ich ďalej spracovávať. Opäť sa spýtame na začiatočné a koncové frekvencie a hodnoty si aktualizujeme v slovníku frquency, počkáme a nastavíme režim SA naspäť do kontinuálneho meracieho režimu ':init1:cont on' .

Nasleduje spracúvavanie dát. Najprv ich rozdelíme pomocou metódy split, odďeľovačom čiarkou na pole stringov. vypočítavame hodnotu gamma pre každú rozdelenú hodnotu z dát = complex(float(data[i]), float(data[i+1])) ďalej pridávame jednotlivé frekvencie, získavame hodnoty čísla Z , impedanciu, admitanciu a fázu a všetko si to ukladáme do jednotlivých zložiek slovníka data.

metóda plot\_data(self):

Metóda triedy agilent zobrazujúca dáta pomocou Gnuplot u do formy grafu.

metóda save\_data(self, fileName=''):

Pokiaľ nie je zadaný názov súboru, do ktorého chceme dáta ukladať, metóda neurobí nič, inak nám uloží dáta do nového súboru, ktorý pomenuje podľa vstupného parametra fileName a to nasledovne : vynechá dva riadky, zapíše # a 140 krát pomlčku (oddeľovač), do ďalšieho riadku zapíše - '# frequency\t\t| Z.real\t\t| Z.imag\t\t| |Z|\t\t\t| phase\t\t\t| admitance\n' , nasleduje ďalší oddeľujúci riadok a potom cyklicky zapisujeme ako do tabuľky všetky hodnoty slovníka data v poradí frekvencia, Z - imaginárna, Z - reálna časť, impedancia, fáza a admitancia pekne po riadkoch až do konca a zavrieme súbor - datafile.close().

**Popis validného testovacieho scenára:**

Meranie sa spúšťa v jednotke(unite) main.py. Vytvoríme objekt a = agilent a p = rlcparams. Nastavíme frekvenciu pomocou metódy objektu a (funkcie) set\_frequency s hodnotami pre center = '8000000' a pre span = '500000' . Voláme metódu measure() = meraj. A po vykonaní celého merania zavoláme metódu plot\_data() na zobrazenie výsledku merania. Do premennej m si uložíme maximálnu hodnotu admitancie - vytiahneme ju príkazommax(a.data['admitance']) zo slovníka data. Zapamätáme si index maximálnej admitancie do premennej maxIndex a nastavíme novú stredovú (center) frekvenciu na hodnotu frekvencie zo zložky frequency v slovníku data pod indexom maxIndex - maximálnej admitancie : a.set\_frequency(center = '%f' % a.data['frequency'][maxIndex]) . Nastavíme aj nové rozpätie merania (span) na hodnotu 2000000 : a.set\_frequency(span = '2000000') . Opäť voláme metódu measure() = meraj a zobrazíme výsledné dáta po jej ukončení.

Nasleduje podobný proces, teda nájdeme si znova najväčšiu admitanciu z nového merania a zapamätáme si jej index a prenastavíme stredovú hodnotu merania na hodnotu frekvencie pod týmto indexom. Nastayme rozpätie na 40 000 Hertzov a v cykle spúšťame opäť meranie pomocou metódy a.measure() . Ak chceme uložíme dáta, inak pokračujeme a voláme metódu data inštancie objektu p (rlcparams) a následne voláme fitovaciu funkciu - taktiež metódu objektu p, v ktorej pomocou zložitých matematických funkcií a pravidiel dostávame optimálny (maximálny) výrez impedancie. Výsledok vypíšeme ako string pomocou príkazu: print str(p.fResonanceFit) + '\t' + str(p.R1) + '\t' + str(p.C0) + '\t' + str(p.C1) + '\t' + str(p.L1).

**Ďalšie možné testovacie scenáre:**

Chceli by sme taktiež otestovať viaceré možné chybové hlášky a to napríklad chybnou inicializáciou triedy agilent : nastavením iného typu diagramu, ako smith chart, napríklad PHAS - ktorý nastavý formát na fázový formát (':calc1:form phas), alebo REAL - reálny formát (':calc1:form real'). ďalej nastaviť sweeping point cez povolenú hranicu 1601 bodov a aj pod najnižšiu povolenú hranicu 2 body - ':sens1:swe:poin 1800' / ':sens1:swe:poin 1'. Vyskúšať nastaviť iný formát dát ako ASCii pomocou príkazu ':form:data \*' na real a real32 (sú iba 3 možné formáty). Ďalej by sme sa chceli zamerať v testoch na nastavenie frekvencií a to presiahnutím rozmedzia pre center aj span (stred a rozpätie) center má rozmedzie od 3E5 do 3E9 (podľa mojich odhadov to znamená 3 na 5 až 3 na 9) TO DO zistiť upresniť, a pre span je to rozmedzie od 0 do 2.9997E9.

Potom by sme sa radi zamerali na série meraní s valídnymi hodnotami - v rámci rozmedzia, ale prenastavenými podľa náhodného výberu - vďaka random generatoru a sledovať výsledky a priebeh meraní.