

Article

Index Matrices—Based Software Implementation of Power Electronic Circuit Design

Nikolay Hinov ^{1,*}, Polya Gocheva ² and Valeri Gochev ³¹ Department of Power Electronics, Technical University of Sofia, 1756 Sofia, Bulgaria² Doctoral School, Technical University of Sofia, 1756 Sofia, Bulgaria; poly.gocheva@gmail.com³ Department of Telecommunications, University of Telecommunications and Posts, 1700 Sofia, Bulgaria; valeripetrovgochev@gmail.com

* Correspondence: hinov@tu-sofia.bg

Abstract: This article aims to present a comprehensive approach with corresponding software that uses index matrices. They have been developed by the authors for an automated model-based design. One of the main goals of this paper is to propose a simple model solving technique for powering electronic devices (particularly, for all subsystems: power supplies, static power converters, electric filters, electrical loads, control systems, etc.). Index matrices, which contain first-order discrete dynamical system parameters, with real numbers as elements, are constructed in order so that voltages, amperages, and discrete values can be calculated. Simulations on a three-phase converter, as well as simulations on buck and boost DC-DC converters with PI controllers, are presented. The function of the proposed software (with examples on the aforementioned devices) is considered, and diagrams of its basic programming classes are shown. The latter draws electronic schemes and their respective graphics, and provides important characteristics. Simulink is used to verify results. Advantages of the proposed approach are a higher speed of calculations (compared to Simulink, due to a lack of differential equations) and a simpler handling of various electronic components. Additionally, a computational scalability is demonstrated.



Citation: Hinov, N.; Gocheva, P.; Gochev, V. Index Matrices—Based Software Implementation of Power Electronic Circuit Design. *Electronics* **2022**, *11*, 675. <https://doi.org/10.3390/electronics11050675>

Academic Editors: Bor-Ren Lin and Anna Richelli

Received: 24 December 2021

Accepted: 20 February 2022

Published: 22 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: index matrices; power electronics; model-based design; software; modeling

1. Introduction

Techniques of solving differential equations solving are widely applied in software for model-based design of real-life systems. For example, power electronic components and devices are modeled in the libraries of one of the most popular graphical programming environments, Simulink, which uses a large number of simulation parameters and model solving modes [1,2]. Theoretically, many classes of dynamical systems can be represented by differential equations.

The first-order discrete dynamical systems have been described by linear systems, which concern consecutive time moments; there exist autonomous (time-independent) simple models (based on constant matrices), in which stability can be easily checked [3]. An example of the application of such a system on a DC-DC boost converter operation is demonstrated in [4]. Software implementations (see [5,6]) are useful with changing dependencies, because many estimations cannot be performed simply in this case; however, generally, power converters can be modeled with straightforward algorithms so that suitable simulations (based on non-constant matrices) can be performed [7–9].

The index matrices have arisen as extensions of the classical matrices. They provide a clear correspondence between elements and indices [10,11]. The index matrices represent a description tool for mathematical and real-life objects. The properties and applications of index matrices have been studied and summarized in [10]. The following types of their elements have been defined: integers, real numbers, Boolean values, predicates, fuzzy values, etc., and additionally, three-dimensional index matrices have been introduced.

Index matrices with real numbers as elements have been used by the authors in power electronic system modeling and simulation [6,12]; for example, single-ended primary-inductor converter example have been considered in [13].

Most of the results in the present paper are obtained by using the authors' .NET software, with its features additionally considered in [5]. It automates the construction and solving of models based on index matrices. The graphics on electronic components (including on their time dependencies) in the presented software are similar to Simulink graphics, but the Simulink models and the proposed index matrix models differ significantly. Differential equations are used in Simulink, in which model configurations are relatively slow processes. The avoidance of the last ones is an important goal of the presented software, because the model-based design techniques, generally, require a large number of simulations (for example, as a part of optimization procedures). Such procedures can be performed faster with the proposed approach (since the linear systems are simpler than the differential equation systems, time-consuming model configurations are reduced).

2. Mathematical Methods and Models

Generally, all dynamical systems are simulated by estimating states at consecutive model time moments. The obtaining of values on system states (which usually are presented by one-dimensional or multidimensional arrays) is known as model solving [1]. No single solving method is optimal for all systems. For example, a typical power conversion system is constituted by subsystems, which have very different levels of detail [2]. Simulink provides a set of solvers, each of which uses a specific technique. According to the way in which the step size is calculated, they are divided into fixed-step and variable-step ones. The fixed-step solvers estimate values at regular model time intervals from the beginning to the end of the simulation. The step size can be specified, and it can be additionally chosen automatically. Generally, decreasing the step size increases the accuracy. The variable-step solvers vary the step size during the simulation. They reduce the step size to increase accuracy when model states are changing rapidly and increase the step size when model states are changing slowly. The approach, which is proposed in the present paper, uses a constant step size.

The apparatus of the indexed matrices (IMs) is not very popular. The respective concept was introduced in 1984 [14]. IMs have been used for a long time as an auxiliary tool for describing transitions of generalized nets only; their values are Boolean and predicate in case of conditions on token passes, and in integers in case of arc capacities [15–17]. Generalized nets are present in areas where modeling with Petri nets is not typical (for example, medicine and biotechnology). Indeed, it is impossible for generalized nets developed for their purposes to be replaced by Petri nets; the difference in the possibilities provided by the two formalisms is especially clear—the classes of tasks to be solved are completely different. The latter fact is also due to the use of IMs. Although basic properties of IMs have been studied in [10,11], a full review on their application is unpublished. Their usage in modeling of electrical circuits is very rare (an example is given in [18]), and their implementation requires additional training and the development of process automation software [19–25].

2.1. Basic Definitions on Index Matrices with Real Number Elements

Proper indices, which determine objects and features, cannot be included in classical matrices; usually respective identifiers are associated to columns of classical matrices. Since the classical matrices of different sizes cannot be summed, the index matrices operation ensures more compact representation. In similar way, there are no limitations on size of IMs in the case of their multiplication.

Generally, each *index matrix with real numbers as elements (R-IM)* has form

$$\bar{A} = \left\langle K, L, \{a_{k,l}\}_{k \in K, l \in L} \subset \mathbb{R} \right\rangle, \quad (1)$$

where

$$K = \{k_1, k_2, \dots, k_m\}, \quad L = \{l_1, l_2, \dots, l_n\}, \tag{2}$$

are finite sets of *indices* [10,11]. In the present paper the following notation on the *first* and the *second index sets* of \bar{A} (K and L respectively) is used:

$$I_1(\bar{A}) = K, \quad I_2(\bar{A}) = L. \tag{3}$$

The usual graphical representation of the above R-IM is

$$\bar{A} = \begin{array}{c|cccc} & l_1 & l_2 & \cdots & l_n \\ \hline k_1 & a_{k_1,l_1} & a_{k_1,l_2} & \cdots & a_{k_1,l_n} \\ k_2 & a_{k_2,l_1} & a_{k_2,l_2} & \cdots & a_{k_2,l_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_m & a_{k_m,l_1} & a_{k_m,l_2} & \cdots & a_{k_m,l_n} \end{array}. \tag{4}$$

Here and below, all R-IMs are denoted by capital letters with additional lines (in order to be deferred from the classical matrices), and their elements by respective small letters. Each element of R-IM \bar{A} is in the same row with its index from $I_1(\bar{A})$, and it is in the same column with its index from $I_2(\bar{A})$. Lines divide indices from elements. Additionally, in (4), the considered indices are included as lower indices of elements of \bar{A} . In all classical matrices, such lower indices are formed by row and column numbers.

Any swap of rows (or columns) of a classical matrix, which participates in a matrix equation, must correspond to a swap of rows (or columns) in other matrices from this equation in order that dependencies can be saved. Conversely, swaps of rows (or columns) in any index matrix do not change. In the context of the present paper, a R-IM on a resistor with resistance R can be represented by

$$\bar{F}_R = \begin{array}{c|cc} & u_R^{cu} & i_R^{cu} \\ \hline c_R & | 1 & -R \end{array} \equiv \begin{array}{c|cc} & i_R^{cu} & u_R^{cu} \\ \hline c_R & | -R & 1 \end{array}. \tag{5}$$

where u_R^{cu} and i_R^{cu} point to current voltage and current amperage respectively on the resistor. Let another R-IM

$$\bar{B} = \langle P, Q, \{b_{p,q}\}_{p \in P, q \in Q} \rangle \tag{6}$$

be given. *Sum* (in index matrix theory the summation is denoted by “ \oplus ”)

$$\bar{C} = \bar{A} \oplus \bar{B} = [K \cup P, L \cup Q, \{c_{r,s}\}_{r \in K \cup P, s \in L \cup Q}], \tag{7}$$

$$c_{r,s} = \begin{cases} a_{r,s}, & \text{if } r \in K \wedge s \in L - Q \vee r \in K - P \wedge s \in L; \\ b_{r,s}, & \text{if } r \in P \wedge s \in Q - L \vee r \in P - K \wedge s \in Q; \\ a_{r,s} + b_{r,s}, & \text{if } r \in K \cap P \wedge s \in L \cap Q; \\ 0, & \text{otherwise;} \end{cases} \tag{8}$$

is a R-IM too. A sum of two index matrices, which have form (5), is presented below:

$$\begin{array}{c|cc} & u_{R_1}^{cu} & i_{R_1}^{cu} \\ \hline c_{R_1} & | 1 & -R_1 \end{array} \oplus \begin{array}{c|cc} & u_{R_2}^{cu} & i_{R_2}^{cu} \\ \hline c_{R_2} & | 1 & -R_2 \end{array} = \begin{array}{c|cccc} & u_{R_1}^{cu} & i_{R_1}^{cu} & u_{R_2}^{cu} & i_{R_2}^{cu} \\ \hline c_{R_1} & | 1 & -R_1 & 0 & 0 \\ c_{R_2} & | 0 & 0 & 1 & -R_2 \end{array}. \tag{9}$$

Since classical matrices of different sizes cannot be summed, it is obvious that in the case of their usage in matrix equations such compact expressions cannot be obtained. The following example supports the last assertion:

$$\begin{pmatrix} 1 & -R_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -R_2 \end{pmatrix} = \begin{pmatrix} 1 & -R_1 & 0 & 0 \\ 0 & 0 & 1 & -R_2 \end{pmatrix}. \tag{10}$$

Indices, which point to resistors, and voltages and amperages, cannot be included in (10); usually, identifiers are associated to the columns in such matrix equations.

