

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra mikroelektroniky

Syntéza pásmových propustí LC pomocí evolučních algoritmů

Vedoucí práce: Doc. Ing. Pravoslav Martinek, CSc.

Student: Bc. Vít Ženíšek

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. ŽENÍŠEK Vít**

Studijní program: Komunikace, multimédia a elektronika
Obor: Elektronika

Název tématu: **Syntéza pásmových propustí LC pomocí evolučních algoritmů**

Pokyny pro vypracování:

1. Prostudujte a ověřte aplikace diferenčních evolučních algoritmů v syntéze pasivních LC filtrů typu pásmová propust.
2. Sestavte návrhový algoritmus a implementujte ho v matematickém programu MAPLE.
3. Zhodnoťte možnosti vícekritériální optimalizace návrhu, zejména z hlediska snížení rozptylu hodnot parametrů obvodových prvků a kompenzace vlivu ztrát reálných induktorů a kapacitorů.
4. Návrhový algoritmus ověřte na příkladech zadaných vedoucím práce.

Seznam odborné literatury:

1. Oplatková, Z., et al: Evoluční výpočetní techniky - principy a aplikace. Nakladatelství BEN, Praha, 2008, ISBN 80-7300-218-3.
2. Coufal, J.: Systém pro návrh filtrů LC na bázi evolučních algoritmů. Diplomová práce, ČVUT FEL, K13131, Praha, 2011.

Vedoucí: **Doc.Ing. Pravoslav Martinek, CSc.**

Platnost zadání: 28. 2. 2015

L.S.

Prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 17. 9. 2013

Čestné prohlášení:

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum: 12. 5. 2014

.....

podpis studenta

Anotace

Tato diplomová práce se zabývá syntézou pásmových propustí LC pomocí evolučních algoritmů. Možnost aplikace diferenčního evolučního algoritmu je otestována na příkladu, který byl zadán vedoucím práce. Diplomová práce se dále věnuje možnosti vícekriteriální optimalizace návrhu z hlediska snížení rozptylu hodnot parametrů obvodových prvků a z hlediska kompenzace vlivu ztrát reálných induktorů a kapacitorů.

Summary

This diploma thesis deals with the synthesis of LC bandpass filters using evolutionary algorithms. Possibility of application of differential evolution algorithm is tested on the example that was specified supervisor. The diploma thesis also describes the possibility of multi-criteria optimization of the design in terms of reducing the variance values of the parameters of the circuit elements and in terms of compensation due to loss of real inductors and capacitors.

Poděkování

Tímto bych chtěl poděkovat vedoucímu mojí diplomové práce, panu Doc. Ing. Pravoslavu Martinkovi, CSc. za množství rad a konzultací, které mi velmi pomohly při tvorbě této diplomové práce.

OBSAH

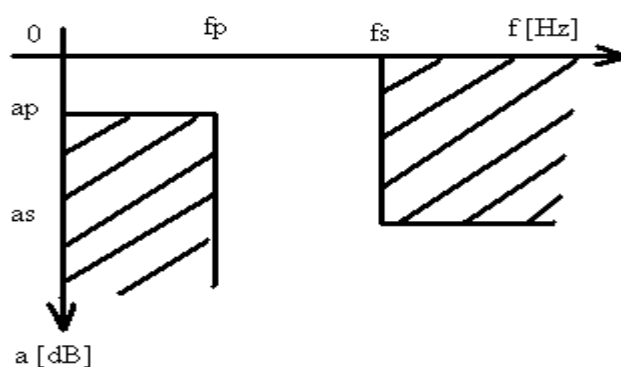
1 Úvod	1
2 Ověření funkčnosti diferenčních evolučních algoritmů v syntéze pasivních LC filtrů typu pásmová propust	4
3 Genetické a evoluční algoritmy – teoretický úvod, sestavení návrhového algoritmu a jeho implementace v programu Maple	9
3.1 Genetické algoritmy	9
3.2 Evoluční algoritmus.....	9
3.3 Diferenční evoluční algoritmus	9
3.4 Sestavení návrhového algoritmu a jeho implementace v programu Maple.....	11
4 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků	17
4.1 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení zakončovacích odporů	17
4.2 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí pomocných proměnných Kon_L, Kon_C.....	21
4.3 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L jako konstanty a optimalizace R1, R2 a C.....	25
4.4 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L na konstantní hodnotu. Optimalizace R1 a C. Výpočet R2 ze vzorce pomocí kk (0,25 0,5 0,75).....	28
4.5 Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L na konstantní hodnotu. Optimalizace R1, R2 a C + Hledání optimálních hodnot zakončovacích odporů pomocí parametru δ_{kk}	33

5	Kompenzace vlivu ztrát reálných induktorů a kapacitorů	39
5.1	Zapojení zadané pásmové propusti se ztrátami v induktorech	39
5.2	Zapojení zadané pásmové propusti se ztrátami v kapacitorech	51
5.3	Zapojení zadané pásmové propusti se ztrátami v induktorech i v kapacitorech	60
6	Závěr	65
	Použité zkratky	67
	Seznam rovnic	68
	Seznam tabulek	68
	Seznam obrázků	69, 70, 71, 72
	Přílohy	72
	Seznam použité literatury	73

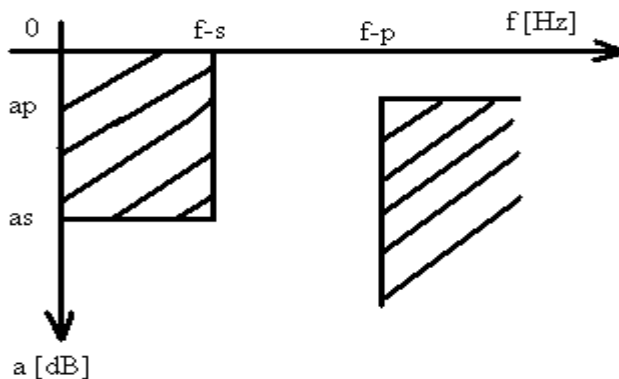
1) Úvod

Frekvenční filtr bychom si mohli představit jako elektronický obvod, který je součástí elektronického systému a má za úkol upravovat frekvenční přenosovou charakteristiku systému. Pod souslovím upravovat frekvenční přenosovou charakteristiku si můžeme představit to, že nám filtr propustí na výstup obvodu jen některé pásmo kmitočtů. Z tohoto hlediska se tedy na kmitočtové filtry můžeme podívat jako na dolní, horní a pásmovou propust a na pásmovou zadrž.

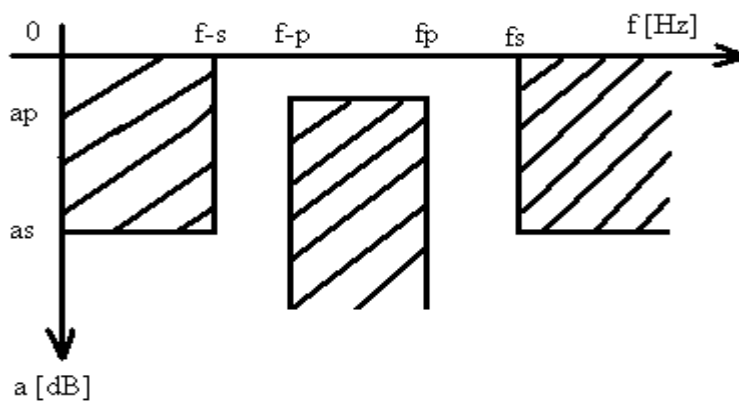
Obr 1.1 - ukázka dolní propusti



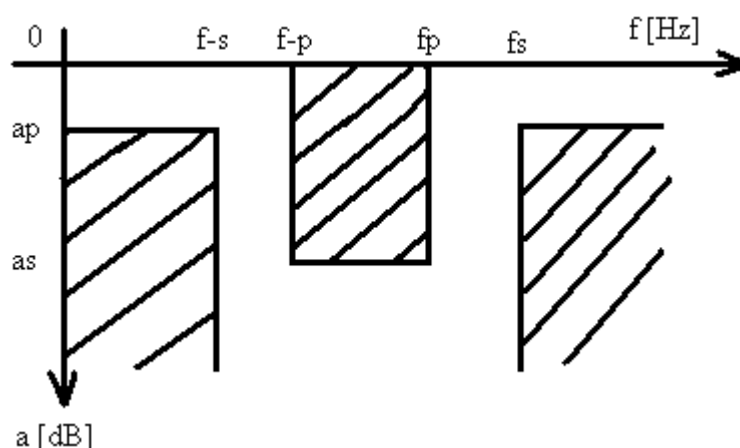
Obr 1.2 - ukázka horní propusti



Obr 1.3 – ukázka pásmové propusti



Obr 1.4 – ukázka pásmové zadržky



V této práci se budu zabývat především pásmovými propustmi. Pásmové propusti jsou jedním z nejčastěji používaných filtrů a to jak v oblasti velmi vysokých frekvencí, tak i v pásmu středních kmitočtů. V řadě případů je výhodné realizovat je jako pasivní filtry, nejčastěji s piezoelektrickými rezonátory, nebo jako LC struktury. V obou případech však návrh může vycházet z LC prototypu. Problematický je zejména návrh úzkých pásmových propustí, kde standardní transformace NDP - PP vede k nepřijatelnému rozptylu hodnot kapacit a indukčností v rezonančních obvodech. Tento problém mohou částečně řešit některé speciální transformace (Nortonova, struktury s reaktančními invertory a další). Zde však platí řada omezení. Mým úkolem je řešit návrh modifikovaných struktur PP a to na bázi diferenčních evolučních algoritmů, které umožňují řešit návrh komplexně, včetně zahrnutí ztrát v pasívních prvcích.

Optimalizace filtru se může při jeho vyšším řádu stát velmi složitým matematickým problémem. Je potřeba hledat další možnosti optimalizace a syntézy a jednou z nich jsou také evoluční algoritmy. Mým úkolem bylo zaměřit se na pásmové propusti typu LC. Ověřit na nich aplikaci diferenčního evolučního algoritmu pomocí programu Maple. Zhodnotit možnosti vícekritériální optimalizace z hlediska snížení rozptylu hodnot obvodových prvků a z hlediska kompenzace vlivu ztrát reálných induktorů a kapacitorů.

Práci jsem rozdělil do šesti kapitol a jejich podkapitol, názvy a umístění jsou zaznamenány v obsahu.

Kapitola 1 se zabývá základním teoretickým úvodem kolem frekvenčních filtrů. Zmiňuji se v ní dále o problémech optimalizace frekvenčních filtrů a uvádím zde, co bylo úkolem práce a popisuji její rozdělení do jednotlivých kapitol.

V kapitole 2 se budu věnovat teoretickému úvodu do problematiky genetických a evolučních algoritmů. Lehce zde popíši vývoj a principy jednotlivých typů genetických algoritmů. Práce není určena k podrobnějšímu popisu této problematiky, takže zde zmíním jen základy a k podrobnějšímu popisu odkáži na odbornou literaturu, která se problematice genetických algoritmů věnuje. Následně naváži na tento teoretický úvod podrobnějším popisem diferenčního evolučního algoritmu, který byl použit v této diplomové práci.

Kapitolu 3 jsem pojal jako ověření funkčnosti diferenčních evolučních algoritmů v syntéze pasivních LC filtrů typu pásmová propust. Na konkrétním zadání jsem se pokusil sestavit návrhový algoritmus a implementoval jsem jej v programu Maple.

Kapitola 4 se věnuje možnosti vícekritériální optimalizace z hlediska snížení rozptylu hodnot obvodových prvků. V jednotlivých podkapitolách se rozebírají různé možnosti snížení rozptylu hodnot obvodových prvků. U každé podkapitoly jsou vloženy průběhy konvergence, porovnání zadané a vypočtené přenosové charakteristiky a zhodnocení výsledků.

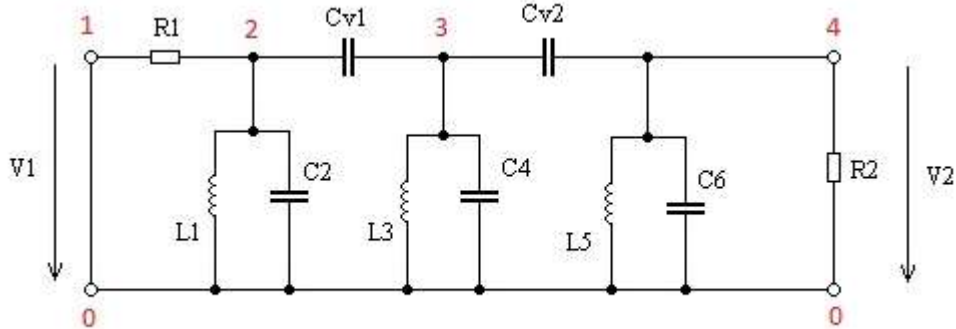
Kapitola 5 pokračuje v pokynech zadání práce. Opět je v ní obsažena možnost vícekritériální optimalizace, tentokrát však z hlediska kompenzace vlivu ztrát reálných induktorů a kapacitorů. Nejprve se zde k zadanému obvodu přidají ztrátové prvky. Provede se optimalizace. Zhodnotí se situace a posléze se budu věnovat možné kompenzaci vlivu připojení těchto ztrátových prvků.

Kapitola 6 obsahuje závěr diplomové práce. V jejím obsahu zhodnotím výsledky diplomové práce, její přínos, případně, jak by se dalo na diplomovou práci navázat.

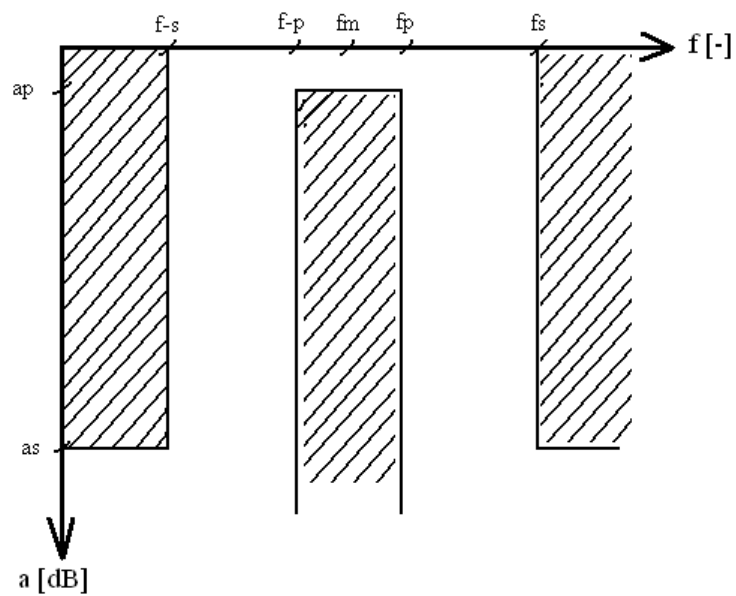
2) Ověření funkčnosti diferenčních evolučních algoritmů v syntéze pasivních LC filtrů typu pásmová propust

Po dohodě s vedoucím práce jsem řešil návrh pásmové propusti 6. řádu podle obr. 2.1

Obr 2.1 – zadaná pásmová propust 6. řádu



Obr. 2.2 – toleranční schéma zadané pásmové propusti



Tab. 2.1 – hodnoty tolerančního schématu zadané pásmové propusti

a_p [dB]	a_s [dB]	f_s [-]	f_p [-]	f_p [-]	f_s [-]	f_m [-]	Δf_p [-]	Δf_s [-]
1	26	0.801	0.951	1.051	1.248	1	0,1	0.447

Tomuto tolerančnímu schématu vyhovuje Čebyševova aproximace se zvlněním propustného pásma 1 dB. Výsledná přenosová funkce je dána vzorcem.

Rovnice 2.1 – obecná přenosová funkce zadané pásmové propusti

$$H(p) = \frac{s^3}{a6 * s^6 + a5 * s^5 + a4 * s^4 + a3 * s^3 + a2 * s^2 + a1 * s^1 + a0 * s^0}$$

Rovnice 2.2 – přenosová funkce zadané pásmové propusti po doplnění koeficientů

$$H(p) = \frac{0,120961p^3}{p^6 + 0,619p^5 + 118,894p^4 + 49,019p^3 + 4,693 * 10^3p^2 + 964,978p + 6,149 * 10^4}$$

Tyto výsledky jsou z programu Syntfil, což je program pro návrh filtrů používaný při výuce na katedře obvodů FEL ČVUT v Praze. Podrobnější informace o prvotním návrhu v Syntfilu jsou k nahlédnutí na konci této diplomové práce v příloze s názvem Návrh v programu Syntfil.

K ověření funkčnosti diferenčního evolučního algoritmu na filtr typu pásmová propust bylo nejdříve potřeba zadat do konkrétního diferenčního evolučního algoritmu zadaný obvod, jeho přenos a jeho modulovou charakteristiku v symbolické formě. K zadání těchto hodnot jsem využil knihovnu Syrup. Knihovna Syrup je knihovna sloužící k symbolické analýze elektrických obvodů.

Pro získání přenosu je potřeba získat hodnoty napětí na vstupu a výstupu obvodu. K tomu je knihovna Syrup ideální, protože obvod je do programu zadávám pomocí uzlových bodů.

V těchto bodech se pak zjistí příslušná hodnota napětí a přenos se vypočítá jako

Rovnice 2.3 – vztah pro výpočet přenosu

$$H = \frac{V_{out}}{V_{in}}$$

V mém zadání vypadal zápis obvodu do programu Maple následovně:

```
with(Syrup) :  
obvod1 := "  
V    1 0  
R1   1 2  
L1   2 0  
C2   2 0  
CV1  2 3  
L3   3 0  
C4   3 0  
CV2  3 4  
L5   4 0  
C6   4 0  
R2   4 0  
.end":
```

Přenos se tedy v tomto případě vypočítá jako podíl napětí mezi body (4-0) a (1-0) – viz Obr. 2.1 na předchozí straně.

Pro možnost ověření funkce programu na filtry typu pásmová propust bylo nezbytné podrobné prostudování informací o účelové funkci a o vyhodnocovací proceduře fitval. Vyhodnocovací procedura fitval je jednou z nejdůležitějších částí programu. Slouží k ohodnocení kvality jedince. Ohodnocení kvality probíhá tak, že se na začátku procedury fitval načtou hodnoty jednotlivých cívek a kondenzátorů, pomocí knihovny Syrup se vypočítá testovací přenosová funkce Htest a vypočtou se odchylky testovací a zadané přenosové funkce. Procedura fitval nabízí širokou možnost vložení optimalizačních podmínek. V předchozích řádcích jsem psal o podmínce minimálních odchylek přenosových charakteristik. Já se v mé diplomové práci zaměřím na další optimalizační podmínky. Konkrétně na snížení rozptylu hodnot obvodových prvků a na kompenzaci vlivu ztrát reálných induktorů a kapacitorů.

Dále jsem se zaměřil na správné nastavení konstant, jako jsou konstanta křížení CR a mutační konstanta F. Přehled o nastavení těchto konstant jsem získal v literatuře [2]. Další nastavení důležitých hodnot je v sekci vstupní data, kde jsem nastavil počet členů v populaci NP=90, maximální počet generací G=400, počet neznámých prvků ND=8 a nakonec horní a dolní mez, ve které se mají pohybovat výsledky – Mh=10 a Md=0,00001.

Díky zabudování vazebních kondenzátorů C_v bylo potřeba přidat pomocnou konstantu $pom3$, aby bylo možné usměrnit hodnoty C_v cca 10x menší než hodnoty $C_{2,4,6}$. V sekci výpočet testovacích vektorů se po-té musel přidat cyklus, který měl za úkol přiřadit vazebním kondenzátorům vypočtená čísla a usměrnit jejich hodnotu na již výše zmíněný vztah mezi C a C_v . Nově napsaný cyklus vypadal takto:

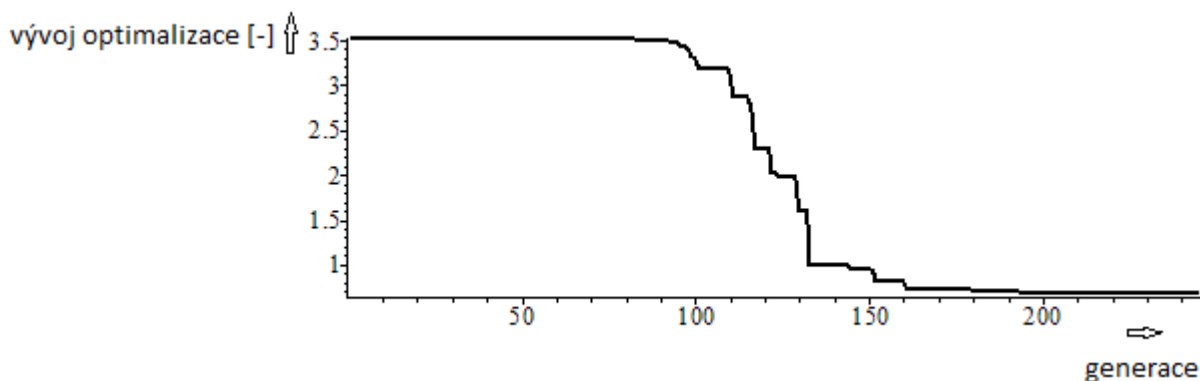
```
for j from NF+1 to ND do
aj:= r():
if aj<CR then U[i][j]:= v[i][j]
else U[i][j]:= X[i][j]
end if;
if U[i][j]<Md then U[i][j]:= 1e-6
elif U[i][j]>Mh then U[i][j]:= pom3()/100
else U[i][j]:= U[i][j]
end if;
end do;
```

Po spuštění programu byla hodnota ohodnocení nejlepšího vektoru po 240 generacích **fit_best = 4.842003443**, což značí dobrý výsledek. Výsledný nejlepší vektor X_{best} měl následující složení – $X_{best}=[8,115909788, 0,002911412269, 6,448794464, 0,003456040366, 7,987994562, 0,002945498907, 0,0002602467637, 0,0002363343150]$

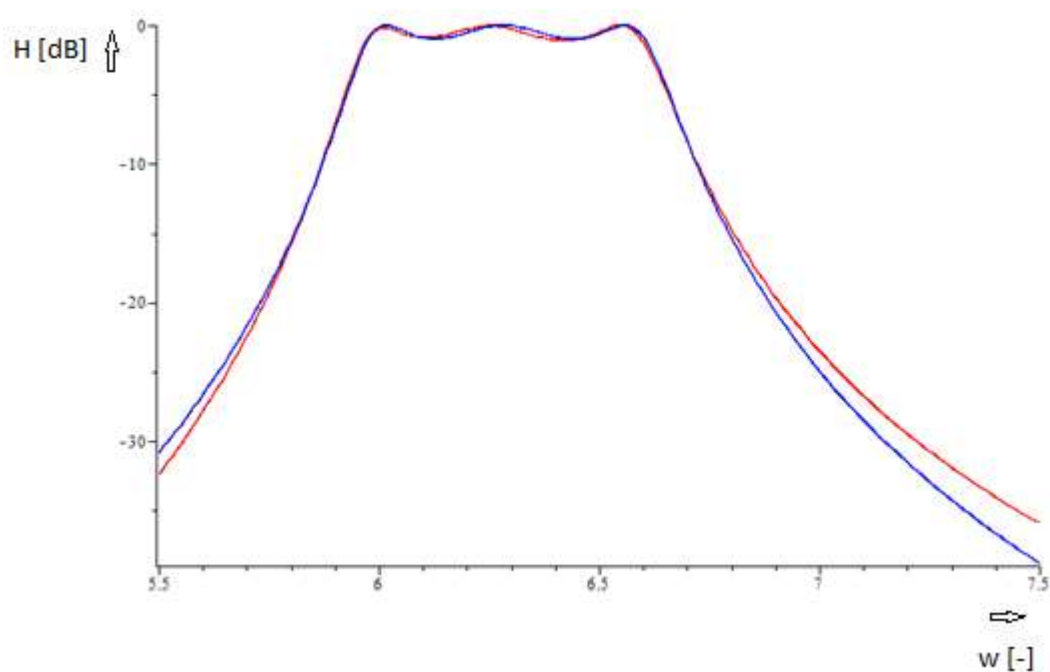
Přiřazení těchto hodnot k jednotlivým prvkům pásmové propusti:

```
L1 := 8.115909788
C2 := 0.002911412269
L3 := 6.448794464
C4 := 0.003456040366
L5 := 7.987994562
C6 := 0.002945498907
CV1 := 0.0002602467637
CV2 := 0.0002363343150
```

Obr. 2.3 - průběh konvergence pro ověření funkce algoritmu na filtr typu PP:



Obr. 2.4 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) pro ověření funkce algoritmu na filtr typu PP:



Výsledky ukazují, že program v první stovce generací lehce stagnoval, ale po-té již začal výrazně konvergovat. Během 60 generací se ohodnocení nejlepšího vektoru velmi přiblížilo nalezení optimálního řešení. Ve zbylých generacích se již hodnoty nejlepšího jedince výrazně neměnily. Závěr je tedy takový, že diferenční evoluční algoritmus se dá na filtr typu pasivní LC pásmová propust bez problémů použít.

3) Genetické a evoluční algoritmy – teoretický úvod, sestavení návrhového algoritmu a jeho implementace v programu Maple.

3.1) Genetické algoritmy

Nejprve by bylo dobré říci, co to vlastně takový genetický algoritmus je. Genetický algoritmus je optimalizační postup, který má svoje základy postavené na objevech v oblasti přírodních věd popsané Charlesem Darwinem. Každý jedinec má své specifické vlastnosti, které se s během programu vyvíjejí. Pro správnost řešení je důležitá různorodost populace, tedy různorodost jednotlivých jedinců. To má v programu za úkol použití pomocných konstant jako jsou křížení a mutace. Populaci se myslí pevně nastavený počet jedinců, kde každý jedinec je právě jedno řešení daného problému. Konkrétněji je tato problematika popsána například v literatuře [1] a [3].

