# Flat Coil Optimizer in the Meaning to Coil Optimization

Lukas Kouril, Martin Pospisilik, Milan Adamek, Roman Jasek

***Abstract*****—** Planar inductors made on a printed circuit board are rather commonly employed today and there are various software applications to help the designers with their proposal. In this paper the authors describe the application called Flat Coil Optimizer developed by them. This application serves for designing proper geometrical dimensions of a single layer spiral coil made on the printed circuit board in order it achieved the desired inductance while its serial resistance is as low as possible. In order to process such optimization, two different approaches have been utilized. Firstly, the optimization can be processed analytically and secondly, Differential evolution can be applied as well. This feature allows the user to decide which method he/she wants to apply and in case of doubts the user can verify the results by employing the other method.

***Keywords*****—**Artificial intelligence, analytical solution, Differential evolution, coil optimization.

## I. Introduction to Flat Coils

CURRENTLY it is rather common to meet flat coils (also called planar inductors) made directly on the printed circuit board (PCB). There are several models and approximations estimating the inductance of these coils. These models usually show some uncertainty as the inductance expression is quite complex, involving the self-inductance of the conductor, mutual inductance among particular current-turns and the influence of the coil surroundings. The list of methods developed in the past is rather wealthy; let us mention for example the Expanded Grover method, Bryan method, Terman method, all mentioned in [7] for rectangle-shaped inductors, or Wheeler, Gleason or Olivei approximations for spiral-shaped coils. The authors of this paper decided to employ the current sheet approximation that is suitable to be implemented in software applications including the one described in this paper. The current sheet method of planar inductor inductance expression can be obtained by approximating the sides of the spirals by symmetrical current sheets of equivalent current densities [8]. The detailed description of this method can be found in [10].

### *A. Flat Coil Description*

A single-layer spiral flat coil is a geometrical spiral-shaped formation made of a copper film on the printed circuit board. This copper spiral is electrically conductive shoving an electrical resistance for the direct current. For the alternating current it shows, except the electrical resistance, also an inductance, which describes the capability of producing a magnetic field around the inductor. From the electrical point of view the inductance acts as complex impedance being proportional to the frequency of the alternating current. Whilst the ideal inductor shows only the inductance and its electrical resistance is negligible, real inductors show also the ineligible resistance that must be minimized as effectively as possible. This phenomenon occurs especially at larger coils or the coils intended for operation under heavy currents. Primarily, such a large spiral coil inductor is intended to be employed in a metal detector, but this method is also suitable for designing smaller loop antennas, Tesla coils, proximity sensors etc. The methods of optimization, described in this paper, are aimed to design the flat coil in order its inductance was as required and its resistance was as small as possible. Spiral coils are ideal for this operation because the length of their conductor is the lowest possible compared to other shapes of the inductors with the same inductance. However, the resistance is determined by the number of current-turns, thickness of the copper clad on the PCB and many other factors. Under the considerations published in [3], the inductance of the single-layer flat spiral coil may be expressed as follows:

$$L = \frac{\mu \cdot n^2 \cdot \left(d_{out} - \frac{w \cdot (n+1) + n \cdot s}{2}\right) \cdot C_1}{2} \cdot \left( \ln\left( \frac{C_2}{\frac{n \cdot (w+s) + w}{2 \cdot d_{out} - (n \cdot (w+s) + w)}} \right) + C_4 \left( \frac{n \cdot (w+s) + w}{2 \cdot d_{out} - (n \cdot (w+s) + w)} \right)^2 \right) \qquad (1)$$

Manuscript received June 3, 2012. This work was supported by the Internal Grant Agency at TBU in Zlin, project No. IGA/FAI/2012/056 and IGA/FAI/2012/053 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

Lukas Kouril is with Tomas Bata University in Zlin, Faculty of Applied Informatics, Department of Informatics and Artificial Intelligence, Namesti T.G.M. 5555, Zlin, Czech Republic (phone: 420-576-035133; e-mail: kouril@fai.utb.cz).

Martin Pospisilik is with Tomas Bata University in Zlin, Faculty of Applied Informatics, Department of Computer and Communication Systems, Namesti T.G.M. 5555, Zlin, Czech Republic (phone: 420-576-035228, e-mail:pospisilik@fai.utb.cz).

Milan Adamek is with Tomas Bata University in Zlin, Faculty of Applied Informatics, Namesti T.G.M. 5555, Zlin, Czech Republic (e-mail:adamek@fai.utb.cz).

Roman Jasek is with Tomas Bata University in Zlin, Faculty of Applied Informatics, Namesti T.G.M. 5555, Zlin, Czech Republic (e-mail:jasek@fai.utb.cz).

Where:

- n – number of current-turns,
- w – conductor width [m],
- dout – outer diameter of the coil [m],
- s – isolation gap between the current-turns [m],
- μ - permeability of the PCB [$Hm^{-1}$],
- $C_1$ – constant, for spiral shape $C_1 = 1.00$,
- $C_2$ – constant, for spiral shape $C_2 = 2.46$,
- $C_4$ – constant, for spiral shape $C_4 = 0.20$.