In [10,11] a *product* of R-IMs (their multiplication is denoted by “ \odot ”) is defined. In case of index matrices \bar{A} and \bar{B} :

$$\bar{D} = \bar{A} \odot \bar{B} = \langle K \cup (P - L), Q \cup (L - P), \{d_{r,s}\}_{r \in K \cup (P-L), s \in Q \cup (L-P)} \rangle, \tag{11}$$

$$d_{r,s} = \begin{cases} a_{r,s}, & \text{if } r \in K \wedge s \in L - P - Q; \\ b_{r,s}, & \text{if } r \in P - K - L \wedge s \in Q; \\ \sum_{t \in L \cap P} a_{r,t} b_{t,s}, & \text{if } r \in K \wedge s \in Q; \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The last sum has a well-known analog in classical matrices theory; in case of $L \equiv P$ such sums are calculated for all elements of $\bar{A} \odot \bar{B}$ (see (10)), i.e.,

$$L \equiv P \Rightarrow \left(\bar{D} = \bar{A} \odot \bar{B} = \langle K, Q, \{d_{r,s}\}_{r \in K, s \in Q} \rangle, d_{r,s} = \sum_{t \in L \equiv P} a_{r,t} b_{t,s} \right). \tag{13}$$

For each multiplication in the present paper, the second index set of its first R-IM is a subset of the first index set of its second R-IM, and the second index set of its second R-IM has one element, denoted by o , only. In case of \bar{A} and \bar{B} it means that $L \subseteq P$ and $Q = \{o\}$; the following statement holds:

$$(L \equiv P \wedge Q = \{o\}) \Rightarrow \left(\bar{D} = \bar{A} \odot \bar{B} = \langle K \cup (P - L), \{o\}, \{d_{r,o}\}_{r \in K \cup (P-L)} \rangle, \tag{14}$$

$$d_{r,o} = \begin{cases} b_{r,o}, & \text{if } r \notin K; \\ \sum_{t \in L} a_{r,t} b_{t,o}, & \text{otherwise.} \end{cases} \tag{15}$$

An example is given below:

$$\begin{array}{c} \frac{-}{c_R} \left| \begin{array}{c} u_R^{cu} \\ 1 \end{array} \right. \frac{i_R^{cu}}{-R} \odot \frac{-}{u_R^{cu}} \left| \begin{array}{c} o \\ u_R(t) \\ i_R(t) \\ u_E(t) \end{array} \right. = \frac{-}{c_R} \left| \begin{array}{c} o \\ u_R(t) - R i_R(t) \\ u_E(t) \end{array} \right. \tag{16} \end{array}$$

The elements of the first R-IM are coefficients of linear dependency; the elements of the second R-IM—voltage or amperage values; and the elements of the third R-IM—resultant values. In the third R-IM, one element is a linear combination and one element (which can be considered as constant voltage in ideal DC voltage source) remains unchanged (index u_E^{cu} is not presented in set $I_2(\bar{F}_R)$).

2.2. Classical Matrix Models on First-Order Discrete Dynamical Systems

In [3] a time-independent first-order discrete dynamical system is defined by

$$X(t + \Delta t) = AX(t) + B, \quad X(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_n(t)]^T, \tag{17}$$

where matrices A and B are constant, and

$$t \in \Theta \equiv \{ t_0, t_0 + \Delta t, t_0 + 2\Delta t, \dots \}, t_0, \Delta t \in \mathbb{R}, \Delta t = \frac{1}{F_d} > 0. \tag{18}$$

Here time set Θ has sampling step Δt and discretization frequency F_d . State vector $X(t + \Delta t)$ depend on its previous one $X(t)$ only. The matrix equation from (17) is replaced by its equivalent one below:

$$C X(t + \Delta t) + D X(t) + E = O, A = -(C)^{-1} D, B = -(C)^{-1} E, \tag{19}$$

where O is a zero matrix, and C is non-singular. It is obvious that the next two matrix equations are equivalent to those ones in (17) and (19):

$$X(t + \Delta t) = A' X'(t), A' = \begin{bmatrix} A & B \end{bmatrix}, X'(t) = \begin{bmatrix} X(t) \\ 1 \end{bmatrix}; \tag{20}$$

$$F X''(t + \Delta t) = O, F = \begin{bmatrix} C & D & E \end{bmatrix}, X''(t + \Delta t) = \begin{bmatrix} X(t + \Delta t) \\ X(t) \\ 1 \end{bmatrix}. \tag{21}$$

In the next section, R-IM analogs of (20) and (21) model proper first-order discrete dynamical systems.

2.3. R-IM Models of Electronic Components, Circuits and Devices

Let an electronic device be modelled and be denoted by D, and let M_D be a finite set of unique indices associated to its components. It is obvious that various divisions can be performed in non-trivial devices, and appropriate hierarchies, which include sub-devices, can be defined. In [12] a two-terminal electronic component $c_m, m \in M_D$, is modelled by R-IM $\bar{F}_m(t)$ so that a relation between its voltages $u_m(t), u_m(t - \Delta t)$ and its amperages $i_m(t), i_m(t - \Delta t)$ at consecutive time moments can be obtained (respective indices $u_m^{cu}, u_m^{pr}, i_m^{cu}$ and i_m^{pr} , which do not depend on time, are used); here $t \in \Theta - t_0$. This relation is given by

$$\bar{F}_m(t) \odot \bar{X}''_m(t) = \bar{O}, I_2(\bar{F}_m(t)) \equiv I_1(\bar{X}''_m(t)). \tag{22}$$

Both values and their respective indices are presented in $\bar{X}''_m(t)$ (see (24)). Here and below \bar{O} denotes a zero R-IM (all its elements are zeros). Examples on basic components are given in [12] and below.

Let an electronic component, denoted by c_L , be an inductor with inductance L. Approximation

$$u_L(t) = L \frac{i_L(t) - i_L(t - \Delta t)}{\Delta t} \tag{23}$$

is a consequence of

$$\bar{F}_L \odot \begin{array}{c|c} \bar{O} & \\ \hline u_L^{cu} & u_L(t) \\ i_L^{cu} & i_L(t) \\ i_L^{pr} & i_L(t - \Delta t) \end{array} = O, \bar{F}_L = \begin{array}{c|ccc} & u_L^{cu} & i_L^{cu} & i_L^{pr} \\ \hline c_L & 1 & -\frac{L}{\Delta t} & \frac{L}{\Delta t} \end{array} \tag{24}$$

Similarly, for a capacitor, denoted by c_C , with capacitance C,

$$\bar{F}_C = \begin{array}{c|ccc} & u_C^{cu} & i_C^{cu} & u_C^{pr} \\ \hline c_C & -\frac{C}{\Delta t} & 1 & \frac{C}{\Delta t} \end{array} \tag{25}$$

gives approximation

$$i_C(t) = C \frac{u_C(t) - u_C(t - \Delta t)}{\Delta t}. \tag{26}$$

It is obvious that (5) models a resistor, denoted by c_R , with resistance R . For an ideal DC voltage source, denoted by c_E , constant voltage E is determined by

$$\bar{F}_E \odot \begin{matrix} - & | & o \\ u_E^{cu} & | & u_E(t) \\ o & | & 1 \end{matrix} = O, \quad \bar{F}_E = \begin{matrix} - & | & o \\ c_E & | & 1 \\ & & E \end{matrix} \cdot \quad (27)$$

The upper parameters L, C, R and E , do not depend on time. However, their time dependences can be presented instead of constants. Let the component, denoted by c_S , be an ideal electronic switch; a joint representation of its two states is

$$\bar{F}_S(t) = \begin{matrix} - & | & u_S^{cu} \\ c_S & | & s_S(t) \\ & & 1 - s_S(t) \end{matrix}, \quad (28)$$

where $s_S : \Theta \rightarrow \{0;1\}$ is a switching function. According to (28), in the closed state of the switch $u_S(t) = 0$, and in its opened state $i_S(t) = 0$.

