Index Matrix-Based Modeling and Simulation of Buck Converter

Nikolay Hinov 1,*, Polya Gocheva 2 and Valeri Gochev 3

1 Department of Power Electronics, Technical University of Sofia, 1756 Sofia, Bulgaria
2 Doctoral School, Technical University of Sofia, 1756 Sofia, Bulgaria; p.gocheva@utp.bg
3 Department of Telecommunications, University of Telecommunications and Posts, 1700 Sofia, Bulgaria;
v.gochev@utp.bg
* Correspondence: hinov@tu-sofia.bg

Abstract: The approach described in this paper handles the parameters and characteristics (analog and discrete ones) of a Buck DC-DC converter (in its power and control parts) in a common manner. The usage of probably complicated differential equations for discrete dynamical systems is avoided by means of index matrix equations, which can be easily understood. Compared to classical matrix models, the proposed index matrix models are more descriptive and also smaller in size. Such functionality is widely applied by the authors, and a new operation is defined and used as well. The relevance of the proposed techniques in power electronics, because of switching topologies and a limited numbers of components, is argued. Respective examples of functioning modes of the considered converter, in which power circuits and controllers are modeled jointly, are given. Estimations of analog values are based on partly linear dependencies, which are shown to be adequate in first-order analyses. Specific expressions for a Buck DC-DC converter are presented. A model-solving technique and an exhaustive search on a parameter space are considered in detail and automated via well-formalized algorithms. Nested parameter intervals and verifications with normal probability distributions are used in an optimization procedure. The full agreement of implementation via MATLAB source code with results obtained via Simulink is demonstrated. The short simulation times of this software (compared to Simulink and a .NET desktop application developed by the authors) justify the search for optimal variants in a wide multi-dimensional space. A max–min procedure with 10,000 simulations in each verification step is presented.

Keywords: index matrix; model-based design; optimization; parameter space; power electronics

MSC: 03C98; 11C20

1. Introduction

Due to the growing need to convert electricity with electronic means, power electronics is a rapidly developing field. This fact requires new developments and an improvement in the tools used to obtain the optimal design of power electronic devices and systems. On the other hand, because of the presence of power and control parts (controllers), which are quite different and mutually influence and complement each other, in many approaches, electronic converters are not susceptible to modeling as a whole unified system. One of the main purposes of the authors’ papers has been to present a joint technique based on the apparatus of index matrices [1,2].

First-order time-invariant dynamical systems are often used as modeling tools [3,4]. Discrete dynamical systems are represented in [5] via equations with constant classical matrices, and this approach does not differ significantly from well-known control system engineering methods, in which derivatives of state variables are written as linear combi-
nations of system states and inputs [4,6]. Both of the aforementioned techniques determine a simple stability analysis of discrete or analog values in multi-input–multi-output systems, including in power electronic devices [7,8].

In [9], two ideal switches, which work synchronously in a Boost DC-DC converter, determine two functioning modes, and, therefore, two different constant matrices in a process of discrete dynamical modeling. In the present paper, each switch is considered to have two modes too, but the transistor and the diode work out of sync. Such a principle requires variants to be checked and, similarly, determines the exponential dependence of the number of variants on the number of switches [1]. However, the latter is not large in power electronic devices; in the following sections, the transistor and the diode determine four modes. Generally, the switching topologies in power converters do not allow an analysis based on constant matrices or constant differential equations. Therefore, computer simulations are widely used; for example, the graphical programming environment Simulink, which is shown in the present paper in a comparison, provides a set of solvers and has power electronic tools [10].

A feature that dynamical systems have in common is that they are all simulated by estimating states at consecutive model time moments. The obtainment of values on system states (which usually are presented by one-dimensional or multi-dimensional arrays) is known as model solving. 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 [11,12]. However, in the present paper, index matrix equations are used in the configuration of electronic components. The discretization step in the algorithm proposed in Section 4 is chosen to be constant; it is obvious that decreasing it increases the accuracy and vice versa (similar dependences are demonstrated in a figure in Section 4). Values on system states are estimated at regular model time intervals from the beginning to the end of the simulation; i.e., a fixed-step solver is constructed. Moreover, the index matrix describing system states can vary in size according to a new kind of multiplication, which is defined for the first time in the present article.

Index matrices (this concept was introduced in 1984) extend classical matrix functionality; their properties have been studied and summarized in [13,14]; basic notes are given in Section 2, too. The index matrices extend the concept of the classical matrices and provide clear correspondence between their elements and indices. Different types of their elements have been proposed; in the present paper, real numbers are used. The index matrices are mainly connected to generalized nets, in which index matrices are basic components [15,16]. Applications of index matrices concern mathematical and real-life objects, for example, databases, but the usage of this apparatus in the modeling of electrical circuits is very rare. A full review of their applications is unpublished as of yet. Indeed, for a long time, they have been used with Boolean values and integers only as an auxiliary tool in generalized net transitions.

Data structures, which implement index matrices in the programming class IndexMatrix, are demonstrated in [1]. There are implementations of all operations from Section 2, except the new one, in this class too. The .NET application, which is presented in [1], concerns analog and digital electronic components and circuits, model solvers, graphics, etc. Its basic feature is the shorter simulation duration compared to Simulink; it has about a 20 times shorter runtime in the case of the considered buck converter and about a 10 times shorter runtime in the case of a three-phase converter. Generally, all model-based design techniques require many simulations [17,18]; the optimization algorithm from Section 5 uses a large number of simulations in a verification process.

The approach discussed in this paper is implemented via MATLAB source code [19]. This choice is natural because classical matrices are presented in the algorithm from Section 4. The core functionality of MATLAB is closely related to matrix manipulation. The obtained runtimes ensure the model’s ability to perform exhaustive searches; this fact is a prerequisite for the application of artificial intelligence techniques, too [2]. The simulation
durations of this implementation, as well as those in the cases of the discussed .NET application and Simulink, are given in Section 6.

2. Basic Definitions of Index Matrices with Real Number Elements

All index matrices with real numbers as elements (R-IM) [13,14] have the form

\[ \begin{bmatrix} a_{k_1,l_1} & a_{k_1,l_2} & \cdots & a_{k_1,l_n} \\ a_{k_2,l_1} & a_{k_2,l_2} & \cdots & a_{k_2,l_n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k_m,l_1} & a_{k_m,l_2} & \cdots & a_{k_m,l_n} \end{bmatrix} \]

with index sets (called the first and second one, respectively)

\[ K = \{k_1, k_2, \ldots, k_m\}, \quad L = \{l_1, l_2, \ldots, l_n\}. \]

In the present paper (as well as in [1]), the following notations of K and L are used:

\[ K = I_1(\overline{A}), \quad L = I_2(\overline{A}). \]

The usual graphical representation of the above index matrix form is

\[ \begin{array}{c|c|c|c} & l_1 & l_2 & \cdots & l_n \\ \hline k_1 & a_{k_1,l_1} & a_{k_1,l_2} & \cdots & a_{k_1,l_n} \\ k_2 & a_{k_2,l_1} & a_{k_2,l_2} & \cdots & a_{k_2,l_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_m & a_{k_m,l_1} & a_{k_m,l_2} & \cdots & a_{k_m,l_n} \end{array} \]

Lines divide indices from elements. All indices from the index set are used like lower indices, too. Each element is positioned in the same row with the index which is identical to its first lower index and in the same column with the index which is identical to its second lower index. The ordering of the indexes explicitly determines the ordering of the elements. Swaps of rows or columns (along with the indexes) of an index matrix do not change it. For comparison, in the classical matrices, row and column indexes are implicitly used, and any swap of elements must generally correspond to respective swaps in other objects in order dependencies (for example, in matrix equations) to be preserved.

Authors’ notations of index and classical matrices are proposed here; they lead to clear expressions, those in algorithms (1), (2) and respectively (4) can be represented by a classical matrix (which is denoted by “CM” in the present paper)

\[ \begin{bmatrix} a_{u_1,v_1} & a_{u_1,v_2} & \cdots & a_{u_1,v_n} \\ a_{u_2,v_1} & a_{u_2,v_2} & \cdots & a_{u_2,v_n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u_m,v_1} & a_{u_m,v_2} & \cdots & a_{u_m,v_n} \end{bmatrix} \]

and ordered sets

\[ U = \{u_1, u_2, \ldots, u_m\}, \quad V = \{v_1, v_2, \ldots, v_n\}, \]

where U and V are permutations of \( I_1(\overline{A}) \) and \( I_2(\overline{A}) \), respectively. There are \( m! \cdot n! \) options for (5) according to all permutations in (6). It is important to note that these variants differ only by ordering. The inverse notation

\[ \text{IM}_{U,V} \quad A' = \overline{A} \]
on \( \mathbf{CM}_{U,V} \mathbf{A} = \mathbf{A}' \) is also used in the present paper.

Brief notes of basic operations (summation, multiplication, transposing and projecting of index matrices—see [12,13]) are given below in order of specific usages (for example, in summation with non-intersecting index sets) and a new definition of a multiplication to be argued.

