



# High Power LED Parametric Modelling Using a Double Polynomial Approximation

K. Zaplatílek<sup>1\*</sup> and J. Leuchter<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, University of Defence, Brno, Czech Republic

<sup>2</sup> Department of Radar Technology, University of Defence, Brno, Czech Republic

The manuscript was received on 9 January 2017 and was accepted  
after revision for publication on 15 June 2017.

## Abstract:

*This article describes calculations for an accurate mathematical model of high power LED modules using double optimal polynomial approximation. The model is based on unique tests of various LED types in a thermal chamber, providing a comprehensive list of parametric temperature profiles. This model was then implemented into MATLAB®&Simulink® and Micro-Cap programs as a Spice compatible electronic circuit model, utilising the newly created algorithm. To define an optimal degree of approximation polynomials, Euclidean norm of residues was used. The new algorithm described in this article was verified using real-life data tested at the author's work site, where the corresponding research takes place. To maximise the test's efficiency, an automated data collection system was created. This article describes one particular tested LED module whose characteristic was modelled in both the absolute and the normalised form for easy comparison.*

## Keywords:

*high power LED module, optimal polynomial approximation, Euclidean norm of residues, MATLAB®&Simulink®, Spice compatible model, Micro-Cap simulator*

## 1. Introduction

High power LED modules have started to spread into many scientific fields and industries. Corresponding research is taking place at numerous universities as well as at various scientific sites, for example the automotive industry and security technologies. Real-life, however, shows that apart from their positive properties such as high energy-efficiency, low weight and low energy consumption, not enough attention is paid to their durability and reliability. In the most common operating conditions, the whole LED

---

\* Corresponding author: Department of Electrical Engineering, University of Defence,  
Kounicova 65, CZ-662 10 Brno, Czech Republic.  
Phone: +420 973 442 709, Fax: +420 973 443 773, E-mail: karel.zaplatilek@unob.cz

module is exposed to a thermal stress that can lead to a reduction in illumination rate or complete damage of the whole LED unit.

Up to now thermal tests of the high power LED modules have not been sufficiently dealt with. In the renowned databases such as Web of Science™, Scopus® or IEEE Xplore®, there are only a few relevant articles and citations addressing this issue. In addition, most of them are compilations, have descriptive nature or are denoted to technological challenges and evaluations [1-5]. On the other hand, there are many articles focusing on model identification and their optimisation [6-9]. Nevertheless, none of these resources, known to us so far, focus on approximation of thermal characteristics of these high power LED modules. This study is based on our two previous studies devoted to the modelling of photovoltaic panels and fuel cells [10] a [11]. These two studies, however, used different model creation methods – stochastic method and a method using optimum logarithmic characteristics.

The method for creating the model in this study is based on extensive tests of the high power LED modules in Vötsch VC<sup>3</sup> 7034 thermal chamber. The chamber is a part of the automated test site equipment [12] in which many LED modules with various performances, illuminations and spectral ranges were tested. Fig. 1 shows the measured thermal profile of Vigan GT-P10WR1499 LED module (10 W, 300 lx, angle 90°, red lighting). The black line represents the parameter dependences  $E = f(T)$  and  $I = const.$ , where  $E$  is the illumination,  $T$  is the temperature and  $I$  is an electrical current running through the module. The illumination has an absolute unit *lux* (lx). For modelling, it is more convenient to normalise the vertical axis for nominal operating point NOP [20 °C, 1 A].