Nodes are modeled by IMs below according to Kirchhoff’s current law in a similar manner. Each component index in the left side of the next three equations denotes an electronic component which is not presented in previous such equations. For the scheme with two-terminal components from Figure 1 the respective R-IMs are

$$\bar{F}_{n_E} = \begin{matrix} - & | & i_E^{cu} \\ n_E & | & 1 \\ & & -1 \end{matrix}; \quad \bar{F}_{n_{S_2}} = \begin{matrix} - & | & i_{S_2}^{cu} \\ n_{S_2} & | & 1 \\ & & 1 \end{matrix}; \quad \bar{F}_{n_L} = \begin{matrix} - & | & i_L^{cu} \\ n_L & | & 1 \\ & & -1 \end{matrix}; \quad \bar{F}_{n_C} = \begin{matrix} - & | & i_C^{cu} \\ n_C & | & 1 \\ & & -1 \end{matrix}; \quad \bar{F}_{n_R} = \begin{matrix} - & | & i_R^{cu} \\ n_R & | & 1 \\ & & -1 \end{matrix}. \quad (29)$$

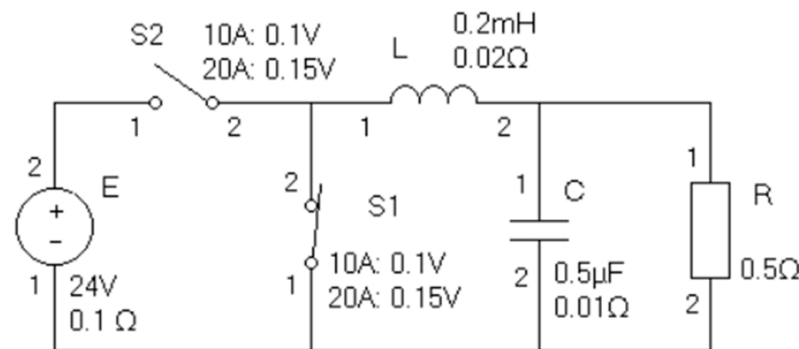


Figure 1. Power electronic circuit of a Buck DC-DC converter.

They model

$$i_E(t) - i_{S_2}(t) = 0; \quad i_{S_2}(t) + i_{S_1}(t) - i_L(t) = 0; \quad i_L(t) - i_C(t) - i_R(t) = 0. \quad (30)$$

The R-IM on the fourth node is omitted, because it gives a linear combination of R-IMs from (29); all elements of $M_{Circ} = \{“E”; “S_2”; “S_1”; “L”; “C”; “R”\}$ are represented in these three R-IMs (the circuit is denoted by “Circ”). For each component, the positive direction of voltage and amperage is from its first to its second terminal, and it determines respective signs in the index matrices from (29) and (31).

Kirchhoff’s voltage law can be applied through index matrices too. For the scheme from Figure 1, respective R-IMs on three loops are

$$\bar{F}_{p_R} = \begin{matrix} - & | & u_R^{cu} \\ p_R & | & 1 \\ & & -1 \end{matrix}; \quad \bar{F}_{p_E} = \begin{matrix} - & | & u_E^{cu} \\ p_E & | & 1 \\ & & 1 \end{matrix}; \quad \bar{F}_{p_L} = \begin{matrix} - & | & u_L^{cu} \\ p_L & | & 1 \\ & & 1 \end{matrix}. \quad (31)$$

They model

$$u_R(t) - u_C(t) = 0; \quad u_E(t) + u_{S_2}(t) - u_{S_1}(t) = 0; \quad u_L(t) + u_C(t) + u_{S_1}(t) = 0. \quad (32)$$

For the considered circuit, denoted by “Circ”, statements

$$\bar{F}_{Circ}(t) = \left(\bigoplus_{m \in M_{Circ}} \bar{F}_m(t) \right) \oplus \left(\bigoplus_{n \in N_{Circ}} \bar{F}_n \right) \oplus \left(\bigoplus_{p \in P_{Circ}} \bar{F}_p \right); \quad (33)$$

$$\bigoplus_{m \in M_{Circ}} \bar{F}_m(t) = \bar{F}_E(t) \oplus \bar{F}_{S_1}(t) \oplus \bar{F}_{S_2}(t) \oplus \bar{F}_L \oplus \bar{F}_C \oplus \bar{F}_R; \quad (34)$$

$$\bigoplus_{n \in N_{Circ}} \bar{F}_n = \bar{F}_{n_E} \oplus \bar{F}_{n_{S_2}} \oplus \bar{F}_{n_L}; \quad \bigoplus_{p \in P_{Circ}} \bar{F}_p = \bar{F}_{p_R} \oplus \bar{F}_{p_E} \oplus \bar{F}_{p_L}. \quad (35)$$

combine models of components, nodes, and loops. Sparse R-IM $\bar{F}_{Circ}(t)$ contains 12 columns, which have indices with subscript “cu”, and 12 rows. It is a part of a bigger model in the present paper. Resistances are associated to the inductor, the capacitor, the DC voltage source and the switches (voltages on closed switches are juxtaposed to amperages), and therefore \bar{F}_L , \bar{F}_C , \bar{F}_E , $\bar{F}_{S_1}(t)$ and $\bar{F}_{S_2}(t)$ are modified. Devices, which are connected to the upper circuit, are depicted in Figure 2 and described below. Their circuits are not considered; only important inputs and outputs are taken into account and shown graphically; voltages on all devices are denoted in the figures. The digital signals are denoted by “s”. The considered buck converter uses five blocks: a constant voltage block, a subtraction block, a proportional-integral (PI) controller, a pulse width modulation (PWM) block, and a logical NOT gate.

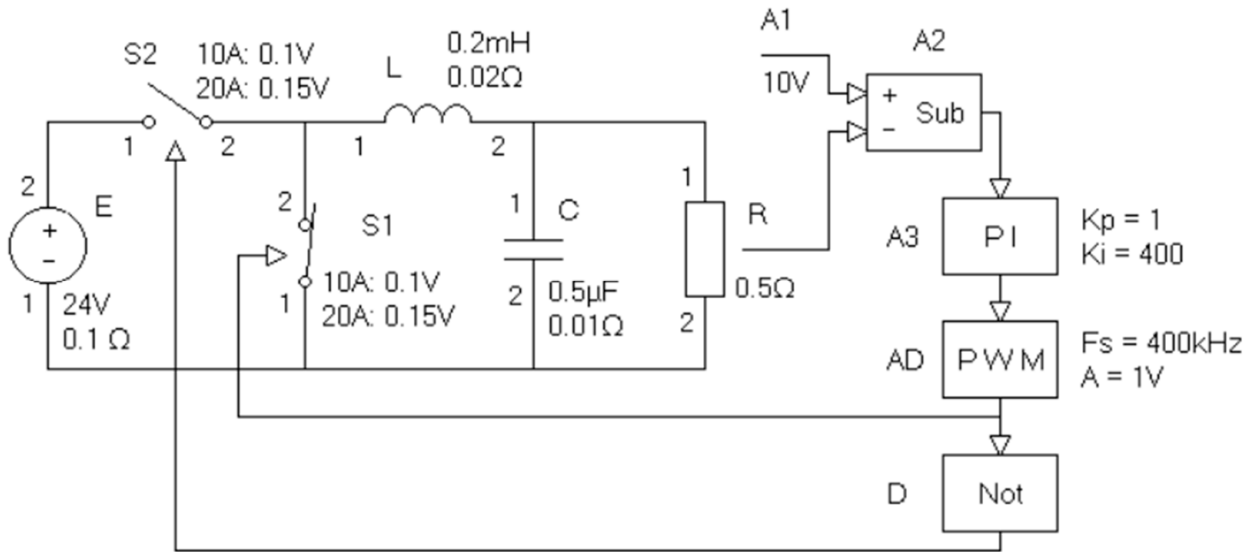


Figure 2. Buck DC-DC converter using proportional-integral (PI) controller and pulse width modulation (PWM).

The constant voltage block is represented by

$$\bar{F}_{A1} = \frac{1}{c_{A1}} \begin{vmatrix} u_{A1}^{cu} \\ -1 \end{vmatrix} \frac{u_{A1,1}^{cu}}{U}. \quad (36)$$

All input and output signals are marked by the respective component names, and the input ones are numbered. In the shown example $U = 10V$.

The subtraction block is represented by

$$\bar{F}_{A2} = \frac{1}{c_{A2}} \begin{vmatrix} u_{A2}^{cu} \\ -1 \end{vmatrix} \frac{u_{A2,1}^{cu}}{1} \frac{u_{A2,2}^{cu}}{-1}. \quad (37)$$

The proportional-integral (PI) controller is modeled by

$$\bar{F}_{A3} = \frac{1}{c_{A3}} \begin{vmatrix} u_{A3}^{cu} & u_{A3}^{pr} & u_{A3,1}^{cu} & u_{A3,1}^{pr} \\ -\frac{1}{\Delta t} & \frac{1}{\Delta t} & \frac{K_p}{\Delta t} + K_i & -\frac{K_p}{\Delta t} \end{vmatrix}. \tag{38}$$

The last index matrix approximates

$$\frac{du_{A3}(t)}{dt} = K_p \frac{du_{A3,1}(t)}{dt} + K_i u_{A3,1}(t), \tag{39}$$

which is obtained from basic definition

$$u_{A3}(t) = K_p u_{A3,1}(t) + K_i \int_0^t u_{A3,1}(\theta) d\theta \tag{40}$$

via differentiation. Proper values of K_p and K_i are shown in Figure 2.