An example of a transpose R-IM is

\[
\mathbf{G} = \mathbf{A}^T = \begin{bmatrix} K, & L, & \{a_{k,l}\}_{k \in K, l \in L} \end{bmatrix}^T = \begin{bmatrix} L, & K, & \{g_{l,k} = a_{k,l}\}_{l \in L, k \in K} \end{bmatrix}.
\] (8)

Each projection of \( \mathbf{A} \) (this one is an index sub-matrix, denoted by \( \mathbf{H} \) below, with notation \( \mathbf{H} \subseteq \mathbf{A} \) in the present paper) has the form

\[
\mathbf{H} \equiv \text{pro}_{R,S} \mathbf{A} = \begin{bmatrix} R, & S, & \{a_{r,s}\}_{r \in R, s \in S} \end{bmatrix} \subseteq \mathbf{A}, \quad R \subseteq K, \quad S \subseteq L.
\] (9)

Let R-IM

\[
\mathbf{B} = \begin{bmatrix} P, & Q, & \{b_{p,q}\}_{p \in P, q \in Q} \end{bmatrix}
\] (10)

be given. Sum

\[
\mathbf{A} \oplus \mathbf{B} = \begin{bmatrix} K \cup P, & L \cup Q, & \{c_{r,s}\}_{r \in K \cup P, s \in L \cup Q} \end{bmatrix}.
\]

\[
c_{r,s} = \begin{cases} 
a_{r,s}, & \text{if } (r \in K \land s \in L - Q) \lor (r \in K - P \land s \in L); 
b_{r,s}, & \text{if } (r \in P \land s \in Q - L) \lor (r \in P - K \land s \in Q); 
a_{r,s} + b_{r,s}, & \text{if } r \in K \cap P \land s \in L \cap Q; 
0, & \text{otherwise}; 
\end{cases}
\] (11)

is a R-IM. Formally, all classical matrix sums can be represented by a proper case of the upper R-IM sum, for which \( K = P \) and \( L = Q \) (see value \( a_{r,s} + b_{r,s} \) in the third case of \( c_{r,s} \)). Oppositely, if there is a given lack of intersection of index sets, this value can be excluded from (11):

\[
(K \cap P = \emptyset \lor L \cap Q = \emptyset) \Rightarrow c_{r,s} = \begin{cases} 
a_{r,s}, & \text{if } r \in K \land s \in L; 
b_{r,s}, & \text{if } r \in P \land s \in Q; 
0, & \text{otherwise}; 
\end{cases}
\] (12)

Here, all elements of \( \mathbf{A} \) and \( \mathbf{B} \) are presented in different parts of \( \mathbf{A} \oplus \mathbf{B} \).

The product of \( \mathbf{A} \) and \( \mathbf{B} \)

\[
\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} K \cup (P - L), & Q \cup (L - P), & \{d_{r,s}\}_{r \in K \cup (P - L), s \in Q \cup (L - P)} \end{bmatrix}
\]
\[
\mathcal{d}_{r,s} = \begin{cases}
    a_{r,s}, & \text{if } r \in K \land s \in L - P - Q; \\
    b_{r,s}, & \text{if } r \in P - K - L \land s \in Q; \\
    \sum_{t \in L \cap P} a_{r,t} b_{t,s}, & \text{if } r \in K \land s \in Q; \\
    0, & \text{otherwise};
\end{cases}
\] 

(13)

is an R-IM, too. Formally, all classical matrix products can be represented by a proper case of the upper R-IM product, for which \( L = P \) (see the sum in its common form).

A new kind of an R-IM product is defined below by the authors:

\[
\overline{A} \otimes \overline{B} = \left[ K, \ Q \cup (L - P), \ \{d'_{r,s}\}_{r \in K, \ s \in Q \cup (L - P)} \right]
\] 

(14)

\[
\mathcal{d}'_{r,s} = \begin{cases}
    a_{r,s}, & \text{if } s \in L - P - Q; \\
    \sum_{t \in L \cap P} a_{r,t} b_{t,s}, & \text{if } s \in Q.
\end{cases}
\] 

(15)

It reduces \( \overline{A} \otimes \overline{B} \) according to its rows (the first index sets of \( \overline{A} \) and \( \overline{A} \otimes \overline{B} \) are identical):

\[
\overline{A} \otimes \overline{B} = \text{pro}_{I_1(\overline{A}), I_2(\overline{A} \otimes \overline{B})}(\overline{A} \otimes \overline{B}).
\] 

(16)

The following statements hold:

\[
L \subseteq P \Rightarrow \overline{A} \otimes \overline{B} = \left[ K \cup (P - L), \ Q, \ \{d^*_{r,s}\}_{r \in K \cup (P - L), s \in Q} \right],
\] 

(17)

\[
L \subseteq P \Rightarrow \overline{A} \otimes \overline{B} = \left[ K, \ Q, \ \{d^*_{r,s}\}_{r \in K, s \in Q} \right], \quad d^*_{r,s} = \sum_{t \in L} a_{r,t} b_{t,s}.
\] 

(18)

Examples of the new R-IM multiplication are given in the next sections.

3. R-IM Models in Electronic Circuit Design

3.1. Models of Electronic Components

Models of basic electronic components are given below. Similar but simpler ones are described in [1], in which ideal components are used; additional resistances and potential barriers are taken into account below.

Let time set \( \Theta \) with sampling step \( \Delta t \) and discretization frequency \( F_d \) be given under the following restrictions:

\[
\Theta \equiv \{ t_0, \ t_0 + \Delta t, \ t_0 + 2\Delta t, \ \ldots \}, \quad t_0, \ \Delta t \in \mathbb{R}, \quad \Delta t = \frac{1}{F_d} > 0.
\]
An R-IM, which concerns an inductor with inductance $L$ and additional resistance $R_L$, is presented at first:

$$\overrightarrow{F}(t) = \begin{bmatrix} u_L^{cu} \\ i_L^{cu} \\ i_L^{pr} \end{bmatrix} = \begin{bmatrix} c_L \\ -1 \\ \frac{L}{\Delta t} + R_L \end{bmatrix} \cdot \frac{L}{\Delta t}.$$  \hspace{1cm} (19)

$c_L$ denotes the inductor. $u_L^{cu}$, $i_L^{cu}$, $u_L^{pr}$ and $i_L^{pr}$ indicate voltages (with "u") and amperages (with "i") in current ($t \in \Theta$, marked by "cu") and previous ($t-\Delta t$, marked by "pr") time moments. This notation is used by the authors in [1] too; index matrix

$$\begin{pmatrix} u_L(t) \\ i_L(t) \\ u_L(t-\Delta t) \\ i_L(t-\Delta t) \end{pmatrix}$$ \hspace{1cm} (19)

where $o$ is a common index, contains all these four indexes. The R-IM equation

$$\overrightarrow{X}_L(t) = \begin{pmatrix} -u_L(t) + \left( \frac{L}{\Delta t} + R_L \right) i_L(t) - \frac{L}{\Delta t} i_L(t-\Delta t) \end{pmatrix} \hspace{1cm} (20)$$

$(\overrightarrow{O}$ is a zero R-IM—its only element is equal to zero) determines equality

$$u_L(t) = L \frac{i_L(t) - i_L(t-\Delta t)}{\Delta t} + R_L i_L(t),$$ \hspace{1cm} (21)

which is an approximation of differential equation

$$u_L(t) = L \frac{d i_L(t)}{d t} + R_L i_L(t).$$ \hspace{1cm} (22)

Equation (21) is based on the new R-IM multiplication (14). For comparison,

$$\overrightarrow{F}_L = \begin{bmatrix} u_L^{cu} \\ i_L^{cu} \\ i_L^{pr} \end{bmatrix} = \begin{bmatrix} c_L \\ -1 \\ \frac{L}{\Delta t} + R_L \end{bmatrix} \cdot \frac{L}{\Delta t} \hspace{1cm} (23)$$

Similarly, in case of a capacitor with capacitance $C$ and additional resistance $R_C$ approximation

$$i_C(t) = C \left[ u_C(t) - R_C i_C(t) \right] - \left[ u_C(t-\Delta t) - R_C i_C(t-\Delta t) \right] \frac{\Delta t}{\Delta t}$$ \hspace{1cm} (24)
In the same manner, index matrix $X_C(t)$ with four elements (see (20)) and an analog of Equation (21) can be formed. The presented approach can be extended to other power components from Figure 1, and common index matrices can be constructed:

Figure 1. Simulink model of buck DC-DC converter with PI controller.

The next R-IM models the diode functioning:

$$ F_D(t) = \begin{cases} 
\frac{u_c^{cu}}{c_D} \left( -1 \right) \frac{i_c^{cu}}{R_D} \Delta t, & \text{if } u_D(t) > U_D^0 \\
\frac{i_D^{cu}}{c_D} \frac{1}{R_D}, & \text{otherwise} 
\end{cases} $$

(28)

It leads to the following statements:

$$ u_D(t) > U_D^0 \Rightarrow u_D(t) = U_D^0 + R_D i_D(t). $$

(29)

$$ u_D(t) \leq U_D^0 \Rightarrow i_D(t) = 0. $$

(30)

Here, $U_D^0$ and $R_D$ denote a threshold voltage (a potential barrier) and a dynamic resistance, respectively.