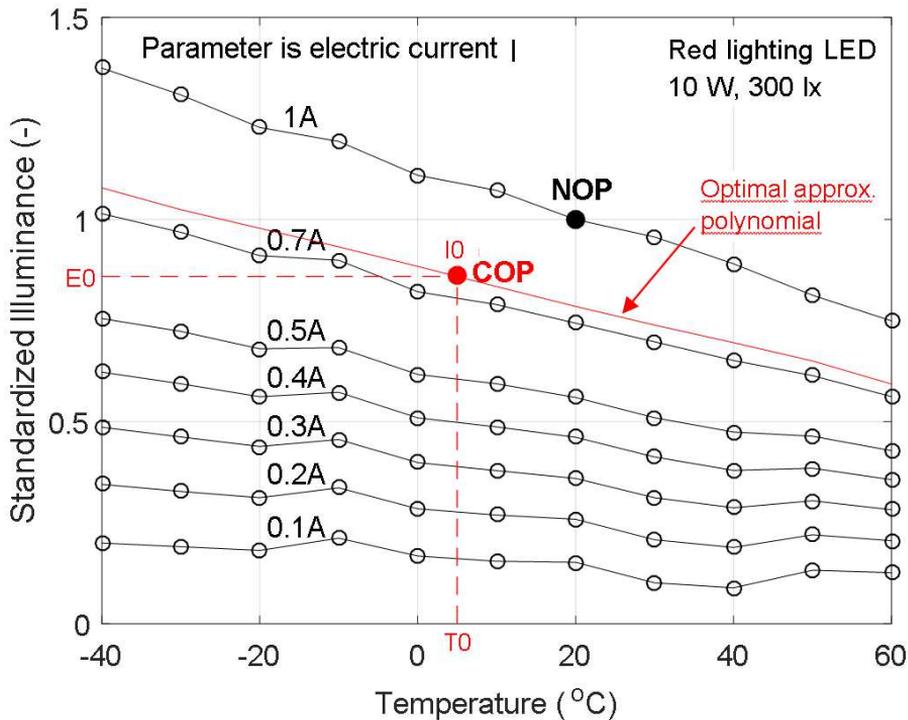


Fig. 1 Temperature parametric profile of the Vigan high power LED

The purpose of the below described algorithm is to calculate a random operating point COP (Calculated Operating Point) in a particular plane, i.e. to determine illumination  $E0$  for a user-defined temperature  $T0$  and an electrical current  $I0$ , see Fig. 1. It is done by performing double approximation using optimal polynomial. The red curve in Fig. 1 represents the profile of the resultant approximation polynomial.

### 2. Basic Principle of Creating the Model

The main purpose of the algorithm is to express the calculated operating point COP analytically in a random point of a plane  $E = f(T), I = const.$ , see Fig. 1. Fig. 2 shows a block schematic of a double polynomial approximation process leading to the desired result.

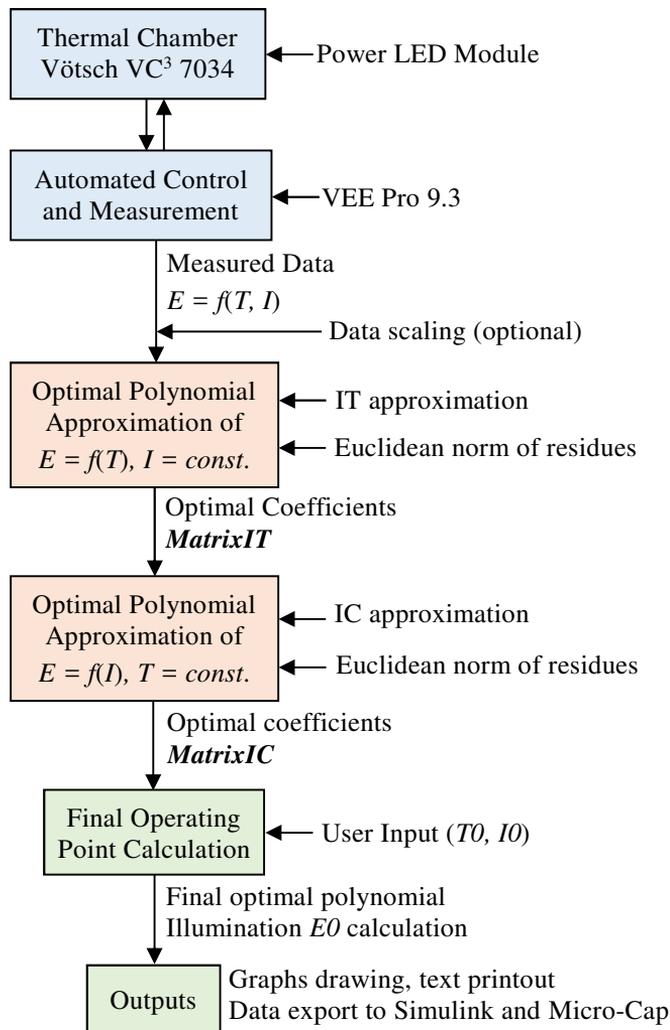


Fig. 2 Double optimal polynomial approximation algorithm

Inputs are the measured data as in Fig. 1 and also user-defined temperature  $TO$  and current  $IO$ . Outputs are optimal approximation polynomial, running via the analytically calculated operating point, and the calculated illumination  $EO$ . Apart from this, the algorithm is able to operate remotely a block model in the Simulink environment, i.e. is able to export the polynomial coefficients and the user-defined values to Simulink. In addition, the algorithm has also another input, the so-called Scale Factor  $SF$  (lx) that ensures normalisation of the measured illuminations in the nominal operating point NOP, see Fig. 1.

### 2.1. IT Approximation

The normalised measured values go into a block of, so called IT Approximation, whose main purpose is to determine optimal approximation polynomial profiles  $E = f(T)$ ,  $I = const$ . Fig.1 shows a total of tested  $M = 7$ , thus 7 optimal polynomials, or let us say their coefficients, shall be determined.