Considering the copper clad thickness is 18μm (the worst case), the resistance may be according to [3] expressed as:

$$R = \frac{35}{36} \cdot \frac{1}{w} \cdot 10^{-3} \cdot \int_{\alpha=0}^{\alpha=2\pi n} \sqrt{\left(\left(\frac{d_{out}}{2}\right) - \frac{\alpha}{2\pi}(w+s)\right)^2 + \left(\frac{w+s}{2\pi}\right)^2}\, d\alpha \quad (2)$$

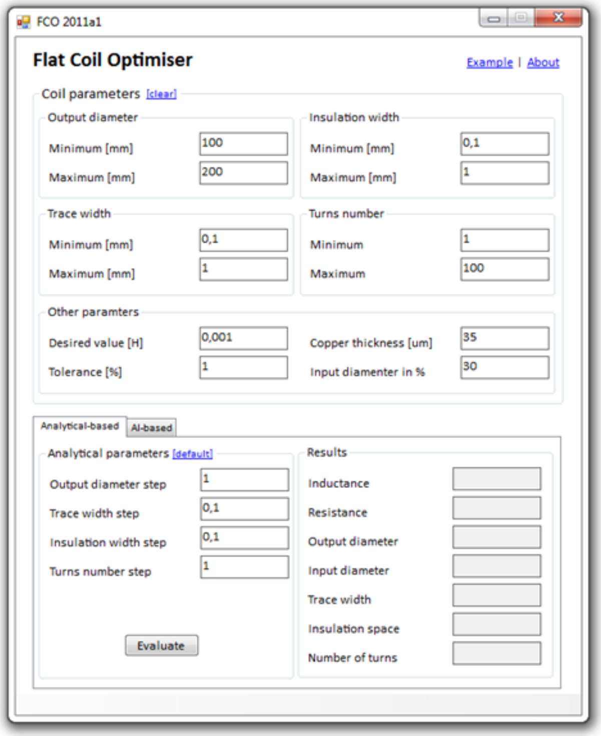
As stated above, the optimization is based on achieving the lowest possible resistance R for the required L with the dimensions of the coil defined by the designer. This can be processed by several types of algorithms. The Flat Coil Optimizer described in this paper employs two different algorithms – the iterative algorithm and the artificial intelligence based algorithm. The results of both can be the easily compared.

## II. Flat Coil Optimizer

The application was created as a result of authors' knowledge experienced during research [1-3], [11] aimed on optimization of coils by artificial intelligence. Flat Coil Optimizer is based on Microsoft .NET Framework 4.0 and is programmed in C# programming language. The application utilizes two optimization methods. These are analytical-based method and AI-based method. The latter method is provided by FSAI library (F# Artificial Intelligence library) which implements Differential evolution in F# programming language. This library can be downloaded for free at Codeplex[1] and used by projects programmed in whatever programming languages allow consuming .NET libraries.

As can be seen in Fig. 1, the application window has two main areas. The first area contains text fields for coil parameters. These are:

- Min / max output diameter in millimeters.
- Min / max trace width of coil in millimeters.
- Min / max insulation width in millimeters.
- Min / max number of coil turns.
- Desired value of inductance in Henry.
- Percentage tolerance of inductance
- Thickness of copper in micrometers.
- Input diameter of coil in percentage.



Flat Coil Optimiser

Coil parameters

Output diameter: Minimum [mm] 100; Maximum [mm] 200

Insulation width: Minimum [mm] 0,1; Maximum [mm] 1

Trace width: Minimum [mm] 0,1; Maximum [mm] 1

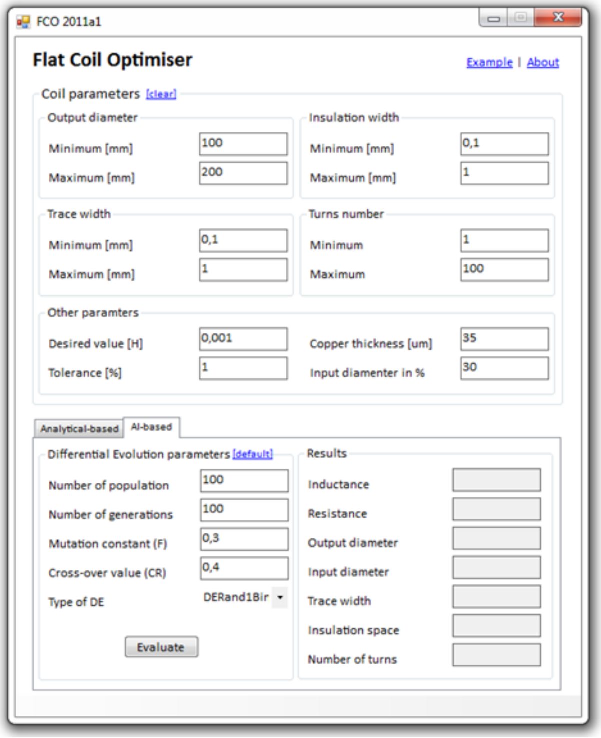
Turns number: Minimum 1; Maximum 100

Other paramters: Desired value [H] 0,001; Tolerance [%] 1; Copper thickness [um] 35; Input diamenter in % 30

Analytical parameters: Output diameter step 1; Trace width step 0,1; Insulation width step 0,1; Turns number step 1

Results: Inductance; Resistance; Output diameter; Input diameter; Trace width; Insulation space; Number of turns

