Abstract—This paper provides an effective new method for design of analog all-pass filters with equal-ripple group delay frequency response. The method is based on combination of the proposed analytical polynomials with a numerical solution by using the Differential Evolution (DE) algorithm. The method is illustrated on practical example.

I. INTRODUCTION

The genetic and evolutionary algorithms were found out as a powerful tool in design and optimization of electrical circuits and systems. Excellent results of their utilization have been obtained e.g. in a transfer function approximation problem solution in many cases of analog or digital filters design. For example, papers [3, 4] have dealt with the approximation of the analog as well as digital filter transfer function with concurrent requirements for magnitude and group delay frequency response by using the Differential Evolution (DE) algorithms.

Generally, the design of equal-ripple group delay of analog all-pass filters represents an exigent mathematical problem. The papers [6, 7] and the thesis [8] have presented different methods for design of analog all-pass filters with equal-ripple group delay frequency response. The methods are mostly based on principle, that nonlinear equation system (describing filter group delay extremes) is generated and this one is solved by classical numerical method, for instance Remez algorithm [7, 8] or another iterative technique [6]. Drawback of the methods is that they are using relatively complicated mathematical background.

Here, we have proposed new interesting method for design of analog all-pass filter with equal-ripple group delay frequency response. The presented design procedure was evolved from the method published in the paper [1], where a design of digital all-pass filters with equal-ripple group delay frequency response is described.

Our presented method uses novel transformation process, where new kind of polynomials derived in digital domain is used as a starting point of the presented procedure and the Differential Evolution (DE) algorithm is used as a numerical solver in analog domain. Design flow will be described on solving the practical example.

The whole design procedure has been programmed in the MATLAB environment.

II. APPROXIMATION POLYNOMIALS

The principle of the method is based on newly proposed type of polynomials \( V_n \), which satisfy the differential equation published in [2]. The differential equation derived in the \( w \)-domain for digital all-pass filters is in the form:

\[
\sqrt{(1-w)\cdot(1-w-\cos(\hat{\omega}c))} \cdot \frac{dF^2}{dw} = n \cdot F \cdot (1 + F^2),
\]

where

\[
F^2 = \frac{1 - (-1)^n \cdot V_n}{1 + (-1)^n \cdot V_n}, \quad w = \cos(\hat{\omega}).
\]

The variable \( \hat{\omega} \) means normalized “digital” frequency:

\[
\hat{\omega} = \omega \cdot T.
\]

Symbol \( T \) labels the sampling interval.

The found polynomials \( V_n \) are defined for each order \( n \) of the all-pass filter in the forms:

\[
V_1 = w - \kappa, \quad V_2 = 2 \cdot (w - \kappa)^2 - (\kappa - 1)^2,
\]

\[
V_n = 2 \cdot (w - \kappa) \cdot V_{n-1} - (\kappa - 1)^2 \cdot V_{n-2},
\]

where \( n > 2 \), variables \( \kappa \) is defined by the formula (6):

\[
\kappa = \frac{1 + \cos(\hat{\omega}c)}{2}.
\]

After evaluation of the polynomial \( V_n \), we can convert this polynomial to the linear form:

\[
V_n = \sum_{i=0}^{n} a_i \cdot w^i = a_n.
\]

Now, the polynomial will be normalized by the term:

\[
b_n = \frac{a_n}{\sum_{i=0}^{n} a_i}.
\]

Afterwards, this final polynomial is included to the function (9), which only “represents” group delay frequency response of the appropriate \( n \)-th order all-pass filter. This function is not real digital all-pass filter group delay function. The function is defined by the term:
The Differential Evolution (DE) algorithm is a parallel direct search method which uses floating-point number representation to find continuous parameters. This algorithm is well-known and popular for solving technical applications. Its principle can be found in [1, 3, 4, 5]. Several variants of DE exist. But, we have found out that the variant denoted like DE/best/1/bin appears as the most efficient version for the solved task. Therefore, we have applied this version in practical examples of analog all-pass filters design.

### III. THE DIFFERENTIAL EVOLUTION ALGORITHM (DE)

Evolutionary algorithms are global minimization algorithms which simulate an evolutioonal process in the nature. We can describe general structure of an evolution program as follows:

```
begin
  t ← 0
  initialize P(t)
  evaluate P(t)
  while termination-condition do
    t ← t+1
    select P(t) from P(t-1)
    alter P(t)
    evaluate P(t)
  end
end
```

These algorithms are probabilistic algorithms which store a number of possible problem solution representations in so called a “population” matrix \( P(t) = \{x'_1, \ldots, x'_n\} \) for iteration \( t \). Each solution \( x'_i \) is evaluated to give some measure of its “fitness”. Then, a new population (iteration \( t+1 \)) is formed by selecting the more fit individuals (select step). Some members of the new population undergo transformations (alter step) by means of “genetic” operators to form new solutions. There are several types of transformations, for example: mutation, crossovering. After some number of generations the program converges – it is hoped that the best individual represents a near-optimum solution.