Polynomial approximation is a standard procedure that is available in the MATLAB environment thanks to *polyfit* and *polyval* functions, see Fig. 3.

```

for k=2:8 % approximation of 7 measured course
    p=polyfit(Data(:,1),Data(:,k),n); % n=5
    Coef(k-1,:)=p; % polynomial coefficients
    f=polyval(p,Data(:,1));
    Norm(k)=sqrt(sum((f-Data(:,k)).^2)); % Euclidean Norm of Residues
end

```

Fig. 3 Main function of an approximation algorithm

It is important to determine an optimisation degree of these polynomials for which the so called Euclidean norm of residues (shortly Norm) [10] is used as shown in the following formula:

$$Norm = \sqrt{\sum_{k=1}^n (p_{fk} - f_k)^2}, \quad (1)$$

where  $p_f$  is the profile of the approximation polynomial and  $f$  is the profile of the original approximated data. The Norm was calculated for each degree of the polynomial in the interval of  $n \in \langle 1, 7 \rangle$  for all seven tested profiles. Fig. 4 shows the dependence of the Norm on the polynomial degree for  $I = 1$  A.

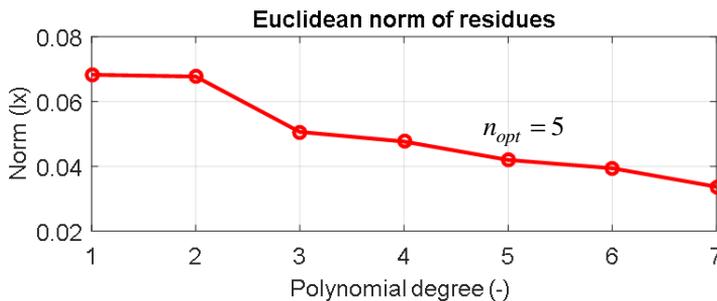


Fig. 4 Determination of optimal degree for approximation polynomial for  $I = 1$  A

The optimal degree can be calculated by an algorithm or it can be defined by a user based on a profile of the Norm. Out of all the seven local optimal degrees, the highest optimal degree is selected. The analysis has proved that the total optimal degree of the approximation polynomials is  $n_{opt} = 5$  (-). For degrees  $n > 7$ , the approximation polynomials were numerically poorly conditioned.

The IT approximation output is, according to Fig. 2, a matrix of the approximation polynomial optimal coefficients, see Fig. 5. In each line of the matrix, the polynomial coefficients of the 5<sup>th</sup> degree are sorted in descending order with the power of independent variable  $T$  in °C, see (2).

$$\begin{matrix}
 & a_{11} & a_{12} & a_{13} & \cdot & \cdot & a_{1n+1} \\
 & a_{21} & a_{22} & a_{23} & \cdot & \cdot & a_{2n+1} \\
 & a_{31} & a_{32} & \cdot & & & \cdot \\
 MatrixIT = & \cdot & \cdot & & & & \cdot & \cdot \\
 & \cdot & \cdot & & & & \cdot \\
 & \cdot & \cdot & & & & \cdot \\
 & a_{M1} & a_{M2} & a_{M3} & \cdot & \cdot & a_{Mn+1}
 \end{matrix} \tag{2}$$

The number of columns in **MatrixIT** corresponds to the number of optimal coefficients  $n + 1$ , where  $n = 5$ . The number of rows is given by the number of the approximated measured profiles, here  $M = 7$ . For example, the equation for optimal approximation polynomial of the 5<sup>th</sup> degree after the first test reads as follows (3):

$$I = a_{11}T^5 + a_{12}T^4 + a_{13}T^3 + a_{14}T^2 + a_{15}T + a_{16} (-). \tag{3}$$

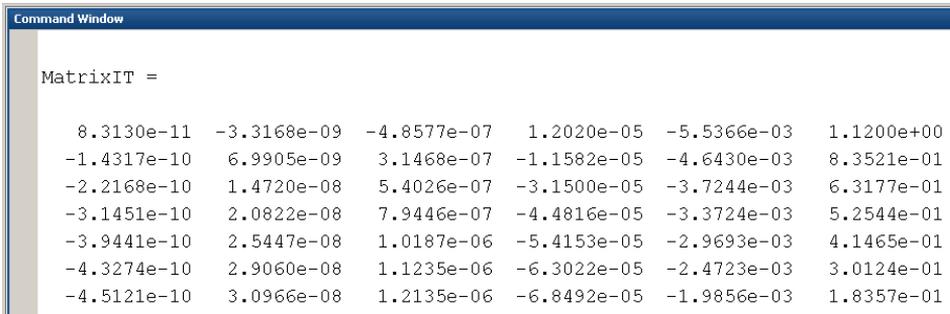


Fig. 5 Matrix for optimal approximation polynomials of IT approximation

**2.2. IC Approximation and Algorithm Outputs**

The purpose of IC approximation is to find optimal approximation polynomials in a cross direction of the measured data, i.e. of the function  $E = f(I), T = const$ . In other words, it is necessary to approximate the optimal coefficients from the IT approximation, or let us say the individual columns of the matrix in Fig. 5.