3.2) Evoluční algoritmus

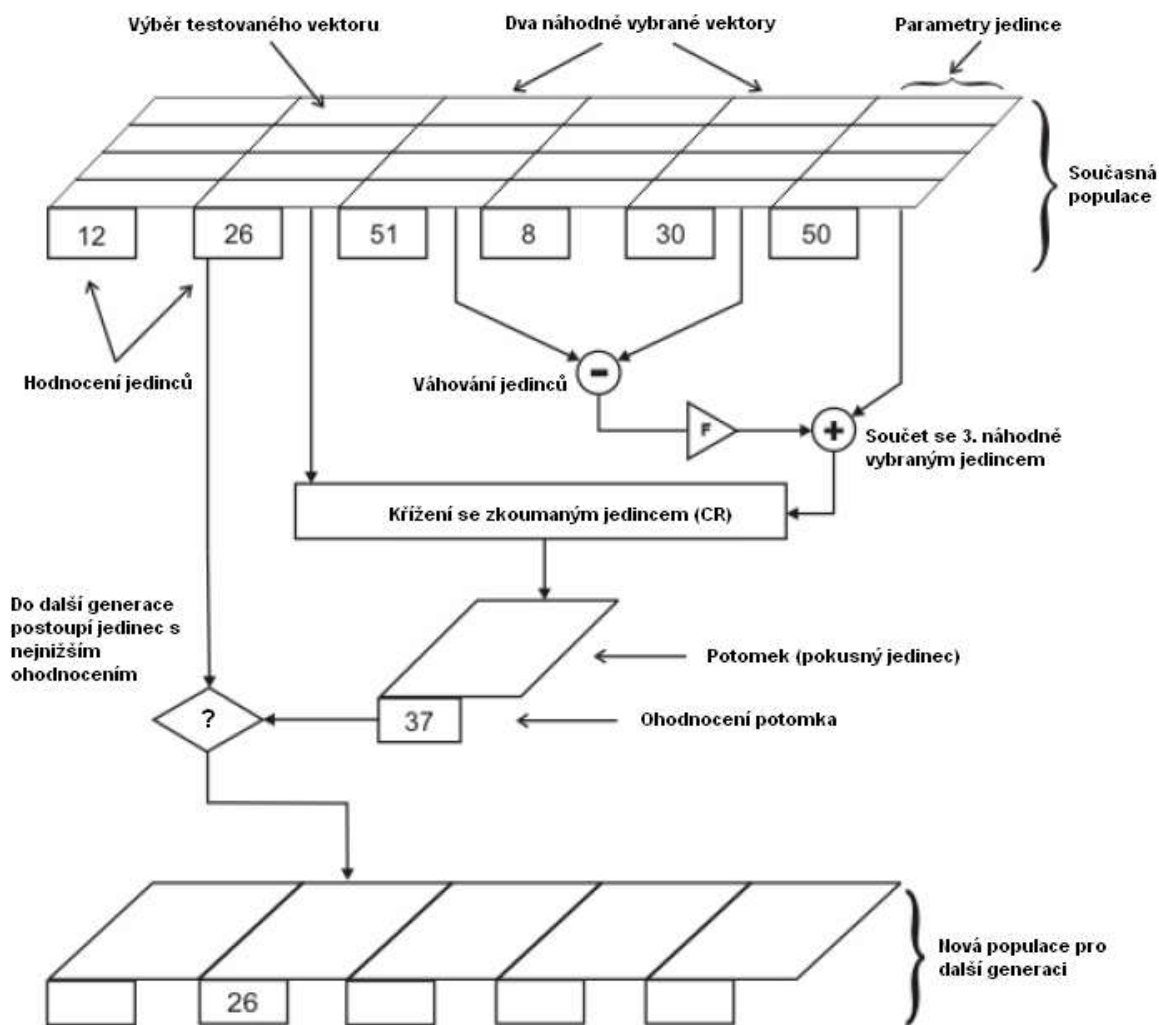
Evoluční algoritmus vychází z teorie evoluce, tak jak je známá z přírodních věd. Jedná se tedy o jeden ze způsobů optimalizace, který využívá (podobně jako v přírodní evoluci) mutaci, rekombinaci a přirozený výběr. První generace je vytvořena náhodným vygenerováním čísel. Postupně se vytváří na sebe navazující generace, které obsahují pevně nastavený počet jedinců. Každý jedinec je vektor, který obsahuje takový počet prvků, kolik je v daném zadání hledaných parametrů. U každé vytvořené generace se porovnává vypočtená přenosová charakteristika se zadanou přenosovou charakteristikou v bodech, které si v programu sami vybereme (v programu jsou tyto “testovací“ body označeny jako dataom). Z porovnání vyjde odchylka, která představuje hodnotu účelové funkce fit. Součástí každé generace je číselné ohodnocení účelové funkce. Snahou programu je najít takové hodnoty obvodových prvků, aby byla odchylka přenosových charakteristik (hodnota účelové funkce) nulová.

3.3) Diferenční evoluční algoritmus

Tento typ genetického a evolučního programování byl poprvé použit roku 1995 pány Kenem Pricem a Rainerem Stormem. Popis tohoto algoritmu by byl téměř totožný s předchozími algoritmy. Největší rozdíl je ve tvorbě potomka. U diferenciální evoluce je k vytvoření potomka zapotřebí 4 rodičů, zatímco u genetických algoritmů postačí jako rodiče 2 jedinci. Novým bodem je v tomto algoritmu pojem diferenciální mutace. Ta generuje zkušební řešení,

což je vlastně přičtení rozdílu dvou náhodně vybraných vektorů ke třetímu vektoru z populace. Podrobněji se popisu pojmů jako je mutace apod. budu věnovat v další kapitole. Pro lepší pochopení diferenčního evolučního algoritmu vkládám ještě jeho schéma, na kterém by měl být vysvětlen jeho princip. Schéma jsem převzal z literatury [4].

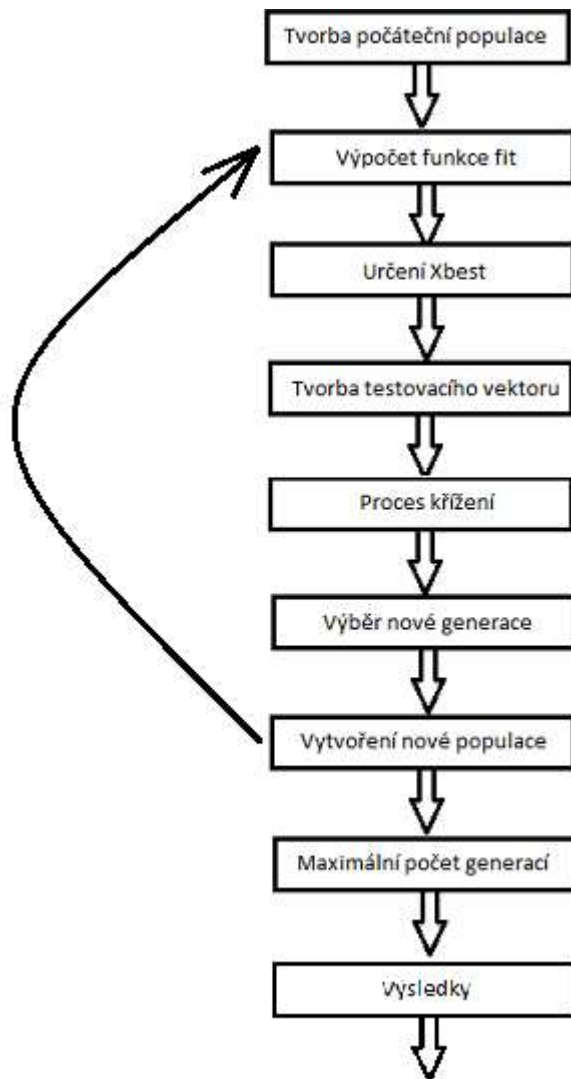
Obr. 3.3.1 - princip diferenčního evolučního algoritmu



3.4) Sestavení návrhového algoritmu a jeho implementace v programu Maple.

V této podkapitole se budu věnovat popisu programu, který byl použit v této diplomové práci. Nejprve bude pomocí vývojového diagramu popsána funkce diferenčního evolučního algoritmu. Vysvětlím jednotlivé části vývojového diagramu. Po-té popíši ty bloky programu, které vývojový diagram neobsahoval.

Obr. – 3.4.1 - vývojový diagram programu z diplomové práce:



Vysvětlení vývojového diagramu:

Tvorba počáteční populace - místo, kde mohu usměrnit generaci první populace pomocí pomocných konstant. - **for i to NP do**

X[i]:= [seq(pom1(k),k=1..NF),pom3()/10,pom3()/10]:
end do:

Pomocné konstanty – konstanty, které mají za úkol zužovat meze generovaných čísel pro “druhy“ jednotlivých parametrů (cívky, kondenzátory, vazební kondenzátory, rezistory) -

pom1:= rand(PMd....PMh)*1e-6: pom1();

pom2:= rand(0....PMh)*1e-6: pom2();

pom3:= rand(1....PMh)*1e-10: pom3();

Výpočet funkce fit – princip spočívá v testování odchylek frekvenční charakteristiky testovaného jedince od referenční charakteristiky dané přenosovou funkcí v bodech, které si v programu zvolím (tyto body jsou v programu označeny jako dataom).

Přenosová funkce – zapsání zadané (referenční) přenosové funkce ve tvaru –

Rovnice 3.4.1 – příklad obecné přenosové funkce

$$H(p) = \frac{s^3}{a_6 * s^6 + a_5 * s^5 + a_4 * s^4 + a_3 * s^3 + a_2 * s^2 + a_1 * s^1 + a_0 * s^0}$$

- tento příklad je uveden pro konkrétní zadání – filtr typu PP 6. řádu.

Určení Xbest – Určení nejlepšího jedince v populaci (Xbest) s nejmenším ohodnocením fmin.

if fmin<best then best:= fmin end if:

for i from 1 to NP do

fmin = fit[i] then Xbest:= X[i]: NFb:= i: end if:

end do:

Xbest – vektor, který má v dané generaci nejmenší hodnotu funkce fit

fmin – představuje nejmenší hodnotu účelové funkce

Tvorba testovacího vektoru – tvorba testovacího vektoru pro každého člena populace. Mohu zvolit mezi verzí testovacích vektorů a – g. V této práci používám pouze verzi f a g a to po dohodě s vedoucím práce. Po vybrání podoby testovacích vektorů je zde prostor pro manipulaci s generovanými čísly. Tím mám na mysli například to, že pokud budu chtít, aby hodnota některých parametrů nebyla menší než nějaké číslo X, tak zde mohu napsat řádek do programu ve smyslu – je-li generované číslo menší než X, pak přepiš toto generované číslo právě na X. Toto uvádím pouze jako příklad, v podstatě si zde mohu manipulovat s čísly, jak chci. Samozřejmě je potřeba testovat a program upravovat, tak aby fungoval co nejlépe –

Rovnice 3.4.2 – verze f, g testovacích vektorů

$$v[i]:= X_{best} + F*(X[R[1]]-X[R[2]]): \quad \text{verze f)}$$

$$v[i]:= X[R[3]] + F*(X[R[1]]-X[R[2]]): \quad \text{verze g)}$$

Proces křížení – křížení jedince ze současné populace s testovaným vektorem. Vznik kříženého vektoru. Aplikace konstanty křížení a mutačního faktoru.

Konstanty F, CR – zde si dovoluji zapojit citaci z literatury [3], stránky 72, 73, 74, pro přesnou definici pojmů mutace a konstanta křížení -

Mutace – Diferenciální evoluce má tu zvláštnost, že k vytvoření dalšího potomka je potřeba ne dvou, ale čtyř rodičů. Pro každého jedince jsou náhodně vybráni tři další nestejní jedinci z populace. Pomocí těchto tří jedinců se pak vytvoří tzv. šumový vektor, který není ničím jiným nežli mutací kombinace oněch tří náhodně vybraných rodičů. Konkrétně je mutace provedena tak, že se rozdíl dvou prvních náhodně vybraných rodičů vynásobí mutační konstantou F a výsledný vektor se přičte ke zbývajícím třetímu.

Rovnice 3.4.3 – vytvoření šumového vektoru

$$\vec{v}_i = \vec{x}_{r1} + F(\vec{x}_{r2} + \vec{x}_{r3})$$

Konstanta křížení – nastavuje se v intervalu (0 -1). V případě, že se nastaví na 0 dojde k tomu, že se mutace nedostane do zkušebního jedince, který díky tomu bude čistou kopií aktuálního (čtvrtého rodiče). Vývoj evoluce se pak zastaví. V případě, že CR bude nastaven na 1, bude zkušební jedinec tvořen pouze ze tří náhodně vybraných rodičů – jedinců

z populace a diferenciální evoluce se bude spíše podobat náhodnému hledání nežli evolučnímu algoritmu.

Výběr nové generace - místo, kde se pomocí for cyklu a příkazu if ukládá a vybírá nová generace. Zjišťuje se zde, je-li následující generace lepší než ta předchozí. Je-li tomu tak, tak se ukládá jako nejlepší poslední generace.

V opačném případě se jako nejlepší bere generace minulá -

for i from 1 to NP do

if fitu[i]<fit[i] then Xnew[i]:= U[i] else Xnew[i]:= X[i] end if:

end do:

for i from 1 to NP do

X[i]:= Xnew[i]:

fit[i]:= fitval(X[i]);

if fit[i] <= fmin then

fmin:= fit[i]

else fmin:= fmin

end if:

end do;

Vytvoření nové populace – do nové generace postupuje lepší z vektorů X[i] a U[i]. Lepší je ten vektor, který má nižší hodnotu funkce fit. –

for i from 1 to NP do

if fitu[i]<fit[i] then Xnew[i]:= U[i] else Xnew[i]:= X[i] end if:

end do:

Po-té je vytvořena nová generace, která je přesunuta zpět na začátek programu. Tak je tomu až do hodnoty maximálního počtu generací.

Maximální počet generací – Po dosažení této nastavené hodnoty se program ukončí a přejde k výpisu výsledků a to i přes špatné výsledky optimalizace.

Výsledky – výpis nejlepšího vektoru s jeho ohodnocením (hodnota fit). Přiřazení optimalizovaných hodnot jednotlivým prvkům. Vykreslení průběhu konvergence a vykreslení přenosových charakteristik (porovnání referenční a vypočtené přenosové charakteristiky).

Bloky knihovna Syrup, Vstupní data a Simplex:

Knihovna Syrup – zapsání konkrétního obvodu do programu Maple pomocí knihovny pro symbolickou analýzu elektrických obvodů. Zadání symbolické analýzy do knihovny Syrup:

with(Syrup):

obvod1:= "

V 1 0

R1 1 2

L1 2 0

C2 2 0

CV1 2 3

L3 3 0

C4 3 0

CV2 3 4

L5 4 0

C6 4 0

R2 4 0

.end":

Vstupní data – nastavení základních konstant a proměnných:

NP – počet členů v generaci

G – počet generací, které proběhnou během chodu programu

ND – počet neznámých parametrů, které se budou optimalizovat

R1, R2 – nastavení zakončovacích rezistorů

kk – konstanta vyjadřující vztah mezi R1, R2 –

Rovnice 3.4.4 – vztah mezi zakončovacími odpory

$$kk = \frac{R2}{R1 + R2}$$

Mh(x), Md(x) – meze pro náhodně generovaná čísla (parametry)

Simplex pro best – Simplex je samostatná optimalizační metoda. Byl jsem na ni upozorněn vedoucím práce a použil jsem ji v těch programech, kde se průběh konvergence v určité fázi programu výrazně zpomalil. Tato metoda byla využita v diplomové práci, která je zároveň doporučenou literaturou pro moji práci. V seznamu literatury je označena pod číslem [2].

Zadání procedury Simplex:

x1:= 'x1': x2:= 'x2': x3:= 'x3': x4:= 'x4': x5:= 'x5': x6:= 'x6': x7:= 'x7': x8:= 'x8':

Xopt:= [seq(x||k,k=1..ND)]:

Xin:= [seq(x||k=Xbest[k],k=1..ND)]:

XS1:= NLPSolve(fitval(Xopt), initialpoint=Xin,

method=nonlinearsimplex,evaluationlimit=12500):

fitb:= XS1[1]:

assign(XS1[2]):

Xbest:= [seq(x||m,m=1..ND)]:

4) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků.

4.1) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení zakončovacích odporů

Nejprve jsem se zaměřil na nastavení zakončovacích odporů. Tedy R1, R2. Postupně jsem testoval programy s různými nastaveními R1, R2. Dodržel jsem však podmínku symetrického zakončení. Tedy $R1 = R2$. Program jsem nechal proběhnout postupně pro hodnoty R1, R2 = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Z výsledků vyplynulo, že hodnoty parametrů obvodových prvků jsou si nejbližší někde mezi R1, R2 e (10 - 27). Nejdříve uvedu hodnoty pro “hrubé” rozložení R1, R2.

R=10

$L1 := 0.07271396069$

$C2 := 0.3461924210$

$L3 := 3.954747968$

$C4 := 0.00001549160180$

$L5 := 0.08342003347$

$C6 := 0.3048183802$

$CV1 := 0.003227828443$

$CV2 := 0.003207308111$

R=20	R=30	R=40
$L1 := 0.1669880466$	$L1 := 0.2175309390$	$L1 := 0.3326616125$
$C2 := 0.1496716253$	$C2 := 0.1156005112$	$C2 := 0.07578459517$
$L3 := 1.147286983$	$L3 := 9.541425562$	$L3 := 4.811341338$
$C4 := 0.01378864065$	$C4 := 0.0002812900273$	$C4 := 0.002385042255$
$L5 := 0.1453635497$	$L5 := 0.2508375033$	$L5 := 0.2917726739$
$C6 := 0.1705440399$	$C6 := 0.1012434420$	$C6 := 0.08562521839$
$CV1 := 0.004208899857$	$CV1 := 0.001202363674$	$CV1 := 0.001457336553$
$CV2 := 0.004238243885$	$CV2 := 0.001190104808$	$CV2 := 0.001459585507$

R=50	R=60	R=70
<i>L1</i> := 0.4175827973	<i>L1</i> := 0.4992976994	<i>L1</i> := 0.5058200985
<i>C2</i> := 0.06058588173	<i>C2</i> := 0.05032679827	<i>C2</i> := 0.04850108904
<i>L3</i> := 9.018155644	<i>L3</i> := 5.243552383	<i>L3</i> := 1.991438622
<i>C4</i> := 0.0009232925674	<i>C4</i> := 0.002584084716	<i>C4</i> := 0.009383213606
<i>L5</i> := 0.3633597185	<i>L5</i> := 0.4376686529	<i>L5</i> := 0.5873240532
<i>C6</i> := 0.06896590111	<i>C6</i> := 0.05690988084	<i>C6</i> := 0.04204439369
<i>CV1</i> := 0.0009489188125	<i>CV1</i> := 0.001138565954	<i>CV1</i> := 0.001724901669
<i>CV2</i> := 0.0009566465700	<i>CV2</i> := 0.001142389116	<i>CV2</i> := 0.001703802058

R=80	R=90	R=100
<i>L1</i> := 0.5864548540	<i>L1</i> := 0.7481430123	<i>L1</i> := 0.7307215524
<i>C2</i> := 0.04242631394	<i>C2</i> := 0.03283368258	<i>C2</i> := 0.03394999498
<i>L3</i> := 6.288482776	<i>L3</i> := 1.978201246	<i>L3</i> := 6.088666338
<i>C4</i> := 0.002252853191	<i>C4</i> := 0.009863320302	<i>C4</i> := 0.002550220447
<i>L5</i> := 0.6623046740	<i>L5</i> := 0.6574228834	<i>L5</i> := 0.8307724398
<i>C6</i> := 0.03789354081	<i>C6</i> := 0.03712963349	<i>C6</i> := 0.03011215536
<i>CV1</i> := 0.0009003303333	<i>CV1</i> := 0.001515129116	<i>CV1</i> := 0.0008199880830
<i>CV2</i> := 0.0009032577618	<i>CV2</i> := 0.001516716374	<i>CV2</i> := 0.0008194436079

Detailnější testování pro hodnoty L a C nejbliže k sobě

R=11	R=12	R=13
<i>L1</i> := 0.08032409291	<i>L1</i> := 0.08205096307	<i>L1</i> := 0.1145269271
<i>C2</i> := 0.3128211027	<i>C2</i> := 0.3048778616	<i>C2</i> := 0.2220193812
<i>L3</i> := 3.044608582	<i>L3</i> := 1.688863750	<i>L3</i> := 5.051735866
<i>C4</i> := 0.001387477965	<i>C4</i> := 0.006090708055	<i>C4</i> := 0.00002834080613
<i>L5</i> := 0.09142323742	<i>L5</i> := 0.1061386734	<i>L5</i> := 0.08848495965
<i>C6</i> := 0.2775824401	<i>C6</i> := 0.2377855490	<i>C6</i> := 0.2843261014
<i>CV1</i> := 0.003497876950	<i>CV1</i> := 0.004708862680	<i>CV1</i> := 0.002394673675
<i>CV2</i> := 0.003493319895	<i>CV2</i> := 0.004311743784	<i>CV2</i> := 0.002629778126

R=14	R=15	R=16
<i>L1</i> := 0.1019965455	<i>L1</i> := 0.1251938926	<i>L1</i> := 0.1163403004
<i>C2</i> := 0.2466917126	<i>C2</i> := 0.2030456677	<i>C2</i> := 0.2162292119
<i>L3</i> := 5.026551020	<i>L3</i> := 5.615030870	<i>L3</i> := 5.507379786
<i>C4</i> := 0.0002510981754	<i>C4</i> := 0.0001338047016	<i>C4</i> := 0.0003211329065
<i>L5</i> := 0.1165467793	<i>L5</i> := 0.1090218963	<i>L5</i> := 0.1334925125
<i>C6</i> := 0.2180624319	<i>C6</i> := 0.2308417193	<i>C6</i> := 0.1903427755
<i>CV1</i> := 0.002417062111	<i>CV1</i> := 0.002197253306	<i>CV1</i> := 0.002162200663
<i>CV2</i> := 0.002406967600	<i>CV2</i> := 0.002212230350	<i>CV2</i> := 0.002148698809

R=17	R=18	R=19
<i>L1</i> := 0.1415232161	<i>L1</i> := 0.1316228909	<i>L1</i> := 0.1582342007
<i>C2</i> := 0.1794545308	<i>C2</i> := 0.1912394601	<i>C2</i> := 0.1606830494
<i>L3</i> := 5.377752454	<i>L3</i> := 7.095421348	<i>L3</i> := 7.349204670
<i>C4</i> := 0.0005123846646	<i>C4</i> := 0.00001612007470	<i>C4</i> := 0.00004703603522
<i>L5</i> := 0.1239470306	<i>L5</i> := 0.1494694451	<i>L5</i> := 0.1384299187
<i>C6</i> := 0.2028684329	<i>C6</i> := 0.1701389610	<i>C6</i> := 0.1818250407
<i>CV1</i> := 0.002112344987	<i>CV1</i> := 0.001788165276	<i>CV1</i> := 0.001707717602
<i>CV2</i> := 0.002118908447	<i>CV2</i> := 0.001790819007	<i>CV2</i> := 0.001716428588

R=20	R=21	R=22
<i>L1</i> := 0.1669880466	<i>L1</i> := 0.1532770566	<i>L1</i> := 0.1600720922
<i>C2</i> := 0.1496716253	<i>C2</i> := 0.1642294824	<i>C2</i> := 0.1570454303
<i>L3</i> := 1.147286983	<i>L3</i> := 8.254909215	<i>L3</i> := 6.644888482
<i>C4</i> := 0.01378864065	<i>C4</i> := 0.00001717710134	<i>C4</i> := 0.0004921929103
<i>L5</i> := 0.1453635497	<i>L5</i> := 0.1745741436	<i>L5</i> := 0.1833929041
<i>C6</i> := 0.1705440399	<i>C6</i> := 0.1456534115	<i>C6</i> := 0.1384429188
<i>CV1</i> := 0.004208899857	<i>CV1</i> := 0.001538124648	<i>CV1</i> := 0.001678917530
<i>CV2</i> := 0.004238243885	<i>CV2</i> := 0.001534978431	<i>CV2</i> := 0.001668099899

R=23	R=24	R=25
<i>L1</i> := 0.1673713616	<i>L1</i> := 0.1740627221	<i>L1</i> := 0.1824951938
<i>C2</i> := 0.1501672080	<i>C2</i> := 0.1441136523	<i>C2</i> := 0.1354197490
<i>L3</i> := 6.677368961	<i>L3</i> := 5.061418580	<i>L3</i> := 1.133301604
<i>C4</i> := 0.0005546862138	<i>C4</i> := 0.001367766772	<i>C4</i> := 0.01490876245
<i>L5</i> := 0.1916437167	<i>L5</i> := 0.2005814702	<i>L5</i> := 0.2078871739
<i>C6</i> := 0.1324480218	<i>C6</i> := 0.1262817148	<i>C6</i> := 0.1198133399
<i>CV1</i> := 0.001637884375	<i>CV1</i> := 0.001845608486	<i>CV1</i> := 0.003804279066
<i>CV2</i> := 0.001627919856	<i>CV2</i> := 0.001826967306	<i>CV2</i> := 0.003796132269

R=26	R=27
<i>L1</i> := 0.1897043827	<i>L1</i> := 0.1971063172
<i>C2</i> := 0.1316005277	<i>C2</i> := 0.1277000389
<i>L3</i> := 2.914529082	<i>L3</i> := 10.42299466
<i>C4</i> := 0.004105116152	<i>C4</i> := 0.00003576127010
<i>L5</i> := 0.2162785541	<i>L5</i> := 0.2244722970
<i>C6</i> := 0.1164920368	<i>C6</i> := 0.1132672526
<i>CV1</i> := 0.002327504743	<i>CV1</i> := 0.001206224608
<i>CV2</i> := 0.002320193444	<i>CV2</i> := 0.001205460548

Z výsledků je vidět, že z hlediska snížení rozptylu hodnot obvodových prvků se jako nejlepší jeví hodnoty 20 a 25. Bylo by tedy nejlepší použít hodnoty zakončovacích odporů $R1 = R2 = 20$ nebo $R1 = R2 = 25$.

Pomocí této optimalizace jsem sice snížil rozptyl hodnot obvodových prvků. Nicméně hodnoty prvků se i přes tuto optimalizaci liší v nejlepším případě o dva řády. Následovat budou další pokusy optimalizace, které budou přispívat ke snížení rozptylu hodnot obvodových prvků.

4.2) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí pomocných proměnných Kon_L, Kon_C.

