IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 24, 2019, accepted February 8, 2019, date of publication February 15, 2019, date of current version March 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899316

Petri Net-Based Specification of Cyber-Physical Systems Oriented to Control Direct Matrix Converters With Space Vector Modulation

REMIGIUSZ WI NIEWSKI, (Member, IEEE), GRZEGORZ BAZYDŁO, PAWEŁ SZCZE NIAK, (Senior Member, IEEE), AND MARCIN WOJNAKOWSKI
Institute of Electrical Engineering, University of Zielona Góra, 65417 Zielona Góra, Poland

Corresponding author: Paweł Szcześniak (p.szczesniak@iee.uz.zgora.pl)

ABSTRACT This paper proposes a Petri-net-based specification of cyber-physical systems dedicated to the control of a direct matrix converter with space vector modulation (SVM) and transistor commutation. The technique employed is further applied for hardware implementation in a programmable logic device [namely, field-programmable gate array (FPGA)]. Contrary to the traditional SVM computation methods, concurrency aspects of the digital devices are highly utilized in the presented solution. Therefore, the hardware system is specified by a live and safe Petri net, which is based on the parallelism. Moreover, such a specification can be easily analyzed and verified against the structural properties in order to avoid formal errors and prototyping mistakes (such as deadlocks or non-reachable states). The proposed idea is illustrated by a case-study example of the real prototype of the SVM algorithm. The system has been specified by a live and safe Petri net, analyzed, verified, and finally implemented in the FPGA device. The obtained results of the physical implementation are presented and discussed.

INDEX TERMS Concurrency, FPGA, cyber-physical system, matrix converter, Petri net, SVM, Verilog.

I. INTRODUCTION

A cyber-physical system (CPS), [1], [2], creates a set of components that interact with each other, integrating cybernetic aspects with physical processes. The operation of such a system is, by nature, concurrent, enabling the execution of multiple operations simultaneously. For example, one cyber-physical system is a power electronic converter, which is used to control digital circuits and must interact with the physical environment. Increasingly, digital control systems for power converters must meet high requirements for advanced modulation and commutation process, as well as control of executive devices. Additionally, it is possible to exchange information about the state of the converter or its failures with the physical environment. The development of converters is related to the progress of control systems and user interfaces as well as power semiconductor components. All these require the use of sophisticated hardware specification algorithms.

Development of power semiconductor devices and the increase in the areas of their application has brought on the development of complex converter configurations and various control schemes [3]. The main stage of the power converter is usually conceived as a combination of elementary semiconductor power devices (transistors and diodes) with passive reactance elements (capacitors, inductors or transformers). Recent advances in semi-conductor technology have led to an increase in the transistor switching frequencies in modern power converters, up to a hundred kilohertz (by application of Silicon Carbide (SiC) [4], [5] or Gallium nitride (GaN) [6] transistors). Increasing switching frequency makes possible a reduction in the size and weight of the reactance elements as well as of the entire converter. At the same time, the number of switching devices is increasing in modern power converters. High switching frequencies, a large number of power switches, as well as a complex control strategy, make necessary the use of parallel processing techniques in the control loops.

An example of a power electronic converter with a large number of power electronics switches and sophisticated control schemes is the matrix converter (MC) [7]. For decades,

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Gomes.

the MC has been an attractive topology for three-phase AC-AC power conversion.

The specificity and complexity of the MC control scheme results directly from its construction as a direct frequency converter. The MC main power stage forms a matrix of bi-directional power electronic switches connected between input and output terminals. Additionally, for good performance, the MC should have a source low pass LC filter, the purpose of which is the minimization of the high frequency components in the input current [8]. Distinctive switch configurations require the generation of appropriate pulse width modulation (PWM) signal patterns for the control of individual transistors with a safe commutation strategy and optimized switching power losses [9], [10]. Of the many existing modulation strategies MCs, the space vector modulation (SVM) methods are the most widespread. Additionally, to ensure the safe switching of transistors and to minimize the losses generated during switching, different switching sequences, called commutation strategies are used [9]. In the scientific literature, commutation strategies are known in which the switching sequences are defined depending on the direction of the load current or voltage on the switch. Both processes, modulation of the control function and commutation of transistors require relatively large computational resources for control devices such as digital signal processors (DSPs) or microcontrollers (μC). In addition, the converter works with a master control algorithm that results from its specific application (Fig. 1). For example, a MC can be a power supply for a variable speed drive system, a power electronic interface for distributed AC energy sources, or a part of an electric power conditioner [11]. Depending on the complexity of external algorithms, additional calculations and transformations as well as measurements of external signals are required. This further adds to the demand for available computing power in control devices. At the same time, the process of modulating the control function and commutation of transistors has a constant structure regardless of the type of application. This relatively complex computation fragment of the entire control algorithm (marked with a red square in Fig. 1.) can be implemented and optimized in CPLD (complex programmable logic devices) or FPGA (field programmable gate array) devices. Considering the above, there is a need to implement the discussed algorithms in FPGA devices and to develop procedures for optimizing and verifying the correctness of the code for the implemented algorithms.