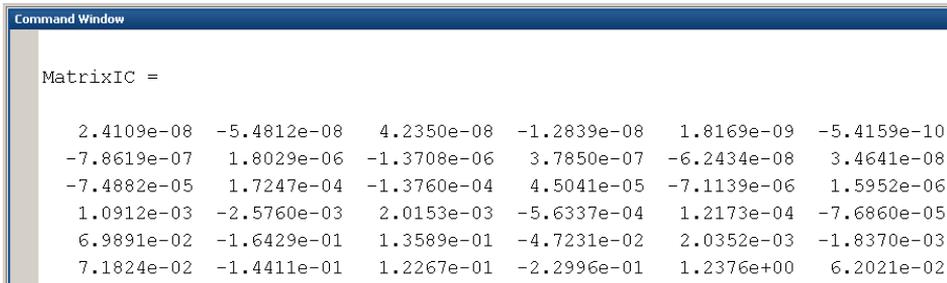
The matrix in Fig. 5 has 6 columns that underwent analogical polynomial approximation, also applying the Norm according to (1), to determine the total optimal degree. The output of this procedure is a matrix of optimal approximation polynomial coefficients of the previous approximation, see numbers in Fig. 6.

Each column in Fig. 6 represents one optimal approximation polynomial for IT approximation coefficients. The number of rows in *MatrixIC* is given by the number of optimal coefficients, here  $n+1 = 6$ . The number of columns corresponds to the number of approximated coefficients from IT approximation, thus 6, see (4). Equation (5) shows the symbolic relation for optimal approximation polynomial, approximating coefficients  $a_1 = [a_{11}, a_{21}, a_{31}, \dots, a_{M1}]$ , see (2) in which electric current  $I$  is an independent variable. If the user sets a particular value of electrical current  $I0$ , the algorithm finds the final approximation polynomial  $a = f(I)$ ,  $I = I0$  using the matrix in Fig. 6.

$$\text{MatrixIC} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1,n+1} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2,n+1} \\ b_{31} & b_{32} & \cdot & \dots & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ b_{n+1,1} & b_{n+1,2} & b_{n+1,3} & \dots & b_{n+1,n+1} \end{pmatrix} \quad (4)$$

For example, the equation for an optimal approximation polynomial of the 5<sup>th</sup> degree after first tests looks as follows (3):

$$a_1 = b_{11}Cur^5 + b_{21}Cur^4 + b_{31}Cur^3 + b_{41}Cur^2 + b_{51}Cur + b_{61}(-). \quad (5)$$



```

Command Window

MatrixIC =

    2.4109e-08   -5.4812e-08    4.2350e-08   -1.2839e-08    1.8169e-09   -5.4159e-10
   -7.8619e-07    1.8029e-06   -1.3708e-06    3.7850e-07   -6.2434e-08    3.4641e-08
   -7.4882e-05    1.7247e-04   -1.3760e-04    4.5041e-05   -7.1139e-06    1.5952e-06
    1.0912e-03   -2.5760e-03    2.0153e-03   -5.6337e-04    1.2173e-04   -7.6860e-05
    6.9891e-02   -1.6429e-01    1.3589e-01   -4.7231e-02    2.0352e-03   -1.8370e-03
    7.1824e-02   -1.4411e-01    1.2267e-01   -2.2996e-01    1.2376e+00    6.2021e-02
  
```

Fig. 6 Matrix for optimal approximation polynomials of IC approximation

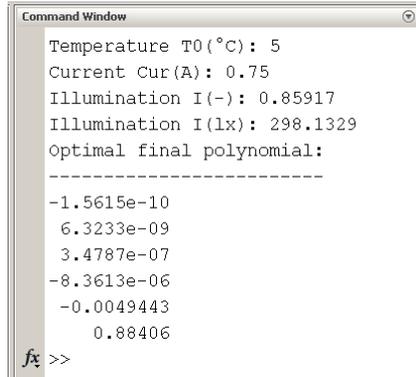
If the final approximation polynomial  $a = f(I)$  is known according to (4) and (5), then the system can already calculate the resultant illuminance  $E0$  for the inputted current and temperature values  $I0$  to  $T0$ , see (2), or (3). Key parts of the MATLAB source code are in Fig. 7. Variable *CoefCoef* is the coefficients matrix according to Fig. 6.

```

%-----
CCoef=zeros(1,6); % koefficients for Cur=Cur0
for k=1:6
    CCoef(k)=polyval(CoefCoef(k,:),Cur0);
end
I0=polyval(CCoef,T0);
%---
  