Fig. 1. Application screenshot #1



Flat Coil Optimiser

Coil parameters

Output diameter: Minimum [mm] 100; Maximum [mm] 200

Insulation width: Minimum [mm] 0,1; Maximum [mm] 1

Trace width: Minimum [mm] 0,1; Maximum [mm] 1

Turns number: Minimum 1; Maximum 100

Other paramters: Desired value [H] 0,001; Tolerance [%] 1; Copper thickness [um] 35; Input diamenter in % 30

Differential Evolution parameters: Number of population 100; Number of generations 100; Mutation constant (F) 0,3; Cross-over value (CR) 0,4; Type of DE DERand1Bin

Results: Inductance; Resistance; Output diameter; Input diameter; Trace width; Insulation space; Number of turns

Fig. 2. Application screenshot #2

The second area is located at the bottom of the main window of the application. This part is divided to two tabs. Each tab is aimed on one of optimization methods. Analytical-

[1] http://fsai.codeplex.com

based method tab is shown in Fig 1. AI-based method tab can be seen in Fig 2.

### *A. Analytical-based method*

Input parameters of analytical-based method are represented by text fields for:

- Step of output diameter
- Step of trace width
- Step of insulation width
- Step of number turns

This method employs a simple iterative algorithm that changes the possible parameter values within the range of excepted rates. In each step the above mentioned expressions are evaluated analytically and the results are stored in a buffer from which is the best result chosen at the end of the iteration. The user can define the steps by which the parameters are changed during the iterative process.

The iterative algorithm has been derived from the more complex algorithm, developed and debugged in Maple software, which enabled to visualize trends occurring when optimizing the inductor in a defined range of parameters.

The full algorithm flowchart is depicted in Fig. 3. The part of the algorithm implemented in the Flat Coil Optimiser software is displayed in a grey rectangle. Parameters according to Table 1 are employed. The parameter describing thickness of the copper clad is not involved in the table as it is directly implemented in the equation for R[$L_R$,j] computation (see Fig. 3).

The algorithm iterates from $L_{MIN}$ to $L_{MAX}$ with the step of $L_{STEP}$ and in each of the iterations it generates the optimal values $D_{OUT}$, W, S and N within the framework of the pre-set boundaries and steps in order the minimum resistance was achieved at the appropriate $L \pm L \cdot \frac{TOL}{100}$. As the result a set of graphs is generated in order to visualise the trends.

The example of the trends for the parameter setting according to Table 2 is depicted in Fig. 4.

The trends depicted in Fig. 4 are influenced not only by the setting of the appropriate parameters but with the order of the iterating cycles implemented in the iterative algorithm as well. This phenomenon is generally not considered a failure because the solution is always found in the framework of the specified parameter and the requirement of achieving the lowest possible resistance always ensure the selection of the proper solution from the range of all possible solutions.

The Flat Coil Optimizer software does not process the graphs as it only evaluates one point of them, according to the desired inductance. The authors however consider implementing the feature of searching for more solutions in the vicinity of the desired inductance so the user could see the trends and therefore assess how critical, concerning the physical limitations, his/her requirements are. How critical the requirements are can be deduced from the graphs depicted in Fig. 4, see text below. Such a computation would unfortunately be highly time consuming, leading to the need of parallelization and optimized implementation.

*Table 1 – Parameters required by the iterative algorithm depicted in Fig. 3*

| Parameter | Description |
|---|---|
| $L_{MIN}$ | Minimum accepted inductance [H] |
| $L_{MAX}$ | Maximum accepted inductance [H] |
| $L_{STEP}$ | Inductance step (defines the number of iterations between $L_{MIN}$ a $L_{MAX}$, that affects the computing time and the resolution of the obtained graphs). |
| $D_{OUT\_MIN}$ | Minimum allowed outer diameter [m] |
| $D_{OUT\_MAX}$ | Maximum allowed outer diameter [m] |
| $D_{OUT\_STEP}$ | Outer diameter step [m] |
| $W_{MIN}$ | Minimum allowed conductor width [m] |
| $W_{MAX}$ | Maximum allowed conductor width [m] |
| $W_{STEP}$ | Conductor width step [m] |
| $S_{MIN}$ | Minimum allowed insulation width [m] |
| $S_{MAX}$ | Maximum allowed insulation width [m] |
| $S_{STEP}$ | Insulation width step |
| **TOL** | Tolerance [%] considering which the target inductance must be achieved. The lower the tolerance is the smaller amount of results for resistance computation is obtained, which leads to making the algorithm run faster but less reliable as there is a risk that no results are obtained at all. |
| $N_{MIN}$ | Minimum number of current turns |
| $N_{MAX}$ | Maximum number of current turns |
| $N_{STEP}$ | Current turns step. Generally may be a real number (for example a coil with 30 ¾ turns can be designed). |
| $D_{IN}$ | Inner diameter of the coil relative to its outer diameter |