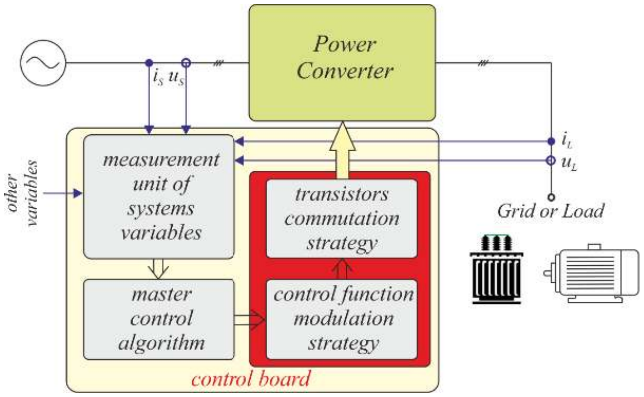
Power
Converter
i_S u_S
i_L
u_L
Grid or Load
other variables
measurement unit of systems variables
transistors commutation strategy
master control algorithm
control function modulation strategy
control board

FIGURE 1. General structure of the power electronic converter control system.

A. OVERVIEW OF EXISTING SOLUTIONS

Nowadays, the most common way to implement digital modulation and commutation algorithms in the MC is to use complex microprocessor systems, often combining DSP and FPGA technologies [12]. Design solutions for research prototypes often use rapid prototyping and diagnostic platforms based on a programming environment equipped with ready-made block control functions with a dedicated real-time hardware simulator, e.g. DSpace, Opal-RT [13], [14]. In the MC the master control and modulation algorithm are in most cases embedded in the DSP, while the FPGA is solely used to execute the functions associated with commutation issues [12]. It should be noted that the DSP is limited in the number of instructions per sampling cycle. This is especially important for very high switching frequencies of several dozen (several hundred) kilohertz, where the time required for performing algorithmic calculations is very short [15]. In addition, the amount of PWM digital signals needed, related to the number of controlled transistors, also significantly increases the number of calculations needed to perform.

Summarizing these considerations, it can be concluded that the use of novel power semiconductors e.g. SiC transistors in matrix converters offers great opportunities associated with reduction of commutation losses and increase in the power density [15], [16].

At the same time, the implementation of a complex SVM algorithm and commutation strategy with very short computing times resulting from a high switching frequency can be difficult to implement in classical DSP systems. Therefore, hardware implementations of the SVM algorithm using programmable logic devices have become more popular. FPGA devices due to their flexibility, parallelism, processing speed and reconfigurability are often the best choice of the implementation platform in various fields of industry and science [17], [18] including SVM algorithm realization [7], [19]–[21]. Generally, there are two main approaches to SVM algorithm hardware implementation. The first one uses a hybrid of connected DSP and FPGA devices, while the second is based on the FPGA only.

In [19] a typical hybrid DSP-FPGA approach is presented. The DSP module calculates parameters and selects modulation mode, while the programmable device is used for PWM pulse generation. The advantage of such an approach is a low-cost system architecture, but the use of external tools to model the FPGA-part can be seen as a drawback. Using specialized tools (e.g., Matlab/Simulink) forces additional conversions which causes inefficient hardware utilization of the FPGA device. Moreover, the aspects of concurrent computation are not considered in the paper.

A similar DSP-FPGA approach, but without any data-transfer bus, is proposed in [20]. Another advantage of the proposed method is the use of the parallelism feature of FPGAs. Unfortunately, the presented approach requires advanced design knowledge of various design and implementation tools. Moreover, the implementation of the whole SVM algorithm in the FPGA is not considered.