```

Fig. 7 Calculation of resultant illumination  $E0$  for input electrical current  $I0$

Fig. 8 shows the final output of the algorithm in a text format.  $T0$  and  $I0$  were input by the user, the algorithm then calculated the resultant illumination in units (–) and (lx), and also the coefficients of the final approximation polynomial running via the provided operating point COP, see the red line in Fig. 1.



```

Command Window
Temperature T0(°C): 5
Current Cur(A): 0.75
Illumination I(–): 0.85917
Illumination I(lx): 298.1329
Optimal final polynomial:
-----
-1.5615e-10
 6.3233e-09
 3.4787e-07
-8.3613e-06
-0.0049443
 0.88406
fx >>

```

Fig. 8 Final output of the algorithm in a text format

As mentioned in the conclusion of Fig. 2, the algorithm can automatically generate and export data to a block model in the Simulink environment, for more details see Chapter 3. Fig. 9 shows some of this source code. Apart from operating point coordinates COP [ $T0$ ,  $I0$ ], final optimal approximation polynomial coefficients are also shown (the total of 6 values,  $n=5$ ). Variable  $SF$  (Scale Factor) is also exported to prevent un-normalizing of  $I0$ , see Chapter 2.

### 3. Model Implementation in Simulink Environment

The resultant block model in a subsystem form is shown in Fig.10. The subsystem consists of six inputs which are the optimal coefficients of the final approximation polynomial (see Figs. 8 and 9), while other inputs are user-defined electrical current  $I0$ , temperature  $T0$  and Scale Factor  $SF$ . The subsystem output is the resultant illumination in unit (–) and (lx).

```

set_param('LED_Simulink_1/Coef1','value',['[',num2str(Coef22(1,:))','']')
set_param('LED_Simulink_1/Coef2','value',['[',num2str(Coef22(2,:))','']')
set_param('LED_Simulink_1/Coef3','value',['[',num2str(Coef22(3,:))','']')
set_param('LED_Simulink_1/Coef4','value',['[',num2str(Coef22(4,:))','']')
set_param('LED_Simulink_1/Coef5','value',['[',num2str(Coef22(5,:))','']')
set_param('LED_Simulink_1/Coef6','value',['[',num2str(Coef22(6,:))','']')
set_param('LED_Simulink_1/T0','value',num2str(T0))
set_param('LED_Simulink_1/Cur0','value',num2str(Cur0))
set_param('LED_Simulink_1/SF','value',num2str(SF))

```

Fig. 9 Export of optimal key values into a block model in Simulink

As mentioned earlier, the algorithm performs this input data export into the model automatically. Fig. 11 shows typical simulation and configuration parameters. The detailed subsystem structure is shown in Fig. 12. Key blocks are *polyval* functions that work in accordance with Fig. 7.