The PWM block is modeled by

$$\bar{F}_{AD}(t) = \begin{cases} \frac{1}{c_{AD}} \begin{vmatrix} s_{AD}^{cu} & 0 \\ -1 & 1 \end{vmatrix}, & \text{if } u_{AD,1}(t_s) > u_{Triang}(t - \Delta t); \\ \frac{1}{c_{AD}} \begin{vmatrix} s_{AD}^{cu} \\ -1 \end{vmatrix}, & \text{otherwise;} \end{cases} \tag{41}$$

where

$$t_s = \{ 0, \pm\Delta t_s, \pm 2\Delta t_s, \pm 3\Delta t_s, \dots \} \cap (t - \Delta t - \Delta t_s, t - \Delta t]. \tag{42}$$

Here the signal

$$u_{Triang} : \Theta \rightarrow [0, A] \tag{43}$$

has a triangular form, period $\Delta t_s = \frac{1}{F_s}$ and amplitude A . The output of the PI controller is checked with *switching* frequency F_s . The voltage on the output of the PI controller and the triangular function forms a *duty cycle*. The last two index matrices represent alternatives $s_{AD}(t) = 1$ and $s_{AD}(t) = 0$, i.e., they form a digital signal. Proper values of F_s and A are shown in Figure 2.

Finally, the logical NOT gate is modeled by

$$\bar{F}_D = \frac{1}{c_D} \begin{vmatrix} s_D^{cu} & 0 \\ -1 & -1 \end{vmatrix}, \tag{44}$$

Values $s_{Inv,1}(t)$ and $s_{Inv}(t)$ belong to set $\{0, 1\}$ too.

All junctions of the considered five blocks are represented by

$$\begin{aligned} & \frac{1}{p_{A2,1}} \begin{vmatrix} u_{A2,1}^{cu} & u_{A1}^{cu} \\ -1 & 1 \end{vmatrix}; & \frac{1}{p_{A2,2}} \begin{vmatrix} u_{A2,2}^{cu} & u_R^{cu} \\ -1 & 1 \end{vmatrix}; & \frac{1}{p_{A3,1}} \begin{vmatrix} u_{A3,1}^{cu} & u_{A2}^{cu} \\ -1 & 1 \end{vmatrix}; & \frac{1}{p_{AD,1}} \begin{vmatrix} u_{AD,1}^{cu} & u_{A3}^{cu} \\ -1 & 1 \end{vmatrix}; \\ & \frac{1}{p_{D,1}} \begin{vmatrix} s_{D,1}^{cu} & s_{AD}^{cu} \\ -1 & 1 \end{vmatrix}; & \frac{1}{p_{S1,1}} \begin{vmatrix} s_{S1}^{cu} & s_{AD}^{cu} \\ -1 & 1 \end{vmatrix}; & \frac{1}{p_{S2,1}} \begin{vmatrix} s_{S2}^{cu} & s_D^{cu} \\ -1 & 1 \end{vmatrix}. \end{aligned} \tag{45}$$

The last two index matrices determine values of the switching functions.

The controller (denoted by ‘‘Contr’’ below) can be modeled by

$$\bar{F}_{Contr}(t) = (\bar{F}_{A1} \oplus \bar{F}_{A2} \oplus \bar{F}_{A2} \oplus \bar{F}_{AD}(t) \oplus \bar{F}_D) \oplus \bar{F}_{P_{Contr}}, \tag{46}$$

where the last index matrix is calculated as a sum of the R-IMs in (45). A common model (denoted by “Buck Conv”) is

$$\bar{F}_{\text{BuckConv}}(t) = \bar{F}_{\text{Circ}}(t) \oplus \bar{F}_{\text{Contr}}(t). \tag{47}$$

A relation between voltages and amperages can be obtained through R-IM equation

$$\bar{F}_{\text{BuckConv}}(t) \odot \bar{X}''_{\text{BuckConv}}(t) = \bar{O}, \quad I_2(\bar{F}_{\text{BuckConv}}(t)) \equiv I_1(\bar{X}''_{\text{BuckConv}}(t)), \tag{48}$$

where both values and their respective indices are presented in $\bar{X}''_{\text{BuckConv}}(t)$.

The equation in (48) is a proper R-IM analog of the matrix equation in (21), i.e., it represents a first-order discrete dynamical system. Sparse R-IM $\bar{F}_{\text{BuckConv}}(t)$ contains 24 columns, which have indices with subscript “cu”, and 24 rows (there are 12 R-IMs in $\bar{F}_{\text{Contr}}(t)$ with 12 indices with subscript “cu”, which are not presented in $\bar{F}_{\text{Circ}}(t)$). All R-IM models in the present section are automatically simplified (since the index matrices in (45) are trivial, all 24 respective variables are automatically reduced to 15 analog ones and 4 discrete ones), generated and solved by the software, which is developed by the authors (see next section).

3. Software Implementation Based on R-IMs

In [5], programming classes, which implement electronic circuit design techniques, and which are based on R-IMs, are presented. Class diagrams of their recent versions are shown in this section. One of the main purposes is the automation of processes of modeling and simulation. An example of joint modeling and simulation of a power circuit and a PI controller is given. All electronic component images in the present paper, except the ones which use Simulink, are obtained by labeling the authors’ methods as Draw().

Each object of class IndexMatrix represents a R-IM through two lists of indices and a two-dimensional array. Static methods Sum() and Multiplication() implement the operations, which are considered in (7) and (11). The method Solve() transforms a R-IM analog of discrete dynamical system model (21) (see (48)) to a R-IM analog of equation (20) in order that state variables can be calculated in a step-wise manner at each simulation step. In this process, the auxiliary method Parts() is used in distinguishing current values from previous ones. Class ClassicalMatrices contain static methods, which implement well-known operations over classical matrixes (multiplication, inversion, and calculation of the sub-matrix), which are used in model solving. Class diagrams of IndexMatrix and ClassicalMatrices are presented in Figure 3.

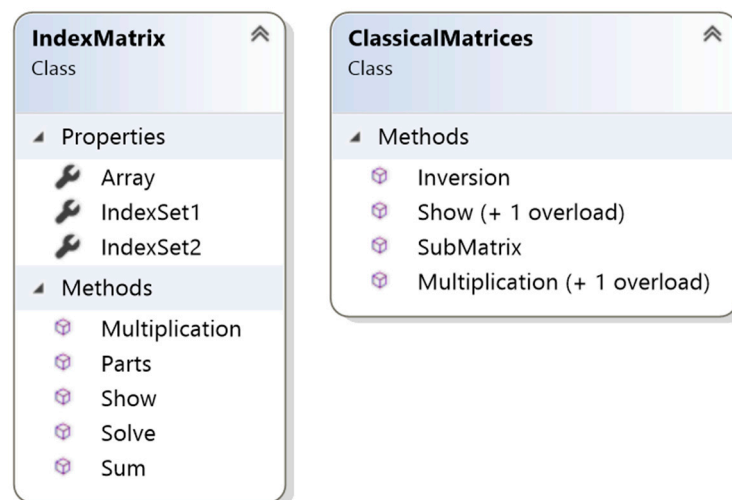


Figure 3. Class diagrams of IndexMatrix and ClassicalMatrices.

Class EC and its subclasses for the considered electronic components are presented in Figure 4. Various specific properties (L, C, R, E, Ki, Kp, etc.) are defined. The method CalculateIM() forms RI-Ms according to the statements, which are given in the paper. The common properties of Position, Rotation and ConnectionPositions are used in drawing (method Draw() is overridden). The members of interfaces, which are used in subclasses of EC, are shown in Figure 4 too.