*Table 2 – Example of parameter settings*

| Parameter | Setting |
|---|---|
| $L_{MIN}$ | $5 \cdot 10^{-5}$ |
| $L_{MAX}$ | $1 \cdot 10^{-3}$ |
| $L_{STEP}$ | $2.5 \cdot 10^{-5}$ |
| $D_{OUT\_MIN}$ | $1.8 \cdot 10^{-1}$ |
| $D_{OUT\_MAX}$ | $2.05 \cdot 10^{-1}$ |
| $D_{OUT\_STEP}$ | $5 \cdot 10^{-3}$ |
| $W_{MIN}$ | $1 \cdot 10^{-3}$ |
| $W_{MAX}$ | $1.5 \cdot 10^{-3}$ |
| $W_{STEP}$ | $1 \cdot 10^{-4}$ |
| $S_{MIN}$ | $2.5 \cdot 10^{-4}$ |
| $S_{MAX}$ | $1.5 \cdot 10^{-3}$ |
| $S_{STEP}$ | $5 \cdot 10^{-5}$ |
| **TOL** | 1 |
| $N_{MIN}$ | 1 |
| $N_{MAX}$ | $1 \cdot 10^{2}$ |
| $N_{STEP}$ | 1 |
| $D_{IN}$ | $0.5 \cdot D_{OUT}$ |

In Fig. 4 the meaning of the graphs is as follows:

a – Dependence of the lowest achievable resistance on the desired inductance,

b – Dependence of the number of current-turns needed on the desired inductance,

c – Outer coil diameter proposal dependence on the desired inductance,

d – Inner coil diameter proposal dependence on the desired inductance,

e – Conductor width proposal dependence on the desired inductance,

f – Insulation gap width proposal dependence on the desired inductance.

It is obvious that the trends depicted in Fig. 4 comply with the theoretical assumptions. For low required inductances there is a lot of space and the width of the conductor and the insulation gap can therefore be close to the maximum allowed number. If the required inductance is increased, the widths must be reduced at size as well as the outer diameter of the inductor is increased. This is the cause of the increase in the minimum achievable resistance, as can be seen in Fig. 4a. Theoretically, if the widths were not changed, the increase of the minimum resistance would copy the increase of the number of current turns needed in order to achieve the desired inductance (see Fig. 4b). In the case presented at Fig. 4a this phenomenon does not occur as the resistance is increased due to reduced conductor widths. On the basis of this finding the authors consider the algorithm to work properly.

### *B. AI-based method*

This method utilizes Differential evolution [4-6] as optimization method. It is a method of artificial intelligence which is inspired by natural processes of evolution where individuals best-adapted to surrounding environment are naturally selected in order to "survivance". The adaptation is expressed as properties (or abilities) of individuals. Differential evolution imitates this process in silico. Contrary to natural evolution, considered properties of individuals are not abilities as strength, hunting etc. which influence real survivance but parameters influencing correctness of final solution of the problem. Except Differential evolution, there are also other methods which imitate natural evolution e.g. SOMA [9], [12] etc.

Optimization proceeds in several generations. Before optimization starts, there is prepared initial population which contains individuals with random values of encoded parameters. These values are generated in scope of specimen which states a number of parameters and boundaries of their values. In this case, the specimen (and individuals based on specimen) is a vector compounded of following parameters with described boundaries:

- *Output diameter*
  Boundaries: Min and max output diameter
- *Trace width*
  Boundaries: Mini and max trace width
- *Insulation space*
  Boundaries: Min and max insulation space
- *Number of turns*
  Boundaries: Min and max number of turns

Boundaries of these parameters are entered through text fields (see fig 1 or 2) located at main application window.

When optimization starts, individuals contained in each generation are subsequently evaluated. It shortly means that parameters of solution encoded as individuals are substituted with mathematical expression of the problem. The difference between expected result and temporary result generally represents the evaluation of current individual thus temporary solution optimality and suitability of individual to "survive". The most-suitable and most-optimal individuals of current generation are passed to the next generation where this process is repeated.

The optimization is processed as transformations of vectors (individuals) ensured by mutation and cross-over (similarly to real evolution). Within optimization and proceeding of individuals there are created several vectors based on current individual and on other three randomly selected individuals. The first vector – differential vector is originated as subtraction of two randomly selected individuals. Then the weighted differential vector is created by multiplication of differential vector and the mutation constant. As the next step, the noise vector is created as addition of the third randomly selected individual to the weighted differential vector. As the last vector, the test vector is originated by cross-over of the noise vector and current individual. The current individual and the test vector are evaluated by the cost function which implements conditions which influence correctness of problem solution. The individual or the test vector with better evaluation is passed to the next generation where the optimization is repeated.

Differential evolution has several parameters which influence the optimization. These can be entered by text fields in the bottom area of Flat Coil Optimizer application windows. The parameters are:

- Number of population
- Number of generations
- Mutation constant
- Cross-over value

The above mentioned parameters represent settings of Differential evolution. Except that, it is necessary to specify the type of Differential evolution. The FSAI library implements four types, which are DERand1Bin, DERand2Bin, DEBest1Bin and DEBest2Bin. These types represent different ways of the noise vector expression.

When the optimization finishes, the requested parameters of coil (encoded as individual) as well as other additionally computed parameters (inductance, resistance and input diameter) are shown in non-editable text fields in the right part of AI-based method tab. Whole AI optimization process realized by Flat Coil Optimizer can be seen in Fig. 5.