The Differential Evolution algorithm belongs to the evolutionary algorithms group. The mentioned algorithm was developed by K. Price and R. Storn and for the first time was presented in 1995. In the conference First International Contest of Evolutionary Computation (1stICEO) held in Nagoya in May 1996, this algorithm turned out to be the best evolution type of algorithm for solving the real-valued functions.

### IV. SOLUTION OF THE PRACTICAL EXAMPLE

The design procedure will be described on solving of the practical example. The goal is to design the 10th order analog all-pass filter with equal-ripple form of its group delay with ripple \( \varepsilon = 0.2 \) on the normalized frequency interval \([0, \Omega_w=1]\).

General transfer function of the even order analog all-pass filter can be defined in the form:

\[
A(s) = \frac{\prod_{i=1}^{\frac{n}{2}} (s - a_i + j \cdot \beta_i) \cdot (s - a_i - j \cdot \beta_i)}{\prod_{i=1}^{\frac{n}{2}} (s + a_i + j \cdot \beta_i) \cdot (s + a_i - j \cdot \beta_i)},
\]

where \( n \) is order of the all-pass filter. This form is advantageous as a matter of the fact that we can simply ensure design of the stable all-pass filter as will be shown later.

Variable \( s \) is a complex variable meaning normalized “analog” frequency defined as:

\[
s = \Sigma + j \cdot \Omega.
\]

The group delay frequency response of the analog all-pass filter is calculated by the formula:

\[
\tau(\Omega) = -\text{Re} \left[ \frac{A(s)}{A'(s)} \right]_{s=j \cdot \Omega},
\]

where \( A(s) \) denotes the transfer function of the all-pass filter and \( A'(s) \) denotes derivative of \( A(s) \) by the variable \( s \).

We found out that for a design of analog all-pass filters with different values of the group delay ripple \( \varepsilon \) on the normalized frequency interval \([0, \Omega_w=1]\) (there is no problem to recalculate parameters of the all-pass filter to have equal-ripple group delay for any frequency interval) is enough to use only one polynomial family (4), (5) generated.
by the parameters: \( n_{all-pass} = 0.1 \) and \( \hat{\omega}_k = 0.8\pi \). It simplifies solution of the problem. Therefore, in the example, we will calculate the function (9) for \( n_{all-pass} = 0.1 \) and \( \hat{\omega}_k = 0.8\pi \). Then, the function (9) is sampled at \( M \) equidistant points and saved as a vector in MATLAB. It means, now the sampled values are independent of the type of frequency axis ("analog" or "digital") and can be located on an analog interval. In this way, the digital interval \([0, \hat{\omega}]\) is transformed on the analog interval \([0, \Omega\) ] = 1].

The values of the function (9) correspond with new even "analog" frequency points. However, the form of the graph of the function (9) is the same. Such an example is plotted in Fig. 1.

Now, the so-called error function will be created for the extremes of the sampled function (9):

\[
e_k = \begin{cases} 
   r(0) + (-1)^k \cdot \varepsilon - \max[r(\Omega_k)] & \text{for even } k \\
   r(0) + (-1)^k \cdot \varepsilon - \min[r(\Omega_k)] & \text{for odd } k 
\end{cases}
\]

\[
\Omega_k = (p_k - \Delta N_1k, p_k + \Delta N_2k),
\]

where \( k = 1, \ldots, n + 1 \), \( p_k \) denotes position (sample) of the appropriate \( k \)th extreme of the sampled function (9), \( \varepsilon \) is a constant value of the group delay, which is approximated in Chebyshev sense. Symbol \( \Omega \) denotes interval in which the appropriate extreme will be searched, \( \Delta N_1k, \Delta N_2k \) are boundaries of the interval \( \Omega \). Mostly, optimum value is \( \Delta N_1k = 40, \Delta N_2k = 70 \) for sampling of equal-ripple at \( M = 1000 \) points. Only \( \Delta N_1k = 0 \) and \( \Delta N_2k = 0 \). Values of \( \Delta N_1k, \Delta N_2k \) have to be re-counted for sampling by using different number of sampled points.

Important note: in case of odd order all-pass filter design, parameter \( k = 0, \ldots, n \). Thus, \( \Delta N_1k = 0 \) and \( \Delta N_2k = 0 \).

As we have mentioned above, DE algorithm is used as numerical solver. Therefore, an objective function will be defined. DE algorithm searches complex poles of the all-pass filter transfer function and constant \( \tau(0) \) to be obtained minimum of the objective function. Thus, the minimization process of the objective function leads to the design of the all-pass filter.