The second group of the discussed hardware implementation techniques uses only the FPGA device (without DSP) to implement the SVM algorithm. It is worth noting that such an implementation (especially when the algorithm uses advanced mathematical functions, e.g., trigonometric functions [22]) is much more difficult because of the severe restrictions of the synthesis tools (e.g., Xilinx ISE Design Suite) [18]. Nevertheless, application of the FPGA results in benefits that cannot be achieved by the traditional DSP, such as parallelism and computation speed.

Using System-On-a-Chip (SoC) for hardware implementation of an SVM algorithm is presented in [7]. The authors show the benefits of using an embedded processor (a part of FPGA from Xilinx Virtex-4 family) instead of a traditional DSP. As the authors state, using an FPGA as the implementation platform results in a very high modulation frequency and shortens the processing time. Furthermore, the whole system is integrated in one single chip. Unfortunately, the presented approach requires conversion to the VHDL code and the use of external tools (e.g., Matlab/Simulink, ISE-EDK). Moreover, utilization of FPGA with integrated processors increases the final cost of the system.

Gulbudak and Santi [21] propose an FPGA-based model predictive controller (MPC) as a matrix converter. The advantages of the presented approach are the use of an FPGA (without DSP) for computation only and utilization of the concurrent feature of the FPGA (some computation tasks are performed simultaneously). Unfortunately, the presented solution is shown at a very general level without the important information regarding the implementation methodology, such as modeling methods or use of special tools or converters.

B. MOTIVATION AND PROBLEM FORMULATION

Summarizing the above discussion, it is worth noting, that a significant trend in the use of programmable logic devices (especially FPGA) as a hardware implementation platform for the SVM algorithm can be observed. The main advantages of FPGAs, such as the parallel task execution feature and very high processing speed, result in the growing popularity of FPGA use in the SVM hardware implementation. Unfortunately, existing solutions usually need external tools (mostly Matlab/Simulink), which requires further translations into the hardware description languages. Moreover, very often the parallel feature of digital devices is not fully exploited. Finally, most of the approaches are based on a FPGA-DSP hybrid structure, where the traditional DSP is responsible for mathematical operations execution.

The paper tries to face the problems mentioned above. The proposed SVM approach is oriented toward the cyber-physical system implemented in a single FPGA. Contrary to the existing solutions, the proposed approach neither requires any additional convertors nor specialized external tools.

There are several approaches for representing the SVM algorithm. The most popular one applies concurrency flowcharts, or block diagrams supplemented by mathematical equations. However, it seems that more efficient options rely on the use of a specialized modeling language, such as Unified Modeling Language [23] or Petri nets [24], [25]. This paper applies safe and live Petri nets to specify the system. Such an approach results in several benefits. First of all, the proposed technique greatly exploits the parallel features of Petri-net-based specification, since most of the operations are performed simultaneously.

Moreover, Petri nets are supported by various analysis and verification methods [26]–[35] which can be applied in order to ensure that the model is formally correct before going on with the next steps of the design process (description on HDLs, synthesis, implementation). It is worth mentioning that the analysis of the reliability and robustness of the system is also possible [27], [28], [36]–[38], as well as the application of the model checking techniques [33], [39], or even detailed analysis of concurrency and sequentiality relations in the system [40].

Furthermore, dedicated decomposition methods (designed specifically for Petri net models) can be applied in order to split the algorithm into several modules [26], [41], [42]. Moreover, each of those modules can be implemented as an independent device, constituting a distributed system [41]–[43]. Additionally, Petri nets allow a comfortable prototyping framework/method enabling further dynamic partial reconfiguration of the targeted FPGA device [18]. This feature is very useful in the case of systems that require modification of part of the system without shutting down the whole system [44]. In other words, a separated part of the algorithm (module) with changed specification can be easily replaced without interruption of other modules [18], [44].

Finally, it should be mentioned that there are several types of Petri nets, e.g., Colored Petri nets, Ordinary Petri nets, Unbounded Petri nets, Hybrid Petri nets, S3PR nets, and interpreted Petri nets [18], [27]–[29], [41], [45]–[50]. Each of them contains unique features and thus can be applied in different areas, such as system engineering, concurrent control system implementation in digital devices, distributed systems or flexible manufacturing systems.