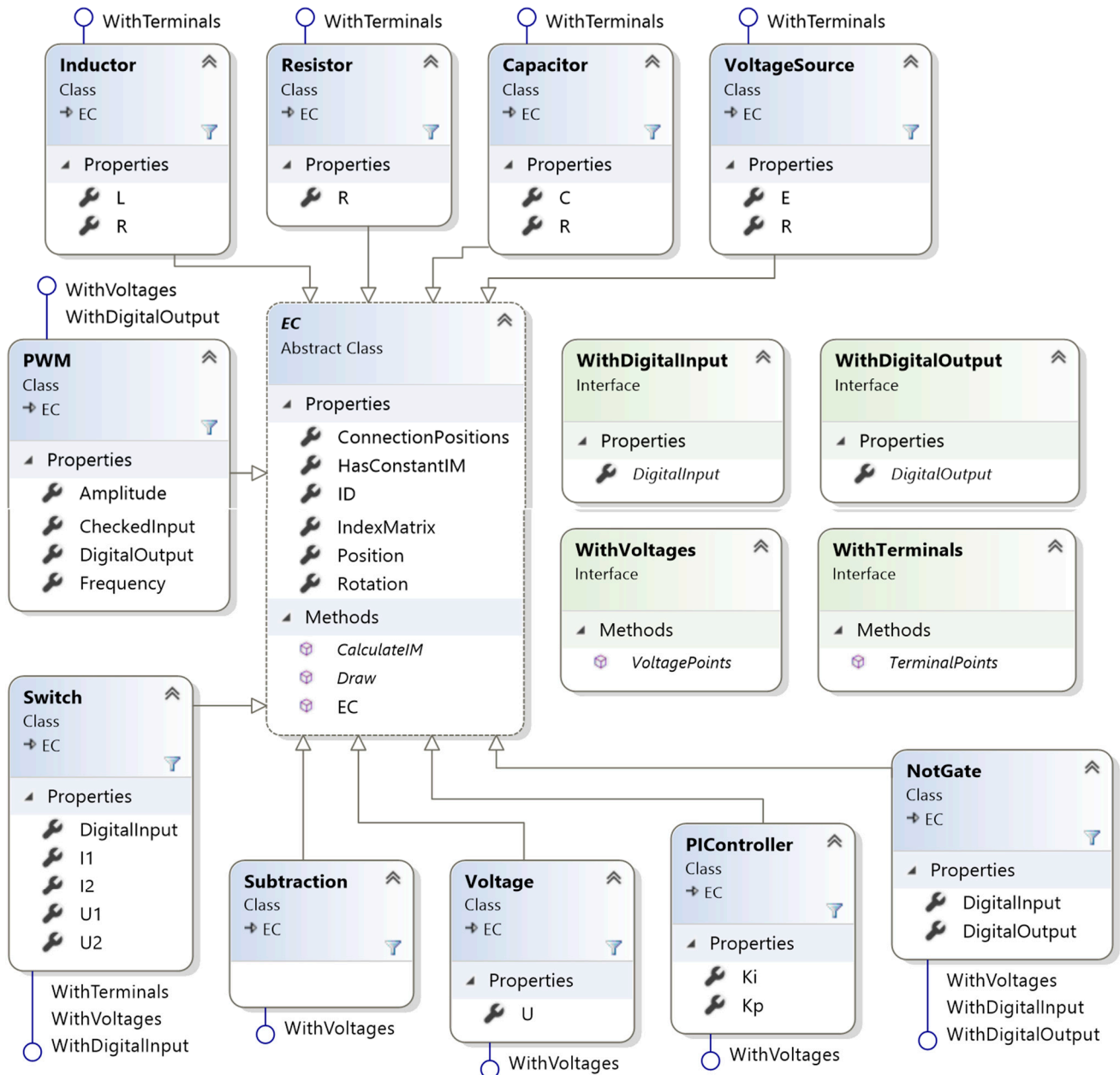


Figure 4. Class diagrams of EC, its subclasses and auxiliary interfaces.

Class diagrams of Circuit and its auxiliary classes are shown in Figure 5. Each object of class Junction represents a connection between two electronic components, and each object of class Node implements a connection (such as the ones modeled in (29)) between two or more components. Finally, class Circuit defines collections of instances of types EC and Junction.

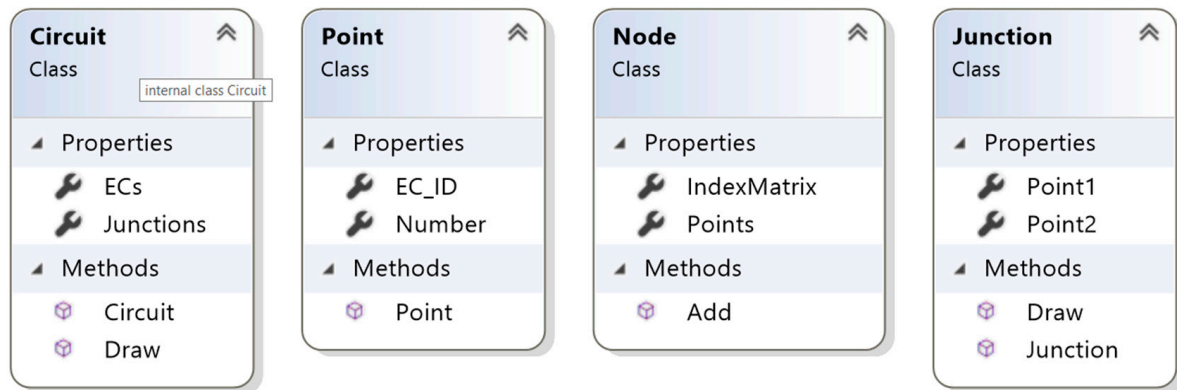


Figure 5. Diagrams of Circuit and its auxiliary classes.

Class Simulation determines in each instance an electronic circuit, a discretization frequency, and a model time interval. Its method Do() invokes private methods in order for proper R-IMs to be constructed and handled; notes on these methods and results of their functioning in case of the considered example are given below:

CalculateConstantIMOnComponents(): performs in field IMAnalog all IMs on objects, which implement interface WithTerminals, and for which property HasConstantIM has value true. A respective view of IMAnalog is given below:

—	u_R^{cu}	i_R^{cu}	u_E^{cu}	i_E^{cu}	o	u_L^{cu}	i_L^{cu}	i_L^{pr}	u_C^{cu}	i_C^{cu}	u_C^{pr}
c_R	1	-0.5	0	0	0	0	0	0	0	0	0
c_E	0	0	1	-0.1	24	0	0	0	0	0	0
c_L	0	0	0	0	0	-1	200.02	-200	0	0	0
c_C	0	0	0	0	0	0	0	0	-1	2.01	1

CalculateIMOnNodes(): represents in IMAnalog a sum of R-IMs on nodes (see (29)). A view of the respective sum is given below:

—	i_E^{cu}	$i_{S_2}^{cu}$	$i_{S_1}^{cu}$	i_L^{cu}	i_C^{cu}	i_R^{cu}
n_E	1	-1	0	0	0	0
n_{S_2}	0	1	1	-1	0	0
n_L	0	0	0	1	-1	-1

CalculateIMOnLoops(): adds in field IMAnalog a sum of R-IMs on loops (see (31)). A view of this sum is given below:

—	u_R^{cu}	u_C^{cu}	u_E^{cu}	$u_{S_2}^{cu}$	$u_{S_1}^{cu}$	u_L^{cu}
p_R	1	-1	0	0	0	0
p_E	0	0	1	1	-1	0
p_L	0	1	0	0	1	1

CalculateConstantIMsOnVoltages(): adds in field IMAnalog a sum of all IMs on objects, which implement interface WithVoltages, do not implement WithDigitalInput and WithDigitalOutput, and for which HasConstantIM has value true (see (36)–(38)). A respective view is given below:

—	$u_{A_1}^{cu}$	o	$u_{A_2}^{cu}$	u_R^{cu}	$u_{A_3}^{cu}$	$u_{A_3}^{pr}$	$u_{A_2}^{pr}$
c_{A_1}	1	-10	0	0	0	0	0
c_{A_2}	1	0	-1	-1	0	0	0
c_{A_3}	0	0	1.004	0	-1	1	-1

Objects of type Junction are used here in order trivial R-IMs from (45) on analog signals to be taken into account in other R-IMs.

CalculateDigitalValues(): performs in the field IMDigital a summation of all IMs on objects, which implement interfaces WithVoltages, WithDigitalInput and WithDigitalOutputs, and for which HasConstantIM has value true (in the considered example there is one such instance of type NotGate).

$$\begin{array}{c|ccc|cc}
 & s_D & s_{AD} & o & s_{S_1} & s_{S_2} \\
 s_D & 1 & 1 & -1 & 0 & 0 \\
 s_{S_1} & 0 & -1 & 0 & 1 & 0 \\
 s_{S_2} & 1 & 0 & 0 & 0 & 1
 \end{array} \quad (53)$$

An invoking of Solve() gives the following result in in field solvedDigital:

$$\begin{array}{c|cc}
 & s_{AD} & o \\
 s_D & -1 & 1 \\
 s_{S_1} & 1 & 0 \\
 s_{S_2} & -1 & 1
 \end{array} \quad (54)$$

The latter two R-IMs determine two solutions, which are presented by

$$\begin{array}{c|c}
 s_{AD} & o \\
 s_D & 1 \\
 s_{S_1} & 0 \\
 s_{S_2} & 1
 \end{array} ; \quad \begin{array}{c|c}
 s_{AD} & o \\
 s_D & 0 \\
 s_{S_1} & 1 \\
 s_{S_2} & 0
 \end{array} \quad (55)$$

CalculateNonConstantIMs(): sums in each element of array iMsAnalog[] IMs on IMAnalog and a sum of all IMs on objects, for which HasConstantIM has the value false. For the described converter, IMs on the switches are added according to the values of sS1 and sS2 in the upper two cases:

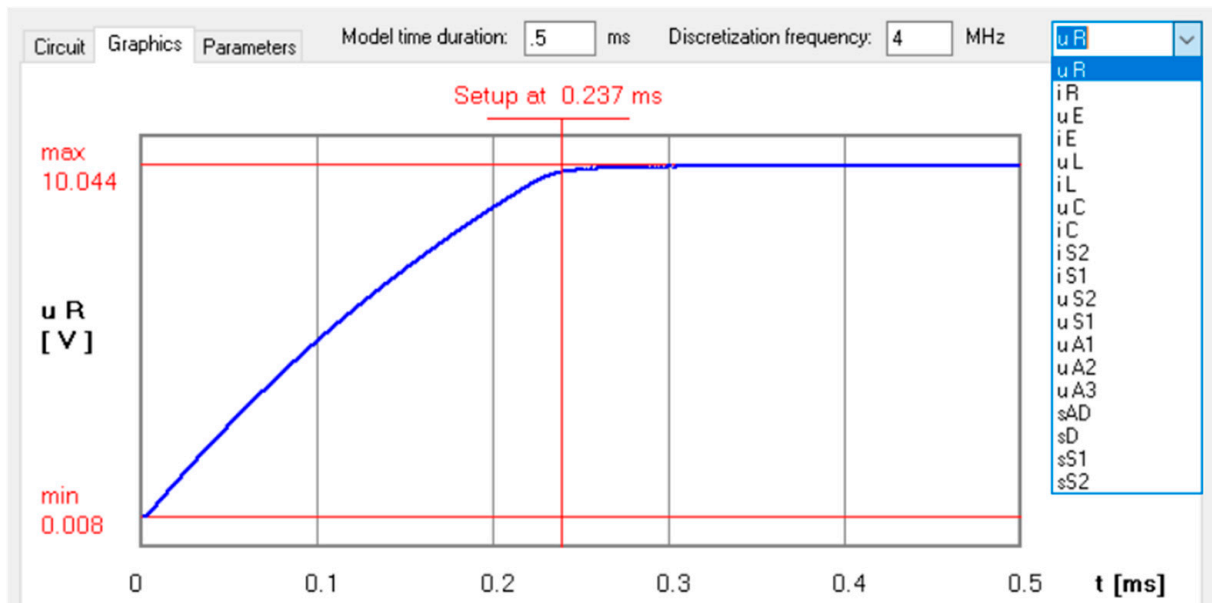
$$\begin{array}{c|cc|cc}
 & u_{S_2}^{cu} & i_{S_2}^{cu} & o & u_{S_1}^{cu} & i_{S_1}^{cu} \\
 c_{S_2} & 1 & -0.05 & -0.05 & 0 & 0 \\
 c_{S_1} & 0 & 0 & 0 & 0 & 1
 \end{array} ; \quad \begin{array}{c|cc|cc}
 & u_{S_2}^{cu} & i_{S_2}^{cu} & o & u_{S_1}^{cu} & i_{S_1}^{cu} \\
 c_{S_2} & 0 & 0 & 0 & 0 & 1 \\
 c_{S_1} & 0 & 0 & -0.05 & 1 & -0.05
 \end{array} \quad (56)$$

In such a way, various R-IMs are constructed at first in iMsAnalog[]. Method Solve() is invoked on its elements in order the values to be calculated in step-wise manner later, and the results of functioning of Solve() are saved in array solvedAnalog[].

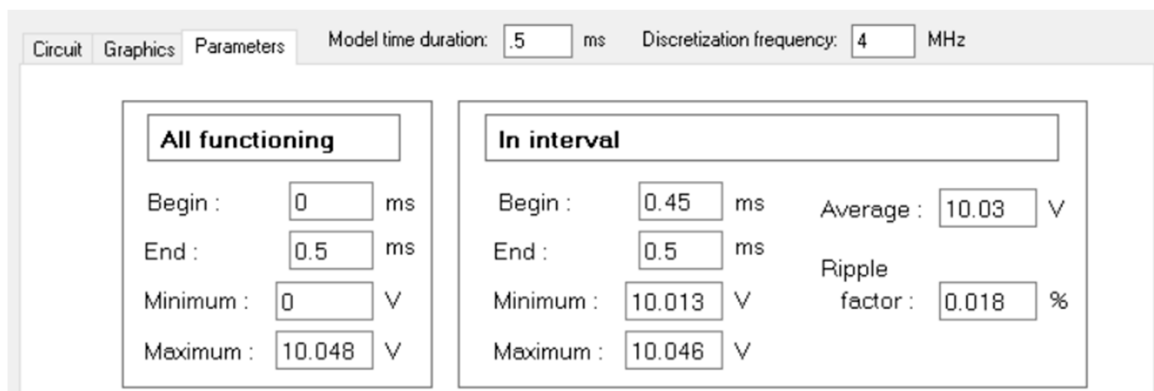
SimulateInModelTime(): calculates for each sampling step non-handled digital values. In the presented example, the signal on the output of the PWM block must be handled (all digital signals depend on it only). The method uses the upper discrete values and elements of solvedAnalog[] in the R-IM analog of state vector estimation (see (17)). For the described converter, all voltages $u_m(t)$ and amperages $i_m(t)$, where $m \in M_{Circ}$, as well as voltages (or respective values "0" and "1") of the components, which are indexed in M_{Contr} , are estimated through their previous ones.

4. Results

Graphics and respective characteristics on the output voltage of the considered converter, which are obtained by authors' software, are presented in [5,6]. Such a graph and the respective characteristics are shown in Figure 6, and similar statistics can be estimated for other values (see the checkbox in this figure).



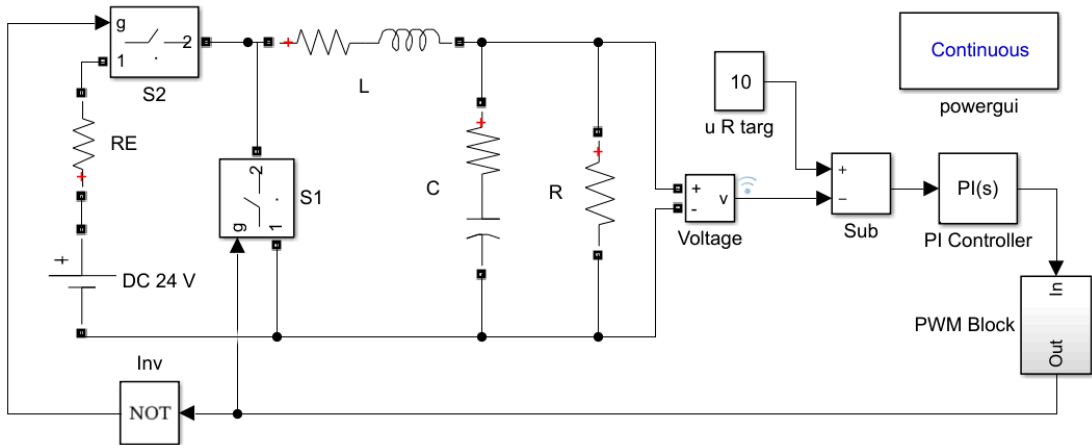
(a)



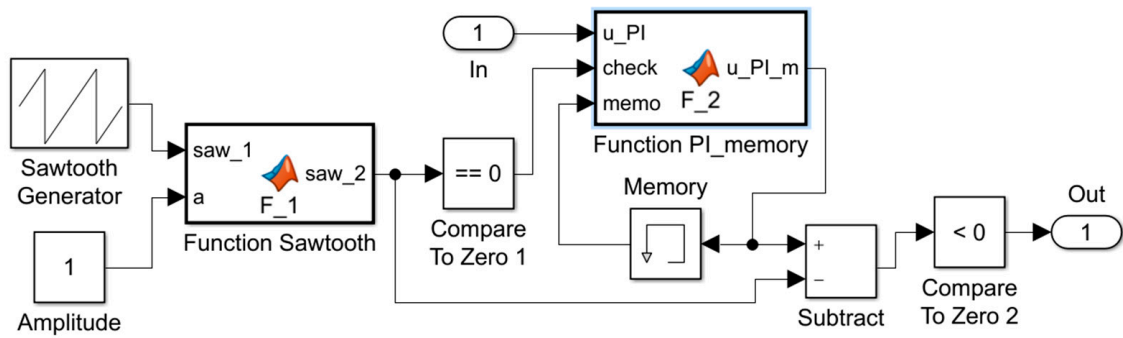
(b)

Figure 6. Output voltage of the buck DC-DC converter from Figure 2: (a) graphics; (b) characteristics.

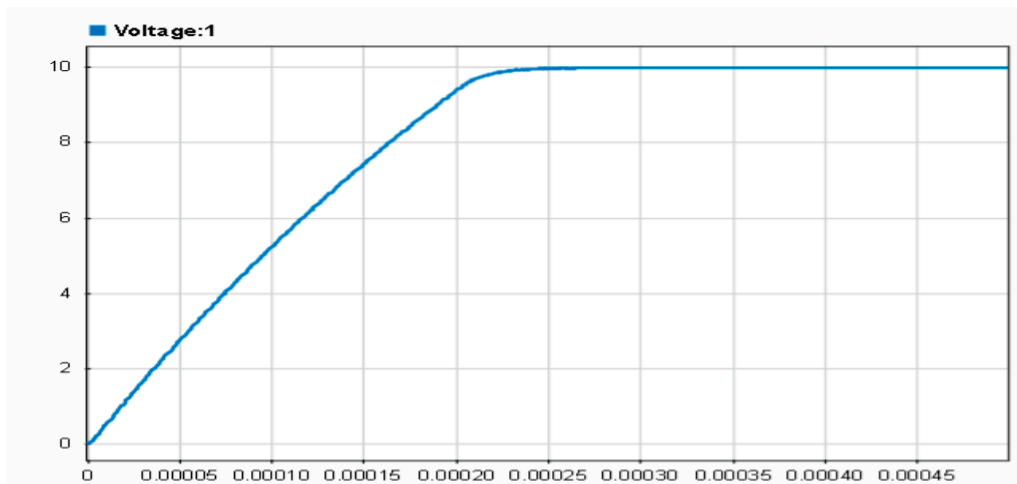
The Simulink model is presented in Figure 7. The same parameters, for example resistance, inductance, capacitance, etc., of the considered buck DC-DC converter are used. The result of its simulation coincides with the result shown in Figure 6. Simulation durations in the authors’ software and Simulink (automatic solver selection is performed in the last one) are presented in Table 1. The same parameters are used in order for step responses to be shown in two ways. Whereas there are many configuration parameters in Simulink, there are only two such ones in the authors’ software: model time duration and discretization frequency (the last one is chosen to be 10 times greater than the switching frequency in the PWM block). Microsoft Visual Studio Community 2017 and MATLAB 2017a are used, and 10,000 and 1000 simulations (in the authors’ software and for the Simulink model, respectively) were performed for each case. All initial values are zero.