## High Power LED Parametric Modelling

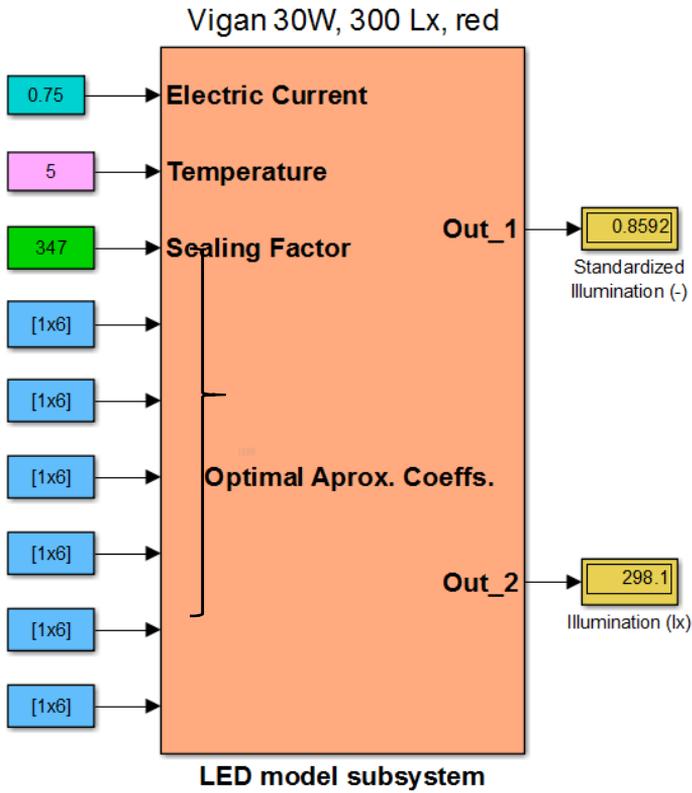


Fig. 10 Block model in Simulink as a subsystem

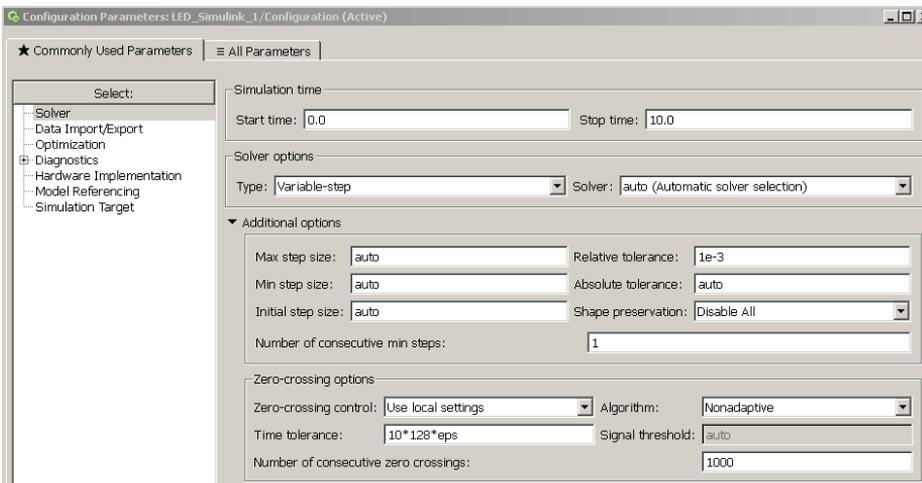


Fig. 11 Settings of simulation parameters in Simulink environment

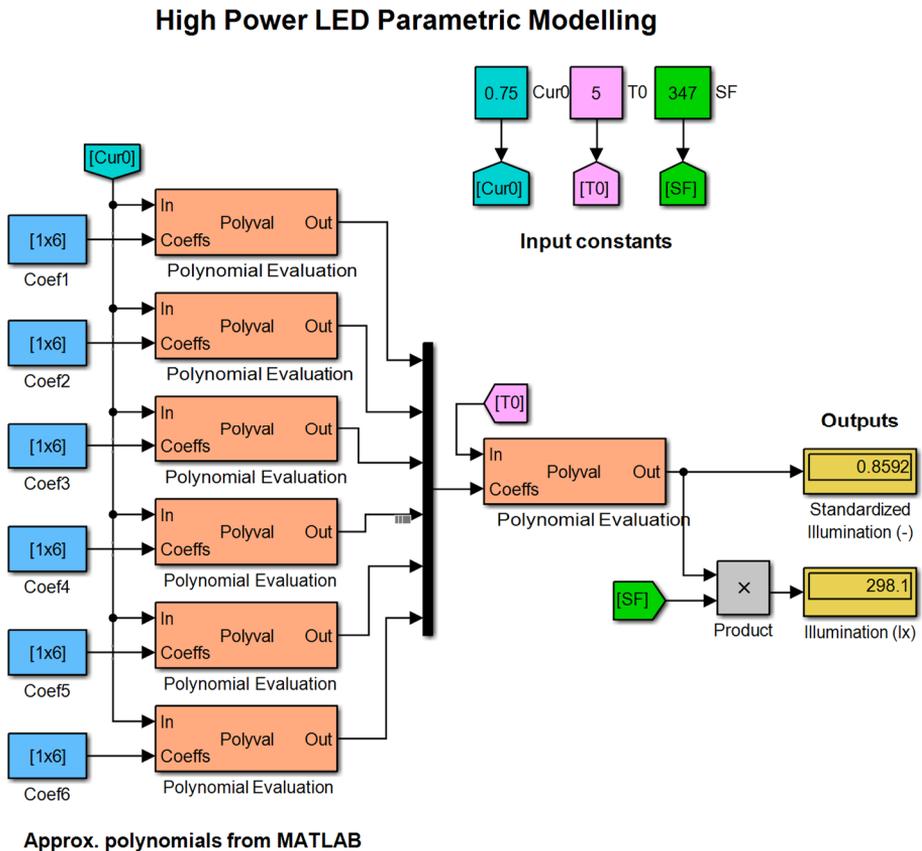


Fig. 12 Detailed block model of the subsystem

#### 4. Model Implementation in Spice Oriented Circuit Simulator

An important aspect of the whole approximation system is the ability to implement the model in a form of Spice compatible circuit model. This is an internationally acknowledged standard for creation of electric and electronic circuits and systems.