The next R-IM models the transistor emitter–collector junction functioning:
\[
\overline{F}_{\text{T} \text{out}}(t) = \begin{cases} 
\frac{1}{c_{\text{out}}} & \frac{u_{\text{T} \text{out}}^{\text{cu}}}{R_{\text{T} \text{out}}} - 1 \frac{i_{\text{T} \text{out}}^{\text{cu}}}{R_{\text{T} \text{out}}} - U_{\text{T} \text{out}}^0, \\
\frac{1}{c_{\text{out}}} & \frac{i_{\text{T} \text{out}}^{\text{cu}}}{R_{\text{T} \text{out}}} - 1
\end{cases}, \quad \text{if } s_{\text{T} \text{in}}(t) = 1 \land u_{\text{T} \text{out}}(t) > U
\]

Similarly to the diode model, it determines the following statements:

\[
s_{\text{T} \text{in}}(t) = 1 \land u_{\text{T} \text{out}}(t) > U_{\text{T} \text{out}}^0 \Rightarrow u_{\text{T} \text{out}}(t) = U_{\text{T} \text{out}}^0 + R_{\text{T} \text{out}} i_{\text{T} \text{out}}(t),
\]

\[
s_{\text{T} \text{in}}(t) = 0 \lor u_{\text{T} \text{out}}(t) \leq U_{\text{T} \text{out}}^0 \Rightarrow i_{\text{T} \text{out}}(t) = 0;
\]

with threshold voltage \(U_{\text{T} \text{out}}^0\) and dynamic resistance \(R_{\text{T} \text{out}}\); signal \(s_{\text{T} \text{in}}(t)\) on the input of the IGBT is digital.

\[
\overline{F}_R = \frac{1}{c_{\text{R}}} \left( \frac{u_{\text{R}}^{\text{cu}}}{R_{\text{R}}} - 1 \right) \frac{i_{\text{R}}^{\text{cu}}}{R_{\text{R}}}; \quad \overline{F}_E = \frac{1}{c_{\text{E}}} \left( \frac{u_{\text{E}}^{\text{cu}}}{R_{\text{E}}} - 1 \right) \frac{i_{\text{E}}^{\text{cu}}}{R_{\text{E}}} - \frac{o}{E}.
\]

The following equations (see (27) and (28)) can be used with the upper R-IMs:

\[
\overline{F}_{\text{PCs}} \otimes \left[ X_{\text{PCs}}(t) \oplus \left( \frac{o}{1} \right) \right] = \overline{O};
\]

\[
\overline{F}_{\text{PC}} \otimes \left[ X_{\text{PC}}(t) \oplus \frac{o}{1} \right] = \overline{O}, \quad \text{PC} \in \{E, T_{\text{out}}, D, L, C, R\}.
\]

The components of the considered control circuit (see Figure 1) can be modeled by R-IMs in a similar manner [1]. The subtraction block with target voltage \(U_{\text{Target}}\), the proportional–integral (PI) controller and the pulse-width modulation (PWM) block are presented by \(\overline{F}_{\text{Sub}}, \overline{F}_{\text{PI}}\) and \(\overline{F}_{\text{PWM}}(t)\), respectively, below:

\[
\overline{F}_{\text{Sub}} = \frac{1}{c_{\text{Sub}}} \left| \frac{u_{\text{Sub}}^{\text{cu}}}{u_{\text{Sub}}^{\text{in}}} - 1 \right| - 1 \frac{o}{U_{\text{Target}}};
\]

\[
\overline{F}_{\text{PI}} = \frac{1}{c_{\text{PI}}} \left| \frac{u_{\text{PI}}^{\text{cu}}}{u_{\text{PI}}^{\text{in}} - 1} \frac{u_{\text{PI}}^{\text{pr}}}{\Delta t} \frac{K_p}{\Delta t} + K_i - \frac{K_p}{\Delta t} \right|; \quad \Delta t = 1.
\]
\[ F_{PWM}^{(t)} = \begin{cases} \frac{c_{PWM}}{1} \frac{s_{PWM_{out}}^{cu}}{0}, & \text{if } u_{\text{Triang}}(t) < u_{PWM_{in}}(\psi); \\ \frac{c_{PWM}}{-1} \frac{s_{PWM_{out}}^{cu}}{0}, & \text{otherwise}; \end{cases} \] (39)

where
\[ \psi \in \left( t - \Delta t_s - \Delta t, t - \Delta t \right), \] (40)
is a time moment from \( \Theta \) (values of \( u_{PWM_{in}} \) are checked for such time moments with sampling step \( \Delta t_s \), and
\[ u_{\text{Triang}}: \Theta \to [0, 1] \] (41)
is a triangular function with period 1 and amplitude \( \Delta t_s = \frac{1}{F_s} \). The two cases of index matrix (40) determine digital signal \( s_{PWM_{out}}(t), t \in \Theta \):
\[ u_{\text{Triang}}(t) < u_{PWM_{in}}(\psi) \Rightarrow s_{PWM_{out}}(t) = 1; \] (42)
\[ u_{\text{Triang}}(t) \geq u_{PWM_{in}}(\psi) \Rightarrow s_{PWM_{out}}(t) = 0. \] (43)

3.2. Models of Component Connections and Electronic Circuits

In [1], nodes and loops are modeled according to Kirchhoff’s laws. The respective sums of R-IMs in case of the present paper are

\[ F_N = \begin{array}{c|cccccc} \hline & i_{\text{cu}}^E & i_{\text{cu}}^T & i_{\text{cu}}^D & i_{\text{cu}}^L & i_{\text{cu}}^C & i_{\text{cu}}^R \\ \hline n_1 & -1 & -1 & 0 & 0 & 0 & 0 \\ n_2 & 0 & 1 & 1 & -1 & 0 & 0 \\ n_3 & 0 & 0 & 0 & 1 & -1 & -1 \\ \hline \end{array}; \] (44)

\[ F_P = \begin{array}{c|cccccc} \hline & u_{\text{cu}}^E & u_{\text{cu}}^T & u_{\text{cu}}^D & u_{\text{cu}}^L & u_{\text{cu}}^C & u_{\text{cu}}^R \\ \hline p_1 & 1 & -1 & 1 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 1 & 1 & 1 & 0 \\ p_3 & 0 & 0 & 0 & -1 & 1 & \end{array}; \] (45)

The first one models nodes according to Kirchhoff’s current law, and the second one models loops according to Kirchhoff’s voltage law. The upper R-IMs are constructed in order conditions

\[ \text{rank} \left( CM_{U_{N}, V_{N}} F_N \right) = \| I_1(F_N) \|; \quad \text{rank} \left( CM_{U_{P}, V_{P}} F_P \right) = \| I_1(F_P) \|. \] (46)
to be satisfied, where \( U_N, V_N, U_P \) and \( V_P \) are permutations of the index sets of \( \overline{F}_N \) or \( \overline{F}_P \) (see (5) and (6)). Here and below, \( \text{rank}(A) \) denotes the rank of classical matrix \( A \), and \( |S| \) denotes the cardinality of set \( S \).

\[
\overline{F}_{PCirc}(t) \otimes \left( \overline{X}_{PCirc} \oplus - \frac{\circ}{\circ} _{1} \right) = \overline{O},
\]

\[
\overline{F}_{PCirc}(t) = \overline{F}_{PCs}(t) \oplus \overline{F}_N \oplus \overline{F}_P,
\]

\[
\overline{X}_{PCirc}(t) = \overline{X}_{PCs}(t)
\]

(47)

represent the considered power circuit with its components and junctions. In order to create a more compact model, the following function of R-IM \( \overline{F}(t) \) is defined below:

\[
X(\overline{F}(t)) = \left[ I_2(\overline{F}(t)), \{0\}, \left\{ x_{m,o}(t) \right\}_{m \in I_2(\overline{F}(t))} \right]
\]

\[
x_{m,o}(t) = \begin{cases} 
  u_e(t), & \text{if } m = u_e^{cu}; \\
  i_e(t), & \text{if } m = i_e^{cu}; \\
  u_e(t-\Delta t), & \text{if } m = u_e^{pr}; \\
  i_e(t-\Delta t), & \text{if } m = i_e^{pr}; \\
  s_e(t), & \text{if } m = s_e^{cu}; \\
  1, & \text{if } m = o.
\end{cases}
\]

(48)

For example (see (19) and (20)),

\[
X(\overline{F}_L(t)) = \left[ - \frac{u_L^{cu}}{o} \frac{i_L^{cu}}{o} \frac{i_L^{pr}}{i_L(t-\Delta t)} \right]^T.
\]

(49)

Indeed,

\[
\left| I_2(\overline{F}_{PCirc}(t)) \right| = 14; \quad \left| I_1\left( \overline{X}_{PCirc}(t) \oplus - \frac{\circ}{\circ} _{1} \right) \right| = 25;
\]

(50)

and 11 columns of \( \overline{X}_{PCirc}(t) \) are not usable in (48). The following statements hold:

\[
\overline{F}_{PC}(t) \otimes \left( \overline{X}_{PC}(t) \oplus - \frac{\circ}{\circ} _{1} \right) = \overline{F}_{PC}(t) \otimes X(\overline{F}_{PC}(t)) = \overline{O},
\]

where \( \text{PC} \in \{ E, T_{out}, D, L, C, R \} \);

(51)

\[
\overline{F}_{PCs}(t) \otimes \left( \overline{X}_{PCs}(t) \oplus - \frac{\circ}{\circ} _{1} \right) = \overline{F}_{PCs}(t) \otimes X(\overline{F}_{PCs}(t)) = \overline{O};
\]

(52)
\[
F_{PCirc}(t) \odot \left( X_{PCirc}(t) \oplus - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) = F_{PCirc}(t) \odot X(F_{PCirc}(t)) = \emptyset. \quad (53)
\]

The upper index matrices, whose identifiers start with “F”, together with the considered function uniquely define the presented power circuit.

