INSERTION LOSS METHOD AND PARTICLE SWARM OPTIMIZATION ALGORITHM IN FILTER DESIGN

Metoda unesenoga gubitka i optimizacijski algoritam roja čestica u dizajniranju filtra

Ante Konjuh, mag. ing. el.
Maritime and technical school, Dubrovnik, Croatia
E-mail: ante.konjuh@gmail.com

Nikša Burum, Ph D
Department of electrical engineering and computing
University of Dubrovnik, Croatia
E-mail: niksa.burum@unidu.hr

Ivan Vilović, Ph D
Department of electrical engineering and computing
University of Dubrovnik
E-mail: ivan.vilovic@unidu.hr, Croatia

Summary
A filter design using insertion loss method (ILM) and Particle Swarm Optimization algorithm (PSO) is presented in the paper. ILM is generally a preferred technique for the filter design with an arbitrary number of elements. This method can be used to create the filter which has different types of frequency responses. However, this procedure is not practical for a large number of elements. PSO algorithm can be a useful procedure for calculating values in filters with a large number of elements. This paper presents the procedure which takes an advantage of ILM for filter design and PSO for determination an unknown component values in filters with large number of elements. This is suitable approach for filter design with an arbitrary cutoff frequency and shape of pass band.

Keywords: filters, insertion loss method, particle swarm optimization

INTRODUCTION / Uvod
Filters are two-port networks used to control the frequency response and they are present in almost every piece of electronic equipment (radios, television, analog-to-digital converters, computers etc.). There are four possible frequency responses: lowpass, highpass, bandpass and bandstop. Filters can be analog or digital. Furthermore, analog filters can be passive or...
active. Passive filters use only resistors, capacitors, and inductors. Very simple analog lowpass or highpass filters can be constructed from resistors and capacitors (RC) network. Both analog and digital filters can be considered as a “black box” (Fig 1.) with signals which are input on one side of the box and output on the other side.

The amplitude of the output signal voltage depends on the filter design and frequency of the input signal and can be found mathematically using transfer function. In other words, transfer function is frequency dependent equation relating the input and output voltage [1].

$$H(\omega) = \frac{V_i}{V_s} \quad (1)$$

Each filter is determined by cutoff frequency, stopband attenuation and frequency response, which can be described by transfer function. Transfer function is often presented in graphical form: as a curve that shows output signal loss (gain) versus frequency. As transfer function becomes more complex determination of the filters element value becomes foreground task in filter design. The ideal transfer characteristic cannot be obtained and the goal of the filter design is to approximate ideal transfer function within an acceptable tolerance. Generally preferred technique for the filter design and analysis is insertion loss method.

**INSERTION LOSS METHOD / Metoda unesenoga gubitka**

The insertion loss method begins with the design of a low-pass filter prototype that is normalized in terms of impedance and cutoff frequency. Normalized design of low-pass filter is then transformed to the filter with desired impedance level, frequency response and cutoff frequency. In this method a filter response is defined by its insertion loss or power loss ratio, $P_{LR}$ [2]:

$$P_{LR} = \frac{P_{inc}}{P_{load}} = \frac{1}{1 - |\Gamma(\omega)|^2} \quad (2)$$

where $\Gamma(\omega)$ is the reflection coefficient at the input point of the filter. Since the input reflection coefficient is equal:

$$\Gamma = \frac{Z_{in} - 1}{Z_{in} + 1} \quad (3)$$

power loss ratio can be written as:

$$P_{LR} = -\frac{|Z_{in} + 1|^2}{2(Z_{in} + Z_{in}^*)} \quad (4)$$

The power loss ratio for a maximally flat low-pass filter is [3]:

$$P_{LR} = 1 + k^2 \left( \frac{\omega}{\omega_c} \right)^{2N} \quad (5)$$

where the passband is the region from $\omega=0$ to $2N$ the cutoff value $\omega_c$, $k^2$ is the passband tolerance and $N$ is number of filter elements (Fig. 2.) For $\omega > \omega_c$ the power loss ratio increases to the power $2N$, which is related to the number of filter elements. In addition, for sharpest $P_{LR}$ characteristic over cutoff frequency larger number of filter elements are needed.

For equal-ripple low-pass filter prototype (i.e. for Chebyshev filter) the power loss ratio is equal:

$$P_{LR} = 1 + k^2 T_N^2 \left( \frac{\omega}{\omega_c} \right) \quad (6)$$

where $T_N(x)$ is a Chebyshev polynomial of order $N$. This will result with a sharper cutoff characteristic, although the passband response will have ripples $(1+k^2)$ of amplitude. Using (5) or (6) we are able to calculate required characteristic of filter and then calculate unknown values of element. However, for large number of $N$ analitical procedure for determining filters elements is very laborious.