Dalším pokusem o snížení rozptylu hodnot obvodových prvků bylo přidání pomocných proměnných Kon_L (kontrola cívek) a Kon_C (kontrola kondenzátorů). Jejich funkce v programu je myšlena jako “ohlídání“ rozptylu hodnot cívek respektive kondenzátorů. V programu si pomocí příkazu **lxmin:= min(seq(abs(L||i),i=1..NF-1))** zjistím minimální hodnotu programem optimalizovaných cívek. Totožným způsobem zjistím i maximální hodnotu optimalizovaných cívek. Obdobně si zjistím maximum a minimum pro kondenzátory. Budu tedy mít hodnoty lxmin, lxmax, cxmin, cxmax. Díky těmto hodnotám mohu v programu na konec bloku vyhodnocovací procedura připsat do řádku účelové funkce další podmínku využívající hodnot pomocných proměnných Kon_L a Kon_C. Hodnoty proměnných budou dány podle vzorců:

Rovnice 4.2.1 – výpočet proměnné Kon_L

$$Kon_L = \exp\left(\frac{lxmax}{lxmin} - 1\right)$$

Rovnice 4.2.2 – výpočet proměnné Kon_C

$$Kon_C = \exp\left(\frac{cxmax}{cxmin} - 1\right)$$

Účelová funkce bude vypadat následovně:

evalf(2e3*abs(delta[j])^2,j=1..Ndo) + 1e3*(delta_h)^2 + 1e-2*Kon_L + 1e-8*Kon_C

Koeficienty 1e-2 pro cívky a 1e-8 pro kondenzátory byly zvoleny empiricky.

Bylo zapotřebí změnit hodnoty řídicích konstant. Konkrétně pak na CR=0,86 a F=0,45. Díky neuspokojivým výsledkům, bylo zapotřebí změnit i hodnoty v nastavení vstupních dat. NP=130 a G=2600. Tím pádem se však velmi výrazně prodloužila doba konvergence. Testováním se také zjistilo, že lepší výsledek získáme změnou verze výpočtu testovacích vektorů f verze f na verzi g.

Rovnice 4.2.3 – Verze f, g testovacího vektoru

Verze f - v[i]:= Xbest + F*(X[R[1]]-X[R[2]]):

Verze g - v[i]:= X[R[3]] + F*(X[R[1]]-X[R[2]]):

Výsledky s použitím pomocných proměnných Kon_L a Kon_C:

Prvky vektoru Xbest při hodnotě fit_best = 6.842616367:

$X_{best} = [0.008082838110, 3.083603648, 0.04151136466,$
 $0.4185095901, 0.008012035945, 3.074307046, 0.1020939723,$
 $0.09314806597]$

Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008082838110$

$C2 := 3.083603648$

$L3 := 0.04151136466$

$C4 := 0.4185095901$

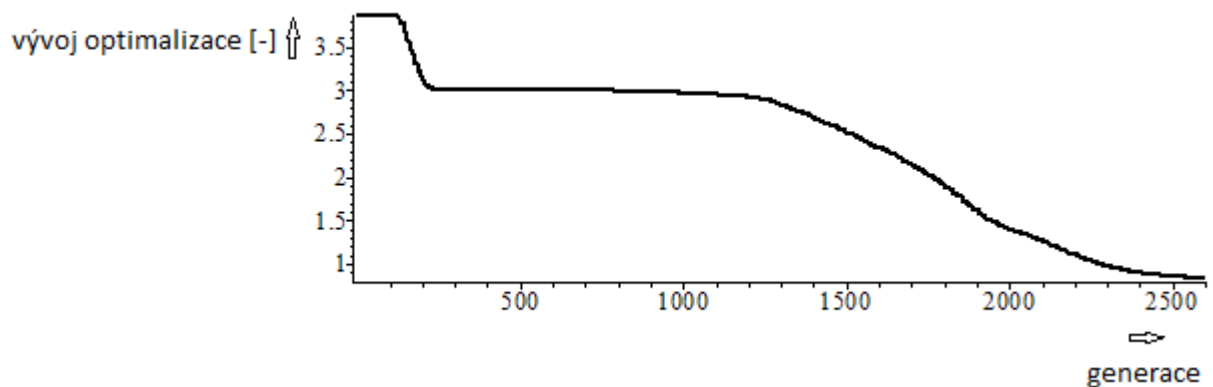
$L5 := 0.008012035945$

$C6 := 3.074307046$

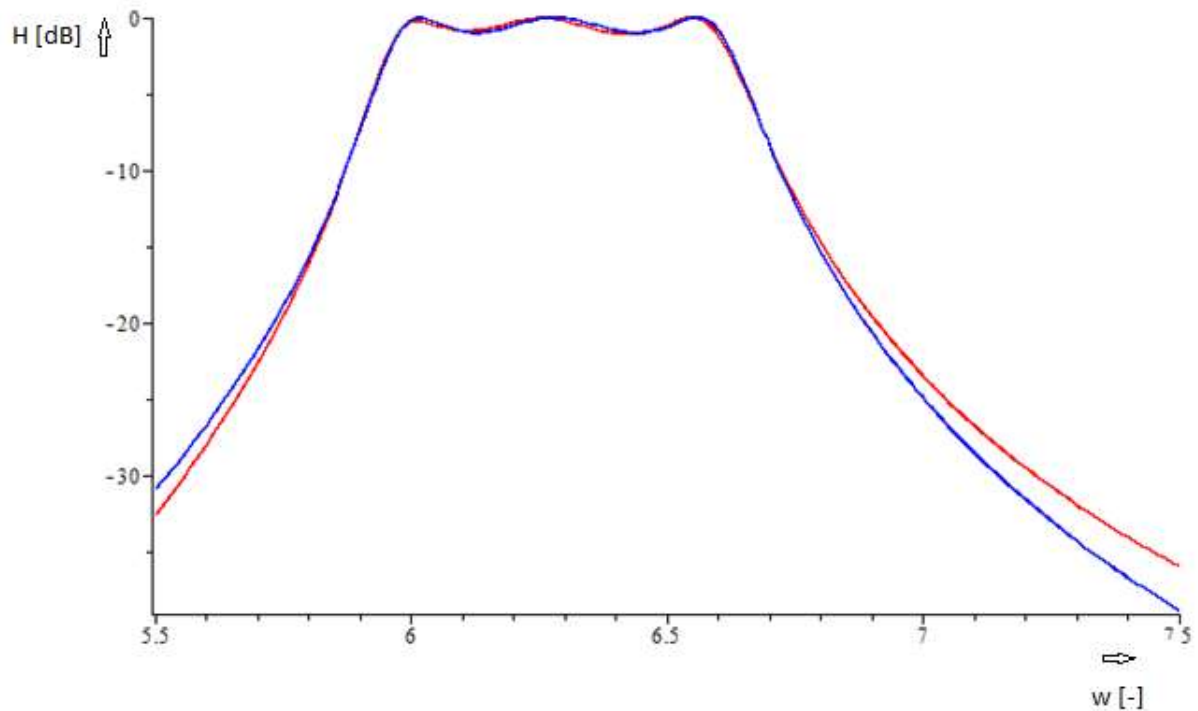
$CV1 := 0.1020939723$

$CV2 := 0.09314806597$

Obr. 4.2.1 - průběh konvergence při použití pomocných proměnných Kon_L a Kon_C:



Obr. 4.2.2 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při použití pomocných proměnných Kon_L a Kon_C:



Z výsledků je patrné, že přenosové charakteristiky se téměř shodují. Průběh konvergence byl velmi pomalý, ale díky velkému počtu nastavených generací došel do přijatelného konce. Z číselných výsledků je vidět, že hodnoty cívek a kondenzátorů se v prvním a třetím rezonančním obvodu velmi přiblížily, avšak v druhém rezonančním obvodu nikoli. Tento problém bude řešen v dalších kapitolách. Jelikož byl průběh konvergence příliš zdlouhavý, tak byla v tomto případě použita pomocná optimalizační metoda Simplex, která byla aplikována na nejlepší vektor X_{best} .

Výsledky s použitím pomocné optimalizační metody Simplex:

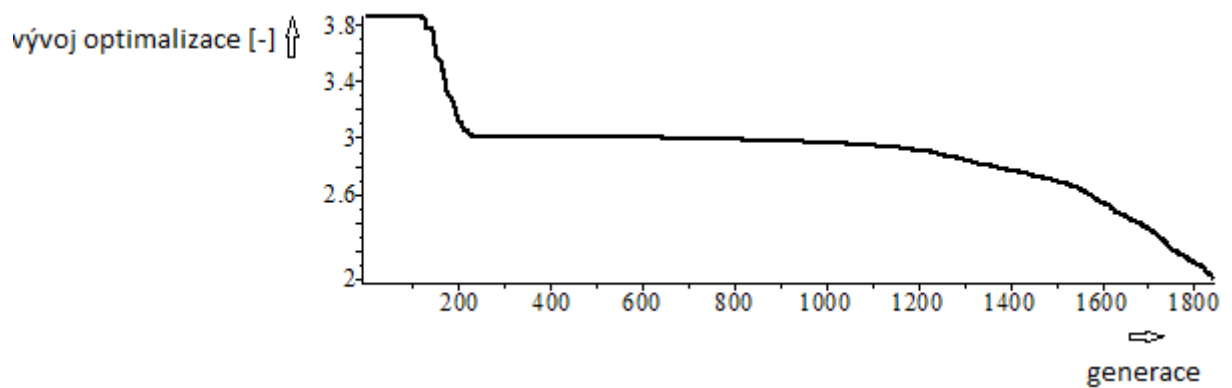
Prvky vektoru X_{best} při hodnotě $fit_best = 15.9047965452434621$:

$$X_{best} = [0.01214207085, 2.043509019, 0.09172134290, \\ 0.1757599746, 0.01254504765, 1.976264858, 0.05860483236, \\ 0.04500746907]$$

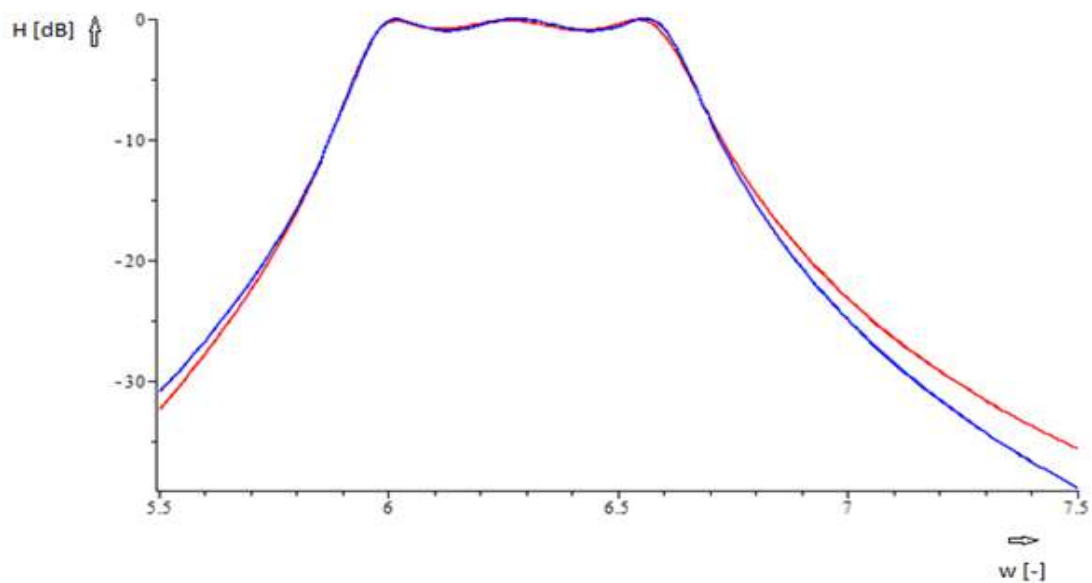
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00841036868696264$
 $C2 := 2.95340444013541$
 $L3 := 0.0575989749102335$
 $C4 := 0.281147119270905$
 $L5 := 0.0084103686866648$
 $C6 := 2.95460194962261$
 $CV1 := 0.0876634107722089$
 $CV2 := 0.0741771938018854$

Obr. 4.2.3 - průběh konvergence při použití pomocných proměnných Kon_L a Kon_C s použitím metody Simplex:



Obr. 4.2.4 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při použití pomocných proměnných Kon_L a Kon_C s použitím metody Simplex:



Aplikace optimalizační metody Simplex na vektor Xbest přinesla velmi podobné výsledky.

Velké pozitivum je ve snížení počtu generací o 800, což velmi urychlilo optimalizaci.

4.3) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L jako konstanty a optimalizace R1, R2 a C.

Po konzultaci s vedoucím práce panem docentem Martinkem jsem zkusil nastavit hodnoty všech tří indukčností na konstantní hodnotu $L1 = L2 = L3 = 1$. Poupravil jsem program k optimalizování hodnot kondenzátorů a zakončovacích odporů. Jelikož se vypočítané hodnoty zakončovacích odporů pohybovaly v jiném číselném rozmezí, tak jsem v programu vytvořil navíc pomocnou proměnnou a nový číselný interval. Tyto prvky jsem následně zakomponoval do bloku výpočet testovacích vektorů. Při přidávání dalšího intervalu jsem se rozhodl vytvořit ještě jeden interval pro vazební kondenzátory.

Vytvoření nových intervalů:

```
Mh:= 500: Md:= 1:
Mh1:= 1: Md1:= 0.0001:
Mh2:= 0.01: Md2:=0.000001:
```

Pomocné proměnné k intervalům:

```
pom1:= rand(PMd...PMh)*1e-6: pom1();
pom2:= rand(1...PMh)*1e-6: pom2();
pom4:= rand(0...PMh1)*1e-6: pom4();
pom3:= rand(1...PMh1)*1e-10: pom3();
```

Vytvoření počáteční populace:

```
X[i]:= [seq(pom2(k),k=1..NF-3),seq(pom4(k),k=3..NF),seq((pom3()/10)(k),k=6..NF+2)]:
```

Zakomponování nových intervalů do výpočtu testovacích vektorů:

```
U[i]:= [seq(pom1(k),k=1..ND)]:
for j from 1 to 2 do
aj:= r():
if aj<CR then U[i][j]:= v[i][j]
else U[i][j]:= X[i][j]
end if;
if U[i][j]<Md then U[i][j]:= pom1()
elif U[i][j]>Mh then U[i][j]:= pom2()/10
else U[i][j]:= U[i][j]
end if;
```



```

end do;
for j from 2 to NF do
aj:= r():
if aj<CR then U[i][j]:= v[i][j]
else U[i][j]:= X[i][j]
end if;
if U[i][j]<Md1 then U[i][j]:= 1e-4
elif U[i][j]>Mh then U[i][j]:= pom4()
else U[i][j]:= U[i][j]
end if;
end do;
for j from NF+1 to ND do
aj:= r():
if aj<CR then U[i][j]:= v[i][j]
else U[i][j]:= X[i][j]
end if;
if U[i][j]<Md2 then U[i][j]:= 1e-6
elif U[i][j]>Mh2 then U[i][j]:= pom3()/100
else U[i][j]:= U[i][j]
end if;
end do;

```

Výsledky:

Prvky vektoru Xbest při hodnotě fit_best = 4.804560921:

$$X_{best} = [86.90288799, 223.8776415, 0.02473421101, 0.02357499390, 0.02236355473, 0.001558123034, 0.001419586190]$$

Po přiřazení hodnot k obvodovým prvkům:

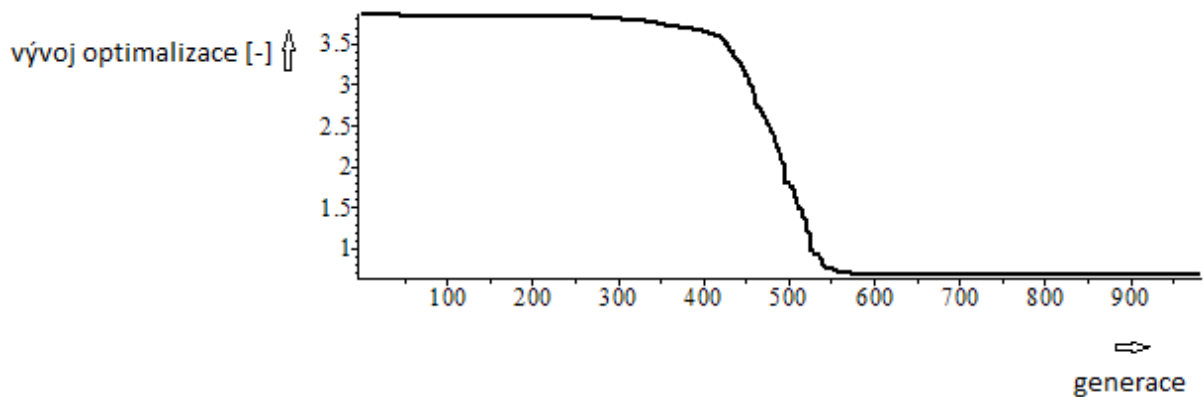
R1 := 86.90288799
R2 := 223.8776415
C3 := 0.02473421101
C4 := 0.02357499390
C5 := 0.02236355473
CVI := 0.001558123034
CVI := 0.001419586190

L1 := 1

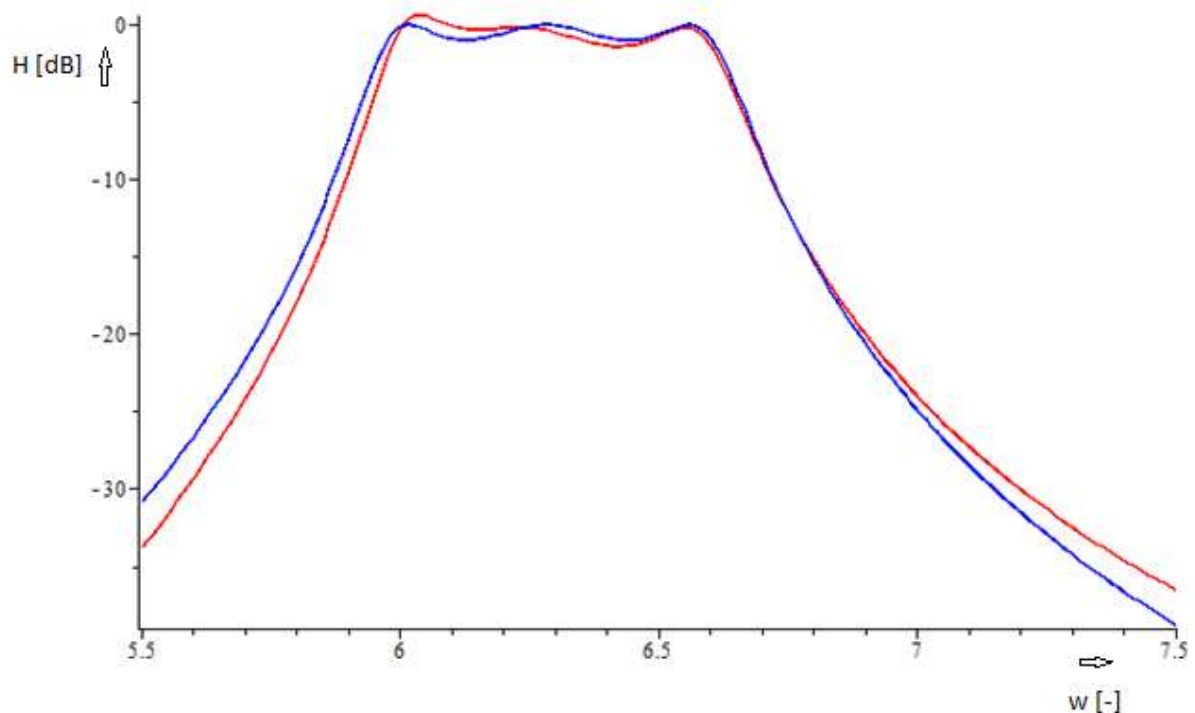
L3 := 1

L5 := 1

Obr. 4.3.1 - průběh konvergence při nastavení shodných indukčností:



Obr. 4.3.2 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při nastavení shodných indukčností:



Výsledky ukazují podstatné zlepšení v problematice snížení rozptylu hodnot mezi kondenzátory. V podstatě není rozdíl ani mezi hodnotami cívek, ale ty jsou v tomto případě nastaveny na konstantní hodnotu. Hodnoty zakončovacích odporů R_1 , R_2 se liší, ale řešením tohoto problému se budu zabývat až v dalších kapitolách. Z grafu průběhu konvergence je zřetelné snížení počtu potřebných generací pro kvalitní optimalizaci zadané pásmové propusti. Graf frekvenční charakteristiky ukazuje, že se nepodařilo dostatečně přesně vyrovnat průběh propustného pásma.

4.4) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L na konstantní hodnotu a optimalizace R1 a C. R2 ze vzorce pomocí kk (0,25 0,5 0,75).

V předchozím výsledku jsem se setkal s problémem rozdílných hodnot u zakončovacích odporů. Jelikož jsou zakončovací odpory zpravidla návrhové veličiny, respektive jejich poměr, tak se zde nabízí možnost využít jejich relaci jako návrhový parametr. Jedna z možností využití relace R1, R2 je taková, že by se společně s kondenzátory optimalizovala i hodnota prvku R1, avšak hodnota druhého zakončovacího odporu R2 by se vypočítala až z výsledku hodnoty R1 a parametru kk. Kde parametr kk je dán vztahem:

Rovnice 4.4.1 – vztah mezi zakončovacími odpory

$$kk = \frac{R2}{R1 + R2}$$

Hodnota R2 by se pak vypočítala jako:

Rovnice 4.4.2 – vztah pro výpočet R2 pomocí parametru kk a hodnoty R1

$$R2 = \frac{(kk * R1)}{abs(1 - kk)}$$

V programu se tedy změní počet optimalizovaných prvků z ND=7 na ND=6. Upraví se vytvoření vektoru počáteční populace na:

```
X[i]:= [seq(pom2(k),k=1..NF-3),seq(pom4(k),k=2..NF),seq((pom3()/10)(k),k=5..NF+2)]
```

Konstanta kk se postupně nastaví na hodnotu kk = 0,5; 0,25; 0,75

Výsledky pro kk = 0,5:

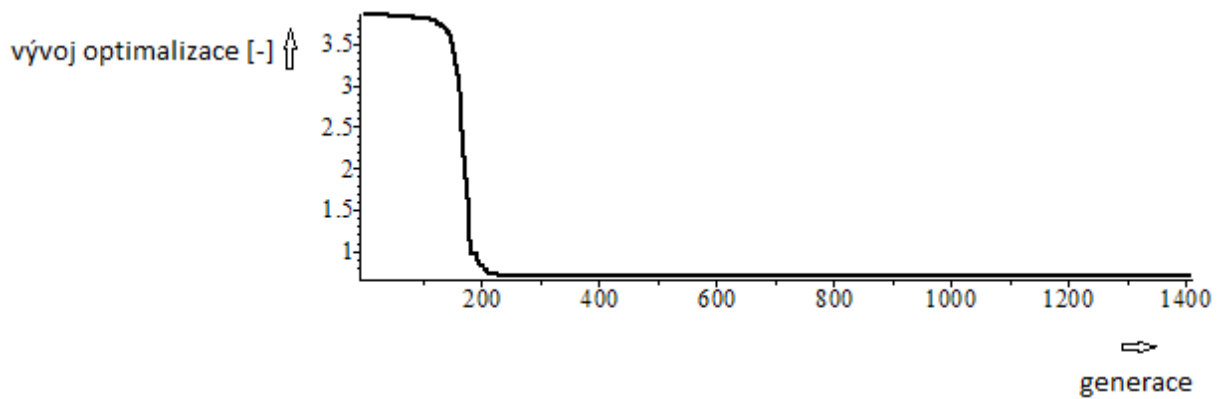
Prvky vektoru Xbest při hodnotě fit_best = 5.057019075:

```
Xbest = [ 124.4523768, 0.02385027128, 0.02191226680,  
0.02369715935, 0.001879007824, 0.001691614819]
```

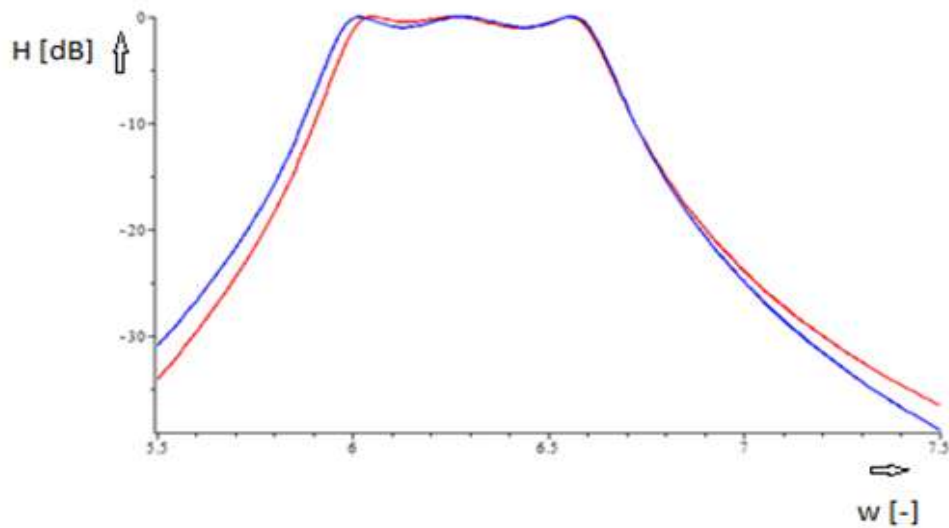
Po přiřazení hodnot k obvodovým prvkům:

$R1 := 124.4523768$
 $C2 := 0.02385027128$
 $C3 := 0.02191226680$
 $C4 := 0.02369715935$
 $CV1 := 0.001879007824$
 $CV1 := 0.001691614819$
 $L1 := 1$
 $L3 := 1$
 $L5 := 1$
 $R2 := 124.4523768$

Obr. 4.4.1 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů
($kk = 0,5$):



Obr. 4.4.2 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($kk = 0,5$):



Výsledky pro $kk = 0,25$:

Prvky vektoru X_{best} při hodnotě $fit_best = 56.58513387$:

$$X_{best} = [278.7861061, 0.02372575007, 0.02188786719, 0.02368277960, 0.001695616945, 0.001964121156]$$