Other common statements on analog values are

\[
F_{Analog}^{CCs}(t) \odot X(F_{Analog}^{CCs}(t)) = \emptyset; \quad F_{Analog}^{CCs}(t) = F_{Sub} \oplus F_{PI} \quad (54)
\]

The abbreviation “CCs” means “control components”. Their connections (including ones with power components) which concern analog values are presented in

\[
F_{Analog}^{Q}(t) = \begin{array}{c|cccccc}
\text{Analog} & u_R^\text{cu} & u_{Sub_{in}}^\text{cu} & u_{Sub_{out}}^\text{cu} & u_{PI_{in}}^\text{cu} & u_{PI_{out}}^\text{cu} & u_{PWM_{in}}^\text{cu} \\
q_1 & -1 & 1 & 0 & 0 & 0 & 0 \\
q_2 & 0 & 0 & -1 & 1 & 0 & 0 \\
q_3 & 0 & 0 & 0 & 0 & -1 & 1
\end{array}. \quad (55)
\]

Statements

\[
F_{Analog}^{CCirc}(t) \odot X(F_{Analog}^{CCirc}(t)) = \emptyset; \quad F_{Analog}^{CCirc}(t) = F_{CCirc}^{CCs} \oplus F_{Q} \quad (56)
\]

represent the considered analog control circuit with its components and junctions in a proper manner.

R-IMs and their equations can be aggregated in

\[
F_{All}^{Analog}(t) \odot X(F_{All}^{Analog}(t)) = \emptyset; \quad F_{All}^{Analog}(t) = F_{PCirc}(t) \oplus F_{Q} \quad (57)
\]

Dependencies on digital signals are presented in

\[
F_{Digital}^{CCs}(t) = F_{PWM}(t); \quad F_{Digital}^{Q}(t) = \begin{array}{c|cc}
\text{Digital} & s_{PWM_{out}}^\text{cu} & s_{T_{in}}^\text{cu} \\
q_1 & -1 & 1
\end{array}; \quad (58)
\]

\[
F_{All}^{Digital}(t) = F_{CCirc}^{CCs}(t) = F_{Digital}^{CCs}(t) \oplus F_{Digital}^{Q} \quad (59)
\]

\[
F_{All, 1}^{Digital} = \begin{array}{c|cccccc}
\text{Digital} & s_{PWM_{out}}^\text{cu} & s_{T_{in}}^\text{cu} \\
q_1 & -1 & 1 & 0 & 0, & \text{if } u_{Triang}(t) < u_{PWM_{in}}^\text{cu}(t); \\
\text{Digital} & -1 & 0 & 1
\end{array}
\]

\[
F_{All, 0}^{Digital} = \begin{array}{c|cccccc}
\text{Digital} & s_{PWM_{out}}^\text{cu} & s_{T_{in}}^\text{cu} \\
q_1 & -1 & 0 & 0, & \text{otherwise}; \\
\text{Digital} & -1 & 0 & 1
\end{array}
\]
under conditions (41) and (42). An R-IM equation based on \( F_{\text{Digital All}}(t) \), similar to this one in (58), can be constructed:

\[
F_{\text{Digital All}}(t) \odot X\left(F_{\text{Digital All}}(t)\right) = \mathcal{O}.
\]  

(60)

R-IMs and their equations can be aggregated in

\[
F_{\text{All}}(t) \odot X\left(F_{\text{All}}(t)\right) = \mathcal{O}; \quad F_{\text{All}}(t) = F_{\text{Analog All}}(t) \oplus F_{\text{Digital All}}(t).
\]  

(61)

The upper statements form a clear mathematical model, which is solved in the next section. However, \( F_{\text{All}}(t) \) is not applied explicitly, whereas variants of \( F_{\text{Analog All}}(t) \) and \( F_{\text{Digital All}}(t) \) are used according to different cases.

4. Model Solving

4.1. Common Notes

\( F_{\text{Analog All}}(t) \) can be divided by columns to three R-IMs in the following way:

\[
F_{\text{Analog All}}(t) = F_{\text{Analog cu}}(t) \oplus F_{\text{Analog pr}}(t) \oplus F_{\text{Analog o}}(t),
\]

where \( I_{\text{cu}}, I_{\text{pr}}, I_{\text{o}} = \{o\} \subset I_{\text{All}}(t) \) are index sets, which concern all current values, all previous values and all constants, respectively. In case of the considered buck converter with a PI controller, \( F_{\text{Analog All}}(t) \) has 17 rows and 23 columns:

\[
I_{\text{cu}}(F_{\text{Analog All}}(t)) = 17, \quad I_{\text{pr}}(F_{\text{Analog All}}(t)) = 23.
\]  

(63)

\[
I_{\text{cu}}(F_{\text{Analog cu}}(t)) = 17, \quad I_{\text{pr}}(F_{\text{Analog pr}}(t)) = 5, \quad I_{\text{pr}}(F_{\text{Analog o}}(t)) = \{o\} = 1;
\]

(64)

\[
I_{\text{cu}}(F_{\text{Analog cu}}(t)) = \{i_{\text{pr}}, u_{\text{pr}}, i_{\text{in}}, u_{\text{in}}, i_{\text{out}}, u_{\text{out}}\};
\]

(65)

i.e., all 23 columns of \( F_{\text{Analog All}}(t) \) (which is sparse—more of its elements are zeros) are divided into three R-IMs with 17, 5 and 1 columns, respectively. (58) and (63) lead to

\[
\left[ F_{\text{Analog cu}}(t) \odot X\left(F_{\text{Analog cu}}(t)\right) \right] \oplus \left[ F_{\text{Analog pr}}(t) \odot X\left(F_{\text{Analog pr}}(t)\right) \right] \oplus F_{\text{Analog o}}(t) = \mathcal{O}.
\]  

(66)

In the last equation, the following equivalencies are used:

\[
X\left(F_{\text{Analog cu}}(t)\right) = \frac{|o|}{1}; \quad F_{\text{Analog o}}(t) \odot X\left(F_{\text{Analog o}}(t)\right) = F_{\text{Analog o}}(t).
\]

(67)

Six classical matrices based on \( h \in \{\text{cu}, \text{pr}, \text{o}\} \) can be defined (see (5) and (6)):
\[ F^\text{Analog}_h(t) = CM_{U, V^\text{Analog}_h(t)}, \quad X^\text{Analog}_h(t) = CM_{V^\text{Analog}_h(t)^\prime}, \quad (68) \]

where \( U = \{u_1, u_2, \ldots, u_{\vert U\vert}\} \), and \( V^\text{h} = \{V^\text{h}_1, V^\text{h}_2, \ldots, V^\text{h}_{\vert V^\text{h}\vert}\} \) are permutations of \( I_1\left(F^\text{Analog}_\text{All}(t)\right) \) and \( I_2\left(F^\text{Analog}_\text{All}(t)\right) \), respectively. Since \( X\left(F^\text{Analog}_0(t)\right) \) is a unit matrix, (67) and (69) derive the following equation:

\[ F^\text{Analog}_{\text{cu}}(t) X^\text{cu}(t) + F^\text{Analog}_{\text{pr}}(t) X^\text{pr}(t) + F^\text{Analog}_0(t) = 0. \quad (69) \]

In case of nonsingular square matrix \( F^\text{Analog}_{\text{cu}}(t) \):

\[ X^\text{cu}(t) = B^\prime(t) X^\text{pr}(t) + J^\prime(t), \quad (70) \]

A respective R-IM equation is

\[ X\left(F^\text{cu}^\text{Analog}(t)\right) = \left[ B(t) \odot X\left(F^\text{pr}^\text{Analog}(t)\right)\right] \oplus J(t), \]

\[ \bar{B}(t) = I_2\left(F^\text{cu}^\text{Analog}(t)\right), \quad I_2\left(F^\text{pr}^\text{Analog}(t)\right), \quad \left\{b_{u_m,v_{pr,n}} = b^\prime_{m,n}\right\}_{m \in 1, 2, \ldots, \vert U\vert, n \in 1, 2, \ldots, \vert V_{pr}\vert}, \quad (71) \]