In this paper we shall use live and safe Petri nets, which are very efficient for control process modeling [41], [43]. Furthermore, such nets can relatively easily split the system into sequential components in order to create a distributed system, or to apply the idea of the FPGA dynamic partial reconfiguration feature [18], [41].

C. MAIN CONTRIBUTIONS OF THE PAPER

The paper proposes application of Petri net theory to the specification of cyber-physical systems designed to control a direct matrix converter with space vector modulation and transistor commutation. The presented approach involves several steps. Initially, the SVM algorithm is described by an interpreted Petri net. Such a specification naturally reflects the parallel relations of the designed system. Moreover, control systems described with Petri nets can be easily decomposed into sequential modules and allows for application of dedicated analysis and verification methods to check the main properties of the system (liveness, safeness, markings, classification, number of the minimal SMCs that cover the net, etc.).

In short, the main contributions can be summarized as follows:

- A Petri-net based specification of a cyber-physical system designed to control an MC with SVM and modulation is proposed. The method is oriented toward implementation in an FPGA device.
- The proposed idea involves the main benefit of Petri nets and digital devices, that is, their parallelism. It means that several tasks are calculated simultaneously, which is not possible in the case of traditional techniques (based on the DSPs), which execute operations sequentially.
- The method applies to live and safe Petri nets, which permit the analysis and verification of the designed system in order to avoid errors and mistakes at the specification stage.
- The applied specification model allows further decomposition of the system into the sequential modules (state machine components, SMCs). These modules can be applied to form a distributed system or to the dynamic partial reconfiguration of the system.
- The presented idea is illustrated by a real-life case study example. The presented cyber-physical system has been specified by a live and safe Petri net, described in the Verilog language and finally implemented in an FPGA.

The structure of the paper is organized as follows. Sections II and III introduce matrix converters and Petri nets, respectively. The main idea of the proposed technique is shown in Section IV, while the subsequent section presents the case-study example. The results of analytical, numerical and hardware verification of the system are presented in Section VI. Finally, Section VII concludes the paper and shows possible directions for future research.

II. MATRIX CONVERTER TOPOLOGY AND SPACE VECTOR MODULATION ALGORITHM

The direct matrix converter (MC) is a power converter topology based on controlled bidirectional switches, as shown in Fig. 2 [7]–[10].

The MC is an alternative to the conventional back-to-back converter. The MC can offer significant advantages in terms of weight and volume (high power density) and it does not require any energy storage element. The matrix converter input/output voltage or current relationships are defined by the equations (1) and (2) [9].

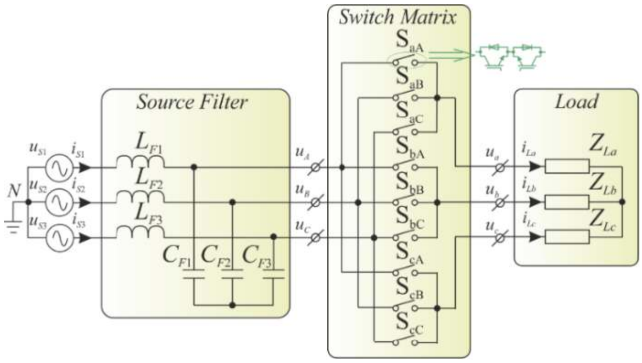

FIGURE 2. Topology of six-bidirectional switch matrix converter with source LC low-pass filter.

$$\begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} = \begin{bmatrix} s_{aA} & s_{bA} & s_{cA} \\ s_{aB} & s_{bB} & s_{cB} \\ s_{aC} & s_{bC} & s_{cC} \end{bmatrix} \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_b(t) \end{bmatrix}$$

$$\begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} = \begin{bmatrix} s_{aA} & s_{aB} & s_{aC} \\ s_{bA} & s_{bB} & s_{cB} \\ s_{cA} & s_{cB} & s_{cC} \end{bmatrix} \begin{bmatrix} i_A(t) \\ i_B(t) \\ i_C(t) \end{bmatrix} \tag{1}$$

$$v_{ABC}(t) = \boldsymbol{D} v_{abc}(t) i_{abc}(t) = \boldsymbol{D}^T i_{ABC}(t) \tag{2}$$

Different modulation strategies for matrix converters, including the low frequency modulation function (Venturini and the scalar method), space vector modulation (SVM) techniques and predictive control, have been previously reported in science literature [10]. The SVM algorithm is most often used in matrix converter applications because of its favorable control properties. The SVM is based on the instantaneous space vector representation of MCs signals.