Po přiřazení hodnot k obvodovým prvkům:

$$R1 := 278.7861061$$

$$C2 := 0.02372575007$$

$$C3 := 0.02188786719$$

$$C4 := 0.02368277960$$

$$CV1 := 0.001695616945$$

$$CVI := 0.001964121156$$

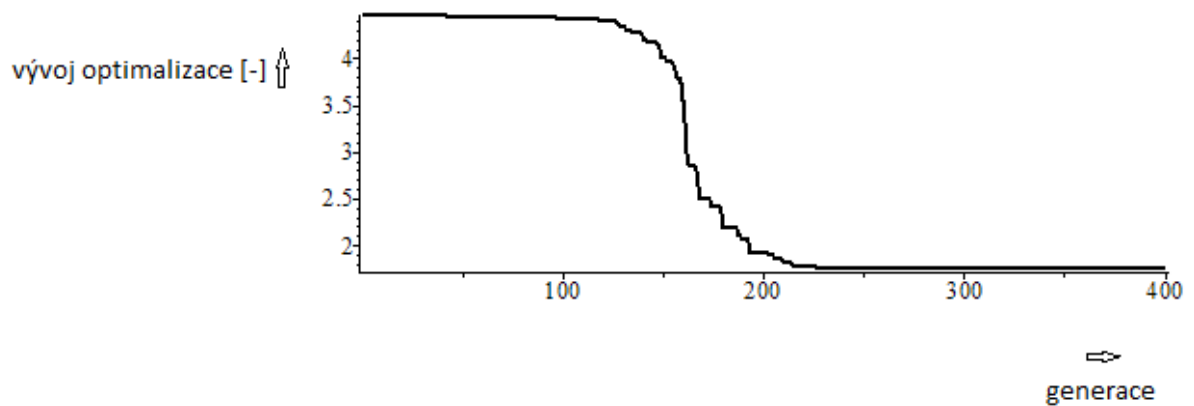
$$L1 := 1$$

$$L3 := 1$$

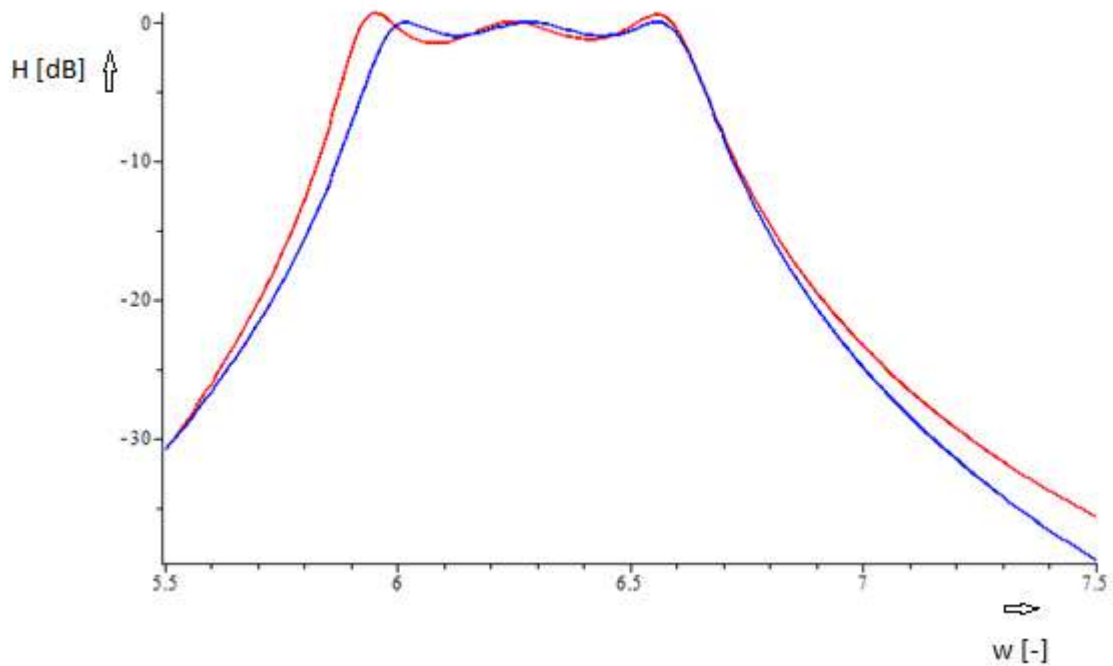
$$L5 := 1$$

$$R2 := 92.92870203$$

Obr. 4.4.3 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů
($kk = 0,25$):



Obr. 4.4.4 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($k_k = 0,25$):



Výsledky pro $k_k = 0,75$:

Prvky vektoru X_{best} při hodnotě $fit_best = 56.58510492$:

$$X_{best} = [92.92513335, 0.02368283885, 0.02188776674, 0.02372579009, 0.001964186435, 0.001695571235]$$

Po přiřazení hodnot k obvodovým prvkům:

$$R1 := 92.92513335$$

$$C2 := 0.02368283885$$

$$C3 := 0.02188776674$$

$$C4 := 0.02372579009$$

$$CV1 := 0.001964186435$$

$$CV1 := 0.001695571235$$

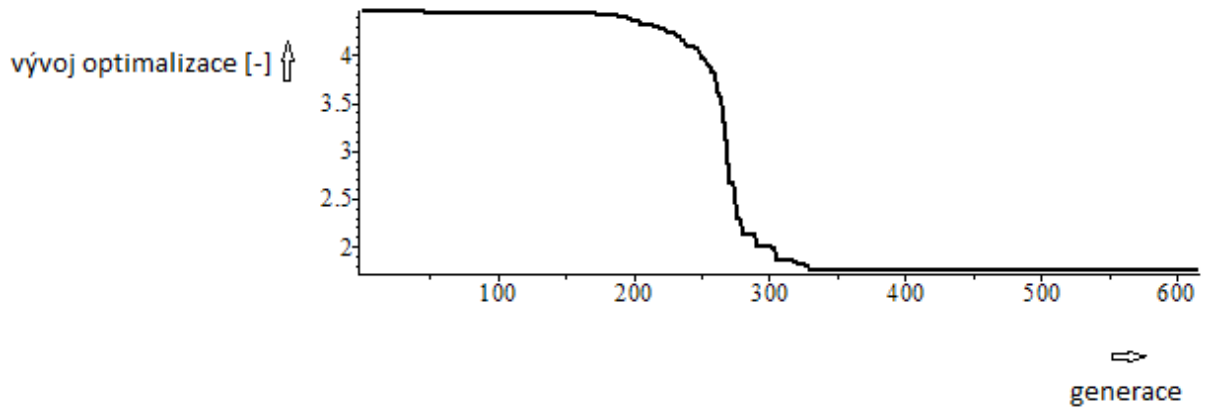
$$L1 := 1$$

$$L3 := 1$$

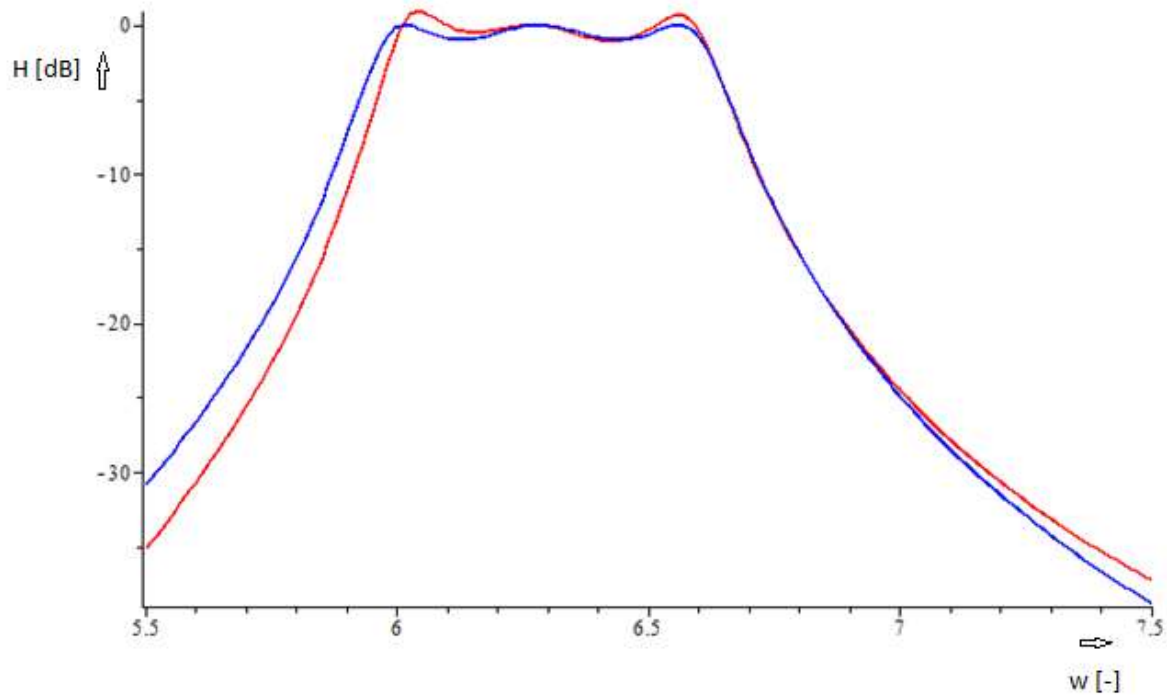
$$L5 := 1$$

$$R2 := 278.7754000$$

Obr. 4.4.5 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů
($kk = 0,75$):



Obr. 4.4.6 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($kk = 0,75$):



Výsledky této kapitoly považuji za částečně uspokojivé, s ohledem na průběh frekvenční charakteristiky propustního pásma. Využití vztahu mezi R_1 , R_2 se ukázalo jako dobrý nápad pro “ohlídání” rozptylu těchto dvou hodnot. Pro symetrické zakončení filtru bylo potřeba nastavit kk na hodnotu 0,5, což je při pohledu na vztah mezi R_1 , R_2 logické –

Rovnice 4.4.3 – vztah mezi zakončovacími odpory

$$kk = \frac{R_2}{R_1 + R_2}$$

Pro ověření správné funkce a pro porovnání výsledků jsem vložil i výsledky pro $kk = 0,25$ respektive pro $kk = 0,75$. Výsledky ukazují správné hodnoty mezi R1 a R2. Tedy $1/3x$ (pro $k = 0,25$) respektive $3x$ (pro $kk = 0,75$) menší (větší) R2 oproti R1. Z výsledných grafů jsou vidět změny důležitých parametrů filtru. Změna je vidět v šířce pásma a ve zvlnění v propustném pásmu. Nejvíce patrné je to v případech kk různých od hodnoty 0,5. V tomto případě tedy pro $kk = 0,25$ respektive $kk = 0,75$. Pro zachování správného průběhu frekvenční charakteristiky by bylo potřeba v programu zrušit podmínku stejných indukčností, aby měl diferenční evoluční algoritmus prostor lépe optimalizovat jednotlivé rezonanční kmitočty.

4.5) Vícekriteriální optimalizace – snížení rozptylu hodnot parametrů obvodových prvků pomocí nastavení hodnot L na konstantní hodnotu a optimalizace R a C + srovnání R pomocí delta_kk .

Na začátku kapitoly 4.4 jsem se zmiňoval o problematice rozdílných hodnot zakončovacích odporů R1, R2. Jako další z možností sblížení jejich optimalizovaných hodnot je další využití jejich společné relace

Rovnice 4.5.1 – vztah mezi zakončovacími odpory

$$kk = \frac{R2}{R1 + R2}$$

Zkusil jsem do vyhodnocení funkce fitval přidat další člen s názvem delta_kk. Program by si tedy nyní měl “ohlídat“ i rozdíl mezi hodnotami zakončovacích odporů. Princip je takový, že parametr delta_kk bude dán “porovnáním“ vypočtené hodnoty kk s optimální hodnotou kk (kk_{opt}). Kterou si bude možno v programu zvolit. Budu-li chtít, aby byla zadaná pásmová propust zakončena symetricky, pak bude muset být $kk_{opt} = 0,5$. ($R1 = R2$)

Ukázka nově vzniklého vyhodnocení funkce fitval:

delta_kk:= evalf(abs(kk/kk_opt-1)), tak aby byl výsledek pro delta_kk = 0 pro správnou hodnotu

(evalf(2e3*abs(delta[j])^2),j=1..Ndo) + 1e3*(delta_h)^2 + 1e-8*PP3 + 1e4*delta_kk

Konstanta parametru delta_kk (1e4) byla zvolena empiricky

Výsledky s parametrem delta_kk(pro $kk_{opt} = 0,5; 0,25; 0,75$):

kk_opt = 0,5:

Prvky vektoru Xbest při hodnotě fit_best = 20.22841337:

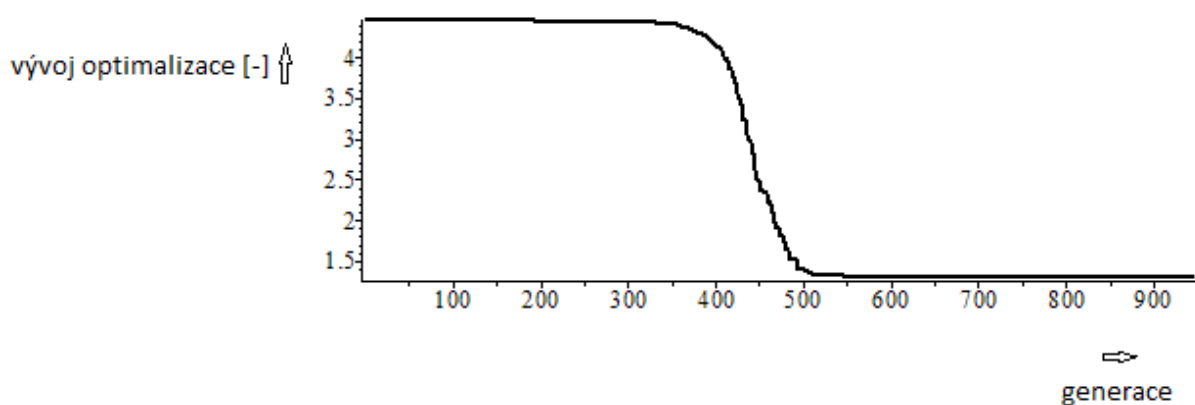
$X_{best} = [124.4693736, 124.4693737, 0.02369710789, 0.02191195189, 0.02385046914, 0.001691529359, 0.001879173512]$

Po přiřazení hodnot k obvodovým prvkům:

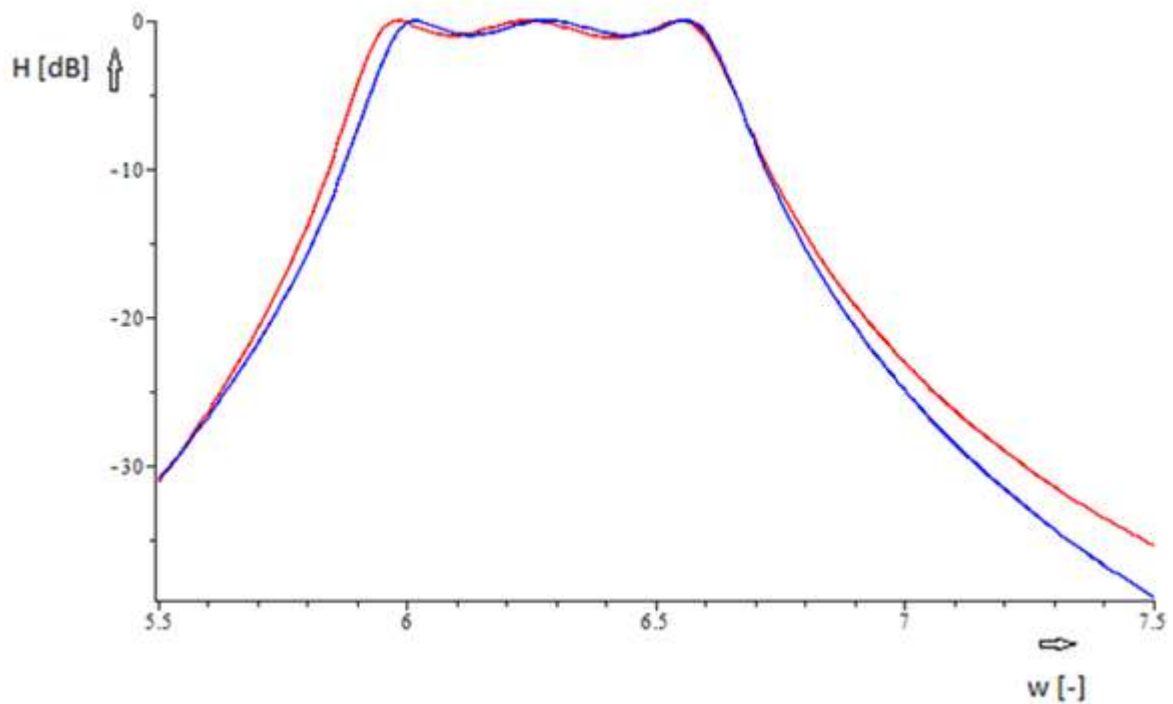
$R1 := 124.4693736$
 $R2 := 124.4693737$
 $C3 := 0.02369710789$
 $C4 := 0.02191195189$
 $C5 := 0.02385046914$
 $CV1 := 0.001691529359$
 $CVI := 0.001879173512$
 $L1 := 1$
 $L3 := 1$
 $L5 := 1$

Obr. 4.5.1 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů

(kk_opt = 0,5):



Obr. 4.5.2 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($kk_{opt} = 0,5$):



$kk_{opt} = 0,25$:

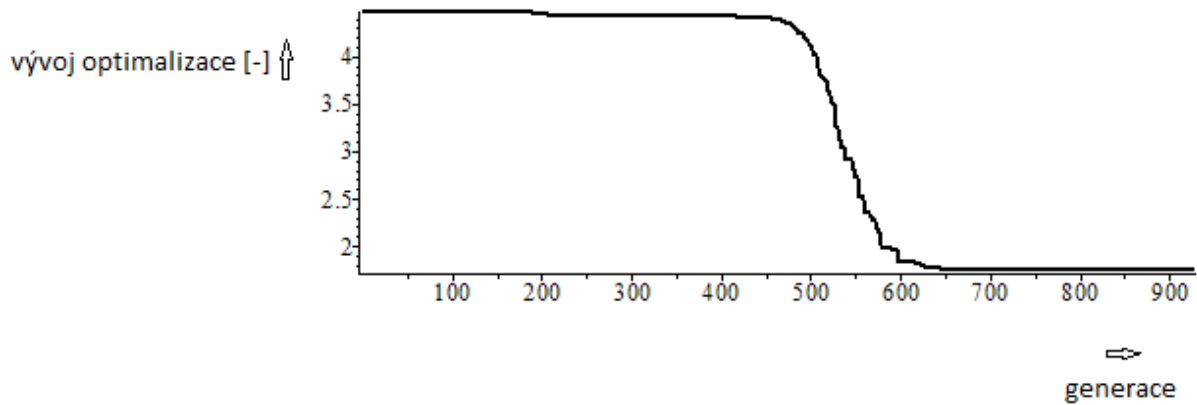
Prvky vektoru X_{best} při hodnotě $fit_best = 56.58509893$:

$X_{best} = [278.7802935, 92.92676422, 0.02372568976, 0.02188790767, 0.02368261182, 0.001695622696, 0.001964142252]$

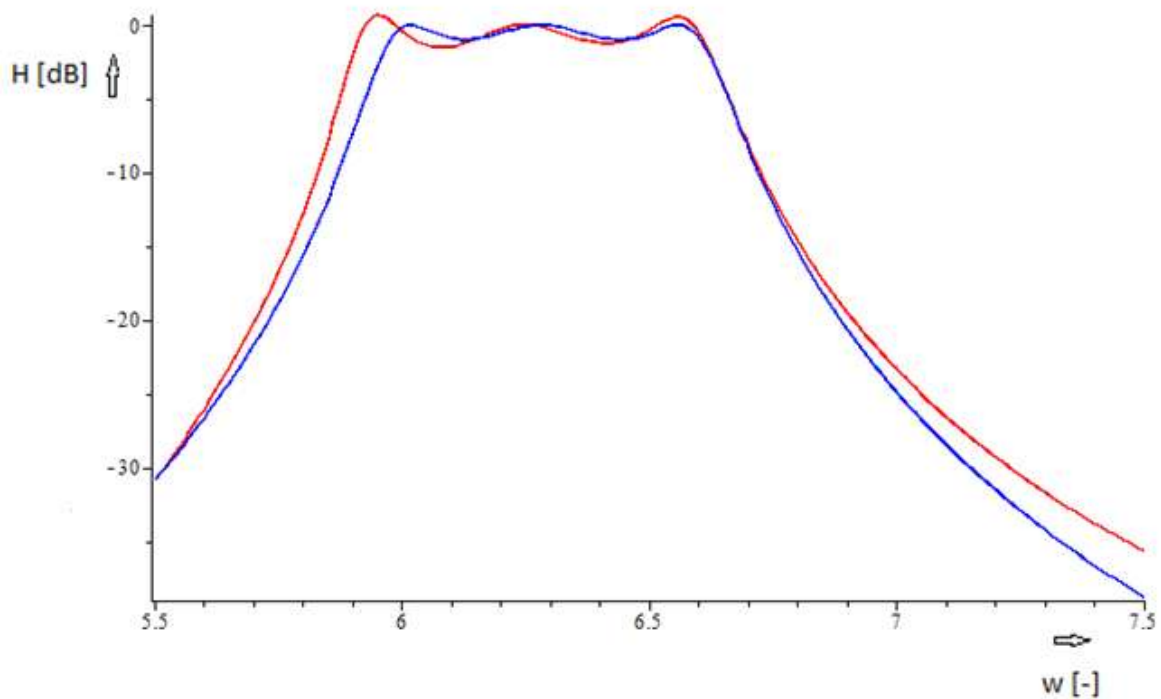
Po přiřazení hodnot k obvodovým prvkům:

$R1 := 278.7802935$
 $R2 := 92.92676422$
 $C3 := 0.02372568976$
 $C4 := 0.02188790767$
 $C5 := 0.02368261182$
 $CVI := 0.001695622696$
 $CVI := 0.001964142252$
 $L1 := 1$
 $L3 := 1$
 $L5 := 1$

Obr. 4.5.3 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů
($kk_{opt} = 0,25$):



Obr. 4.5.4 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($kk_{opt} = 0,25$):



$kk_{opt} = 0,75$:

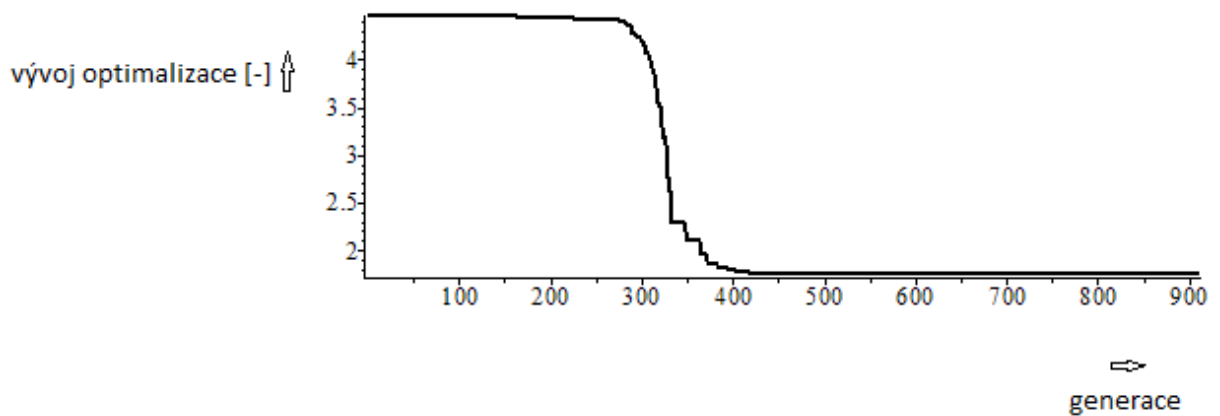
Prvky vektoru X_{best} při hodnotě $fit_{best} = 56.58506327$:

$$X_{best} = [92.92490701, 278.7747212, 0.02368253868, 0.02188803932, 0.02372572909, 0.001964173677, 0.001695596297]$$

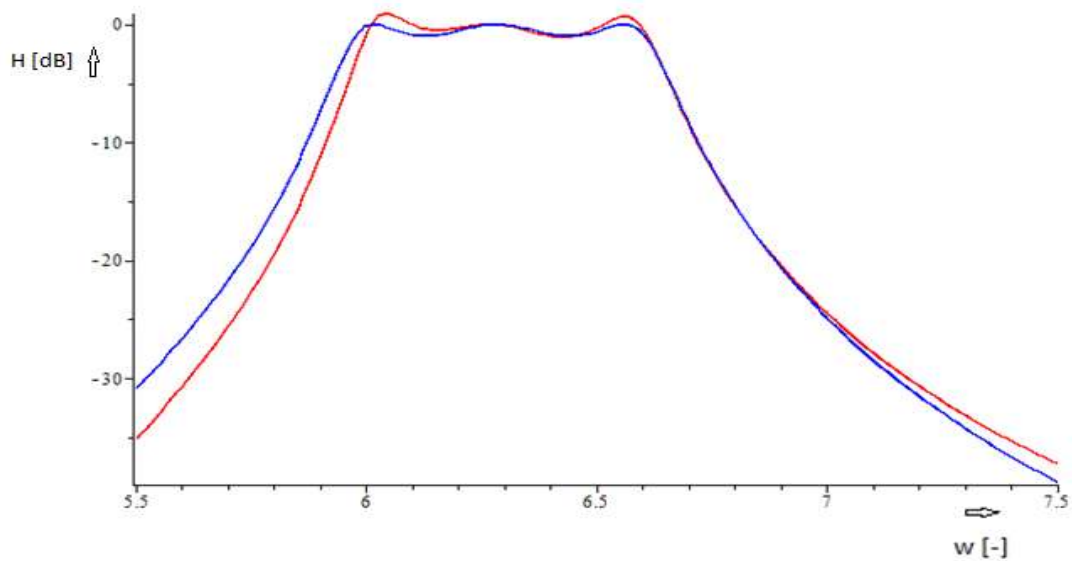
Po přiřazení hodnot k obvodovým prvkům:

$R1 := 92.92490701$
 $R2 := 278.7747212$
 $C3 := 0.02368253868$
 $C4 := 0.02188803932$
 $C5 := 0.02372572909$
 $CV1 := 0.001964173677$
 $CVI := 0.001695596297$
 $L1 := 1$
 $L3 := 1$
 $L5 := 1$

Obr. 4.5.5 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů
($kk_{opt} = 0,75$):



Obr. 4.5.6 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při hledání optimálních hodnot zakončovacích odporů ($kk_{opt} = 0,75$):



Druhá “metoda“ pro minimalizaci rozdílu hodnot mezi zakončovacími odpory se také jeví jako částečně úspěšná. Z programátorského hlediska mi přijde nápad s rozšířením vyhodnocovací procedury fitval o člen delta_kk jako velmi zajímavý. Pro symetrické zakončení filtru je potřeba nastavit parametr kk_opt na hodnotu 0,5. Pro úplnost opět vkládám i výsledky pro $\text{kk_opt} = 0,25$ a $\text{kk_opt} = 0,75$. Podobně jako v předchozí kapitole, tak i zde jsou vidět změny v šířce pásma a ve zvlnění v propustném pásmu filtru pro hodnoty $\text{kk_opt} = 0,25$ respektive $\text{kk_opt} = 0,75$. Problém je totožný jako v předchozí kapitole, tedy podmínka stejných indukčností. Pro lepší průběh frekvenční charakteristiky je potřeba zrušit podmínku totožných indukčností.