At the author’s work site, there is a professional circuit simulator Micro-Cap that allows work with Spice models [13]. To implement a model of the power LED module into the circuit simulator means to replace the numerical approach with a circuit approach, i.e. with a voltage current approach. To create a model in Micro-Cap, the so called “Analogue Behavioural Voltage Source NFV” is used, belonging to a group of functional power supplies, see Fig. 13. Each functional power supply represents one optimal approximation polynomial, or its coefficients shown in Fig. 6.

Input to the circuit model is the user-defined temperature  $T_0$  and the electrical current  $I_0$  in the form of a constant voltage supply (battery symbol). Temperature can also be inputted as AC power supply, allowing for a subtle wobble (fluctuation) of the LED module temperature. The change of both temperature excitations is ensured via manually operated (animated) switch Sw.

### High Power LED temperature parametric model

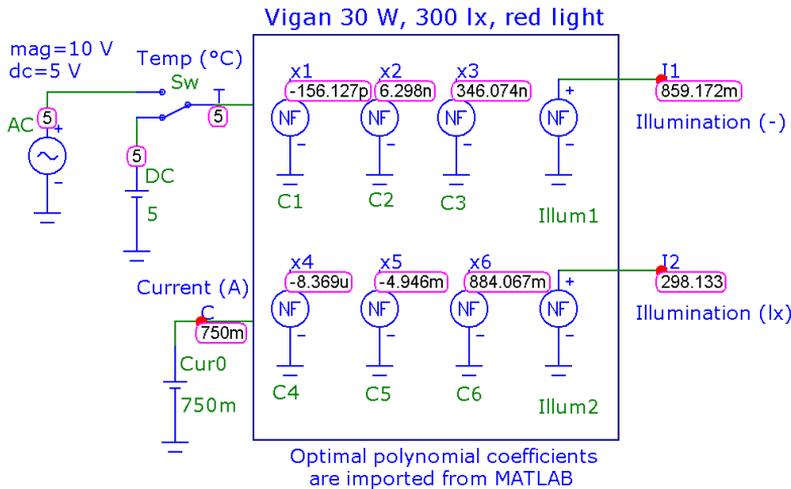


Fig. 13 Spice compatible circuit model in Micro-Cap

The output of the circuit model is illumination in units (–) and (lx) it is analogical to block model in Simulink. Individual coefficients of the functional power supplies NFV are defined by Spice commands “.define” as a text according to Fig. 14. The circuit model in Fig. 13 can have a schematic symbol assigned and the whole LED module can be used as another function block for simulation purposes of other more complex circuit structures. Values of the optimal approximation coefficients can be inputted into the functional power supplies NFV manually or they can be imported into the program by Spice language commands.

```

.define a1 2.4109e-08
.define b1 -5.4812e-08
.define c1 4.235e-08
.define d1 -1.2839e-08
.define e1 1.8169e-09
.define f1 -5.4159e-10
Coef2
.define a2 -7.8619e-07
.define b2 1.8029e-06
.define c2 -1.3708e-06
.define d2 3.785e-07
.define e2 -6.2434e-08
.define f2 3.4641e-08

```

Fig. 14 Polynomial coefficients definition using a Spice command

By selecting the *Transient* type of analysis, the whole model will be simulated, see Fig. 15. The analysis output shows the response of the LED module to a temperature fluctuation, see Fig. 13. The temperature fluctuated in the range of  $\pm 10$  °C with 5 °C

offset and a frequency of 1 Hz. Fig. 15 shows COP movement caused by temperature changes.