The transform from the three-phase line-to-line output voltages $u_{ab}(t)$, $u_{ba}(t)$, $u_{ca}(t)$ to the complex vector $\underline{u}_{OL}$ is given by (3) while the transformation of the MC input currents $i_A(t)$, $i_B(t)$, $i_C(t)$ is given by (4).

$$\underline{u}_{OL} = \frac{2}{3}\left(u_{ab}(t) + e^{-\frac{j2\pi}{3}} u_{bc}(t) + e^{\frac{j2\pi}{3}} u_{ca}(t)\right) = |u_{OL}| e^{j\alpha_{OL}(t)} \tag{3}$$

$$\underline{i}_S = \frac{2}{3}\left(i_A(t) + e^{-\frac{j2\pi}{3}} i_B(t) + e^{\frac{j2\pi}{3}} i_C(t)\right) = |i_S| e^{j\beta_I(t)} \tag{4}$$

In the MC with SVM the output voltage reference space-vectors $\underline{u}_{OL}$ and input current reference space-vectors $\underline{i}_S$ are constructed by selecting four ‘‘active’’ and one ‘‘zero’’ switch combinations in the sequence period T_{Seq} [9]. Allowed ‘‘active’’ and ‘‘zero’’ switch configurations and the corresponding values of output voltages and grid currents for individual output or input terminals are listed in Table 1.

Voltage and current vectors with non-zero amplitude are created during ‘‘active’’ switch configurations. The zero configurations are applied to complete time interval T_{Seq}. In Fig. 3 the output voltage and input current vectors corresponding to the ‘‘active’’ switch configurations are shown,

TABLE 1. Switch configuration in the MC SVM algorithm.

No	a	b	c	u_{ab}	u_{bc}	u_{ca}	i_A	i_B	i_C
0_A	A	A	A	0	0	0	0	0	0
0_B	B	B	B	0	0	0	0	0	0
0_C	C	C	C	0	0	0	0	0	0
+1	A	B	B	u_{AB}	0	-u_{AB}	i_a	-i_a	0
-1	B	A	A	-u_{AB}	0	u_{AB}	-i_a	i_a	0
+2	B	C	C	u_{BC}	0	-u_{BC}	0	i_a	-i_a
-2	C	B	B	-u_{BC}	0	u_{BC}	0	-i_a	i_a
+3	C	A	A	u_{CA}	0	-u_{CA}	-i_a	0	i_a
-3	A	C	C	-u_{CA}	0	u_{CA}	i_a	0	-i_a
+4	B	A	B	-u_{AB}	u_{AB}	0	i_h	-i_h	0
-4	A	B	A	u_{AB}	-u_{AB}	0	-i_h	i_h	0
+5	C	B	C	-u_{BC}	u_{BC}	0	0	i_h	-i_h
-5	B	C	B	u_{BC}	-u_{BC}	0	0	-i_h	i_h
+6	A	C	A	-u_{CA}	u_{CA}	0	-i_h	0	i_h
-6	C	A	C	u_{CA}	-u_{CA}	0	i_h	0	-i_h
+7	B	B	A	0	-u_{AB}	u_{AB}	i_c	0	-i_c
-7	A	A	B	0	u_{AB}	-u_{AB}	0	i_c	-i_c
+8	C	C	B	0	-u_{BC}	u_{BC}	-i_c	i_c	0
-8	B	B	C	0	u_{BC}	-u_{BC}	-i_c	0	i_c
+9	A	A	C	0	-u_{CA}	u_{CA}	0	-i_c	i_c
-9	C	C	A	0	u_{CA}	-u_{CA}	i_c	-i_c	0

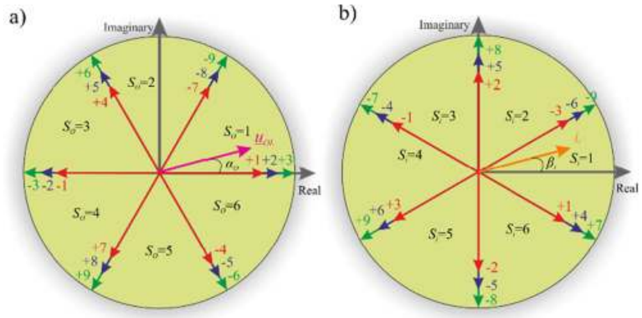

FIGURE 3. Graphical interpretation of: a) sectors and direction of the output non-zero voltage vectors, b) sectors and directions of the input line non-zero current vectors.