**PARTICLE SWARM OPTIMIZATION ALGORITHM / Optimizacijski algoritam roja čestica**

The PSO algorithm was introduced by James Kennedy and Russel C. Eberrhart in 1995 [4]. The authors were inspired by swarm of bees or flock of birds in search
for food. The main goal for the swarm of bees is to find the place with the greatest concentration of flowers with the most nectar. This is equal to global optimum solution. Compared with genetic algorithm that is based on Darwin’s theory of natural selection and competition between chromosomes, the model of the swarm intelligent behavior in the process of optimization is based on Newton’s mechanical principles. Hence, it provides concise formulation and simple application in different optimization problems.

The PSO, although originally invented for research on simulating the movement of the swarm in 2-dimensional space, as an optimization method can be applied in n-dimensional space (see e.g. [5]). The particles are defined with its own position \( x \), velocity \( v \), and personal best result so far \( pbest \). The key element of the entire optimization is the changing of particle’s velocity. For the \( k+1 \) particle movement, the \( i \)-th coordinate component of velocity of the \( i \)-th particle, we can write

\[
v_{ij}^{k+1} = c_0 v_{ij}^{k} + c_1 rand_1 (pbest_{ij} - x_{ij}^{k}) + c_2 rand_2 (gbest_{ij} - x_{ij}^{k})
\]

(7)

In the above equation \( j = 1,2,...,m \), where \( m \) is the size of the swarm; \( j = 1,2,...,n \), where \( n \) is dimension of the space; \( c_0, c_1, \) and \( c_2 \) are positive constants that scale the old velocity and increase new velocity toward \( pbest \) (local best result) or \( gbest \) (global best result), respectively. \( rand_1 \) and \( rand_2 \) represent random numbers that are uniformly distributed in the interval \([0,1]\). The parameter \( c_0 \) is called “inertial weight” and it determines if the particle will stay on its current trajectory or if it will be strongly pulled toward \( pbest \) or \( gbest \). Its value is between 0 and 1. The new particle location is given by

\[
x_{ij}^{k+1} = x_{ij}^{k} + \Delta t_{ij}^{k+1}
\]

(8)

The new velocity is applied after some time-step \( \Delta t \), which is usually one. In other words, particles exchange information about results they obtained, so they know the best of all results so far. According to this information they accelerate in the direction of the global best result \( gbest \) and at the same time toward its own best result \( pbest \), so their trajectory is altering between these two goals depending on which direction prevails.

A proper selection of parameter values is very important to obtain qualitative result. Various authors have proposed different number of particles, maximum velocity, inertial weights and other constants. After running PSO algorithm with different parameters we got the best result when inertial weight \( c_0 \) was changed linearly from 0.9 to 0.2 during the run of algorithm.

In this way, particles at the beginning are less pulled toward \( pbest \) and \( gbest \), but after a number of iterations they are more rapidly pulled toward these values, which are illustrated in Fig. 3 for three different values of \( c_0 \). Higher value of \( c_0 \) means faster move toward \( gbest \), faster convergence, but less accuracy.

For the constants \( c_i \) and \( c_p \), value of 2 is used, but in our case where very little change in coordinates may result in great change in cost function value, the time step needs to be chosen carefully. Considering chosen values for \( c_0, c_1, c_2 \) and examining equations (7) and (8), we have chosen 0.4 for the time step value.

We carefully selected population size among large populations with a lot cost function evaluations and longer computation time, and smaller populations that give the result much faster. It was determined by many parametric studies [5] that relatively small populations can sufficiently explore the space under consideration, so population of 30 particles is used in our algorithm. Among the suggested boundary conditions, introduced by various authors, we have selected so-called “reflecting walls” to avoid moving the particle out from the given space [5].

**DETERMINING VALUES OF THE VARIOUS FILTER’S COMPONENTS USING PSO ALGORITHM / Određivanje raznih komponenata filtra s pomoću PSO algoritma**

PSO can be applied in the filter design as powerful optimization method to find out value of elements. In our algorithm, for N element filter, particle has following characteristics:

![Fig. 3 Cost function for different inertial weights. Slika 3. Ovisnost ciljne funkcije o inertnim težinama](image-url)
Agent{
  g_value : array[1..N]
  g_velocity: array[1..N]
  R_value
  R_velocity
  pbest_g_value : array[1..N]
  pbest_R_value
  pbest_fitness}

where $g_{value}$ and $R_{value}$ represent the value of reactive elements and impedance of the load, respectively. The $g_{velocity}$ and $R_{velocity}$ represent the velocity of these filter elements. $pbest_{x}$ are the values of elements for the best fitness function. Fitness function is difference between the desired $P_{LR}$ and $P_{LR}$ obtained using assumed values of various elements.