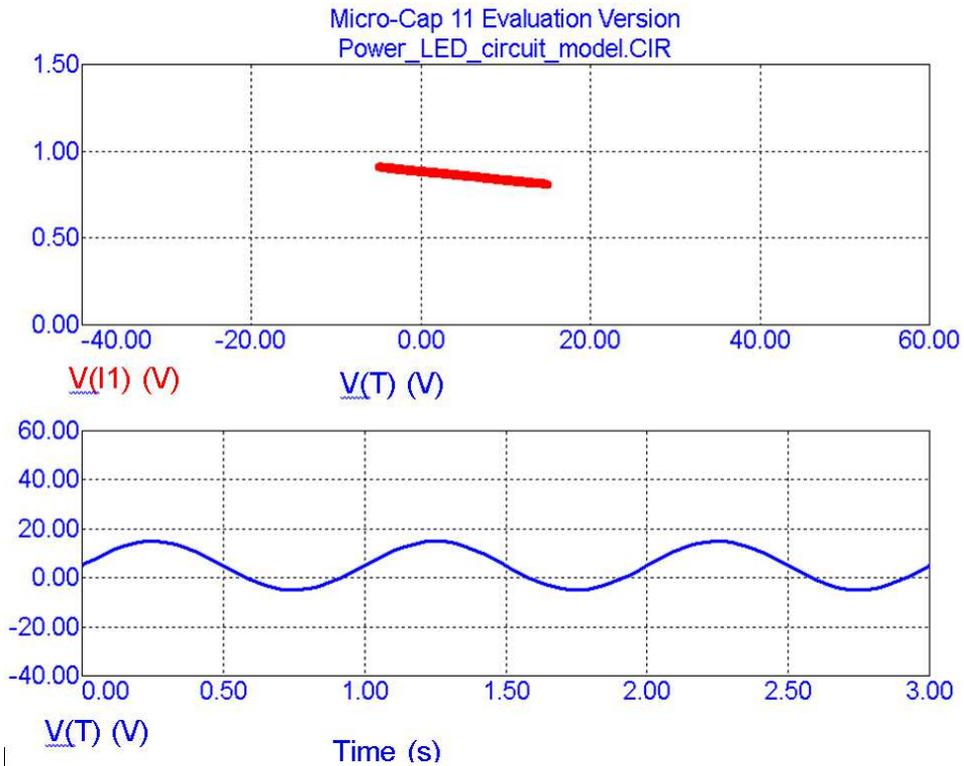


Fig. 15 Transient analysis in case of AC temperature excitation (symbol  $T$  is time)

## 5. Conclusion

This article described the creation of a brand new algorithm for modelling high power LED modules using double optimal polynomial approximation. Inputs to the model were measured module temperature profiles, i.e. parametric dependencies of illumination on temperature, where the parameter was electric current. Other inputs were the user-defined temperature, electrical current and so called Scale Factor, ensuring normalisation of the input data. The model output is the calculated illumination and optimal approximation polynomial. The model was implemented and tested in MATLAB&Simulink environment and also in Micro-Cap environment representing a category of so called Spice compatible circuit simulators. Approximation errors were evaluated by the well-known function of Euclidean norm of residues. Multiple practical experiments and simulations proved the high accuracy and usability of this model's outputs.

## Acknowledgements

The work presented in this paper has been supported by research program of Ministry of Defence, VÝZKUM FVT (DZRO K-217 and K-215), Czech Republic. MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc.

## References

- [1] WINDER, S. *Power Supplies for LED Driving*. Elsevier, 2008, 235 p.
- [2] KHAN, N.M. *Understanding LED Illumination*. CRC Press, 2014, 225 p.
- [3] LENK, R. and LENK, C. *Practical Lighting Design with LEDs*. IEEE Press, 2011, 235 p.
- [4] Ultralow-Power LED-Enabled On-Chip Optical Communication Designed in the III-Nitride and Silicon CMOS Process Integrated Platform. *IEEE Design & Test of Computers*, 2014, vol. 31, no. 1, p. 36-45.
- [5] YUNG, K.C., SUN, B. and JIANG, X. Prognostic-based Qualification of High-Power White LEDs Using Lévy Process Approach. *Mechanical Systems and Signal Processing*, 2016, vol. 82, p. 206-216.
- [6] JANCZAK, A. *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models. A Block-Oriented Approach*. Springer, 2009, 199 p.
- [7] COMINETTI, R., FACCHINEI, F. and LASSERRE, B. J. *Modern Optimization Modelling Techniques*. Springer, 2012, 267 p.
- [8] ZHENING, L., SIMAI, H. and SHUSHONG, Z. *Approximation Methods for Polynomial Optimization. Models, Algorithms and Applications*. Springer, 2012, 125 p.
- [9] BOGUSLAVSKIY, A.J. *Dynamic Systems Models. New Methods of Parametric and State Estimation*. Springer Basel, 2016, 201 p.
- [10] ZAPLATÍLEK, K. and LEUCHTER, J. Optimal Polynomial Approximation of Photovoltaic Panel Characteristics Using a Stochastic Approach. *Advances of Military Technology*, 2013, vol. 8, no. 2, p. 43-51.
- [11] ZAPLATÍLEK, K. and LEUCHTER, J. Fuel Cell 3-D Modelling Using a Logarithmic Approximation in MATLAB®&Simulink®. *Advances in Military Technology*, 2016, vol. 11, no. 1, p. 53-62.
- [12] *Vötsch industrial technique* [on line]. Wien, Austria [cited 2017-01-01]. Available from: <<http://www.v-it.com>>.
- [13] *Spectrum Software* [on line]. Sunnyvale, CA USA [cited 2017-01-01]. Available from: <<http://www.spectrum-soft.com>>.