(a)



(b)



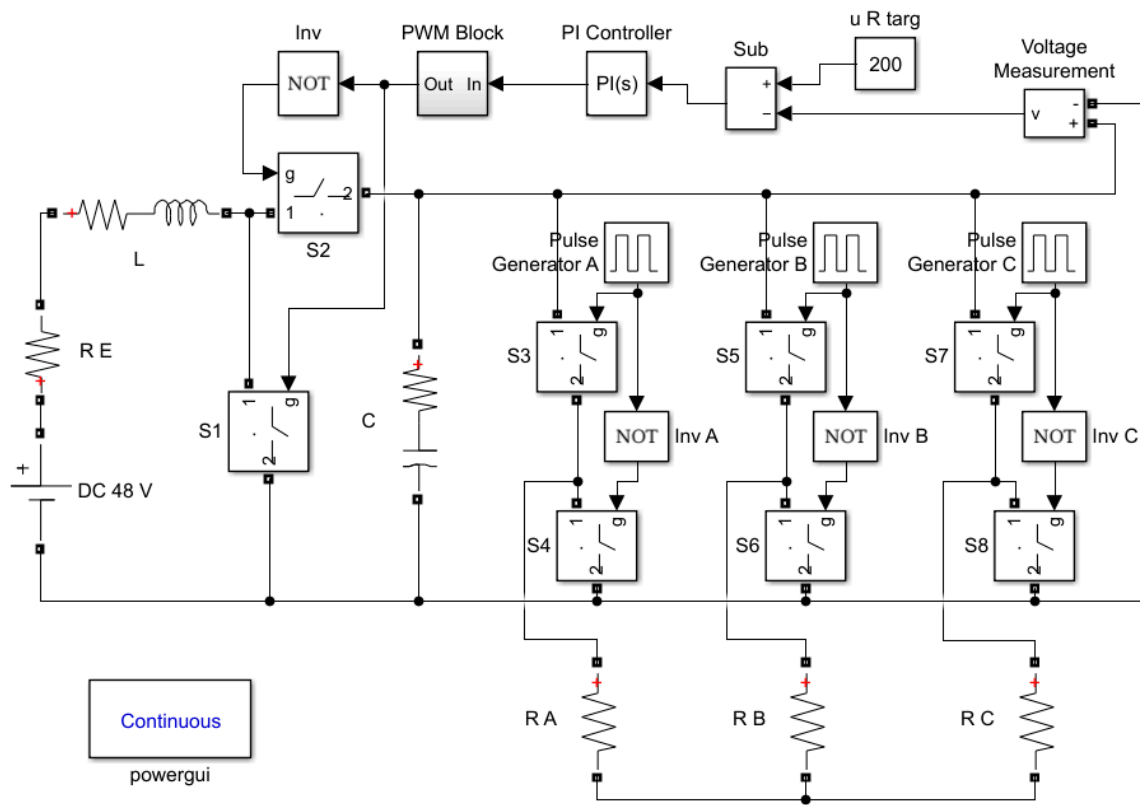
(c)

Figure 7. Simulink model of the buck DC-DC converter from Figure 2: (a) graphical model of the converter; (b) graphical model of the PWM block; (c) output voltage.

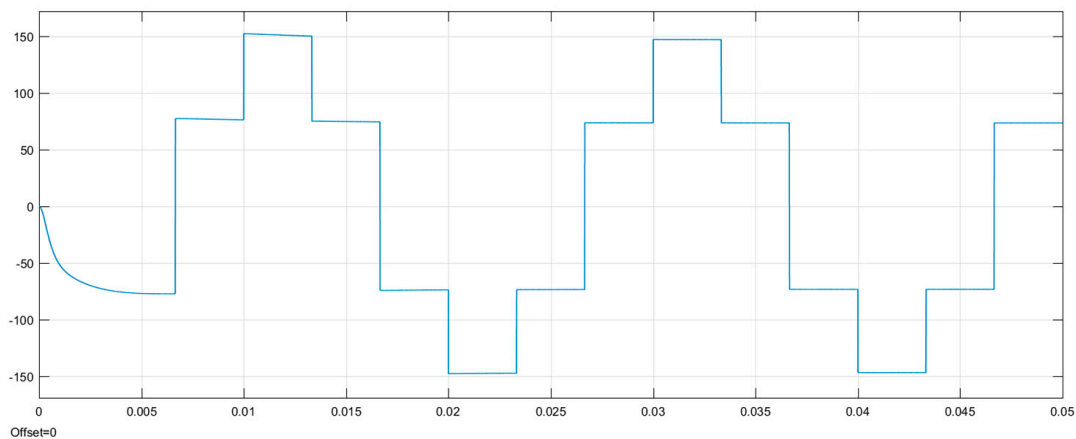
Table 1. Simulation durations for the buck DC-DC converter from Figure 2.

Model time duration [ms]	0.2	0.4	0.6	0.8	1
Simulation duration in Simlink [ms]	510	530	547	567	595
Simulation duration in the authors' software [ms]	18.1	19.9	21.9	23.9	26.1

The Simulink model of a three-phase converter is presented in Figure 8. The scheme contains a boost DC-DC converter, in which the positions of a switch and an inductor are shifted (compared to Figure 7). The devices of the same name have the same functionality; their parameters are shown more clearly in Figure 9, which presents the respective model in the authors' software. Simulation durations in the latter one and Simulink (the same circuit parameters are used) are presented in Table 2. 1000 and 100 simulations (in the authors' software and for a Simulink model, respectively) were performed for each case.



(a)



(b)

Figure 8. Simulink model of a three-phase converter (a) graphical model of the converter; (b) output voltage on phase B.

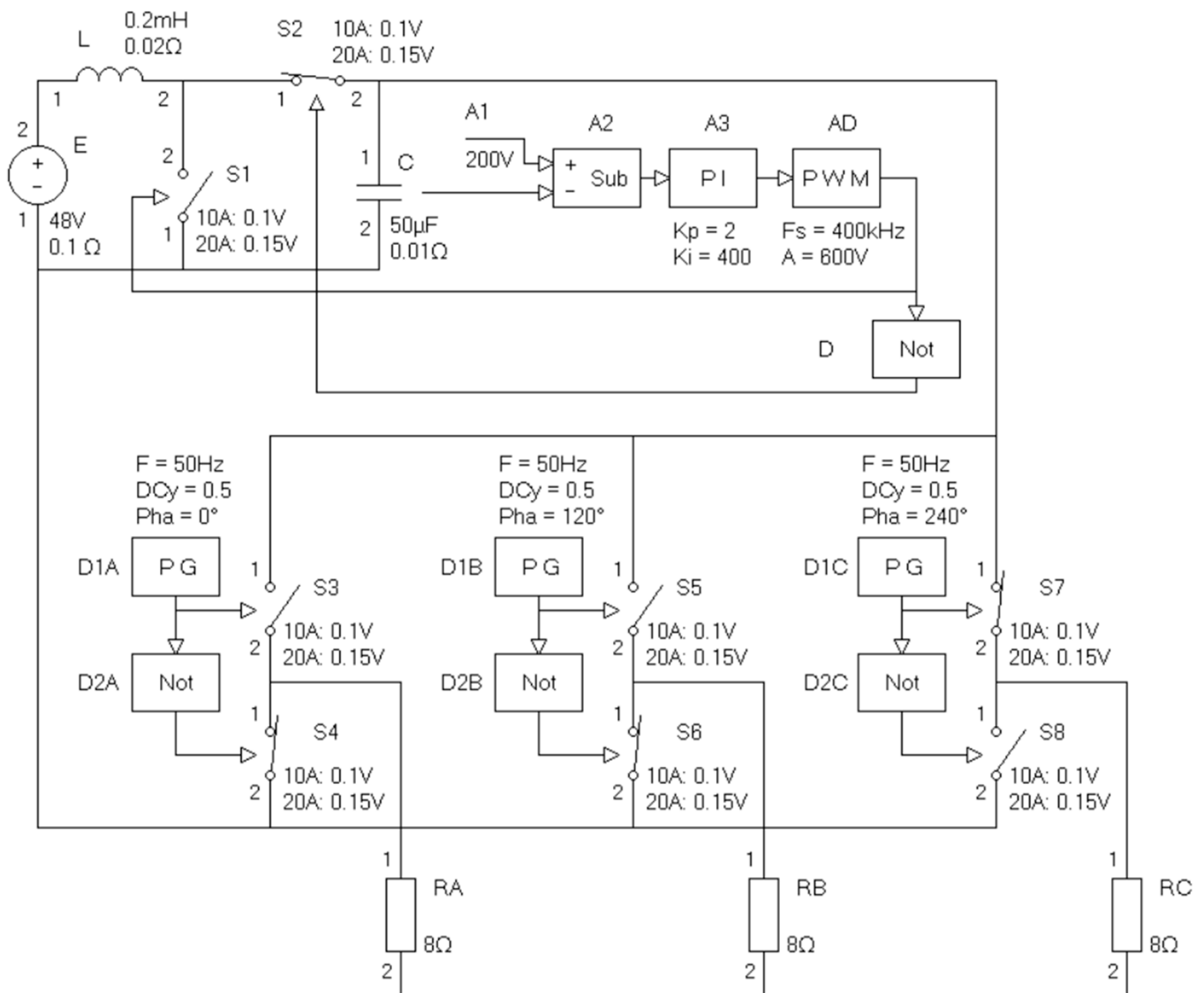


Figure 9. Three-phase converter using proportional–integral (PI) controller and pulse width modulation (PWM).