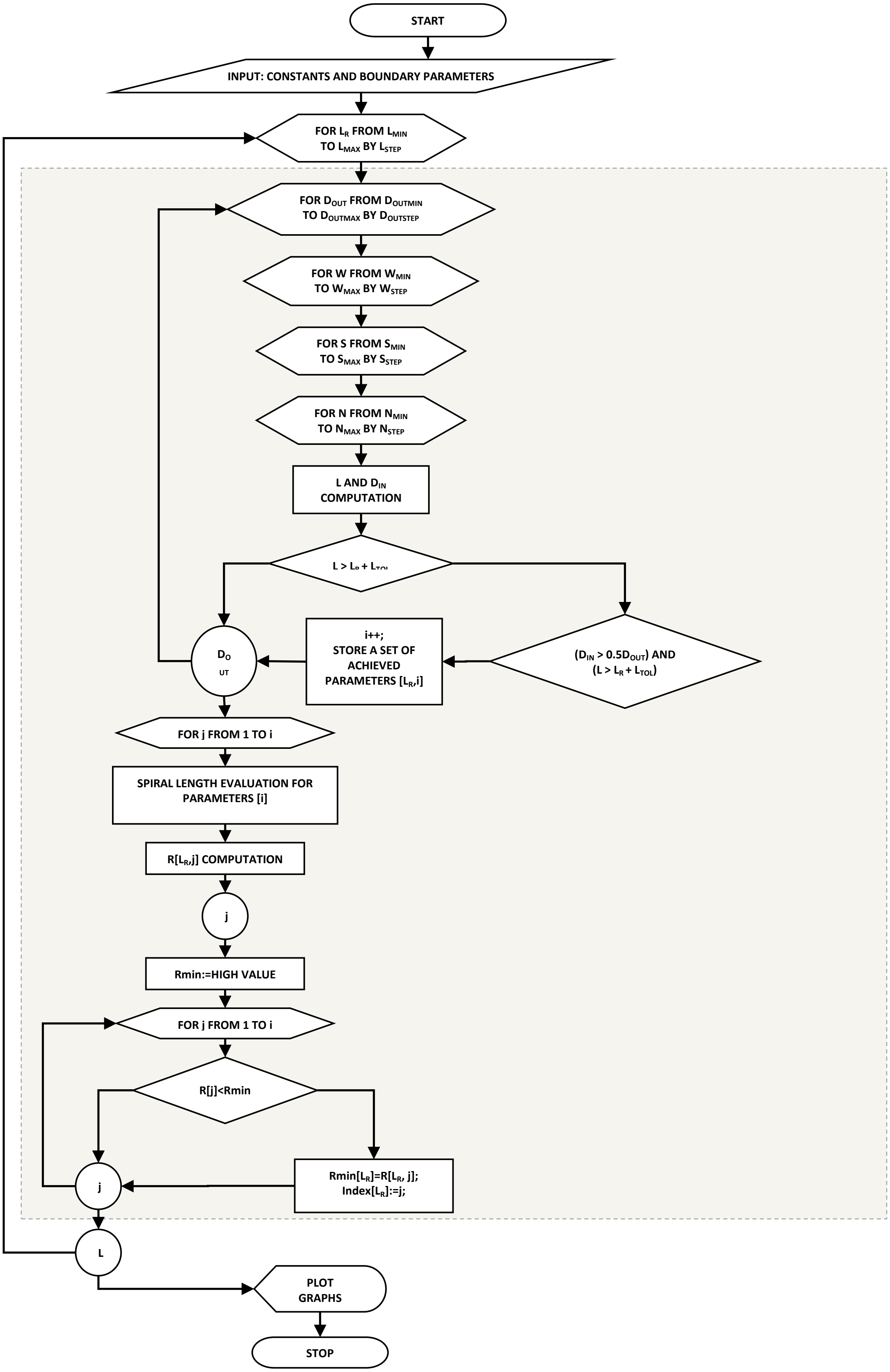

START
INPUT: CONSTANTS AND BOUNDARY PARAMETERS
FOR $L_R$ FROM $L_{MIN}$ TO $L_{MAX}$ BY $L_{STEP}$
FOR $D_{OUT}$ FROM $D_{OUTMIN}$ TO $D_{OUTMAX}$ BY $D_{OUTSTEP}$
FOR W FROM $W_{MIN}$ TO $W_{MAX}$ BY $W_{STEP}$
FOR S FROM $S_{MIN}$ TO $S_{MAX}$ BY $S_{STEP}$
FOR N FROM $N_{MIN}$ TO $N_{MAX}$ BY $N_{STEP}$
L AND $D_{IN}$ COMPUTATION
$L > L_R + L_{TOL}$
$(D_{IN} > 0.5D_{OUT})$ AND $(L > L_R + L_{TOL})$
i++; STORE A SET OF ACHIEVED PARAMETERS $[L_R,i]$
$D_O$ UT
FOR j FROM 1 TO i
SPIRAL LENGTH EVALUATION FOR PARAMETERS [i]
$R[L_R,j]$ COMPUTATION
j
Rmin:=HIGH VALUE
FOR j FROM 1 TO i
R[j]<Rmin
$Rmin[L_R]=R[L_R, j]$; $Index[L_R]:=j$;
j
L
PLOT GRAPHS
STOP