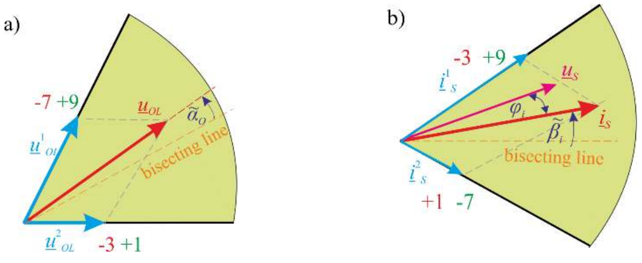

FIGURE 4. Synthesis description of: a) reference output voltage vector $\underline{u}_{OL}$, b) reference input current vector $\underline{i}_S$.

respectively. The complex coordinate system is divided into six sectors S_O for output voltages and six sectors Si for source current. The resulting voltage vector creates an α_O angle while the current vector creates a β_i angle between the origin of the coordinate system.

The synthesis process of reference output voltage and source current space vectors is presented in Fig. 4, according to dependence (5).

$$\underline{u}_{OL} = \underline{u}^1_{OL} + \underline{u}^2_{OL} = d_I \underline{u}_I + d_{II} \underline{u}_{II} + d_{III} \underline{u}_{III} + d_{IV} \underline{u}_{IV} \tag{5}$$

where: $d_k = t_k/T_{Seq}$, $k = I, II, III, IV$, and $d_0 = 1 - d_I - d_{II} - \mathrm{d_{III}} - \mathrm{d_{IV}}$. Relative switching times in each sequence period T_{Seq} are defined by the following equations [9]:

$$d_I = (-1)^{S_O+S_i+1} \frac{2q\cos\left(\tilde{\alpha}_O - \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\sqrt{3}\cos\varphi_i} \tag{6}$$

$$d_{II} = (-1)^{S_O+S_i} \frac{2q\cos\left(\tilde{\alpha}_O - \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\sqrt{3}\cos\varphi_i} \tag{7}$$

$$d_{III} = (-1)^{S_O+S_i} \frac{2q\cos\left(\tilde{\alpha}_O + \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\sqrt{3}\cos\varphi_i} \tag{8}$$

$$d_{IV} = (-1)^{S_O+S_i+1} \frac{2q\cos\left(\tilde{\alpha}_O + \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\sqrt{3}\cos\varphi_i} \tag{9}$$

For equations (6)–(9), new angles $\tilde{\alpha}_O$ and $\tilde{\beta}_i$ are defined relative to the bisecting line of the current sector of the complex coordinate system and reference voltage and current space vectors, respectively. The new angles are limited to the following values: $\frac{\pi}{6} < \tilde{\alpha}_O < \frac{\pi}{6}$; $\frac{\pi}{6} < \tilde{\beta}_i < \frac{\pi}{6}$. The switching pattern of the switches depends on the position of the reference vectors in a sector S_O and S_i and is summarized in Table 2 [9].

TABLE 2. Selection of the switching configurations for each combination of output voltage S_O and input current S_i sectors.

Index of d_k	Sector of the output voltage vector S_o	Sector of the input current vector S_i					
		1	2	3	4	5	6
I	1	+9	-8	+7	-9	+8	-7
II		-7	+9	-8	+7	-9	+8
III		-3	+2	-1	+3	-2	+1
IV		+1	-3	+2	-1	+3	-2
I	2	-6	+5	-4	+6	-5	+4
II		+4	-6	+5	-4	+6	-5
III		+9	-8	+7	-9	+8	-7
IV		-7	+9	-8	+7	-9	+8
I	3	+3	-2	+1	-3	+2	-1
II		-1	+3	-2	+1	-3	+2
III		-6	+5	-4	+6	-5	+4
IV		+4	-6	+5	-4	+6	-5
I	4	-9	+8	-7	+9	-8	+7
II		+7	-9	+8	-7	+9	-8
III		+3	-2	+1	-3	+2	-1
IV		-1	+3	-2	+1	-3	+2
I	5	+6	-5	+4	-6	+5	-4
II		-4	+6	-5	+4	-6	+5
III		-9	+8	-7	+9	-8	+7
IV		+7	-9	+8	-7	+9	-8
I	6	-3	+2	-1	+3	-2	+1
II		+1	-3	+2	-1	+3	-2
III		+6	-5	+4	-6	+5	-4
IV		-4	+6	-5	+4	-6	+5

The distribution of zero vectors in a given switching sequence is determined by the d_{IV} sign, which is summarized in Table 3. The advantage of SVM is the ability to control the input displacement angle φ_i between the source current and the voltage as is shown in Fig. 4b. The disadvantage of the matrix converter is voltage gain, the maximum value of which is equal $q_{max} = \frac{\sqrt{3}}{2} \approx 0.866$.