The mathematical formulation of the objective function for the all-pass filter is defined by:

\[
\begin{align*}
F(x) &= \sum_{k=1}^{n+1} \left[ \left( \varepsilon + \max(r(\Omega_k)) - \min(r(\Omega_k)) \right) \right] + \max[r(\Omega)] - \min[r(\Omega)].
\end{align*}
\]

Vector \( x = [x_1, \ldots, x_{11}] \) is mapped:

\[
x_1 = \alpha_1, x_2 = \alpha_2, x_3 = \alpha_3, x_4 = \alpha_4, x_5 = \alpha_5, x_6 = \beta_1, x_7 = \beta_2, x_8 = \beta_3, x_9 = \beta_4, x_{10} = \beta_5, x_{11} = \tau(0).
\]

\[
P = \min_{x \in F} \left[ P_{\min}(0) + \max_{x \in F} \left[ P_{\max}(0) \right] \right] - \min_{x \in F} \left[ P_{\min}(0) \right].
\]

\[
P = \begin{cases} 
   20000 - 100 \cdot x_i & \text{if } x_i < 0 \\
   0 & \text{otherwise}
\end{cases}
\]

\[
\min_{x \in F} \left[ P_{\min}(0) \right] = 0.1. \]

Symbol \( N \) labels number of the unknown searched variables and \( x_1 \) are elements of the vector \( x \) (real and imaginary parts of all-pass filter transfer function complex poles) and \( x_6 \) is constant \( \tau(0) \).

The penalty function \( P_1 \) is included into the objective function to guarantee stability of the analog all-pass filter. As is known, the real parts of the complex poles must be located in the left part of the complex plane in the variable \( s \). The penalty function \( P_2 \) ensures positive values of initial group delay \( \tau(0) \). Here we have used penalty functions created according to the principle published in [4].

The key factor of the design procedure is good choice of the value \( \tau_{min}(0) \). In general, the \( n \)th order all-pass filter is defined by:

\[
\text{TABLE I} \quad \text{VALUES OF THE INITIAL GROUP DELAY}
\]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \tau_{min}(0) ) for appropriate order ( n ) and ripple ( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \tau_{min}(0) = 9.86 )</td>
</tr>
<tr>
<td>2</td>
<td>( \tau_{min}(0) = 19.63 )</td>
</tr>
<tr>
<td>3</td>
<td>( \tau_{min}(0) = 25.28 )</td>
</tr>
<tr>
<td>4</td>
<td>( \tau_{min}(0) = 20.13 )</td>
</tr>
<tr>
<td>5</td>
<td>( \tau_{min}(0) = 21.77 )</td>
</tr>
<tr>
<td>6</td>
<td>( \tau_{min}(0) = 23.31 )</td>
</tr>
<tr>
<td>7</td>
<td>( \tau_{min}(0) = 24.40 )</td>
</tr>
<tr>
<td>8</td>
<td>( \tau_{min}(0) = 25.28 )</td>
</tr>
</tbody>
</table>
The vector $x_{opt}$ for which the objective function $F(x)$ has minimum, is the wanted solution of the analog all-pass filter design problem.

A. Results of the Analog All-pass Filter Design

The design task stated above was solved using the DE algorithm (version DE/best/1/bin) initial settings: NP=150, CR=0.9, F=0.9, variables range: $x_1 - x_{10} \in (0...2)$, $x_{11} \in (10...40)$, $\Delta N_1 = 40$, $\Delta N_2 = 70$. The algorithm found the parameters of the all-pass filter arranged in the Tab. II.

<table>
<thead>
<tr>
<th>PARAMETERS OF THE PROPOSED ALL-PASS FILTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1 = 0.20785349613198$</td>
</tr>
<tr>
<td>$\beta_1 = 1.00950461318312$</td>
</tr>
<tr>
<td>$\alpha_2 = 0.35448524212411$</td>
</tr>
<tr>
<td>$\beta_2 = 0.80711110677064$</td>
</tr>
<tr>
<td>$\alpha_3 = 0.11860437154075$</td>
</tr>
<tr>
<td>$\beta_3 = 0.20206399490238$</td>
</tr>
<tr>
<td>$\alpha_4 = 0.14505884019310$</td>
</tr>
<tr>
<td>$\beta_4 = 0.58579691206979$</td>
</tr>
</tbody>
</table>

The obtained transfer function was simulated in the MATLAB environment to be achieved resultant group delay frequency response. This response is shown in the Fig. 2.

![Fig. 2. Group delay frequency response of the 10th order analog all-pass filter with $\epsilon=0.2$.](image)

V. CONCLUSION

Here, we have presented new method for design of analog all-pass filter with equal-ripple form of the group delay frequency response. A usage of the evolutionary algorithm represents unconventional approach at solving of the presented problem.

This way, we can propose $n^{th}$ order analog all-pass filter with equal-ripple form of the group delay frequency response in the various interval $[0, \Omega_3]$ and with the variable ripple $\epsilon$.

Moreover, it is obvious, that we are able to include other optimization criteria to the objective function, for example non-ideal parameters of operational amplifiers, resistors and capacitors which can be used for practical implementation of a proposed all-pass filter. Actually, we can take into account real properties of bipolar or MOS transistors (it depends on used IC technology) to design the all-pass filter on a transistor level as to realize the all-pass filter by an integrated circuit as well.

REFERENCES


2007 IEEE Congress on Evolutionary Computation (CEC 2007)