According to (65) in the discussed example, \( \bar{B}(t) \) has 17 rows and 5 columns, and \( \bar{J}(t) \) has 17 rows and 1 column. They uniquely define all voltages and amperages of analog components. Index sub-matrices can be used, too:

\[ \text{pro}_{L, \{0\}} X\left(F^\text{cu}^\text{Analog}(t)\right) = \left[ B_1(t) \odot X\left(F^\text{pr}^\text{Analog}(t)\right)\right] \oplus \bar{J}(t), \quad I \subseteq I_2\left(F^\text{cu}^\text{Analog}(t)\right) \]

\[ B_1(t) = \text{pro}_{L, I_2(\bar{B}(t))} \bar{B}(t), \quad \bar{J}(t) = \text{pro}_{L, \{0\}} \bar{J}(t), \quad (72) \]

Such an approach is basic in one of the presented algorithms, in which I depend on

\[ I_{\text{cu, 1}}^\text{Analog} = \{u^\text{cu}_e \mid u^\text{pr}_e \in I_2\left(F^\text{pr}^\text{Analog}(t)\right)\} \cup \{i^\text{cu}_e \mid i^\text{pr}_e \in I_2\left(F^\text{pr}^\text{Analog}(t)\right)\}; \quad (73) \]

this index set denotes all current values, whereas the respective previous ones are included in (69)–(73).

\[ B_1(t), \bar{J}(t), F^\text{Analog}_\text{All}(t) \] and many others from the upper R-IMs depend on time. Alternative constant index matrices can be constructed; such ones are described in (29),
Index set Analog can be defined as related to values in conditions on time-dependent R-IMs in case of the considered converter

\[ i_{\text{Analog}}^{\text{cu},2} = \left\{ u_0^D, u_0^\text{cu}_\text{out}, u_0^\text{cu}_{\text{PWM}_{\text{in}}} \right\} \]  

A scheme according to the conditions is given below:

\[ (s_{\text{in}}(t) = 0 \lor u_{\text{out}}^t(t) \leq U_0^{\text{out}}) \land u_0^t(t) \leq U_0^D \iff F_{\text{in}}^r(t) = F_{\text{in}}^0 \lor \]  

\[ \land \overline{F}_{\text{D}}(t) = \overline{F}_{\text{D}}^0 \iff \overline{F}_{\text{Analog}}^{\text{All}}(t) = \overline{F}_{\text{Analog}}^{\text{All},0,0} \supseteq \overline{F}_{\text{out}}^0 \land \overline{F}_{\text{Analog}}^{\text{All},0,0} \supseteq \overline{F}_{\text{D}}^0; \]  

\[ (s_{\text{in}}(t) = 0 \lor u_{\text{out}}^t(t) \leq U_0^{\text{out}}) \land u_0^t(t) > U_0^D \iff F_{\text{in}}^r(t) = F_{\text{in}}^1 \lor \]  

\[ \land \overline{F}_{\text{D}}(t) = \overline{F}_{\text{D}}^1 \iff \overline{F}_{\text{Analog}}^{\text{All}}(t) = \overline{F}_{\text{Analog}}^{\text{All},0,1} \supseteq \overline{F}_{\text{out}}^0 \land \overline{F}_{\text{Analog}}^{\text{All},0,1} \supseteq \overline{F}_{\text{D}}^0; \]  

\[ (s_{\text{in}}(t) = 1 \land u_{\text{out}}^t(t) > U_0^{\text{out}}) \land u_0^t(t) \leq U_0^D \iff F_{\text{in}}^r(t) = F_{\text{in}}^1 \lor \]  

\[ \land \overline{F}_{\text{D}}(t) = \overline{F}_{\text{D}}^1 \iff \overline{F}_{\text{Analog}}^{\text{All}}(t) = \overline{F}_{\text{Analog}}^{\text{All},1,0} \supseteq \overline{F}_{\text{out}}^1 \land \overline{F}_{\text{Analog}}^{\text{All},1,0} \supseteq \overline{F}_{\text{D}}^1; \]  

\[ (s_{\text{in}}(t) = 1 \land u_{\text{out}}^t(t) > U_0^{\text{out}}) \land u_0^t(t) > U_0^D \iff F_{\text{in}}^r(t) = F_{\text{in}}^1 \lor \]  

\[ \land \overline{F}_{\text{D}}(t) = \overline{F}_{\text{D}}^1 \iff \overline{F}_{\text{Analog}}^{\text{All}}(t) = \overline{F}_{\text{Analog}}^{\text{All},1,1} \supseteq \overline{F}_{\text{out}}^1 \land \overline{F}_{\text{Analog}}^{\text{All},1,1} \supseteq \overline{F}_{\text{D}}^1 \]  

(relation \( \supseteq \)) on index matrices is presented in (9)). The upper variants of \( \overline{F}_{\text{Analog}}^{\text{All}}(t) \) (which are \( \overline{F}_{\text{Analog}}^{\text{All},0,0}, \overline{F}_{\text{Analog}}^{\text{All},0,1}, \overline{F}_{\text{Analog}}^{\text{All},1,0} \) and \( \overline{F}_{\text{Analog}}^{\text{All},1,1} \)) are constructed in order for all the functioning modes to be handled via respective variants of \( \overline{F}_{\text{in}}^r(t) \) and \( \overline{F}_{\text{in}}^1(t) \) through (73) (all these calculations can be performed in advance).

\[ \overline{F}_{\text{All}}^{\text{Digital}}(t) \odot X\left(\overline{F}_{\text{All}}^{\text{Digital}}(t)\right) = \overline{\Omega} \]  

describes digital signals. The common statement

\[ X\left(\overline{F}_{\text{cu}}^{\text{Digital}}(t)\right) = \text{IM}_{\text{cu},\{\text{r}\}} \left[ - \left( \text{CM}_{\text{U}}, \text{V}_{\text{cu}} \overline{F}_{\text{cu}}^{\text{Digital}}(t) \right)^{-1} \text{CM}_{\text{U},\{\text{r}\}} \overline{F}_{\text{o}}^{\text{Digital}}(t) \right]; \]  

Determines all digital values; here, \( \text{U} \) and \( \text{V}_{\text{cu}} \) are permutations of the first and the second index sets of \( \overline{F}_{\text{cu}}^{\text{Digital}}(t) \), respectively (\( \overline{F}_{\text{cu}}^{\text{Digital}}(t) \) and \( \overline{F}_{\text{o}}^{\text{Digital}}(t) \) are defined similarly to these ones in (63)).
4.2. Model-Solving Algorithm

An algorithm (Algorithm 1), which is presented in case of the considered example, is given below. Its first three steps are implemented once, and its next three steps are performed cyclically until statement $t \leq t_{\text{end}}$ holds.

**Algorithm 1: Single simulation of device functioning**

Step 1: Specifying a time interval and a time step

Let the functioning of the model be simulated in time set (18) with a proper initial time $t_0$ (for a convenience, it is set to zero below) and a proper time step $\Delta t$. It is appropriate for the end model time $t_{\text{end}}$ to be used, too. The choice of $\Delta t$ determines a proper accuracy; Figure 2 shows the results of simulations depending on this choice (relative to a switching interval).

Step 2: Initialization

Initial values corresponding to index sets $\mathcal{I}_{\text{Analog}, 1}$ and $\mathcal{I}_{\text{Analog}, 2}$ must be set. In case of zero values of voltages and amperages at $t_0$ for the power components of the considered DC-DC converter, the following R-IM can be used:

$$\begin{bmatrix}
    u_{\text{D}}^\text{cu} \\
    u_{\text{T}_{\text{out}}}^\text{cu} \\
    i_{\text{L}}^\text{cu} \\
    i_{\text{C}}^\text{cu} \\
    u_{\text{P}_{\text{in}}}^\text{cu} \\
    u_{\text{PWM}_{\text{in}}}^\text{cu}
\end{bmatrix}^\text{T} =
\begin{bmatrix}
    0 \\
    0 \\
    0 \\
    0 \\
    0 \\
    K_{\text{P}} U_{\text{Target}}
\end{bmatrix}; \quad (81)
$$

Here, $u_{\text{P}_{\text{out}}}^\text{cu}(t_0) = u_{\text{PWM}_{\text{in}}}^\text{cu}(t_0)$ holds.

Step 3: Specifying constant R-IMs

Such R-IMs are calculated in advance and explored in the next steps.