Table 2. Simulation durations for the three-phase converter from Figures 8 and 9.

Model time duration [ms]	20	40	60	80	100
Simulation duration in Simlink [s]	2.93	5.03	7.47	9.37	11.54
Simulation duration in the authors’ software [ms]	333	448	544	637	751

5. Discussion

The presented approach effectively automates electronic circuit modeling. The components, junctions, nodes, and loops are represented by index matrices with real number elements, and sums of such matrices are implemented in the proposed .NET application for power systems’ design. The operation of PID-regulator, which theoretically can be modeled by second order differential equation, has likewise been properly implemented by the authors; the achieved results are identical to the results obtained by Simulink. The chosen approach is appropriate in power electronics since the number of components in circuits is not large, and the number of switches is not high, and furthermore, the combinations of closed and opened switch states raise exponentially with reason of their number in the

case of non-interaction, but many of these combinations are forbidden by the operation of the circuits. For example, there are two combinations of switch states for the buck DC-DC converter from Figure 2 (the respective number in case of non-interaction of its two switches is four) and there are six such combinations of switch states for the three-phase converter from Figure 9 (the respective number in case of non-interaction of its 8 switches is 256). The presented approach suggests a fast simulation, and results can be obtained automatically in an exhaustive way. It can be extended to handle problems in the wide area of the first-order discrete dynamical systems and their applications.

6. Conclusions

The paper presents an integrated approach to the modeling and design of power electronic devices, based on an index matrix tool. Compared to other applied methods for studying the behavior of electronic converters [19–25], the proposals are characterized by versatility, reduced requirements for computational resources, robustness of the solution, and a high potential for algorithmization. Thus, it is highly suitable for the purposes of training in power electronics, as well as in research and engineering practice. However, an open-source software system, which users can customize according to their knowledge, capabilities, and needs, is proposed. One natural development of this approach is its incorporation into an expert system that automates the design of power electronic devices and systems.

Author Contributions: N.H., P.G. and V.G. were involved in the full process of producing this paper, including conceptualization, methodology, modeling, validation, visualization and preparing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Bulgarian National Scientific Fund, grant number KII-06-H57/7/16.11.2021, and the APC was funded by KII-06-H57/7/16.11.2021.

Acknowledgments: This research was carried out within the framework of the project “Artificial Intelligence-Based modeling, design, control and operation of power electronic devices and systems”, KII-06-H57/7/16.11.2021, Bulgarian National Scientific Fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karris, S. *Introduction to Simulink with Engineering Applications*; Orchard Publications: San Antonio, TX, USA, 2006. Available online: neuron.eng.wayne.edu/auth/ece4340/Simulink_Introduction.pdf (accessed on 23 December 2021).
2. Araujo, E.; Leite, A.; Freitas, D. Modelling and simulation of power electronic systems using a bond graph formalism. In Proceedings of the 10th Mediterranean Conference on Control and Automation MED2002, Lisbon, Portugal, 9–12 July 2002.
3. Galor, O. *Discrete Dynamical Systems*; Springer-Verlag: Berlin/Heidelberg, Germany, 2007.
4. Gocheva, P.; Hinov, N.; Gochev, V. Matrix based estimation of current and voltage magnitude in electronic circuits. *AIP Conf. Proc.* **2018**, *2048*, 060026.
5. Gochev, V.; Gocheva, P.; Hinov, N. NET implementation of electronic circuit design. *AIP Conf. Proc.* **2021**, *2333*, 070016.
6. Hinov, N.; Gocheva, P.; Gochev, V. Index Matrices Based Modelling of a DC-DC Buck Converter with PID Controller and GUI on It. In Proceedings of the 34th International Conference on Information Technologies InfoTech 2020—Proceedings, Varna, Bulgaria, 17–18 September 2020.
7. Cerny, D.; Dobes, J. Adaptive sparse matrix indexing technique for simulation of electronic circuits based on λ -calculus. In Proceedings of the 2015 European Conference on Circuit Theory and Design (ECCTD), Trondheim, Norway, 24–26 August 2015; pp. 1–4. [[CrossRef](#)]
8. Cerny, D.; Dobes, J. Functional Programming Languages in Computer Simulation of Electronics Circuits. In Proceedings of the Computational Science and Computational Intelligence (CSCI) 2014 International Conference, Las Vegas, NV, USA, 10–13 March 2014; Volume 1.
9. Xia, W.; Liu, J.; Cao, M.; Johansson, K.H.; Başar, T. Generalized Sarymsakov Matrices. *IEEE Trans. Autom. Control* **2018**, *64*, 3085–3100. [[CrossRef](#)]
10. Atanassov, K. Index Matrices: Towards an augmented matrix calculus. In *Studies in Computational Intelligence*; Springer International Publishing: Cham, Switzerland, 2014; Volume 573.
11. Traneva, V.; Traneva, S. *Index Matrices as a Tool for Managerial Decision Making*; Publishing House of the Union of Scientists: Sofia, Bulgaria, 2017. (In Bulgarian)

12. Gochev, V.; Hinov, N.; Gocheva, P. Mathematical Modelling of Basic Electronic Components with Index Matrices. In Proceedings of the 33th International Conference on Information Technologies InfoTech 2019—Proceedings, Varna, Bulgaria, 19–20 September 2019.
13. Hinov, N.; Gocheva, P.; Gochev, V. Representation with Index Matrices of Single Ended Primary Inductor Converter Functioning. In Proceedings of the International Conference on High Technology for Sustainable Development HiTech 2019, Sofia, Bulgaria, 10–11 October 2019.
14. Atanassov, K. Conditions in generalized nets. In Proceedings of the XIII Spring Conference of the Union of Bulgarian Mathematical, Sunny Beach, Bulgaria, 6–9 April 1984; pp. 219–226.
15. Atanassov, K. *Generalized Nets*; World Scientific: Singapore, 1991.
16. Atanassov, K. *On Generalized Nets Theory. "Prof. Marin Drinov"*; Academic Publishing House: Sofia, Bulgaria, 2007.
17. Alexieva, J.; Choy, E.; Koycheva, E. Review and bibliography on generalized nets theory and applications. In *A Survey of Generalized Nets, Raffles KvB Monograph*; Choy, E., Krawczak, M., Shannon, A., Szmidt, E., Eds.; World Scientific: Singapore, 2007; Volume 10, pp. 207–301.
18. Vardeva, I.; Staneva, L. *Automated Design of Electronic Circuits Modeled with Generalized Networks*; Industrial Technologies; University "Prof. Dr. Asen Zlatarov": Burgas, Bulgaria, 2014; Volume I, pp. 74–77. Available online: www.researchgate.net/publication/306346628_Avtomatizirano_proektirane_na_elektronni_shemi_modelirani_s_obobseni_mrezi (accessed on 23 December 2021). (In Bulgarian)
19. Ali, U.; Raza, H.; Ahmed, Y. On Normalized Laplacians, Degree-Kirchhoff Index and Spanning Tree of Generalized Phenylene. *Symmetry* **2021**, *13*, 1374. [[CrossRef](#)]
20. Kallioras, N.A.; Nordas, A.N.; Lagaros, N.D. Deep Learning-Based Accuracy Upgrade of Reduced Order Models in Topology Optimization. *Appl. Sci.* **2021**, *11*, 12005. [[CrossRef](#)]
21. Benômar, Y.; Croonen, J.; Verrelst, B.; Van Mierlo, J.; Hegazy, O. Model-Based Control System Design of Brushless Doubly Fed Reluctance Machines Using an Unscented Kalman Filter. *Energies* **2021**, *14*, 8222. [[CrossRef](#)]
22. Belhaj, F.Z.; El Fadil, H.; Idrissi, Z.E.; Koundi, M.; Gaouzi, K. Modeling, Analysis and Experimental Validation of the Fuel Cell Association with DC-DC Power Converters with Robust and Anti-Windup PID Controller Design. *Electronics* **2020**, *9*, 1889. [[CrossRef](#)]
23. El Fadil, H.; Giri, F.; Chaoui, F.; El Magueri, O. Accounting for Input Limitation in the Control of Buck Power Converters. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *56*, 1260–1271. [[CrossRef](#)]
24. Duda, J. Lyapunov matrices approach to the parametric optimization of a time delay system with a PI controller. In Proceedings of the 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 29 August–1 September 2016; pp. 1206–1210. [[CrossRef](#)]
25. Akundi, A.; Tseng, T.-L.B.; Almeraz, C.N.; Lopez-Terrazas, R.J.; Luan, H. A Novel Approach to Behavior Design for Model Based Systems Engineering Application Using Design Structure Matrix. In Proceedings of the 2021 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 15 April–15 May 2021; pp. 1–7. [[CrossRef](#)]