Fig. 3. Full iterative algorithm flowchart

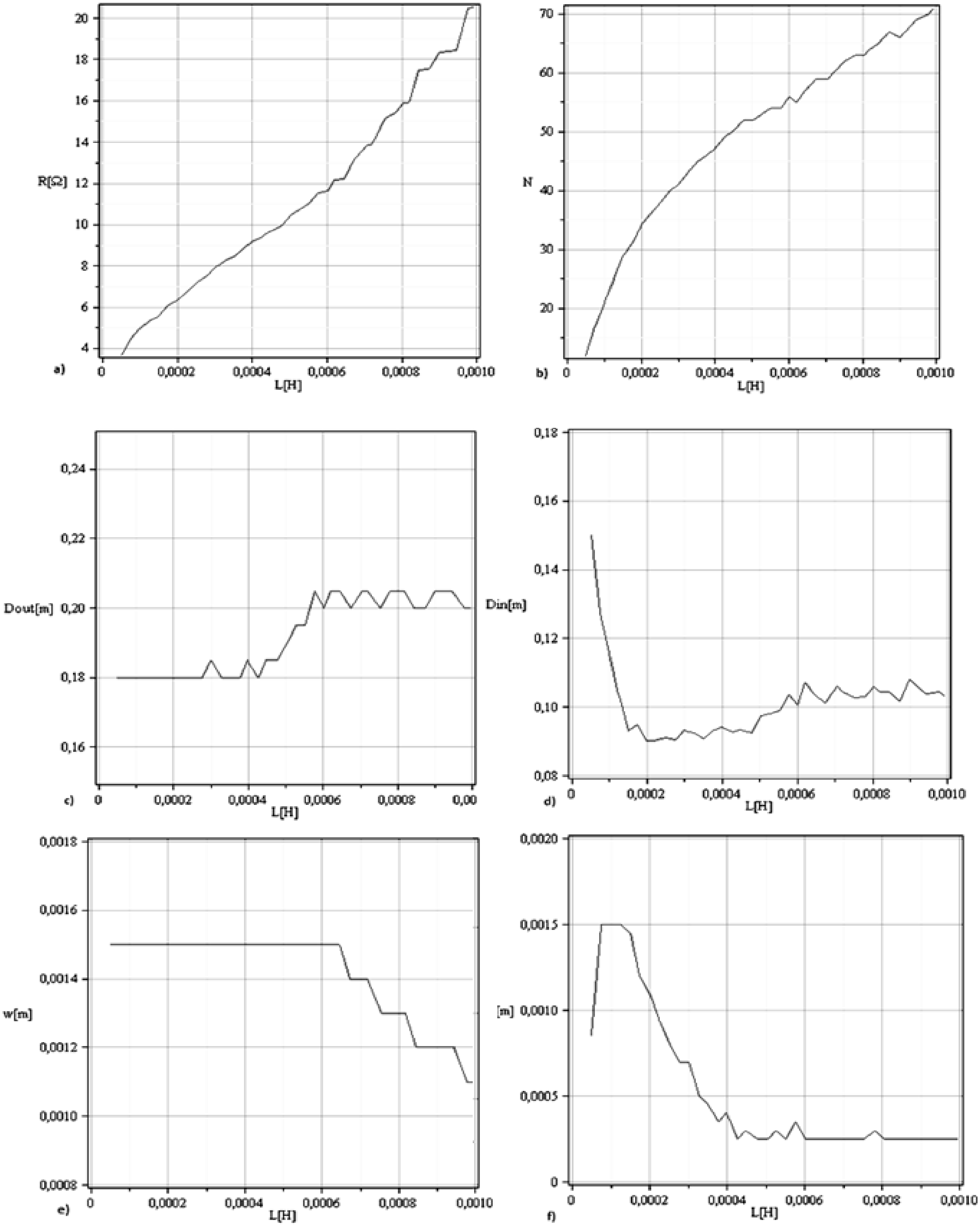


Fig. 4. Iterative algorithm results for different desired inductances. See text above for description.