Four variants of $\mathcal{F}_{\text{Analog}}(t)$ are described in (76)–(79). The following variants of index matrices from (73) can be presented:

$$\mathcal{B}_{\text{I}, m, n}, \quad \mathcal{J}_{\text{I}, m, n}, \quad m \in \{0, 1\}, \quad n \in \{0, 1\},$$

$$\mathcal{I} \in \{\mathcal{I}_{\text{Analog}, 1}, \mathcal{I}_{\text{Analog}, 2}\}, \quad \mathcal{I}_{\text{Analog}, 1} - \mathcal{I}_{\text{Analog}, 2} \subseteq \mathcal{I}_1 \subseteq \mathcal{I}_2 \left(\mathcal{F}_{\text{cu}}^\text{Analog}(t)\right) - \mathcal{I}_{\text{Analog}, 2}; \quad (82)$$

$$\mathcal{B}_{\text{I}, m, n} \quad \text{and} \quad \mathcal{J}_{\text{I}, m, n} \quad \text{are variants of} \quad \mathcal{B}_{\text{I}} \quad \text{and} \quad \mathcal{J}_{\text{I}} \quad \text{respectively in case of} \quad \mathcal{F}_{\text{Analog}}(t) = \mathcal{F}_{\text{Analog}}^\text{Digital All, m, n}; \quad (83)$$

($\mathcal{I}_1$ contains all indexes related to analog values which should be investigated except for those values which are related to $\mathcal{I}_{\text{cu}, 2}$; the last ones are calculated separately).

Two variants of $\mathcal{F}_{\text{Digital}}(t)$ are described in (60). The following variants of index matrix $X\left(\mathcal{F}_{\text{Digital}}(t)\right)$ can be calculated in advance according to (60) and (81):
\[ X_{\text{Digital}}^{cu,1} = \text{IM} \begin{pmatrix} s_{PWM_{\text{out}}}^{cu} \end{pmatrix} \begin{pmatrix} s_{T_{\text{in}}}^{cu} \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} = \begin{pmatrix} cu^{PWM_{\text{out}}} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \]

\[ X_{\text{Digital}}^{cu,0} = \text{IM} \begin{pmatrix} s_{PWM_{\text{out}}}^{cu} \end{pmatrix} \begin{pmatrix} s_{T_{\text{in}}}^{cu} \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} = \begin{pmatrix} cu^{PWM_{\text{out}}} \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \]

(trivial equivalence \( s_{PWM_{\text{out}}}^{cu}(t) = s_{T_{\text{in}}}^{cu}(t) \) holds).

Step 4: Current time setting

Add \( \Delta t \) to current time \( t \).

Step 5: Calculation of digital values

Generally, these values depend on analog and other digital values. All described analog values are continuous; small values of \( \Delta t \) determine near their values, which can be used like replaceable ones. An interrelation between digital values has been described in (81). For the considered converter, the signal at the input of the presented PWM block is continuous, and \( u_{PWM_{\text{in}}}^{cu}(\psi) \) for current time \( t \) is modeled by either \( u_{PWM_{\text{in}}}^{cu}(t - \Delta t) \) or an earlier value (see (41), where sampling moment \( \psi \) depends on triangular function (42)). The latter and \( u_{PWM_{\text{in}}}^{cu}(\psi) \) determine \( F_{\text{Digital}}^{cu} \) in (60) and, therefore, all digital values through (85) or (86).

Step 6: Calculation of analog values

Analog values which concern time \( t - \Delta t \), as near to respective values for time \( t \), have been used in the previous step. Oppositely, analog values at time \( t \), which are related to conditions of other values (and, therefore, indexed in \( F_{\text{Analog}}^{cu,2} \)), are used directly in this step in order for variants of \( F_{\text{Analog}}^{cu} \) to be chosen. For the considered converter, \( F_{\text{Analog}}^{cu,2} \) has three elements. For example, if for the analog values at time \( t \), which are indexed in \( F_{\text{Analog}}^{cu,2} \) and estimated by

\[
\begin{align*}
 u_{D}^{cu} & | \begin{pmatrix} o \end{pmatrix} \begin{pmatrix} u_{D}(t) \end{pmatrix} \\
 u_{T_{\text{out}}}^{cu} & | \begin{pmatrix} o \end{pmatrix} \begin{pmatrix} u_{T_{\text{out}}}(t) \end{pmatrix} \\
 u_{PWM_{\text{in}}}^{cu} & | \begin{pmatrix} o \end{pmatrix} \begin{pmatrix} u_{PWM_{\text{in}}}(t) \end{pmatrix}
\end{align*}
\]

the inequalities in (76) hold, then the following assignments must be implemented:

\[
\begin{align*}
 B_{\text{F}}^{\text{cu}}(t) & = B_{\text{F}}^{\text{cu},0,0}, \\
 J_{\text{F}}^{\text{cu}}(t) & = J_{\text{F}}^{\text{cu},0,0}
\end{align*}
\]

See (83) and (84) for the upper notations (here, \( F_{\text{Analog}}^{\text{cu,0}} \)). Similar statements can be presented according to (77)–(79). Since all considered analog values are continuous, only one of the cases (76)–(79) can be applied. \( B_{\text{F}}^{\text{cu}}(t) \) and \( J_{\text{F}}^{\text{cu}}(t) \) are used in the following calculations. In case of \( I' = I_{\text{cu,1}}^{\text{Analog}} - I_{\text{cu,2}}^{\text{Analog}} = I_{\text{cu,1}}^{\text{Analog}} \):
Since \( u_C(t) \equiv u_R(t) \), the output voltage of the considered DC-DC buck converter with a PI controller can be investigated. In case of \( I' = I_2^{\text{Analog}}(t) - I_1^{\text{Analog}}(t) \), all analog values can be investigated. On each iteration, elements of \( I' \) can be modified under constraints in (83) in order for proper sets of current voltages and amperages to be estimated.

Step 7: Time check

If \( t + \Delta t \leq t_{\text{end}} \) holds, then Steps 4, 5 and 6 must be performed; otherwise, the algorithm must be terminated.

### 4.3. Single Simulation Results

Figure 2 presents output voltage simulation results for the considered converter. Algorithm 1 is implemented through MATLAB source code (without Simulink to be used). In case of Figure 2b–d, the described model varies on a sampling frequency (relative to a switching frequency). A comparison with the MATLAB Simulink model (see Figure 2a) has been made. The simulations are performed with the following parameters:

\[
E = 20 \text{ V}; \quad U_{\text{Target}} = 10 \text{ V}; \quad R = 100 \Omega; \quad F_s = 400 \text{ kHz};
\]

\[
R_E = 100 \text{ m}\Omega; \quad R_L = 20 \text{ m}\Omega; \quad R_C = 10 \text{ m}\Omega;
\]

\[
U_{\text{out}}^{0} = 500 \text{ mV}; \quad R_T = 20 \text{ m}\Omega; \quad U_{\text{D}}^{0} = 500 \text{ mV}; \quad R_D = 50 \text{ m}\Omega;
\]

\[
L = 10 \mu\text{H}; \quad C = 5 \text{ mF}; \quad K_P = 200; \quad K_i = 200.
\]
Figure 2. Output voltage simulation results for Simulink and the proposed R-IM-based software. Modeling with (a) Simulink; (b) R-IM-based software with ratio $F_d/F_s = 600$; (c) R-IM-based software with ratio $F_d/F_s = 60$; (d) R-IM-based software with ratio $F_d/F_s = 6$.

The full meaning of these parameters is given in (19), (26), (29), (32), (35) and (38)–(40); the uncommented ones are as follows: $U_{\text{Target}}$ — the output target voltage; $K_p$ and $K_i$ — the coefficients for proportional and integral terms in a PI controller.

A step response on the output is presented on Figure 2. The last one displays the fact that large changes in the $F_d/F_s$ ratio keep this step response almost unchanged. Initial values from (82) and time $t_{\text{end}} = 0.9 \text{ ms}$ are used. The graphs are zoomed (see whole graphs in Figure 3) in order for the similarity between cases to be shown more clearly.

Characteristics of the output signal are defined below. The maximum

$$U_{R, \text{max}}^S = \max_{\theta \in [0, t_{\text{end}}]} u_R^S(\theta)$$  \hspace{1cm} (93)

(s denotes the performed simulation) for the whole time interval and the deviation

$$\Lambda_{\text{end}} U_R^S = U_R^S \left( 0.9 t_{\text{end}} \right)$$  \hspace{1cm} (94)

for an end part of this interval are estimated (notation
\[
\Delta U_R^S(t) = \max_{\theta \in [t, t_{end}]} \left| U_R^S(\theta) - U_{Target} \right|
\]  

(95)

is used here. Statements

\[
U_{R, max}^S < 0.03 \text{ V} ; \quad \Delta_{end} U_R^S < 0.01 \text{ V} ;
\]  

(96)

hold in all cases, which are shown in this subsection.

Figure 3. Simulation results for an output voltage, parameter distributions (for “successful” cases) and nested parameters. Intervals in an optimization procedure: (a) at the first iteration; (b) at the second iteration; (c) at the 39th iteration.

5. Optimization

An iterative algorithm, which determines parameter intervals in order for the proper output voltage requirements to be met, is given below (Algorithms 2 and 3).

Algorithm 2: Determining final parameter intervals at once
Step 1: Specifying fixed parameters

Let the input voltage, the target voltage, the output resistance, the switching frequency and the simulation duration be fixed through

\[ E = 20 \text{ V}; \quad U_{\text{Target}} = 10 \text{ V}; \quad R = 100 \Omega; \quad F_s = 1 \text{ MHz}; \quad t_{\text{end}} = 0.11 \text{ ms}; \quad (97) \]

(in this case, \( 0.9 \ t_{\text{end}} = 0.1 \text{ ms} \)), and let the additional parameters in (91) and (92) have the same constant values.