5) Kompenzace vlivu ztrát reálných induktorů a kapacitorů

5.1) Zapojení zadané pásmové propusti se ztrátami v induktorech

Pro tento konkrétní případ je z počátku vhodné nastavit jakost induktorů na přibližně desetinásobek Q-pólů přenosové funkce. Což je v mém případě $QL = 10 \cdot 40 = 400$. První věc, kterou je potřeba udělat je změna schématu. Prvotně zadaný obvod do programu Maple pomocí knihovny Syrup byl bezztrátový. Nejdříve budu řešit ztráty v induktorech, takže do schématu přidám k ideálním cívkám sériově ztrátové odpory. Jejich hodnota bude dána vztahem –

Rovnice 5.1.1 – vztah pro výpočet RQ

$$RQ = \frac{w \cdot L}{QL}$$

Zadání obvodu pomocí knihovny Syrup se změní takto:

V 1 0

R1 1 2

L1 2 20

RQ1 0 20 (omN*L1)/QL

C2 2 0

CV1 2 3

L3 3 30

RQ3 0 30 (omN*L3)/QL

C4 3 0

CV2 3 4

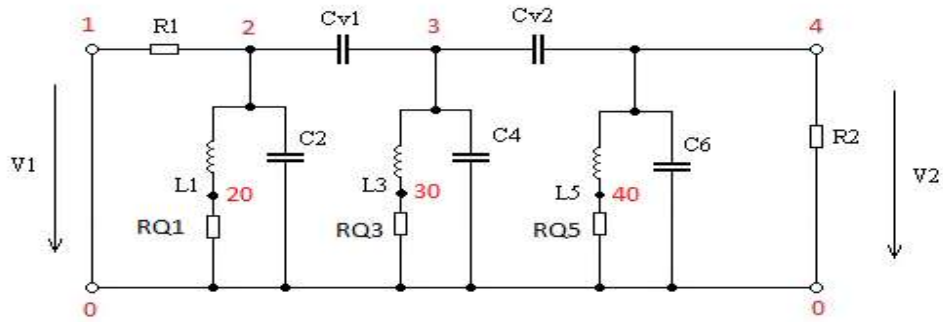
L5 4 40

RQ5 0 40 (omN*L5)/QL

C6 4 0

R2 4 0

Obr. 5.1.1 – schéma zadané pásmové propusti se ztrátami v induktorech:



Výsledky pro QL = 400:

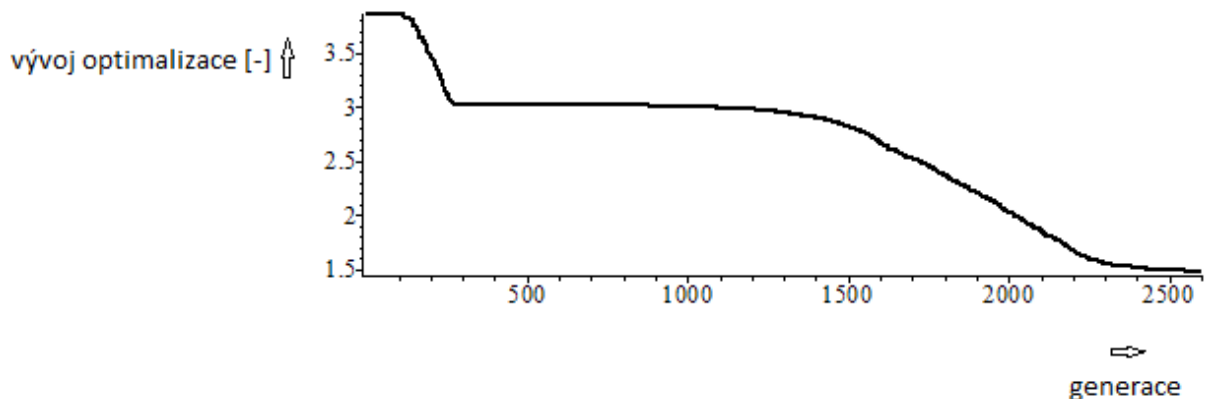
Prvky vektoru X_{best} při hodnotě $fit_best = 29.81669183$:

$$X_{best} = [0.008350948881, 2.962149758, 0.04530818787, 0.3771010244, 0.008321388129, 2.977552375, 0.09290601257, 0.09361884594]$$

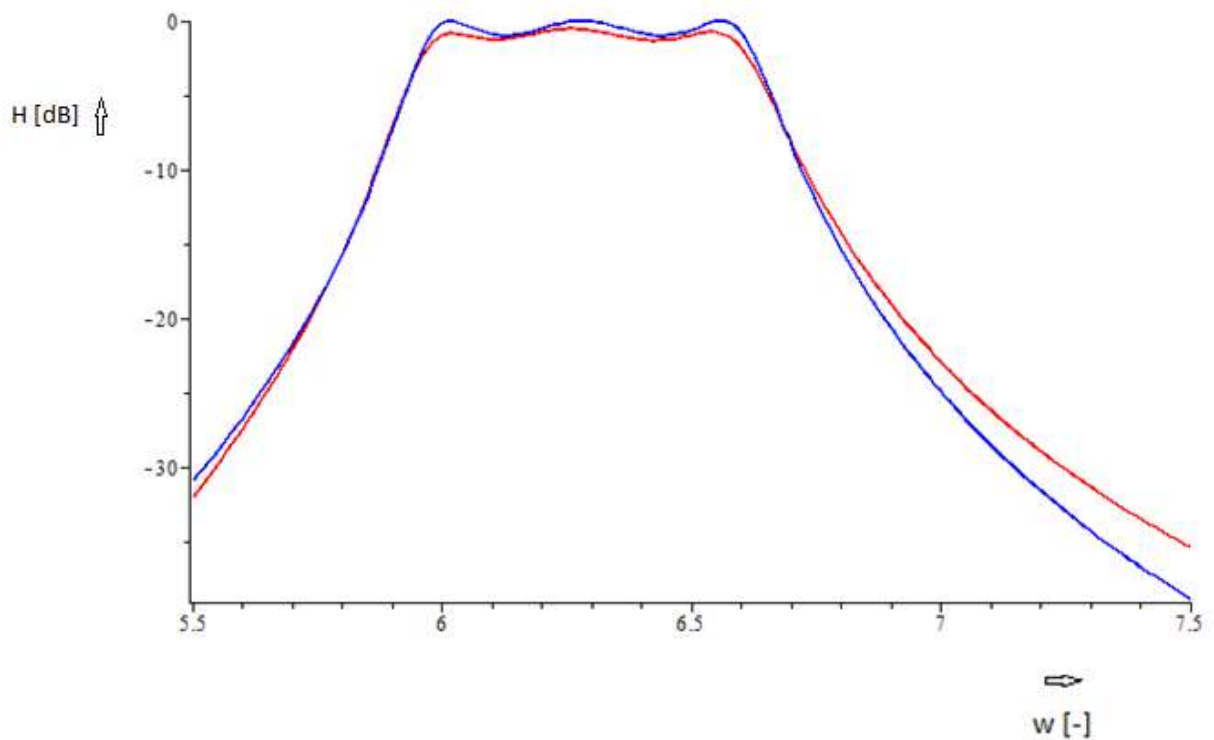
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008350948881$
 $C2 := 2.962149758$
 $L3 := 0.04530818787$
 $C4 := 0.3771010244$
 $L5 := 0.008321388129$
 $C6 := 2.977552375$
 $CV1 := 0.09290601257$
 $CV2 := 0.09361884594$
 $RQ1 := 0.0001311763983$
 $RQ3 := 0.0007116993509$
 $RQ5 := 0.0001307120591$

Obr. 5.1.2 - průběh konvergence při jakosti QL = 400:



Obr. 5.1.3 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 400:



Výsledky pro QL = 400 s použitím pomocné optimalizační metody Simplex:

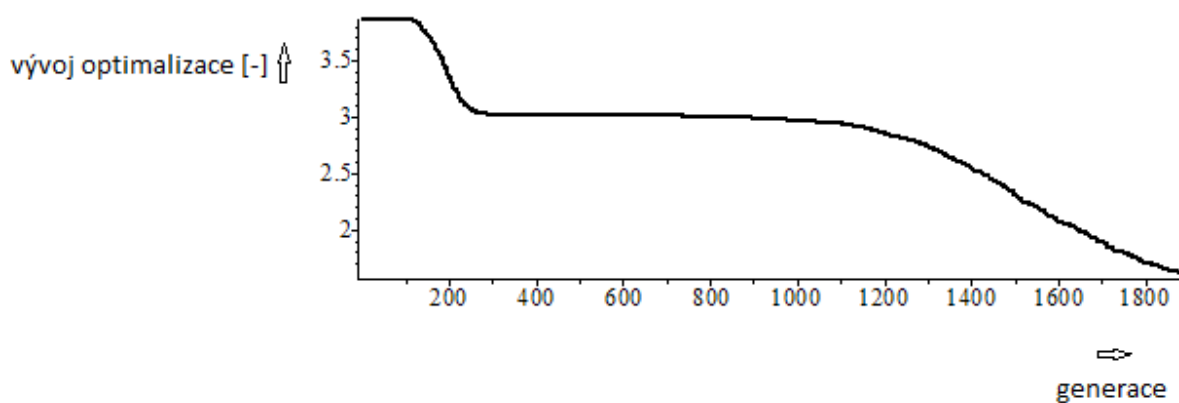
Prvky vektoru Xbest při hodnotě fit_best = 27.8661347784391111:

$X_{best} = [0.008855924087, 2.813922109, 0.05843970130,$
 $0.2797495922, 0.009211962156, 2.693855387, 0.07826336725,$
 $0.07824796121]$

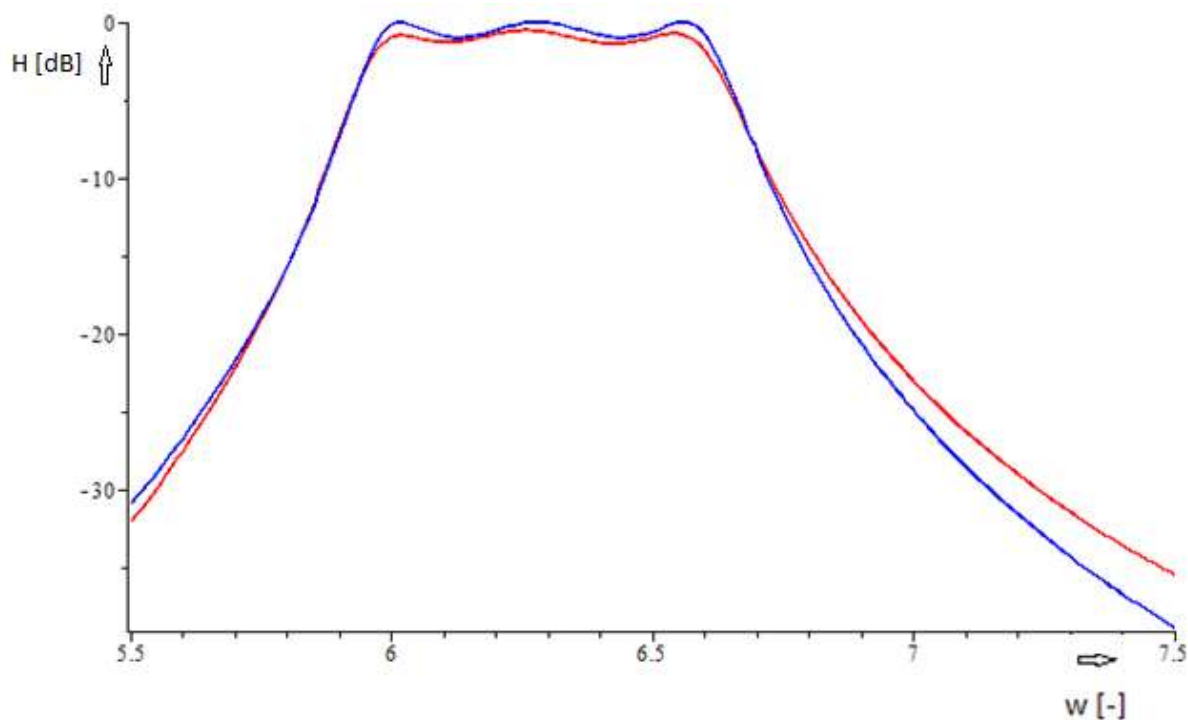
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00825440121928083$
 $C2 := 2.98184911907415$
 $L3 := 0.0318202820625104$
 $C4 := 0.578872412041811$
 $L5 := 0.00825074661876449$
 $C6 := 2.98321174198822$
 $CV1 := 0.111826915545924$
 $CV2 := 0.111862551562584$
 $RQ1 := 0.000129659831168307$
 $RQ3 := 0.000499831821878953$
 $RQ5 := 0.000129602424837629$

Obr. 5.1.4 - průběh konvergence při jakosti QL = 400 s použitím metody Simplex:



Obr. 5.1.5 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 400 s použitím metody Simplex:



Po konzultaci s vedoucím práce otestuji obvod ještě pro QL = 200, 300, 600 a porovnám jednotlivé výsledky.

Výsledky pro QL = 200:

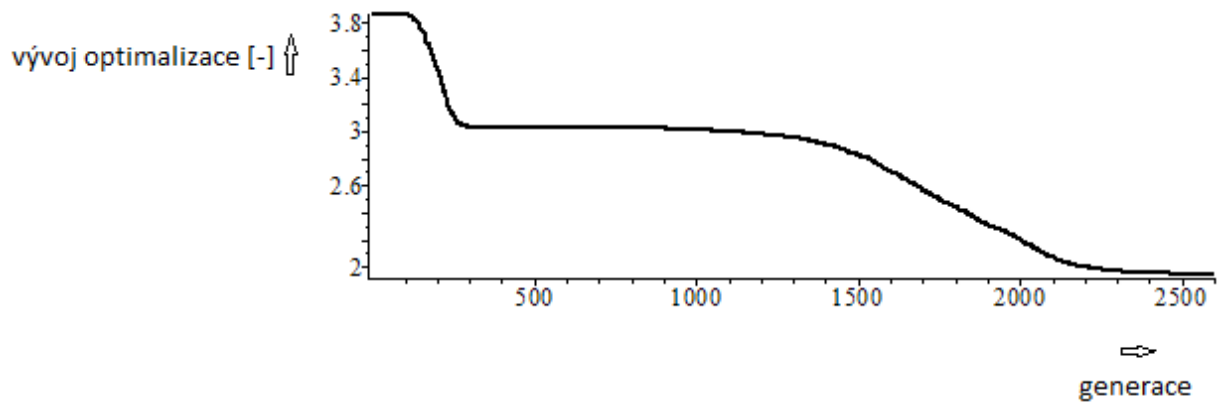
Prvky vektoru Xbest při hodnotě fit_best = 88.24597337:

$$X_{best} = [0.008778139274, 2.817393790, 0.04490676575, \\ 0.3844955719, 0.008780524213, 2.817131105, 0.09260771606, \\ 0.09202741216]$$

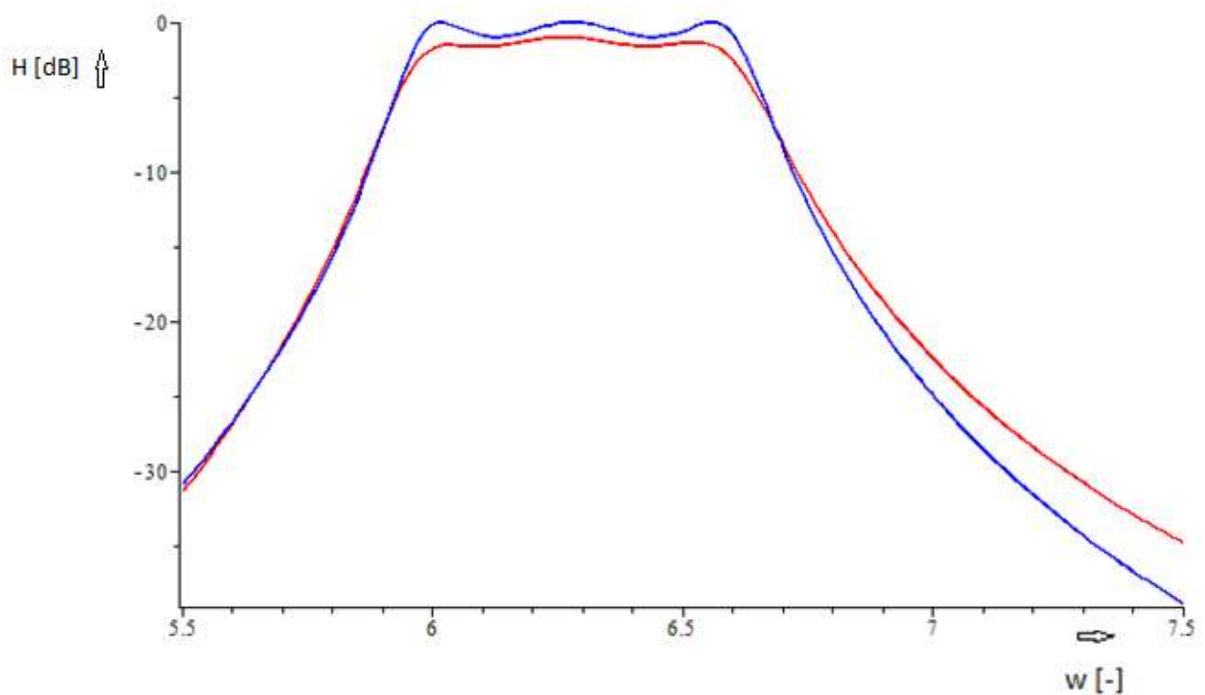
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008778139274$
 $C2 := 2.817393790$
 $L3 := 0.04490676575$
 $C4 := 0.3844955719$
 $L5 := 0.008780524213$
 $C6 := 2.817131105$
 $CV1 := 0.09260771606$
 $CV2 := 0.09202741216$
 $RQ1 := 0.0002757733786$
 $RQ3 := 0.001410787654$
 $RQ5 := 0.0002758483037$

Obr. 5.1.6 - průběh konvergence při jakosti QL = 200:



Obr. 5.1.7 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 200:



Výsledky pro QL = 200 s použitím pomocné optimalizační metody Simplex:

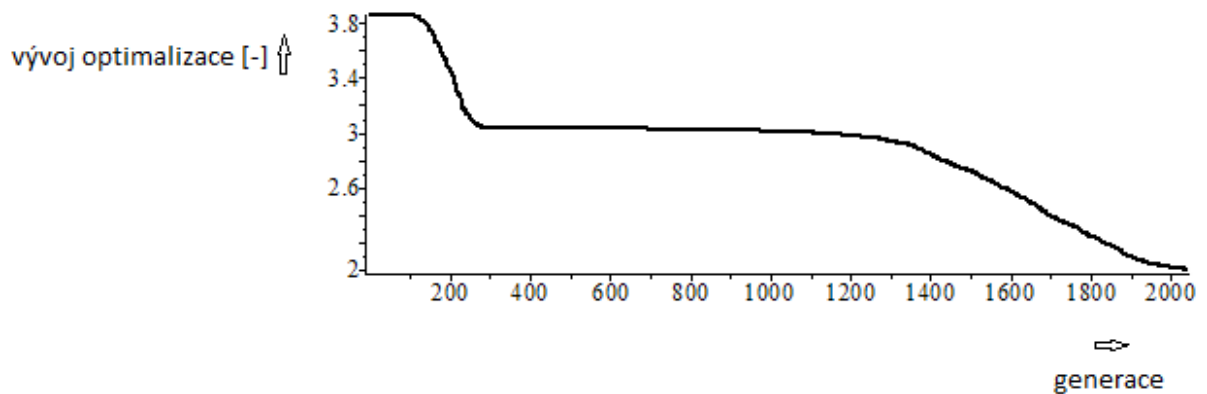
Prvky vektoru Xbest při hodnotě fit_best = 94.6470418435071963:

$X_{best} = [0.009333266333, 2.667147799, 0.06423071436,$
 $0.2482452113, 0.009294532674, 2.677367680, 0.07448821208,$
 $0.07438692233]$

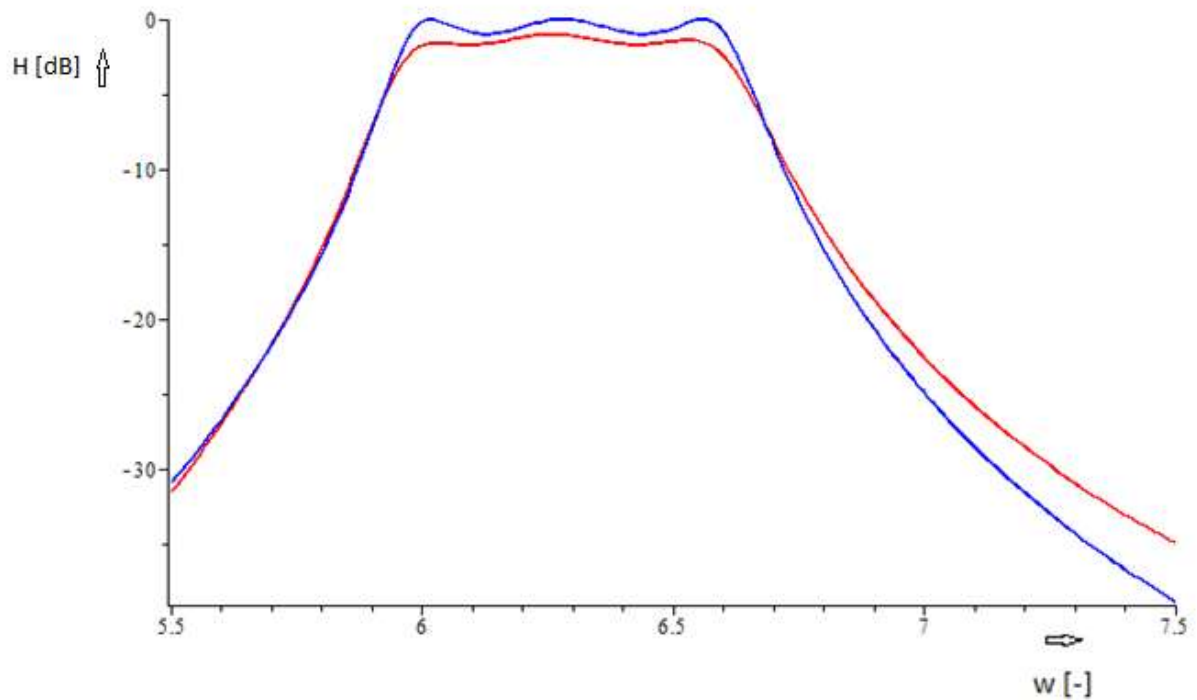
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00855228865749634$
 $C2 := 2.90405014576217$
 $L3 := 0.0570788731283471$
 $C4 := 0.281204790617021$
 $L5 := 0.00855229059627164$
 $C6 := 2.90412552977806$
 $CV1 := 0.0831638459749635$
 $CV2 := 0.0832887051250968$
 $RQ1 := 0.000268678072212780$
 $RQ3 := 0.00179318568518613$
 $RQ5 := 0.000268678133121203$

Obr. 5.1.8 - průběh konvergence při jakosti QL = 200 s použitím metody Simplex:



Obr. 5.1.9 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti $QL = 200$ s použitím metody Simplex:



Použitím Simplexové metody jsem získal téměř totožný výsledek jako bez jejího použití. Bylo však zapotřebí o 500 generací méně.