For filter with arbitrary number of elements (Fig. 4), input impedance is calculated using following equations:

\[
Z_{in} = \sum_{k=N}^{} Z_{in(k)}
\]

(9)

where:

\[
Z_{in(k)} = j\omega g_k + Z_{in(k+1)} \quad \text{for odd } k
\]

(10a)

\[
Z_{in(k)} = \frac{Z_{in(k+1)}}{1 + j\omega g_k Z_{in(k+1)}} \quad \text{for even } k
\]

(10b)

\[
Z_{in(N+1)} = R_T
\]

(10c)

Power loss ratio is calculated from (4) and then the result is evaluated using the following fitness function:

\[
fitness = \sum_{k=0}^{M} \left| \frac{P_{LRd}}{P_{LRc}} \right|
\]

(11)

where $M$ is maximal number of test points, as $P_{LRd}$ and $P_{LRc}$ represent the desired and obtained power loss ratio, respectively. The PSO algorithm searches for the minimum of the fitness function.

For filter with arbitrary number of elements (Fig. 4), input impedance is calculated using following equations:

\[
Z_{in} = \sum_{k=N}^{} Z_{in(k)}
\]

(9)

where:

\[
Z_{in(k)} = j\omega g_k + Z_{in(k+1)} \quad \text{for odd } k
\]

(10a)

\[
Z_{in(k)} = \frac{Z_{in(k+1)}}{1 + j\omega g_k Z_{in(k+1)}} \quad \text{for even } k
\]

(10b)

\[
Z_{in(N+1)} = R_T
\]

(10c)

Power loss ratio is calculated from (4) and then the result is evaluated using the following fitness function:

\[
fitness = \sum_{k=0}^{M} \left| \frac{P_{LRd}}{P_{LRc}} \right|
\]

(11)

where $M$ is maximal number of test points, as $P_{LRd}$ and $P_{LRc}$ represent the desired and obtained power loss ratio, respectively. The PSO algorithm searches for the minimum of the fitness function.

RESULTS / Rezultati

The comparison between analytically calculated $P_{LR}$ and $P_{LR}$ obtained by the PSO method will be illustrated in following examples. First, we consider $P_{LR}$ for maximally flat low-pass filter prototype (see Fig. 5 to Fig. 7) for different number of elements. It can be seen that there is a good agreement between $P_{LR}$ calculated analytically (e.g. 5) and by the PSO method.

As a next step, we investigated equal-ripple low-pass filter prototype and calculated $P_{LR}$ with values of various filter’s components computed by PSO algorithm (see Fig. 8 to Fig. 10). In all of these cases we obtained
a good agreement between ideal characteristic and obtained by the PSO method for $P_{LR}$.

The PSO algorithm was run with 200 agents. The inertial weight was changed from 0.7 to 0.5 during algorithm run. The parameters $c_1$ and $c_2$ were set to 1.5 and 2, respectively. The results are presented in the Table 1 for the flat filter with 7 elements. It is obvious good agreement between desired and obtained results. For this case the mean square deviation ($mse$) was $9.23 \cdot 10^{-4}$. The mse is calculated for values of the $\omega/\omega_c$ ratio between 0 and 1.1. For the higher values of the $\omega/\omega_c$ ratio deviation between ideal and PSO obtained results is higher. The impact on the filter functioning is negligible because the value of power loss ratio is enough high for those values of the $\omega/\omega_c$ ratio.

**Table 1.** The comparison of desired and PSO obtained results of flat low pass filter with 7 elements (Fig. 5)

<table>
<thead>
<tr>
<th></th>
<th>$g_1$</th>
<th>$g_2$</th>
<th>$g_3$</th>
<th>$g_4$</th>
<th>$g_5$</th>
<th>$g_6$</th>
<th>$g_7$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired</td>
<td>0.4450</td>
<td>1.2470</td>
<td>1.8019</td>
<td>2.0000</td>
<td>1.8019</td>
<td>1.2470</td>
<td>0.4450</td>
<td>1</td>
</tr>
<tr>
<td>PSO</td>
<td>0.4274</td>
<td>1.2207</td>
<td>1.8457</td>
<td>1.9774</td>
<td>1.8254</td>
<td>1.2435</td>
<td>0.4971</td>
<td>0.9980</td>
</tr>
</tbody>
</table>

**CONCLUSION / Zaključak**

The paper presents an approach for filter design that is based on insertion loss method and particle swarm optimization algorithm. Insertion loss method is generally preferred technique for designing a filter with an arbitrary number of elements and can be used to create a filter with frequency response. However, this procedure is not practical for filter with large number of elements. In that case numerical techniques are preferred. Particle Swarm Optimization (PSO) algorithm represents an evolutionary computation technique that was used for determining the values for various filter’s components.
A computer program was developed and tested in designing maximally flat and equally-rippled filter prototypes. It was shown that the developed program gives good results and could be commercially used for filter design. It should be emphasized that proposed designing procedure is suitable for filters with large number of elements when a sharp cutoff characteristic is required and when the filter’s frequency response is given in a graphical way.

ACKNOWLEDGEMENT / Zahvala

This material is based upon work supported by Croatian Ministry of Science, Education and Sports, under Project No. 036-0361566-1570 and Project No. 275-0361566-3136.

REFERENCES / Literatura