Step 2: Specifying output voltage requirements

For the described DC-DC converter, the following requirements are taken (see (94)–(96)):

\[ U_{R, \text{max}}^S < 0.2 \text{ V} ; \quad \Delta U_{R, \text{end}}^S < 0.01 \text{ V} ; \quad (98) \]

Step 3: Specifying initial parameter intervals

Let the parameters from (93) vary in order for minimal values of \( U_{R, \text{max}}^S \) and \( \Delta U_{R, \text{end}}^S \) to be obtained, and let their initial values be in the following ranges:

\[ L = 0.2 + 2 \mu\text{H}; \quad C = 0.1 + 1 \text{ mF}; \quad K_p = 0 + 20; \quad K_i = 0 + 500. \quad (99) \]

Step 4: Simulations with constant parameter intervals

Simulations, which use random uniformly distributed values according to (100), must be implemented. They are performed until the number of “successful” ones (in this context, the requirements from (99)) is lower than \( n_{\text{Successful}} \). In the present paper,

\[ n_{\text{Successful}} = 50 \quad (100) \]

(see Figure 3). In case of the initial intervals, the number of all simulations is 939 (the results are presented in Figure 3a). Nested parameter intervals have been used in this algorithm, and the number of simulations presented in Figure 3b is 214 (the number of “successful” ones is equal to \( n_{\text{Successful}} \), too). The next respective numbers of simulations are 185, 178 and 113. The 23rd simulation has 50; i.e., it is equal to \( n_{\text{Successful}} \). However, 38 variants of parameter intervals are used in order for 10% parameter tolerances to be obtained (see (108)–(110)). Distributions for “successful” simulations are presented in the second and third columns in Figure 3.

Step 5: Determining nested intervals

The following intervals can be defined:

\[ \left[ \omega_{\text{min}}, \omega_{\text{max}} \right], \quad \omega \in \Omega = \{ C, \ L, \ K_i, \ K_p \} \quad (101) \]

The intervals are updated according to “successful” simulations only. Let

\[ \left\{ \omega_1, \omega_2, \ldots, \omega_{n_{\text{Successful}}} \right\}, \quad \left( i < j \Rightarrow \omega_i \leq \omega_j, \quad i, j \in 1, 2, \ldots, n_{\text{Successful}} \right) \quad (102) \]

present the values of parameter \( \omega \) in such simulations in an ordered manner.

In case of the initial ranges, which are presented in (100) and Figure 3a, new intervals are formed by

\[ \left[ \omega_{\text{min}}, \omega_{\text{max}} \right] = \left[ \omega_1, \omega_{n_{\text{Successful}}} \right], \quad \omega \in \Omega = \{ C, \ L, \ K_i, \ K_p \} . \quad (103) \]

They are shown with red lines in Figure 3a and are used in the next iteration (see the ranges of the parameters in Figure 3b).

In case of the next (non-initial) intervals, a single element \( \mu \) from \( \Omega \) is selected in order for the new interval \( \left[ \mu_{\text{min}}, \mu_{\text{max}} \right] \) to be set (the ranges of the other parameters remain the same). The percentage of values
μ1, μ2, ..., μn_{Successful}, which do not belong to [μ_{min}, μ_{max}], is denoted by λ. In the present consideration, λ is constant:

\[ \lambda = 10\%. \] (104)

One of the endpoints of the respective parameter interval [μ^{old}_{min}, μ^{old}_{max}] from the previous iteration is updated in [μ_{min}, μ_{max}]. A pair containing an interval and a ratio is defined for each ω ∈ Ω:

\[ (\omega_{min}, \omega_{max}), \ \frac{ζ}{ω} = \begin{cases} \left( \omega_{min}, \omega_{min} (1 - \lambda) n_{Successful} \right), & \text{if } \frac{ζ}{ω} \text{ begin } < \frac{ζ}{ω} \text{ end}; \\ \left( \omega_{max}, \omega_{max} + \left( \omega_{max} - \omega_{min} \right) \lambda n_{Successful} \right), & \text{otherwise}; \end{cases} \]

where \(|x|\) denotes the integer part of x. Parameter μ is selected in order for the following equality to be satisfied:

\[ \sigma_{ψ} ≥ \frac{ζ_{μ} \min - ζ_{max}}{ζ_{μ} \max - ζ_{min}} ≥ 1 ≤ 1.15. \] (105)

Inequality \( \sigma_{ψ} > \frac{1 + \sigma}{1 - \sigma} \) is discussed in Step 6. L_{max} is updated at the second iteration to 1.405 μH (since \( \lambda = 10\% \) of 50 is equal to 5, the parameter range is reduced on the base of the highest 5 values on induction—see the second graph on Figure 3b).

Step 6: Termination check

In practice, the parameters can be realized with given tolerances. In the present paper,

\[ \sigma = 10\%. \] (107)

is the used tolerance for each parameter; it means that only ranges, which satisfy

\[ \sigma_{ψ} = \frac{ω_{max}}{ω_{min}} > \frac{1 + \sigma}{1 - \sigma} = \frac{1.1}{0.9} = 1.22, \quad ω ∈ Ω = \{ C, \ L, \ K_1, \ K_p \}; \] (108)

Are applied. Otherwise, if inequality (109) does not hold for any new interval in Step 4, the algorithm is terminated. For the presented example, after the last allowed parameter changes, the ranges of the parameters (see Figure 3c) are

\[ L = 0.974 + 1.205 \mu H; \quad C = 0.4037 + 0.5003 \mu F; \quad K_p = 14.76 + 18.32; \quad K_1 = 58.88 + 72.42; \] (109)

Algorithm 3: Optimizing final parameter intervals

Step 1: Setting a number of simulation series and a current simulation series index

A number of simulation series m in the present paper has been set by

\[ m = 10 \] (110)

in order for sufficiently distinguishable graphics to be obtained (see Figure 4b,c). The current simulation series index is initialized by

\[ i = 1 \] (111)
Step 2: Applying an algorithm with a fixed input voltage and a fixed output resistance

Algorithm 2 must be applied.

Step 3: Verifying results

The final parameter intervals obtained in Step 2 through Algorithm 2 have been used here. Normal distributions based on the final parameter intervals (notation $[\omega_{\min}, \omega_{\max}]$ from (102) are used here) with means and standard deviations

$$
\mu_\omega = \frac{\omega_{\max} + \omega_{\min}}{2}, \quad \sigma_\omega = \frac{\omega_{\max} - \omega_{\min}}{6}, \quad \omega \in \Omega = \{C, L, K_1, K_2\},
$$

respectively, must be constructed and applied. $\sigma_\omega$ ensures 99.7% of the data within $[\omega_{\min}, \omega_{\max}]$. Simulations with a fixed input voltage and a fixed output resistance are performed in order levels

$$
\Delta U^i_R (\theta) = \max_{s \in S} = \{ s_{i,1}, s_{i,2}, \ldots, s_{i,n} \} \Delta U^i_R (\theta), \quad \theta \in [t_0, 0.9 t_\text{end}]
$$

under notation (96) to be obtained ($s_{i,1}, s_{i,2}, \ldots, s_{i,n}$ are separate simulations); in the present paper,

$$
n = 10000.
$$

The magenta dotted line on Figure 4a presents the resulting curve from (114).

Step 4: Increasing the current number of simulation series

Increment the current simulation series index by 1.

Step 5: A check on the current number of simulation series

If $i \leq m$ holds, then Steps 2, 3 and 4 must be performed; otherwise, the execution of these steps must be terminated.

Step 6: Choosing an optimal parameter intervals

It is obvious that $m$ simulation series have been performed in the previous steps. Different levels (114) have been obtained. Optimal final parameter intervals, which correspond to a proper

$$
k \in \{1, 2, \ldots, m\},
$$

based on the simulation series with the lowest level

$$
\Delta U^k_R (0.9 t_\text{end}) = \min_{i \in \{1, 2, \ldots, m\}} \Delta U^i_R (0.9 t_\text{end})
$$

must be chosen; here

$$
U^S_{R, \max} = \max_{s \in S} = \{ s_{i,1}, s_{i,2}, \ldots, s_{i,n} \} U^S_{R, \max}.
$$

In the present paper

$$
U^\text{fixed}_{R, \max} = 10.2 \text{ V}.
$$

This process is shown in Figure 4b,c, where 10 lines (one continuous and nine dotted ones) present variants of the dotted curve from Figure 4a. The red ones represent simulations, for which the following statement holds:

$$
U^S_{R, \max} > U^\text{fixed}_{R, \max}.
$$

The blue ones represent simulations, for which
Mathematics 2023, 11, 4756

\[ U_{R, \text{max}}^S \leq U_{R, \text{max}}^{\text{fixed}}, \]

and the line, which corresponds to the simulations with an optimal parameter intervals (see (117)), is continuous. Generally, the discussed example gives the following constraints:

\[ U_{R, \text{max}}^S \leq U_{R, \text{max}}^{\text{fixed}}; \quad \Delta U_{R}^{S'}(0.9 t_{\text{end}}) < 0.01 \text{ V}; \]

in case of 10,000 simulations for normally distributed parameters from intervals with at least 10% tolerance for \( L, C, K_P, \) and \( K_I \).