Výsledky pro $QL = 300$:

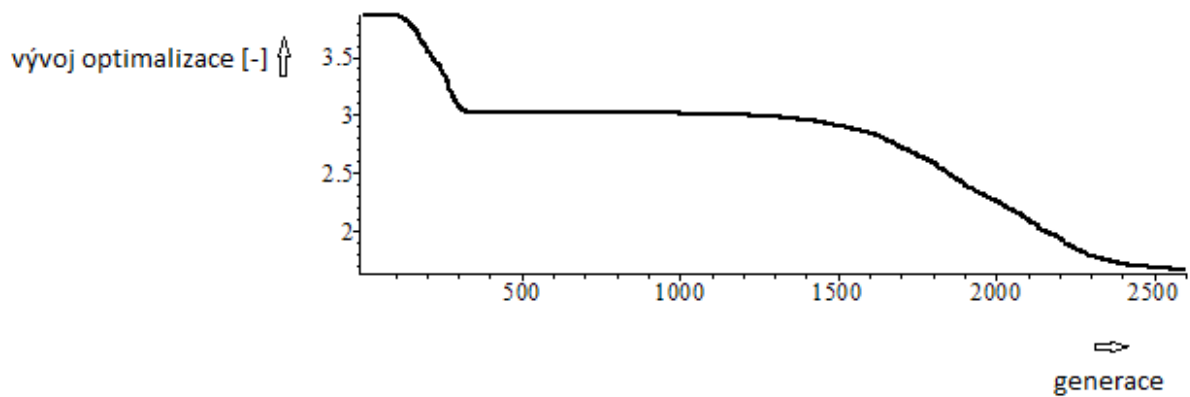
Prvky vektoru X_{best} při hodnotě $fit_best = 46.44559261$:

$$X_{best} = [0.008573259825, 2.889459822, 0.04655647257, 0.3670846954, 0.008607826632, 2.875016340, 0.09027408973, 0.09129028175]$$

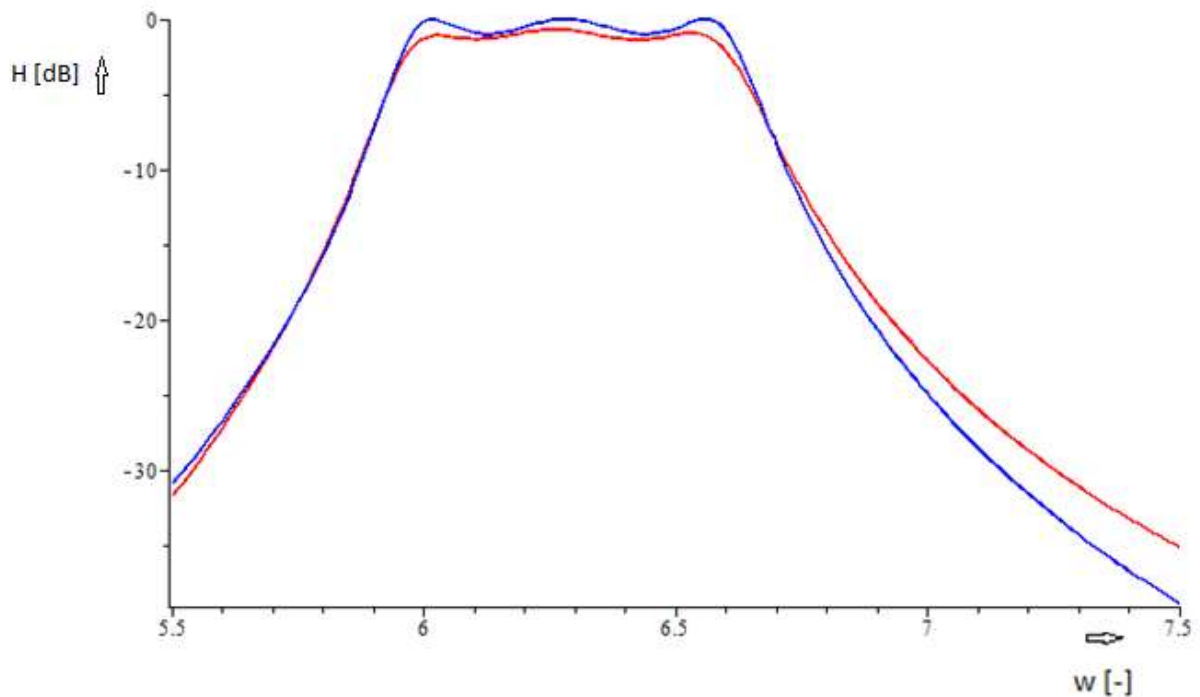
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008573259825$
 $C2 := 2.889459822$
 $L3 := 0.04655647257$
 $C4 := 0.3670846954$
 $L5 := 0.008607826632$
 $C6 := 2.875016340$
 $CV1 := 0.09027408973$
 $CV2 := 0.09129028175$
 $RQ1 := 0.0001795579339$
 $RQ3 := 0.0009750764816$
 $RQ5 := 0.0001802818995$

Obr. 5.1.10 - průběh konvergence při jakosti QL = 300:



Obr. 5.1.11 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 300:



Výsledky pro QL = 300 s použitím pomocné optimalizační metody Simplex:

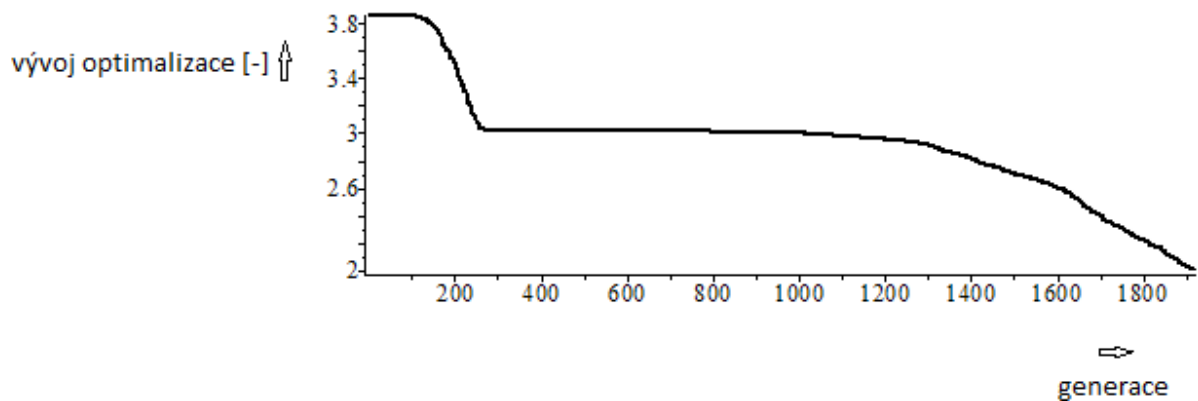
Prvky vektoru X_{best} při hodnotě $fit_best = 51.8638553000059090$:

$X_{best} = [0.01128274150, 2.202577222, 0.07787593264,$
 $0.2080097218, 0.01110543012, 2.231454866, 0.05913197665,$
 $0.06132470479]$

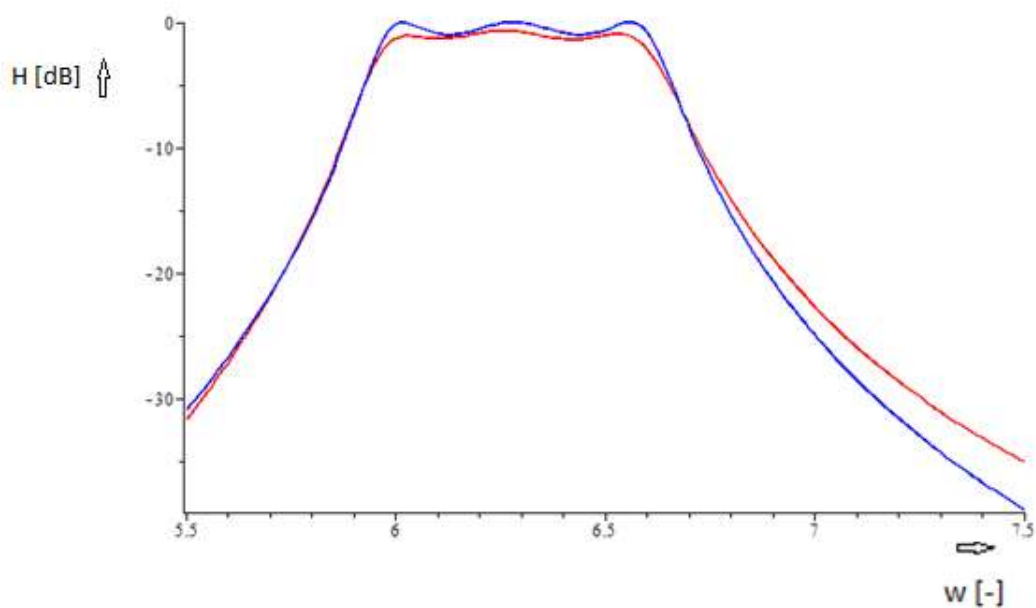
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00865716694119305$
 $C2 := 2.86838225237262$
 $L3 := 0.0573017621769334$
 $C4 := 0.282674438240385$
 $L5 := 0.00865716705412803$
 $C6 := 2.86948285303518$
 $CV1 := 0.0811143818769580$
 $CV2 := 0.0818959608247685$
 $RQ1 := 0.000181315280474882$
 $RQ3 := 0.00120012530096640$
 $RQ5 := 0.000181315282840187$

Obr. 5.1.12 - průběh konvergence při jakosti $QL = 300$ s použitím metody Simplex:



Obr. 5.1.13 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti $QL = 300$ s použitím metody Simplex:



Použitím Simplexové metody se v tomto případě zkrátila doba konvergence o 700 generací.

Výsledky pro QL = 600:

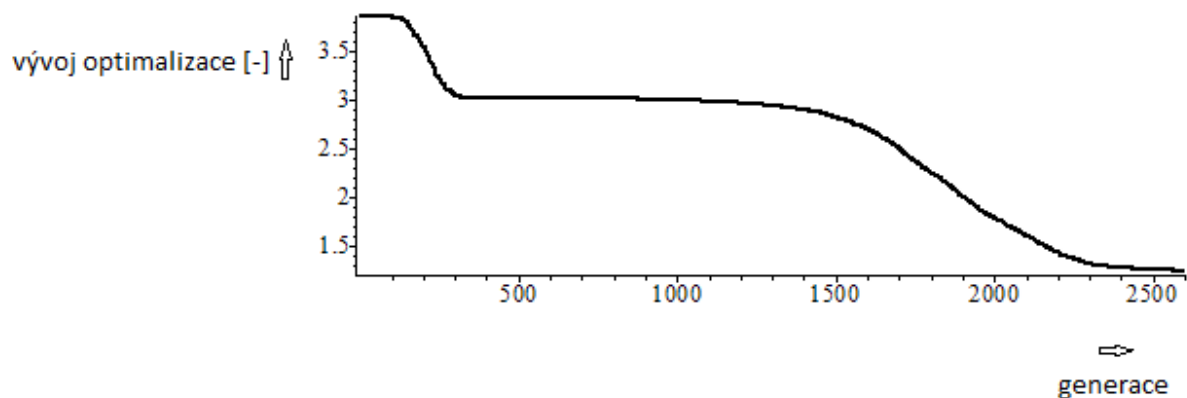
Prvky vektoru Xbest při hodnotě fit_best = 16.98764059:

$X_{best} = [0.008287966172, 2.986576456, 0.04253903208,$
 $0.4075578096, 0.008219238057, 3.009687535, 0.09623853182,$
 $0.09626501063]$

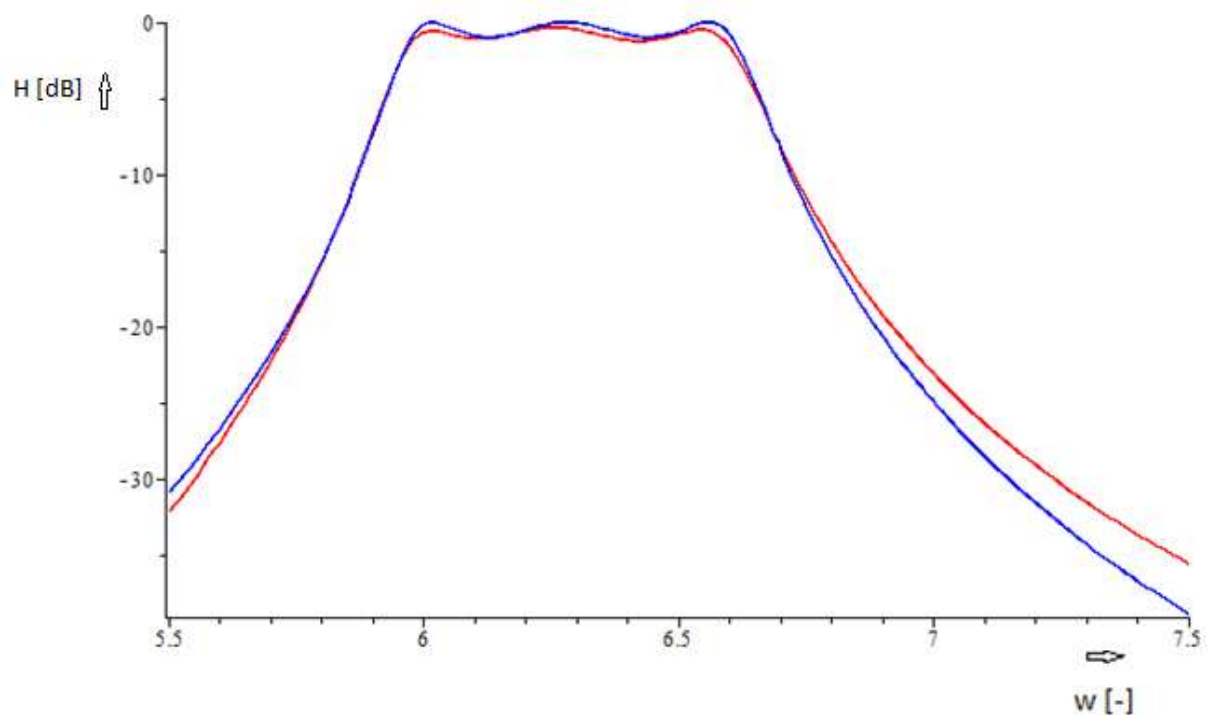
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008287966172$
 $C2 := 2.986576456$
 $L3 := 0.04253903208$
 $C4 := 0.4075578096$
 $L5 := 0.008219238057$
 $C6 := 3.009687535$
 $CV1 := 0.09623853182$
 $CV2 := 0.09626501063$
 $RQ1 := 0.00008679137878$
 $RQ3 := 0.0004454677022$
 $RQ5 := 0.00008607165964$

Obr. 5.1.14 - průběh konvergence při jakosti QL = 600:



Obr. 5.1.15 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 600:



Výsledky pro QL = 600 s použitím pomocné optimalizační metody Simplex:

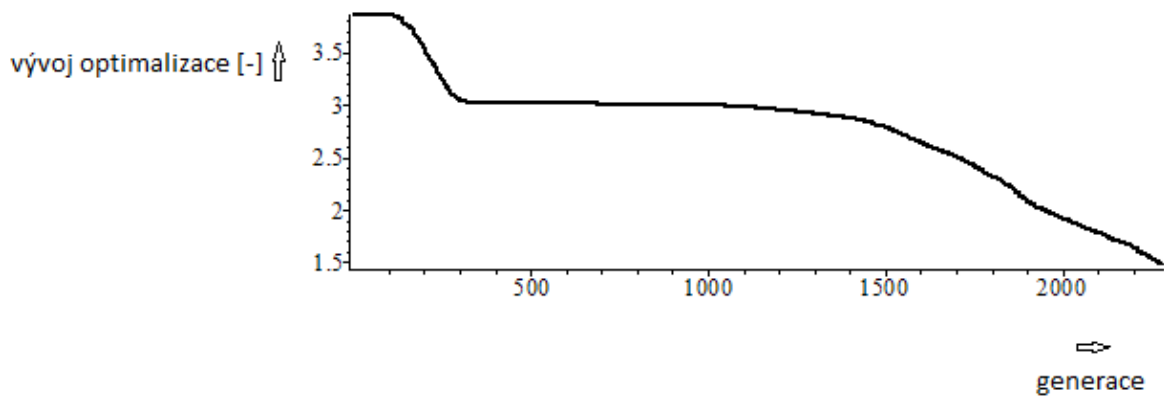
Prvky vektoru Xbest při hodnotě fit_best = 24.0699243343002145:

$X_{best} = [0.009187078259, 2.709479638, 0.05742298173,$
 $0.2882127867, 0.009175619200, 2.707489107, 0.07961937862,$
 $0.07553368657]$

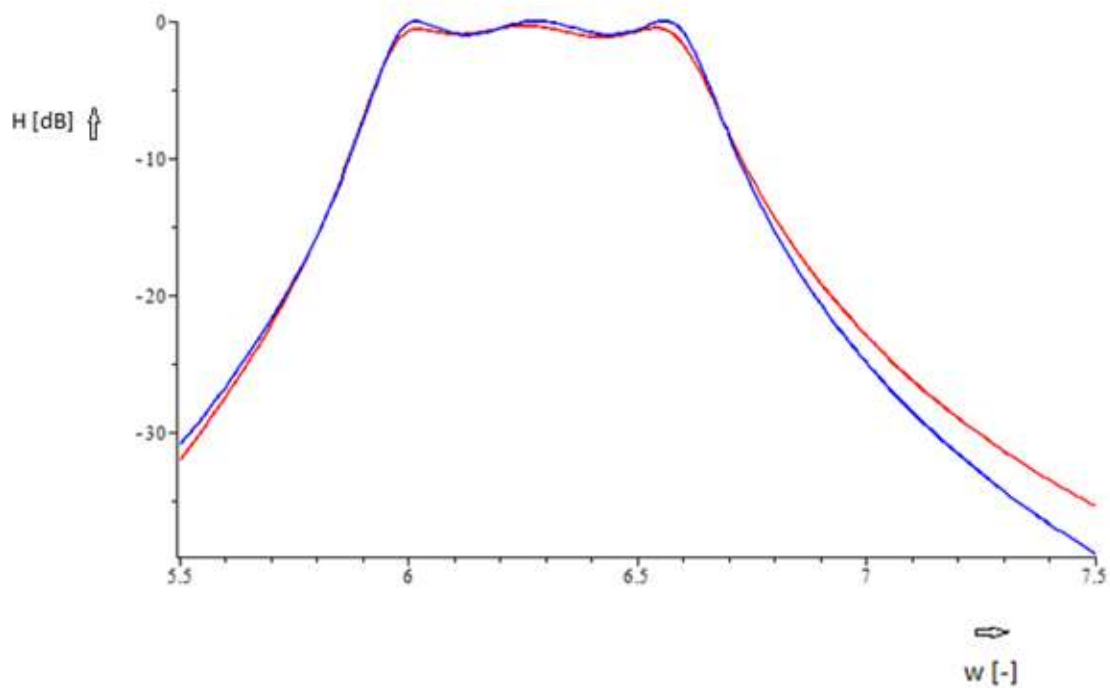
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00845637364649571$
 $C2 := 2.93852097418154$
 $L3 := 0.0566490281297789$
 $C4 := 0.286024076074868$
 $L5 := 0.00845637364650285$
 $C6 := 2.93864026017242$
 $CV1 := 0.0822198130089963$
 $CV2 := 0.0822174254600770$
 $RQ1 := 0.0000885549377295124$
 $RQ3 := 0.000593227235240346$
 $RQ5 := 0.0000885549377295873$

Obr. 5.1.16 - průběh konvergence při jakosti QL = 600 s použitím metody Simplex:



Obr. 5.1.17 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 600 s použitím metody Simplex:



Pro QL = 600 jsem použitím Simplexové metody zkrátila dobu konvergence o 300 generací.

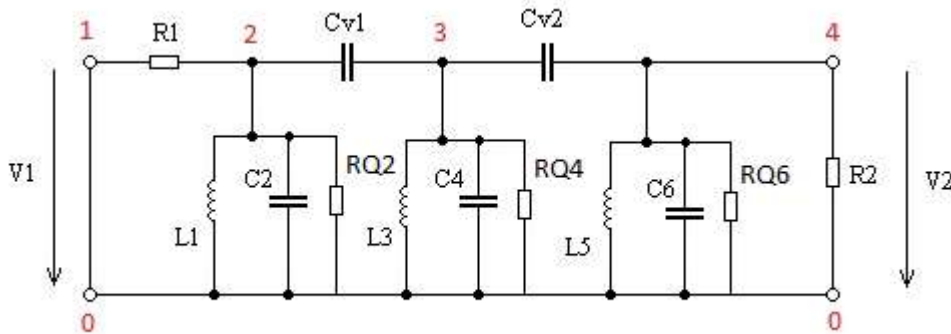
5.2) Zapojení zadané pásmové propusti se ztrátami v kapacitorech

V praxi by se mohla hodnota jakosti u kondenzátorů pohybovat kolem hodnoty $Q_C = 10000$. První hodnota pro Q_C bude tedy 10000. Po-té jsem se rozhodl otestovat obvod pro hodnoty $Q_C = 5000$ a $Q_C = 15000$. Ke kondenzátorům připojím paralelně ztrátové odpory. Jejich hodnota bude dána vztahem -

Rovnice 5.2.1 – vztah pro výpočet RQ

$$RQ = \frac{Q_C}{\omega * C}$$

Obr. 5.2.1 – schéma zadané pásmové propusti se ztrátami v kapacitorech:



Zápis obvodu pomocí knihovny Syrup bude vypadat následovně:

V 1 0
R1 1 2
L1 2 20
C2 2 0
RQ2 2 0 (QC)/(omN*C2)
CV1 2 3
L3 3 30
C4 3 0
RQ4 3 0 (QC)/(omN*C4)
CV2 3 4
L5 4 40
C6 4 0
RQ6 4 0 (QC)/(omN*C6)
R2 4 0

Výsledky pro QC = 10000:

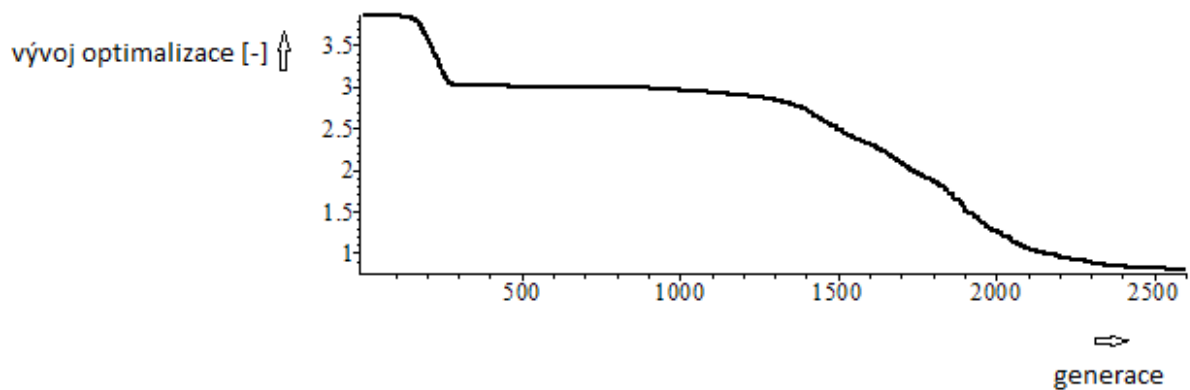
Prvky vektoru Xbest při hodnotě fit_best = 6.314100113:

$X_{best} = [0.008104075836, 3.066459619, 0.03782530919,$
 $0.4691681776, 0.008024321915, 3.067223300, 0.1072980213,$
 $0.09733932324]$

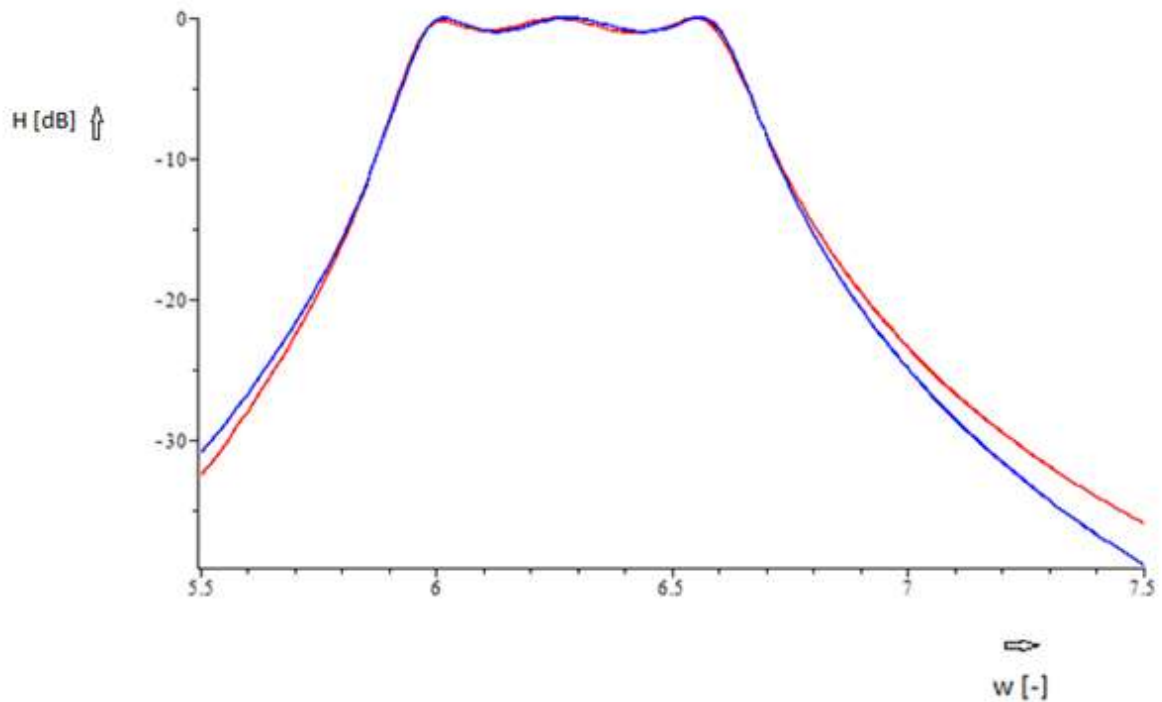
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008104075836$
 $C2 := 3.066459619$
 $L3 := 0.03782530919$
 $C4 := 0.4691681776$
 $L5 := 0.008024321915$
 $C6 := 3.067223300$
 $CV1 := 0.1072980213$
 $CV2 := 0.09733932324$
 $RQ2 := 519.0185519$
 $RQ4 := 3392.279159$
 $RQ6 := 518.8893260$

Obr. 5.2.2 - průběh konvergence při jakosti QC = 10000:



Obr. 5.2.3 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 10000:



Výsledky pro QC = 10000 s použitím pomocné optimalizační metody Simplex:

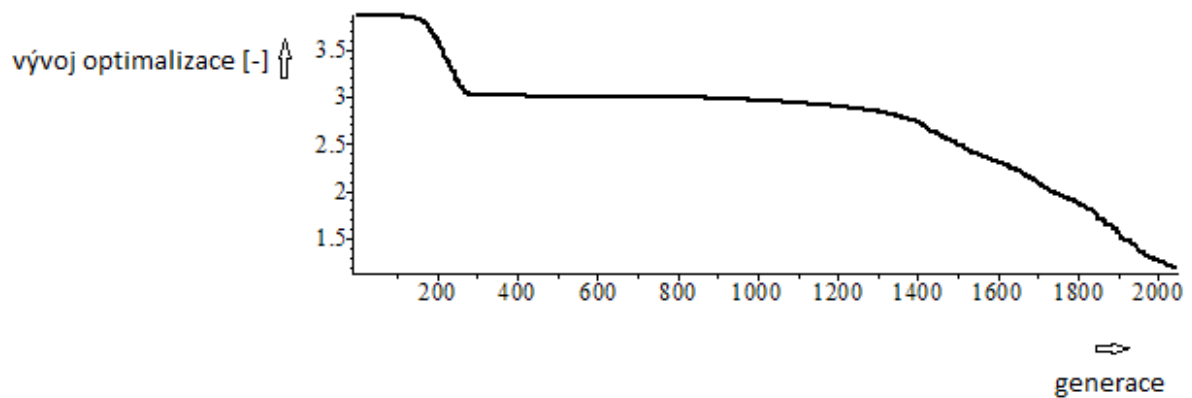
Prvky vektoru Xbest při hodnotě fit_best = 12.4919743800323513:

$X_{best} = [0.008798275952, 2.824023191, 0.05465274096, 0.3039963245, 0.008842656190, 2.812590331, 0.08782837368, 0.07414271377]$

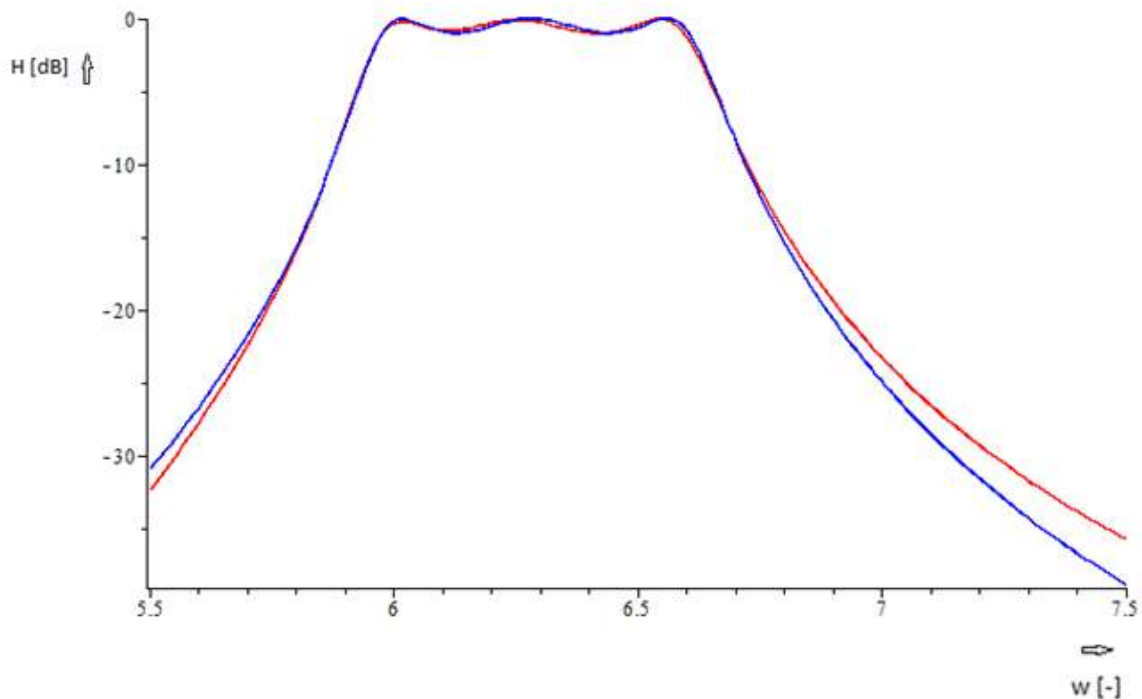
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00830635747105960$
 $C2 := 3.00673229307593$
 $L3 := 0.0540480214938632$
 $C4 := 0.302981476931812$
 $L5 := 0.00830635747106034$
 $C6 := 2.98174868193392$
 $CV1 := 0.0894788354758772$
 $CV2 := 0.0788219080373187$
 $RQ2 := 529.59709541674$
 $RQ4 := 5255.6238895416$
 $RQ6 := 534.03450732471$

Obr. 5.2.4 - průběh konvergence při jakosti QC = 10000 s použitím metody Simplex:



Obr. 5.2.5 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 10000 s použitím metody Simplex:



Použitím Simplexu se pro QC = 10000 zkrátila doba konvergence o 500 generací

Výsledky pro QC = 5000:

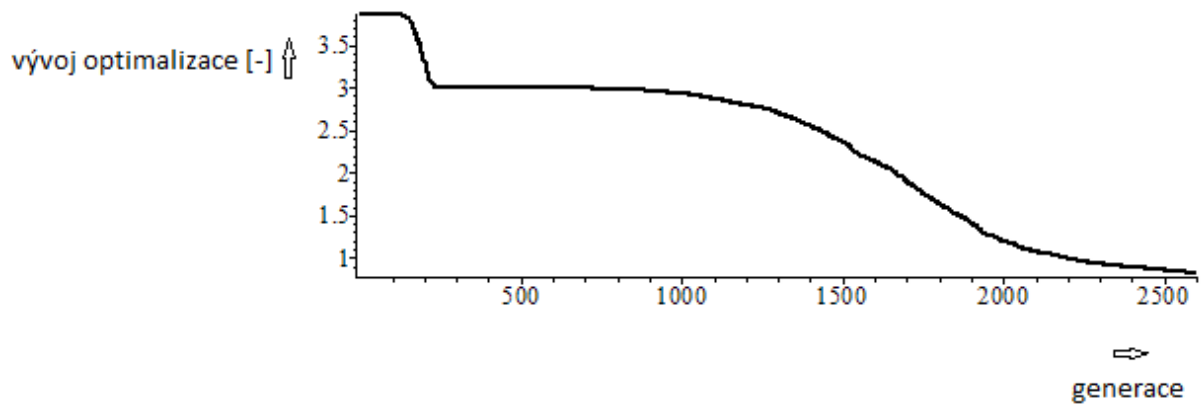
Prvky vektoru Xbest při hodnotě fit_best = 6.675051144:

$X_{best} = [0.008030632940, 3.097277339, 0.03898062125,$
 $0.4524466563, 0.008123542976, 3.030628474, 0.1055202317,$
 $0.09587288558]$

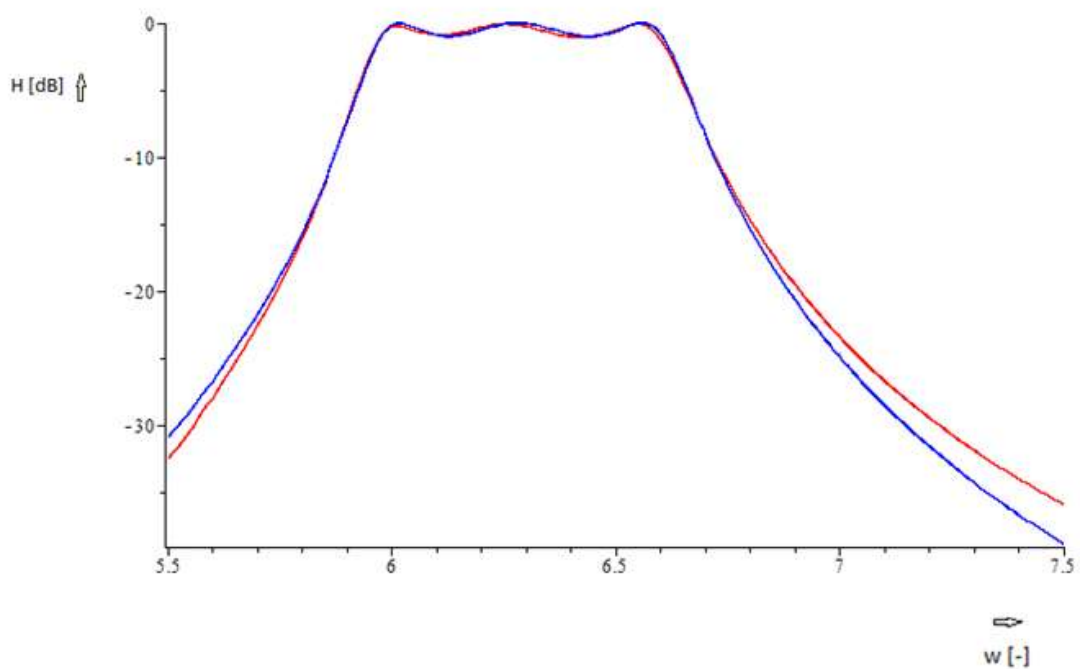
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008030632940$
 $C2 := 3.097277339$
 $L3 := 0.03898062125$
 $C4 := 0.4524466563$
 $L5 := 0.008123542976$
 $C6 := 3.030628474$
 $CV1 := 0.1055202317$
 $CV2 := 0.09587288558$
 $RQ2 := 256.9271745$
 $RQ4 := 1758.825498$
 $RQ6 := 262.5774562$

Obr. 5.2.6 - průběh konvergence při jakosti QC = 5000:



Obr. 5.2.7 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 5000:



Výsledky pro QC = 5000 s použitím pomocné optimalizační metody Simplex:

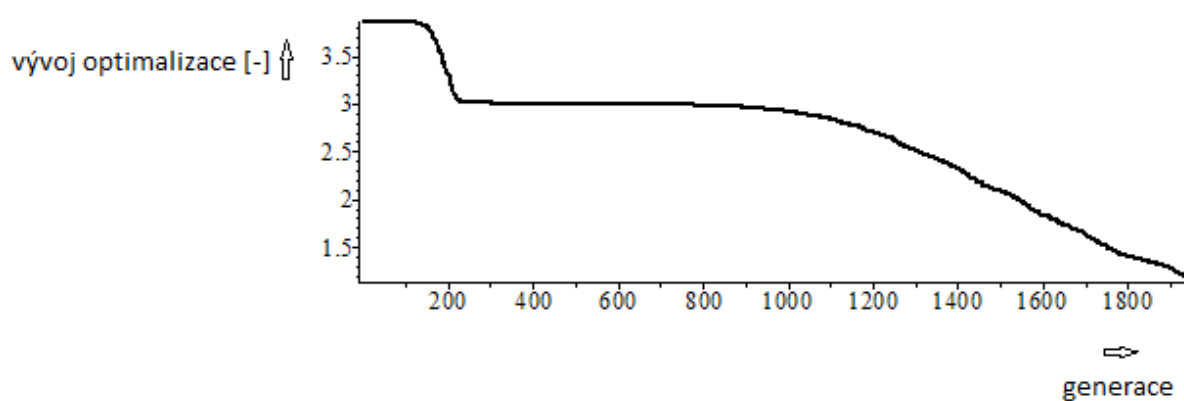
Prvky vektoru Xbest při hodnotě fit_best = 7.69210967791765921:

$X_{best} = [0.008650039574, 2.869692317, 0.05537220297,$
 $0.2978252433, 0.008751068860, 2.847053182, 0.07417678512,$
 $0.08776883311]$

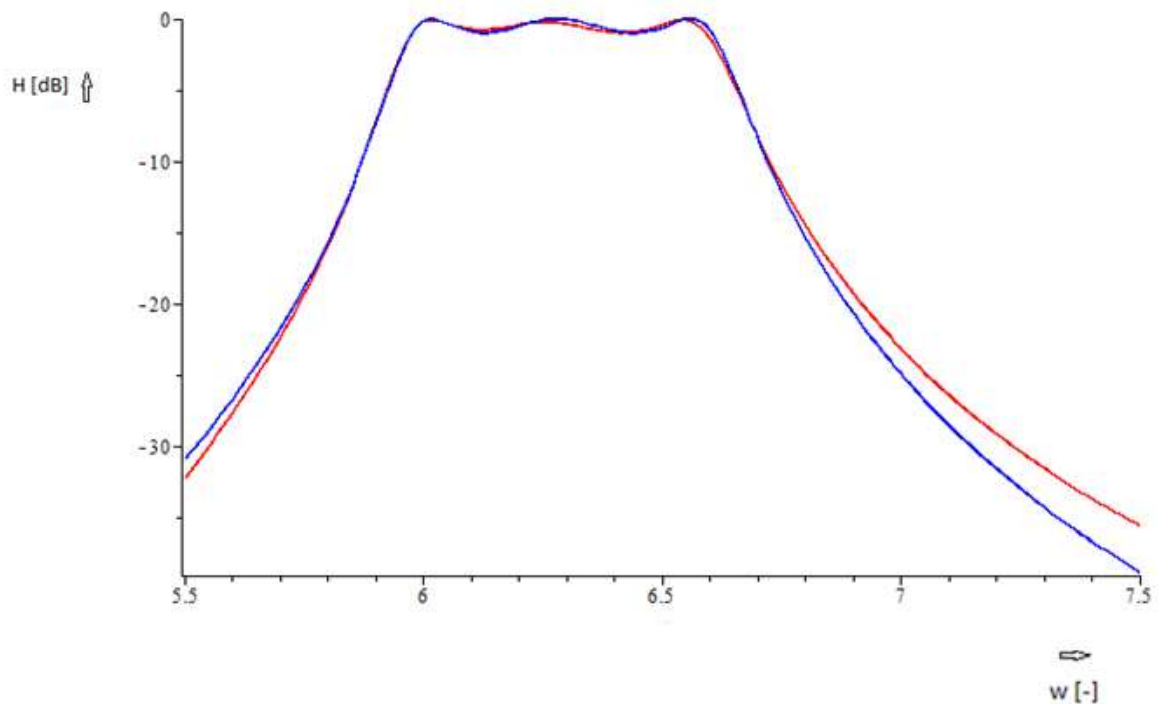
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00965220015117650$
 $C2 := 2.53576125702068$
 $L3 := 0.0297434449826941$
 $C4 := 0.630488996030077$
 $L5 := 0.00737661672160379$
 $C6 := 3.35358796652084$
 $CV1 := 0.101055974152489$
 $CV2 := 0.125015901879701$
 $RQ2 := 313.95004152648$
 $RQ4 := 1262.7949882417$
 $RQ6 := 237.41090235817$

Obr. 5.2.8 - průběh konvergence při jakosti QC = 5000 s použitím metody Simplex:



Obr. 5.2.9 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 5000 s použitím metody Simplex:



Metoda Simplex zkrátila v tomto případě dobu konvergence o 700 generací

Výsledky pro QC = 15000:

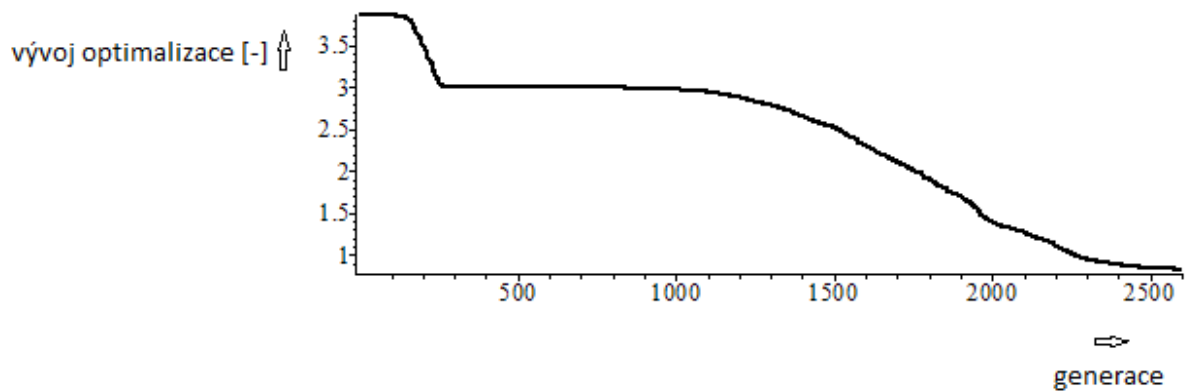
Prvky vektoru Xbest při hodnotě fit_best = 6.682531259:

$X_{best} = [0.008086177592, 3.044374494, 0.04036439226, 0.4335287005, 0.008094408601, 3.077639400, 0.09335969268, 0.1042903091]$

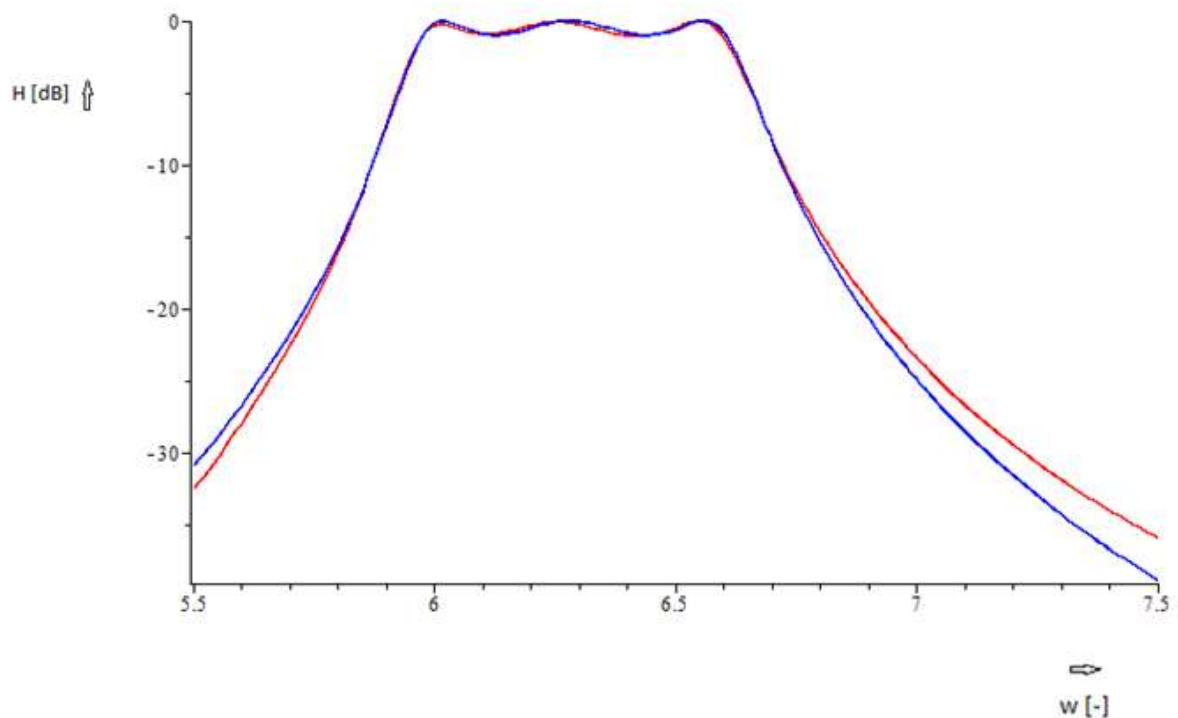
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008086177592$
 $C2 := 3.044374494$
 $L3 := 0.04036439226$
 $C4 := 0.4335287005$
 $L5 := 0.008094408601$
 $C6 := 3.077639400$
 $CV1 := 0.09335969268$
 $CV2 := 0.1042903091$
 $RQ2 := 784.1755839$
 $RQ4 := 5506.726875$
 $RQ6 := 775.6997607$

Obr. 5.2.10 - průběh konvergence při jakosti QC = 15000:



Obr. 5.2.11 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 15000:



Výsledky pro QC = 15000 s použitím pomocné optimalizační metody Simplex:

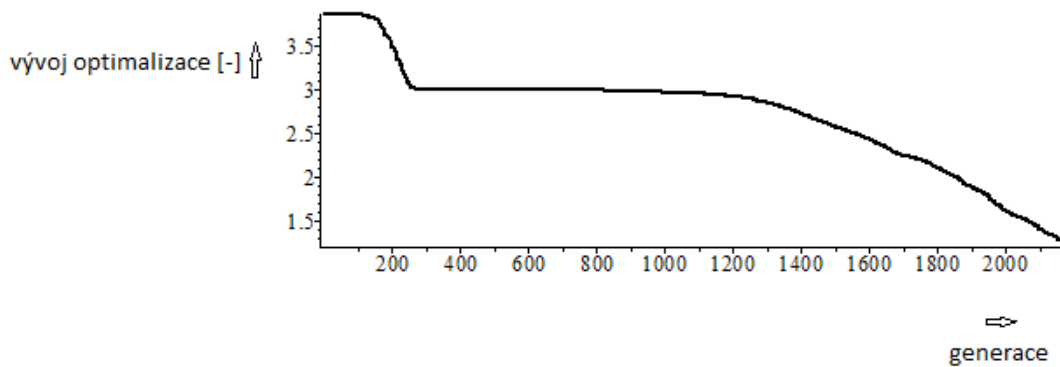
Prvky vektoru X_{best} při hodnotě $fit_best = 5.66969636528114673$:

$$X_{best} = [0.008675386586, 2.887381231, 0.05830285912, \\ 0.2777156982, 0.008534484873, 2.903925390, 0.08227775184, \\ 0.07637455953]$$

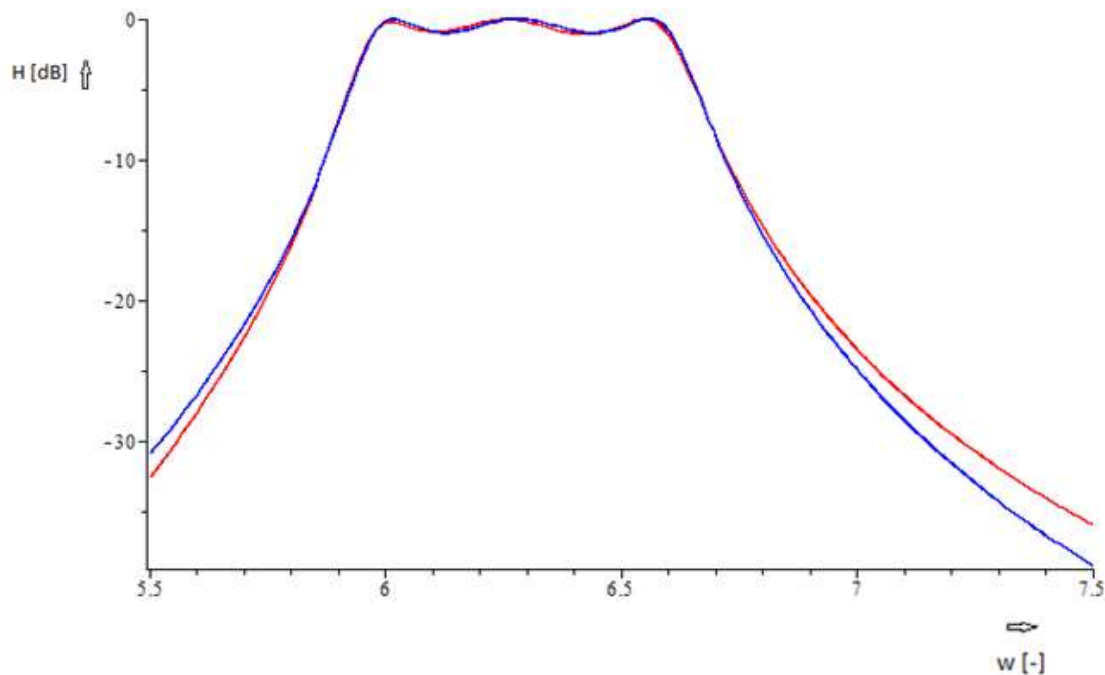
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00832400793515065$
 $C2 := 2.97072090456918$
 $L3 := 0.0308478229432338$
 $C4 := 0.599000325367562$
 $L5 := 0.00769482166250887$
 $C6 := 3.19066246446806$
 $CV1 := 0.117237512932468$
 $CV2 := 0.110260927562961$
 $RQ2 := 80399832547156$
 $RQ4 := 3987.5354527846$
 $RQ6 := 748.60160287421$

Obr. 5.2.12 - průběh konvergence při jakosti QC = 15000 s použitím metody Simplex:



Obr. 5.2.13 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QC = 15000 s použitím metody Simplex:

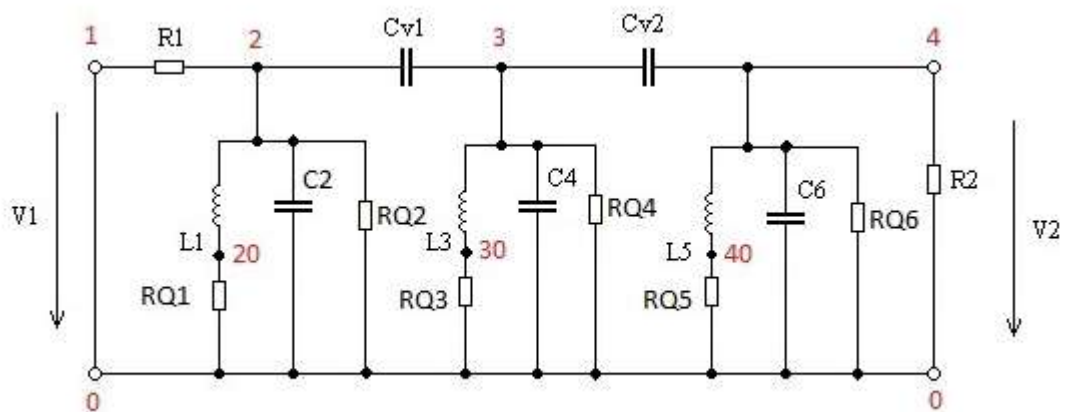


Simplex zkrátil dobu konvergence o 500 generací.

5.3) Zapojení zadané pásmové propusti se ztrátami v induktorech i v kapacitorech

Nyní otestuji zadanou pásmovou propust se ztrátovými odpory jak v induktorech, tak v kapacitorech. Testována bude jedna varianta. Tím je myšleno nastavení $Q_L = 400$ a $Q_C = 10000$.