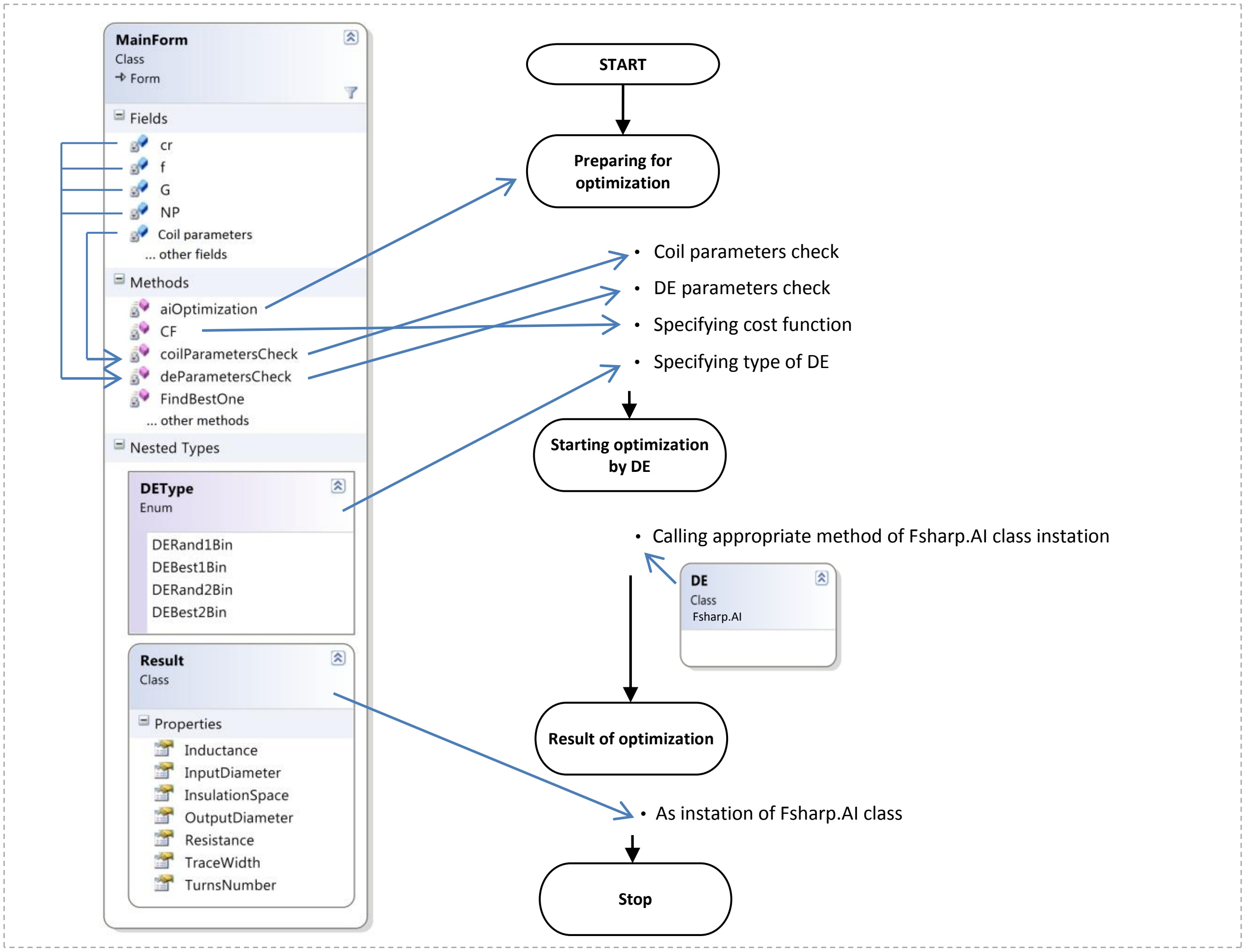


Fig. 5. Simplified scheme of AI optimization implementation

### C. *Application requirements*

With regard to utilization of Microsoft .NET Framework 4.0, its presence is required on the computer indented to run Flat Coil Optimizer. The application was developed for Microsoft Windows 7 primarily thus other operation systems are not directly supported but possibilities of compatibility with other operating systems are not omitted.

## III. Application Utilization

The application was created in order to help the electronics circuit designers to create flat coils on the PCBs under the consideration their high Q is required. This application is suitable for low-frequency designs because in the meantime it does not support parasitic capacitances among the conductors that occur at high frequencies, creating the prevailing factor to reduce the coil's performance. However, the authors are continuing in their research and currently, improving of this drawback is their main goal.

The possibility of utilization of two different approaches (analytical approach and artificial intelligence) enables the user, in case of doubts, to verify the results obtained by one of the method by employing the second one.

## IV. Further Research

The further research consists in implementing the following features:

- parasitic capacity and resonant frequency estimation,
- other shapes than spiral implementation,
- multilayer coils design,
- trends evaluation for more inductances in the vicinity of the desired inductance.

### A. *Parasitic capacity estimation*

This problem is complicated by the fact that the capacity is spread across the whole geometric shape of the coil and moreover the two neighbouring conductors cannot be treated as two different ones as the current via one of them is determined by the current flowing via the second one. This could probably be solved by evaluation of the voltage difference between the two neighbouring conductors relatively to the total voltage on the coil. This percentage could be applied for proper reducing of the computed capacity in order it complied with reality but another problem rises up when this solution is applied. In flat coils every single turn is of different length. For spiral-shaped coils the voltage percentage would change fluently across the whole length of the conductors. Moreover not only the capacity between the two of the conductors can be considered at one point. In fact all the pieces of the conductor in all of the turns influence each other. The authors did not find the proper approximation in order to estimate the capacitance properly yet.

### B. *Other shapes than spiral implementation*

According to [8] the equation (1) for expressing the inductance of the flat coil can be easily modified for other possible shapes by changing the constants $c_1$ to $c_4$. More details can be found in [8].

### C. *Multilayer coils design*

Previously the authors created algorithms for optimising double-layered coils. Details are described in [2]. The algorithms are based on the fact that the inductance of the coils that are concentrically placed one above the other, close enough and fed with the same current can be expressed according to the equation (3).

$$L = L_1 + L_2 \pm 2M \tag{3}$$

Where $L_1$ and $L_2$ correspond to the inductances of the coils (can be replaced by 2L as the coils are identical) and M is their mutual inductance. Due to the mutual inductance the total inductance can be increased more than twice when two coils are implemented, but the practical computation shown that the results are quite ambiguous [2]. More layers than two are needed in order to achieve better results.

### D. *Trends evaluation*

As described above in the text, evaluation of trends describing how the parameters are evolved when the desired inductance is increased can show how critical, considering the physical limitations, the user's demands are.