Figure 4. Tolerance simulation results according to Algorithm 3: (a) on Step 3; (b) on Step 6; (c) on Step 6 in a zoom mode.

The final parameter intervals have been obtained with a fixed input voltage of 20 V and a fixed output resistance of 100 Ω. Such intervals (based on these two values) are used below, but different input voltages and output resistances are used in Algorithm 3. Graphs of characteristic \( \Delta U_{R}^{S'}(0.9 t_{\text{end}}) \) depending on the last two parameters (with one constant parameter) are given in Figure 5.

Algorithm 3 can be modified in order for various input voltages and output resistances to be used simultaneously, too. In this case, random normally distributions on these parameters with means and standard deviations

\[ \mu_\omega = \frac{\omega_{\text{max}} + \omega_{\text{min}}}{2}, \quad \sigma_\omega = \frac{\omega_{\text{max}} - \omega_{\text{min}}}{6}, \quad \omega \in \Omega = \{E, R\}, \]

(122)
can be used in Step 3 of the algorithm. The respective results are presented in Figure 6.
Figure 5. Output voltage tolerances for various input voltages or output resistances: (a) $E = 11, 13, 15, 17, 19 \text{ V}; R = 100 \Omega$ (colored from red to blue); (b) $E = 21, 23, 25, 27, 29 \text{ V}; R = 100 \Omega$ (colored from blue to red); (c) $E = 20 \text{ V}; R = 1, 3, 10, 30, 100 \Omega$ (colored from red to blue).

Figure 6. Tolerance simulation results according to Algorithm 3 with various input voltages and various output resistances ($E = 18 \pm 22 \text{ V}; R = 50 \pm 150 \Omega$) (a) on Step 6; (b) on Step 6 in a zoom mode.

The notations in Figure 6a are the same as the ones in Figure 4b where 10 lines (one continuous and nine dotted ones) present 10 variants. Since three graphs in Figure 6a determine tolerances on $\Delta U_R^S \left(0.9 t_{end}\right)$ greater than 0.018 V, only seven graphs are presented in Figure 6b. The discussed example gives the following constraints:

$$\left< k_{S_R, max} \right> < 0.2 \text{ V}; \quad \Delta U_R^S \left(0.9 t_{end}\right) \approx 0.012 \text{ V}. \quad (123)$$

also in case of 10,000 simulations. Normal distributions of the input voltage and the output resistance under conditions (123) with parameters defined by

$$E_{\text{min}} = 18 \text{ V}; \quad E_{\text{max}} = 22 \text{ V}; \quad R_{\text{min}} = 50 \Omega \quad R_{\text{max}} = 150 \Omega \quad (124)$$

are used in these calculations.

for a Buck DC-DC converter without a PI controller $\left(L = 167 \mu\text{H}; C_{\text{min}} = 188 \text{nF}\right)$.

In [20], conventional methods estimate the inductance $L$ and the minimal capacitance $C_{\text{min}}$ in a similar Buck DC-DC converter in the following manner:
\[
L = \frac{U_{\text{Target}} (E - U_{\text{Target}})}{\Delta i_L F_s E}, \quad C_{\text{min}} = \frac{\Delta i_L}{8 F_s \Delta u_R}, \quad \Delta i_L = (0.2 + 0.4) i_R. \quad (125)
\]

In case of \( \Delta u_R = 0.02 \) V, these formulas give the following results:

\[
L = 125 + 250 \, \mu\text{H}; \quad C_{\text{min}} = 125 + 250 \, \text{nF}; \quad (126)
\]

which significantly differ from the values given in Figure 3c. Indeed, \( K_p \) and \( K_i \) have not been taken into account here (there is not a PI controller in the Buck DC-DC converter presented in [20]). Output voltage simulation results for the scheme, which is considered in the present paper, are shown on Figure 7. They are based on (126) and (127) with \( \Delta i_L = 0.3 i_R \) and \( C = 5 \, C_{\text{min}} \); Algorithm 3 with fixed values of \( L \) and \( C \) has been performed in the calculation of optimal values of the coefficients for the proportional and the integral terms in the PI controller.

![Figure 7. Output voltage based on an inductance and a minimal capacitance both determined.](image)

The main advantage of PI controllers is that they can be implemented without detailed knowledge of systems. Algorithms of tuning PI controllers are presented in [21]. They consider devices that are regulated as black boxes. Therefore, they cannot change parameters \( L \) and \( C \). Generally, estimations of \( K_p \) and \( K_i \) are empirically based on the simulation of a large number of simulations in these conventional algorithms. There exist well-established approaches to derive initial coefficients. The Ziegler–Nichols method is used below in order for a PI controller to be tuned [21,22]. The parameters obtained in Algorithm 3 are used (see Figure 3c):

\[
L = 1.1 \, \mu\text{H}; \quad C = 450 \, \mu\text{F}. \quad (127)
\]

A proportional (P) controller with coefficient \( K_p = 32 \) is modeled at first; the value of \( P \) starts from 0 and increases until the system starts to show consistent and stable oscillation \( (\Delta u_R = 0.01 \) V). The next proportional controller is modeled with \( K_p = 0.45 \times 32 = 14.4 \). The final controller is proportional–integral with coefficients \( K_p = 14.4 \) and \( K_i = 172.8 \), which are close to these ones in Figure 3c. The results are presented in Figure 8.
Figure 8. Tuning PI controller according to Ziegler–Nichols method. Graphics with: \( K_p = 32 \), \( K_i = 0 \) (red one); \( K_p = 14.4 \), \( K_i = 0 \) (blue one); \( K_p = 14.4 \), \( K_i = 172.8 \) (green one); (a) maximal values; (b) oscillation at time 0.3 ms.

6. Simulation Duration

Three software applications modeling the discussed power converter are as follows: the Simulink model from Figure 1 and two index matrix-based models: the .NET one (which is described in [1]) and MATLAB one (which implements the algorithms from the previous two sections using MATLAB source code without using Simulink). Their runtimes are shown in Table 1. There is a slight difference in the .NET model: ideal switches have been used; all programming methods defined in the .NET software are presented in [1]. A single simulation of the converter operation is taken into account when calculating the durations. Each runtime considers both the time to run the simulation and the time to configure the model. For example, the Simulink model runtime is formed as a sum of execution times of two functions: set_params() and sim().

<table>
<thead>
<tr>
<th>Model Time Duration (ms)</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime in Simulink (ms)</td>
<td>622</td>
<td>728</td>
<td>848</td>
<td>943</td>
<td>1042</td>
</tr>
<tr>
<td>Runtime in the authors’ .NET software (ms)</td>
<td>18.6</td>
<td>23.3</td>
<td>27.7</td>
<td>33.1</td>
<td>37.5</td>
</tr>
<tr>
<td>Runtime in the authors’ MATLAB software (ms)</td>
<td>2.7</td>
<td>6.4</td>
<td>9.8</td>
<td>12.8</td>
<td>16.2</td>
</tr>
</tbody>
</table>

Runtimes for Simulink and .NET models determine the following linear regressions:

\[
t_{\text{run, Simulink}}^{\text{Simulink}} = 520 + 1055 t_{\text{model time}, \text{Simulink}};
\]

\[
t_{\text{run, .NET}}^{\text{.NET}} = 14 + 48 t_{\text{model time}, .NET};
\]

where all durations are measured in milliseconds. These expressions indicate the presence of a significant configuration time in these two models, especially for the first one. Here and below, \( t_{\text{model time}} \) denotes a duration of model time. For the third model:

\[
t_{\text{run, MATLAB}}^{\text{MATLAB}} = 33 t_{\text{model time}, \text{MATLAB}};
\]

i.e., the runtime is proportional to the model time duration (this fact is conditioned by the pre-calculation of matrices—see Sections 4.1 and 4.2). There are commensurate coefficients (48 and 33 approximately) in the .NET and MATLAB models.
7. Conclusions

The proposed new definition of an R-IM product guarantees convenient index matrix equations with zero R-IM on the right-hand side. Index sets can determine which analog values should be calculated at any time in the model; additional values do not affect the calculation procedure.

The simulations produce results that are virtually identical to those from Simulink. At the same time, large changes in the ratio between the sampling rate and switching frequency keep the output characteristics almost unchanged. Threshold voltages (potential barriers) and dynamic resistances have been successfully included in the models.

The performed optimization procedures based on exhaustive searches with nested intervals give very good results—small peaks and small deviations after settling. It is shown that when the choice of inductance and capacity is made, the tuning of the PID controller by means of the proposed algorithm is close to that which has been achieved with conventional methods.

The discussed MATLAB implementation based on the pre-computation of indexed matrices and the selection of variants results in small computation times.

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 KTI-06-H57/7/16.11.2021, and the APC was funded by KTI-06-H57/7/16.11.2021.

Data Availability Statement: Data are contained within the article.

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”, KTI-06-H57/7/16.11.2021, Bulgarian National Scientific Fund.

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

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.