Obr. 5.3.1 – schéma zadané pásmové propusti se ztrátami v induktorech a kapacitorech:



Zadání obvodu pomocí knihovny Syrup:

```

V 1 0
R1 1 2
L1 2 20
RQ1 20 0 (omN*L1)/QL
RQ2 2 0 QC/(omN*C2)
C2 2 0
CV1 2 3
L3 3 30
RQ3 30 0 (omN*L3)/QL
C4 3 0
RQ4 3 0 QC/(omN*C4)
CV2 3 4
L5 4 40
RQ5 40 0 (omN*L5)/QL
RQ6 4 0 QC/(omN*C6)
C6 4 0
R2 4 0

```

Výsledky pro QL = 400 a QC = 10000:

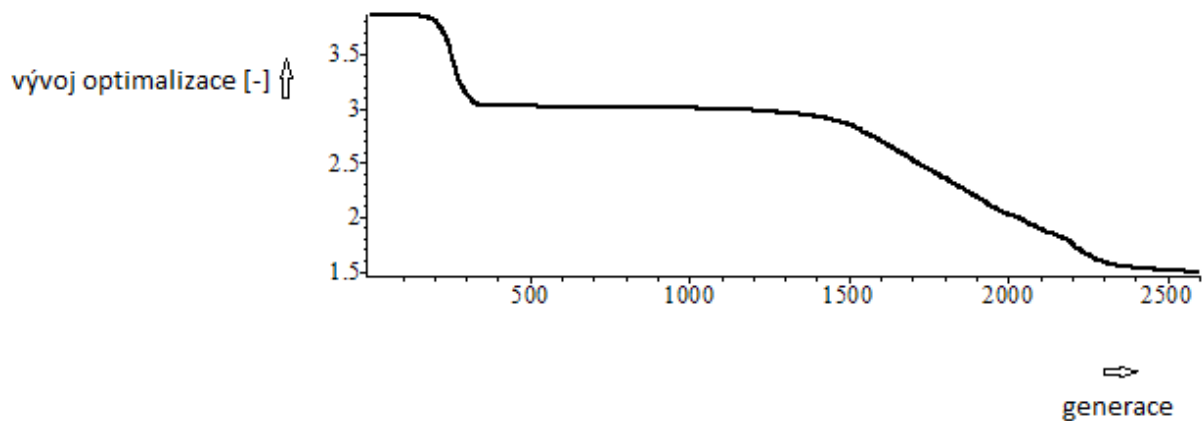
Prvky vektoru Xbest při hodnotě fit_best = 30.97774987:

$X_{best} = [0.008362483019, 2.961459743, 0.04430013672,$
 $0.3882426036, 0.008366070464, 2.957455945, 0.09425608119,$
 $0.09385682760]$

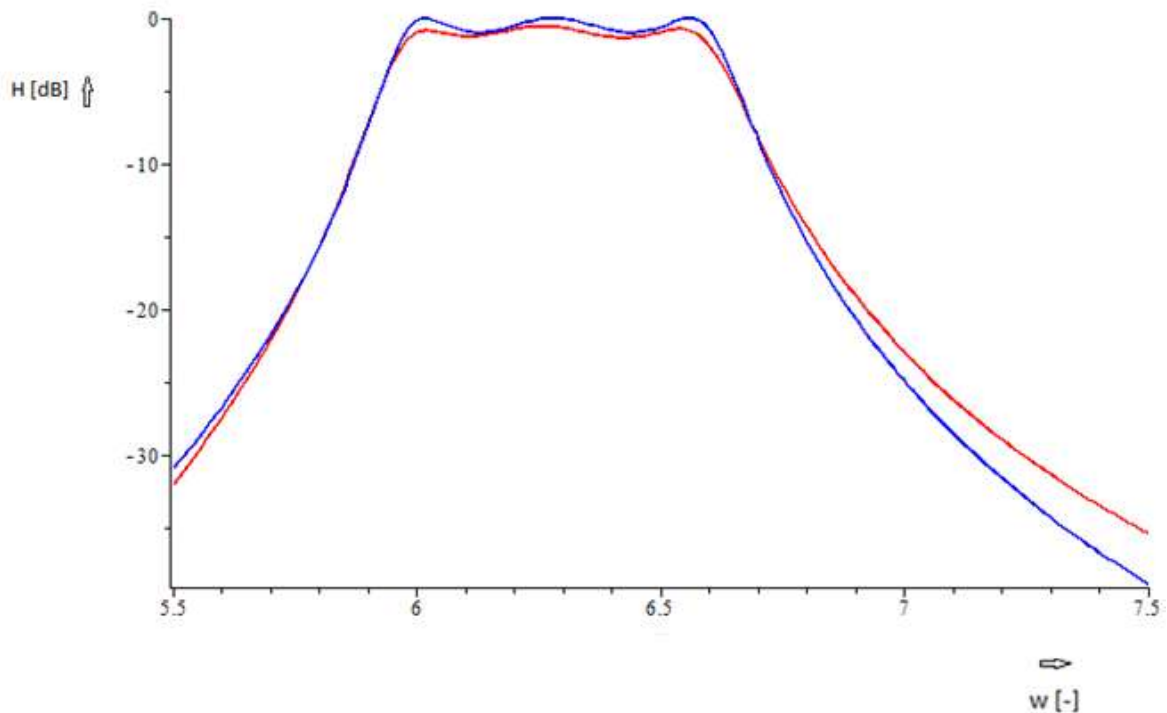
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.008362483019$
 $C2 := 2.961459743$
 $L3 := 0.04430013672$
 $C4 := 0.3882426036$
 $L5 := 0.008366070464$
 $C6 := 2.957455945$
 $CV1 := 0.09425608119$
 $CV2 := 0.09385682760$
 $RQ1 := 0.0001313575761$
 $RQ2 := 537.4205862$
 $RQ3 := 0.3545804693$
 $RQ4 := 4099.368324$
 $RQ5 := 0.0001314139276$
 $RQ6 := 538.1481451$

Obr. 5.3.2 - průběh konvergence při jakosti QL = 400 a QC = 10000:



Obr. 5.3.3 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti QL = 400 a QC = 10000:



Výsledky pro QL = 400 a QC = 10000 s použitím pomocné optimalizační metody Simplex:

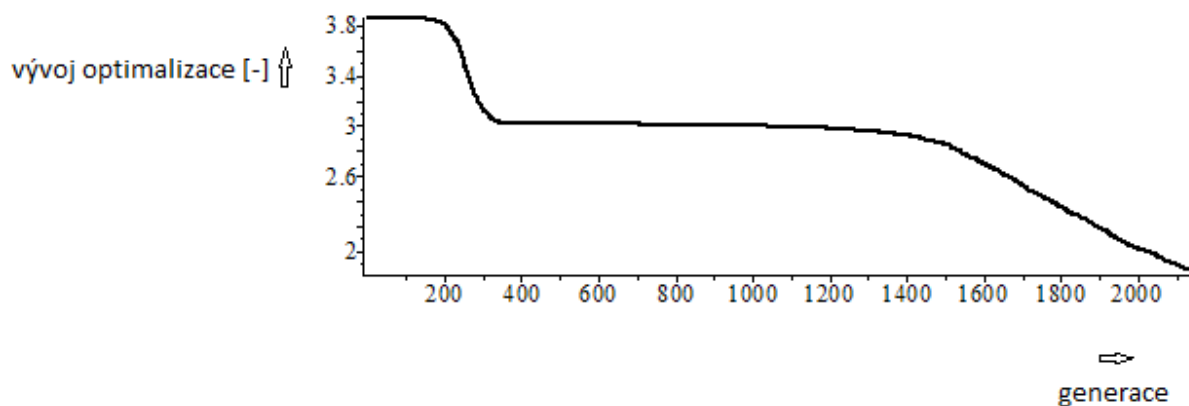
Prvky vektoru Xbest při hodnotě fit_best = 29.7954857401111504:

$X_{best} = [0.01059098067, 2.335204949, 0.07117045500,$
 $0.2295244754, 0.01047042049, 2.379426712, 0.06437050592,$
 $0.06543543882]$

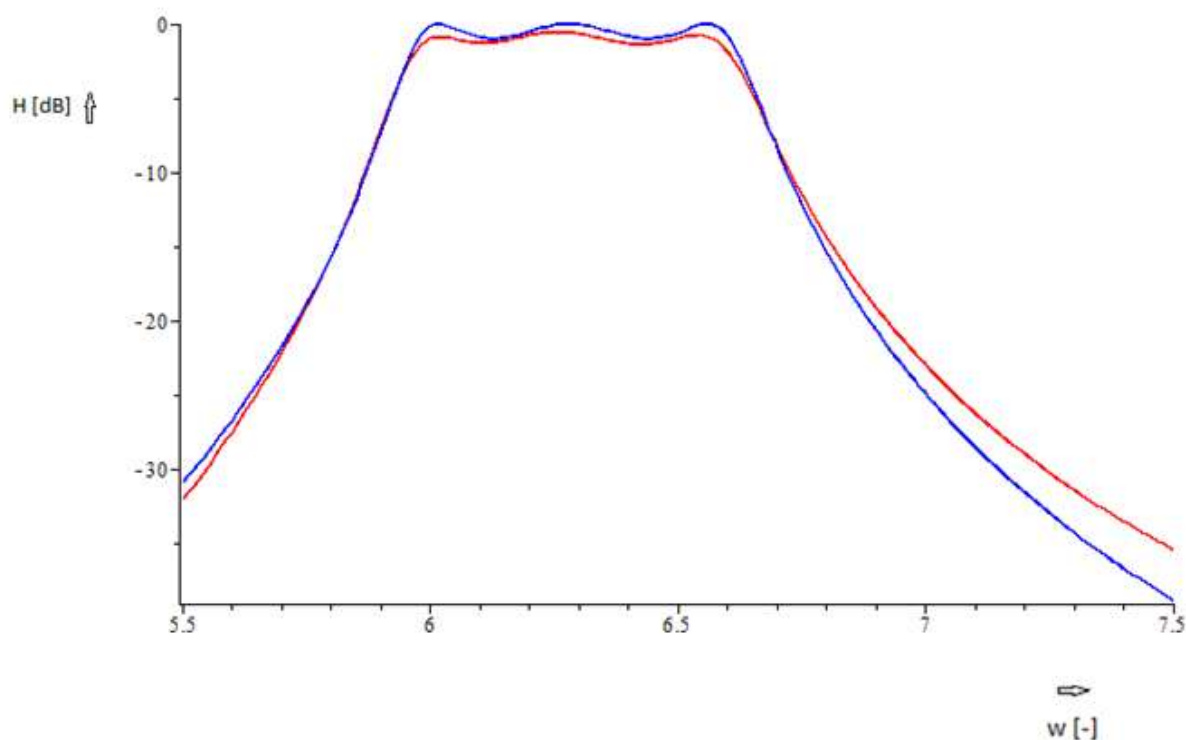
Po přiřazení hodnot k obvodovým prvkům:

$L1 := 0.00828054368154798$
 $C2 := 2.98055410014599$
 $L3 := 0.0371511337478368$
 $C4 := 0.480670265546227$
 $L5 := 0.00828054368153149$
 $C6 := 2.98060637994415$
 $CV1 := 0.103364291234910$
 $CV2 := 0.103373492349787$
 $RQ1 := 0.000130070476005386$
 $RQ2 := 533.977702643292$
 $RQ3 := 0.422812487409341$
 $RQ4 := 3311.10439958541$
 $RQ5 := 0.000130070476005127$
 $RQ6 := 533.968336681149$

Obr. 5.3.4 - průběh konvergence při jakosti $QL = 400$ a $QC = 10000$ s použitím metody Simplex:



Obr. 5.3.5 - porovnání zadané a vypočtené přenosové charakteristiky (modrá – zadaná; červená - vypočtená) při jakosti $QL = 400$ a $QC = 10000$ s použitím metody Simplex:



Použitím Simplexové metody jsem v tomto případě ušetřil 500 generací.

Přidáním ztrátových prvků k induktorům a kapacitorům se značně rozšířil programem Maple vypočtený přenos obvodu. Tím pádem se velmi prodloužilo i testování odchylek v přenosové charakteristice a tím i celá doba průběhu konvergence. Po konzultaci s vedoucím práce jsem převzal z literatury [2] pomocnou optimalizační metodu Simplex, kterou jsem aplikoval ve vhodnou chvíli na vektor X_{best} . Vhodnou chvílí je myšlen okamžik, kdy se program sice

blížil ke správnému výsledku, ale pomocí aplikace metody Simplex se cesta programu za výsledkem zkrátila o několik stovek generací. Přesný údaj o kolik se počet generací snížil, uvádím po každém použití Simplexu pod grafem porovnání přenosových charakteristik.

Přidáním ztrátových prvků se v LC filtru změní poměr ztrát k reaktanční složce (souhrn reaktancí induktorů a kapacitorů). Při změně tohoto poměru se zhoršuje průběh frekvenční charakteristiky a snižuje se násobná konstanta přenosu. Čím bude menší Q cívek a kondenzátorů, tím více bude zhoršení na frekvenční charakteristice vidět. Dobře to ukazují např. obrázky Obr. 5.1.7 na straně 43 pro $QL = 200$ a Obr. 5.1.15 na straně 49 pro $QL = 600$, kde je jasně vidět lepší průběh frekvenční charakteristiky pro případ s větším QL . Za pozitivní však lze považovat to, že algoritmus se snaží zachovat hodnotu zvlnění frekvenční charakteristiky v propustním pásmu a zčásti i šířku propustného pásma. Přitom se charakteristika “posouvá směrem dolů“, k vyšším hodnotám útlumu.

6) Závěr

Na začátku práce jsem nejprve prostudoval a poté ověřil funkčnost diferenčních evolučních algoritmů v syntéze pasivních LC filtrů typu pásmová propust. Upravil jsem diferenční evoluční algoritmus pro možnost aplikace na konkrétní zadání pro moji diplomovou práci. Přenosovou charakteristiku, kterou jsem zadával do programu Maple jsem získal pomocí programu Syntfil. Bližší informace jsou k nahlédnutí v příloze s názvem – Návrh v programu Syntfil.

Jedním ze dvou hlavních bodů diplomové práce je zhodnotit možnost vícekriteriální optimalizace návrhu z hlediska snížení rozptylu hodnot obvodových prvků. Zaměřil jsem se celkem na 5 možností řešení tohoto problému. První nápad na možné snížení rozptylu hodnot prvků byl v testování nastavení zakončovacích odporů, tak, aby byl rozptyl hodnot prvků co nejmenší. Testoval jsem za podmínky symetrického zakončení filtru, tedy při hodnotách zakončovacích odporů $R_1 = R_2$. Výsledky ukázaly, že hodnoty prvků se v případech $R_1 = R_2 = 20$ respektive 25 skutečně přiblížily.

Dalším způsobem, jak realizovat snížení rozptylu hodnot bylo rozšíření vyhodnocení účelové funkce o členy Kon_L a Kon_C . Význam těchto parametrů je v “ohlídání” rozptylu hodnot jednotlivých cívek a kondenzátorů. Bližší informace jsou v kapitole 4.2. Výsledky ukazují, že tato metoda funguje, ale hůře než metoda předchozí. Přesto ji však pokládám za funkční, protože oproti výsledkům z kapitoly 2 je kvalitnější z hlediska snížení rozptylu hodnot obvodových prvků. Doba konvergence byla u této metody velmi dlouhá. Po konzultaci s vedoucím práce jsem použil optimalizační metodu Simplex, kterou jsem získal z diplomové práce, která byla určena jako doporučená literatura [2] pro moji diplomovou práci. Po jejím použití jsem dostal téměř totožné výsledky jako po optimalizaci bez metody Simplex. Doba konvergence se však snížila o 800 generací.

Třetí způsob vyzkoušený pro snížení rozptylu prvků je nastavení hodnot všech tří indukčností na konstantní hodnotu $L_1 = L_2 = L_3 = 1$. Poupřavil jsem program k optimalizování hodnot kondenzátorů a zakončovacích odporů. Jelikož se vypočítané hodnoty zakončovacích odporů pohybovaly v jiném číselném rozmezí, tak jsem v programu vytvořil navíc pomocnou proměnnou a nový číselný interval. Tyto prvky jsem následně zakomponoval do bloku s

názvem výpočet testovacích vektorů. Při přidávání dalšího intervalu jsem se rozhodl vytvořit ještě jeden interval pro vazební kondenzátory. Výsledek ukazuje výrazné zlepšení v problematice snížení rozptylu prvků. Další kapitoly se zabývají modifikacemi tohoto návrhu.

V dalším pokusu využívám totožného principu jako v předchozí kapitole. Snažím se však řešit rozptyl hodnot zakončovacích odporů, které nejsou nyní pevně nastaveny, ale jsou optimalizovány. Podrobnější popis problematiky je v kapitole 4.4. Využívám zde vztahu mezi zakončovacími odpory R_1 , R_2 –

Rovnice 6.1 – vztah mezi zakončovacími odpory

$$kk = \frac{R_2}{R_1 + R_2}$$

Problematiku rozptylu hodnot zakončovacích odporů řeším ještě jednou metodou. Opět využívám vztahu mezi R_1 , R_2 . Nyní však trochu jiným způsobem než v předchozí kapitole. Rozšiřuji vyhodnocení účelové funkce o parametr Δ_{kk} , který vyhodnocuje rozptyl R_1 , R_2 podle nastaveného parametru kk_{opt} , který určuje poměr mezi R_1 , R_2 . Podrobněji je myšlenka vysvětlena v kapitole 4.5. Výsledek optimalizace u těchto dvou metod je u hodnot kondenzátorů obdobný jako u metody z kapitoly 4.3. Je zde však rozdíl v hodnotách zakončovacích odporů, které se pohybují podle nastavení konstanty kk , respektive kk_{opt} .

Druhým bodem diplomové práce byla kompenzace vlivu ztrát reálných induktorů a kapacitorů. Tuto problematiku jsem pro přehlednost pojal nejprve přidáním ztrátových odporů k induktorům. Poté jsem přidal ztrátové odpory pouze ke kapacitorům a až na konec jsem optimalizoval obvod se ztrátovými odpory v induktorech a kapacitorech zároveň. Po dohodě s vedoucím práce jsem testoval program pro několik různých jakostí QL a QC. Průběhy konvergencí byly v těchto případech vždy velmi zdlouhavé. V těchto momentech jsem využil pomocnou optimalizační metodu Simplex. Výsledky ukazují pro nízké hodnoty jakostí cívek a kondenzátorů zhoršení průběhu frekvenční charakteristiky, což potvrzuje teoretické předpoklady.

Použité zkratky

NDP – normovaná dolní propust

PP – pásmová propust

Q – jakost

QC – jakost kondenzátoru

QL – jakost cívky

Seznam rovnic

Rovnice 2.1 – obecná přenosová funkce zadané pásmové propusti.....	5
Rovnice 2.2 – přenosová funkce zadané pásmové propusti po doplnění koeficientů.....	5
Rovnice 2.3 – vztah pro výpočet přenosu.....	5
Rovnice 3.4.1 – příklad obecné přenosové funkce.....	12
Rovnice 3.4.2 – verze f, g testovacích vektorů.....	13
Rovnice 3.4.3 – vytvoření šumového vektoru.....	13
Rovnice 3.4.4 – vztah mezi zakončovacími odpory.....	15
Rovnice 4.2.1 – výpočet proměnné Kon_L.....	21
Rovnice 4.2.2 – výpočet proměnné Kon_C.....	21
Rovnice 4.2.3 – Verze f, g testovacího vektoru.....	21
Rovnice 4.4.1 – vztah mezi zakončovacími odpory.....	28
Rovnice 4.4.2 – vztah pro výpočet R2 pomocí parametru kk a hodnoty R1.....	28
Rovnice 4.4.3 – vztah mezi zakončovacími odpory.....	32
Rovnice 4.5.1 – vztah mezi zakončovacími odpory.....	33
Rovnice 5.1.1 – vztah pro výpočet RQ.....	39
Rovnice 5.2.1 – vztah pro výpočet RQ.....	51
Rovnice 6.1 – vztah mezi zakončovacími odpory.....	66

Seznam tabulek

Tab. 2.1 – hodnoty tolerančního schématu zadané pásmové propusti.....	5
--	---

Seznam obrázků

Obr. 1.1 - ukázka dolní propusti.....	1
Obr. 1.2 - ukázka horní propusti.....	1
Obr. 1.3 - ukázka pásmové propusti.....	1
Obr. 1.4 - ukázka pásmové zádrže.....	2
Obr. 2.1 - zadaná pásmová propust 6. řádu.....	4
Obr. 2.2 - toleranční schéma zadané pásmové propusti.....	4
Obr. 2.3 - průběh konvergence pro ověření funkce algoritmu na filtr typu PP	8
Obr. 2.4 - porovnání zadané a vypočtené přenosové charakteristiky pro ověření funkce algoritmu na filtr typu PP.....	8
Obr. 3.3.1 - princip diferenčního evolučního algoritmu.....	10
Obr. 3.4.1 - vývojový diagram programu z diplomové práce.....	11
Obr. 4.2.1 - průběh konvergence při použití pomocných proměnných Kon_L a Kon_C	22
Obr. 4.2.2 - porovnání zadané a vypočtené přenosové charakteristiky při použití pomocných proměnných Kon_L a Kon_C	23
Obr. 4.2.3 - průběh konvergence při použití pomocných proměnných Kon_L a Kon_C s použitím metody Simplex	24
Obr. 4.2.4 - porovnání zadané a vypočtené přenosové charakteristiky při použití pomocných proměnných Kon_L a Kon_C s použitím metody Simplex	24
Obr. 4.3.1 - průběh konvergence při nastavení shodných indukčností.....	27
Obr. 4.3.2 - porovnání zadané a vypočtené přenosové charakteristiky při nastavení shodných indukčností	27
Obr. 4.4.1 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk = 0,5).....	29
Obr. 4.4.2 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk = 0,5).....	29

Obr. 4.4.3 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk = 0,25).....	30
Obr. 4.4.4 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk = 0,25).....	31
Obr. 4.4.5 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk = 0,75).....	32
Obr. 4.4.6 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk = 0,75).....	32
Obr. 4.5.1 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,5).....	34
Obr. 4.5.2 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,5).....	35
Obr. 4.5.3 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,25).....	36
Obr. 4.5.4 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,25).....	36
Obr. 4.5.5 - průběh konvergence při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,75).....	37
Obr. 4.5.6 - porovnání zadané a vypočtené přenosové charakteristiky při hledání optimálních hodnot zakončovacích odporů (kk_opt = 0,75).....	37
Obr. 5.1.1 - schéma zadané pásmové propusti se ztrátami v induktorech	40
Obr. 5.1.2 - průběh konvergence při jakosti QL = 400.....	40
Obr. 5.1.3 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 400.....	41
Obr. 5.1.4 - průběh konvergence při jakosti QL = 400 s použitím metody Simplex	42
Obr. 5.1.5 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 400 s použitím metody Simplex	42
Obr. 5.1.6 - průběh konvergence při jakosti QL = 200.....	43

Obr. 5.1.7 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 200.....	43
Obr. 5.1.8 - průběh konvergence při jakosti QL = 200 s použitím metody Simplex	44
Obr. 5.1.9 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 200 s použitím metody Simplex	45
Obr. 5.1.10 - průběh konvergence při jakosti QL = 300	46
Obr. 5.1.11 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 300.....	46
Obr. 5.1.12 - průběh konvergence při jakosti QL = 300 s použitím metody Simplex	47
Obr. 5.1.13 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 300 s použitím metody Simplex	47
Obr. 5.1.14 - průběh konvergence při jakosti QL = 600.....	48
Obr. 5.1.15 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 600.....	49
Obr. 5.1.16 - průběh konvergence při jakosti QL = 600 s použitím metody Simplex	50
Obr. 5.1.17 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QL = 600 s použitím metody Simplex	50
Obr. 5.2.1 - schéma zadané pásmové propusti se ztrátami v kapacitorech.....	51
Obr. 5.2.2 - průběh konvergence při jakosti QC = 10000.....	52
Obr. 5.2.3 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QC = 10000.....	53
Obr. 5.2.4 - průběh konvergence při jakosti QC = 10000 s použitím metody Simplex.....	54
Obr. 5.2.5 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QC = 10000 s použitím metody Simplex	54
Obr. 5.2.6 - průběh konvergence při jakosti QC = 5000.....	55
Obr. 5.2.7 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti QC = 5000.....	55

Obr. 5.2.8 - průběh konvergence při jakosti $QC = 5000$ s použitím metody Simplex	56
Obr. 5.2.9 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti $QC = 5000$ s použitím metody Simplex.....	57
Obr. 5.2.10 - průběh konvergence při jakosti $QC = 15000$	58
Obr. 5.2.11 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti $QC = 15000$	58
Obr. 5.2.12 - průběh konvergence při jakosti $QC = 15000$ s použitím metody Simplex	59
Obr. 5.2.13 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti $QC = 15000$ s použitím metody Simplex	59
Obr. 5.3.1 - schéma zapojení zadané pásmové propusti se ztrátami v induktorech a kapacitorech.....	60
Obr. 5.3.2 - průběh konvergence při jakosti $QL = 400$ a $QC = 10000$	61
Obr. 5.3.3 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti $QL = 400$ a $QC = 10000$	62
Obr. 5.3.4 - průběh konvergence při jakosti $QL = 400$ a $QC = 10000$ s použitím metody Simplex.....	63
Obr. 5.3.5 - porovnání zadané a vypočtené přenosové charakteristiky při jakosti $QL = 400$ a $QC = 10000$ s použitím metody Simplex	63

Přílohy – Návrh v programu Syntfil (umístěno na konci desek)

Seznam použité literatury

- [1] Oplatková, Zelinka, et all.: Evoluční výpočetní techniky – principy a aplikace. Nakladatelství BEN, Praha, 2008. ISBN 80-7300-218-3
- [2] Coufal, J.: Systém pro návrh filtrů LC na bázi evolučních algoritmů. Diplomová práce, ČVUT FEL, K13131, Praha, 2011.
- [3] Zelinka Ivan. Umělá inteligence v problémech globální optimalizace. Nakladatelství BEN, Praha, 2002. ISBN 80-7300-069-5
- [4] Corne, D., Dorigo, M., Glover, F.: New ideas in Optimization. McGraw-Hill International, UK, 1999.
- [5] Bičák J., Laipert M., Vlček M. Lineární obvody a systémy. Nakladatelství ČVUT, Praha, 2007. ISBN 978-80-01-03649-5
- [6] Kobliha, O. Návrhový systém SYNTFIL [online].2004, ČVUT FEL Praha, Katedra teorie obvodů. Dostupné z: <http://syntfil.feld.cvut.cz/>.
- [7] Maplesoft, Maple, Online Help [online]. 2011, dostupné z: <http://www.maplesoft.com/support/help/Maple/view.aspx?path=Optimization/NLPSolve>.
- [8] Barchánek, J.: Optimized design of ARC filters using Differential Evolutionary Algorithms. Master thesis, ČVUT FEL, 2013.