## V. Conclusion

In this paper the application that enables the designer to create resistance-optimized single-layer flat coils on the PCBs is described. In the application the authors implemented two different approaches. This is advantageous because the results of both approaches, the iterative analytical algorithm and the artificial intelligence-based algorithm, can be compared in order to verify the results. Currently, the application is fully functional but its utilization is restricted to the applications where the parasitic capacitances of the conductors on the PCB are negligible. The further research of the authors is focused on implementing all the features described in the chapter "Further research". The most important problem consists in parasitic capacity estimation as the absence of this feature limits the utilization of this software for applications employing higher frequencies.

## Acknowledgment

This paper is supported by the Internal Grant Agency at TBU in Zlin, project No. IGA/FAI/2012/056 and IGA/FAI/2012/053 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

## References

[1] M. Pospisilik, L. Kouril., M. Adamek, "Comparison of Differential Evolution and Self-Organising Migrating Algorithm at Flat Spiral Inductor Optimization" in *Proc. XIII Annu. Int. Conf. Internet, Competitiveness and Organizational Security – Process Management and the Use of Modern Technologies*, Zlin, Czech Republic, 2011, pp. 323-327. ISBN 978-80-7454-012-7.

[2] M. Pospisilik, L. Kouril, I. Motyl, M. Adamek, "Single and Double Layer Spiral Planar Inductors Optimisation with the Aid of Self-Organising Migrating Algorithm" in *Proc. 11th WSEAS Int. Conf. on Signal Processing, Computational Geometry and Artificial Vision (ISCGAV '11) – Recent Advances in Signal Processing, Computational Geometry and System Theory*, Florence, Italy, 2011, pp. 272-277. ISBN 978-1-61804-027-5.

[3] M. Pospisilik, L. Kouril, M. Adamek, "Planar Inductor Optimised by Evolutionary Algorithm" in *Proc. 17th Int. Conf. on Soft Computing MENDEL 2011*, Brno, Czech Republic, 2011, pp. 38-43. ISBN 978-80-214-4302-0.

[4] J. Lampinen, I. Zelinka, "Mechanical Engineering Desing Optimization by Differential Evolution" in *New Ideas of Optimization*, 1st London: McGraw-Hill, 1999. ISBN 007-709506-5.

[5] I. Zelinka, *Umela inteligence v problemech globalni optimalizace*. Praha: BEN – technicka literatura, 2002. ISBN 80-7300-069-5.

[6] I. Zelinka, Z. Oplatkova, M. Seda, P. Osmera, F. Vcelar, *Evolucni vypocetni techniky – principy a aplikace*. Praha: BEN – technicka literatura, 2008. ISBN 80-7300-218-3.

[7] H.-M. Greenhouse, "Design of Planar Rectangular Microelectronic Inductors", *IEEE Trans. on Parts, Hybrids and Packaging*, vol. PHP-10, no. 2, June 1974.

[8] S. Mohan, M. Hershenson, S. Boyd, T. Lee, "Simple Accurate Expressions for Planar Spiral Inductances", *IEEE J. of Sold-state Circuits*, vol. 34, no. 10, Oct. 1999.

[9] P. Varacha, "Innovative Strategy of SOMA Control Parameter Setting" in *Proc. 12th WSEAS Int. Conf. on Neural Networks, Fuzzy Systems, Evolutionary Computing & Automation – Recent Researches on Neural Networks, Fuzzy Systems, Evolutionary Computing and Automation*. Timisoara, 2011, pp. 70-75. ISBN 978-960-474-292-9.

[10] Delka krivky (2012, May 15) [Online]. Avaílable: http://www.math.muni.cz/~xschlesi/dp/web/i22.html

[11] L. Kouril, M. Pospisilik, M. Adamek, R. Jasek, "Coil Optimization with Aid of Flat Coil Optimizer" in *Proc. 5th WSEAS World Congress on Applied Computing Conference*, Faro, Portugal, 2012, pp. 124-127. ISBN 978-1-61804-089-3.

[12] P. Varacha, "Neural Network Synthesis via Asynchronous Analytic Programming" in *Proc. 12th WSEAS Int. Conf. on Neural Networks, Fuzzy Systems, Evolutionary Computing & Automation – Recent Researches on Neural Networks, Fuzzy Systems, Evolutionary Computing and Automation*. Timisoara, 2011, pp. 92-97. ISBN 978-960-474-292-9

**Lukas Kouril** was born in Czech Republic. He completed Master's degree in Engineering Informatics at the Tomas Bata University in Zlin, Faculty of Applied Informatics in Czech Republic. Today is a PhD student at Tomas Bata University in Zlin. His research area is the application of artificial intelligence to the optimization.
He is a researcher at the regional research center CEBIA-Tech in the Czech Republic where he is concerned with biomedical signal processing.

**Martin Pospisilik** was born in Czech Republic. He completed Master's degree at the Czech Technical University in Prague and since 2008 he is with Tomas Bata University in Zlin, Faculty of Applied Informatics, employed as assistant and PhD student. His research area is the design, construction and optimization of electrical circuits.
He is a researcher at the regional research center CEBIA-Tech in the Czech Republic where he is concerned with biomedical signal processing.