TABLE 3. Selection of the "zero" switching configurations for each combination of output voltage S_O and input current S_I sectors and sign of d_{IV}.

	Sector of the output voltage vector S_o					
Sector of the input current vector S_i	$S_o = 1 \vee 4$		$S_o = 2 \vee 5$		$S_o = 3 \vee 6$	
$S_i = 1 \vee 4$	0_B	0_A	0_B	0_A	0_B	0_A
$S_i = 2 \vee 5$	0_A	0_C	0_A	0_C	0_A	0_C
$S_i = 3 \vee 6$	0_C	0_B	0_C	0_B	0_C	0_B
	$d_{IV}>0$	$d_{IV}<0$	$d_{IV}>0$	$d_{IV}<0$	$d_{IV}>0$	$d_{IV}<0$

Safe commutation of individual transistors of the matrix converter is connected with the fulfillment of the following rules: i) should not cause a short circuit between the two input phases, because the consequent high circulating current might destroy the switches, ii) should not cause an interruption of the output current, because the consequent overvoltage might likely destroy the switches [9]. In order to meet these rules, several commutation strategies have been proposed [51], [52]. The most common solution is the four-step commutation strategy. In this method the direction of current flow through the commutation cells can be controlled (Fig. 5, top), and the current direction is used to determine which device in the active switch cell is not current conducting. The commutation process as a timing diagram and a state diagram is shown at the bottom of Fig. 5. An identical switching sequence occurs between any two bidirectional switches in each output phase of the MC.

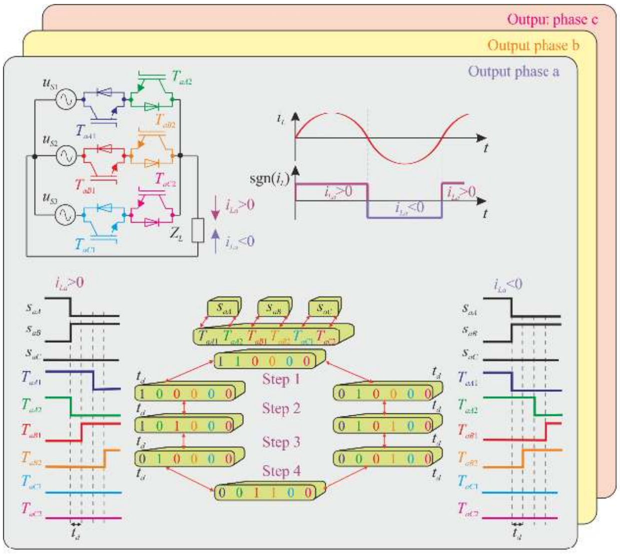

FIGURE 5. Four step MC commutation of bi-directional switches: general commutation circuit and definition of the current direction (top), diagram of four-step commutation process based on current direction for two bi-directional switches (bottom).

Commonly in MC control systems, the commutation process is carried out with FPGAs due to the limited computing power of DSPs, while the modulation and control part of the control algorithm is implemented on DSP devices [12]. Digital controllers for power converters are being more and more implemented in FPGAs due to the increasing complexity of modulation and commutation algorithms, higher switching frequencies, and concurrence requirements. In this paper, due to the high switching frequency of PWM signals, both the modulation and commutation processes will be implemented in the FPGA device.

III. PETRI NETS

Let us introduce the main definitions and notations necessary to explain the proposed idea. The presented definitions correspond to the notations from [18], [28], [41], [43], and [53]–[57].

Definition 1: A Petri net PN is a 4-tupe: $N = (P, T, F, M_0)$ where P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs, $M_0 : P \to \mathbb{N}$ is an initial marking. Sets of input and output places of a transition are defined as: $t = \{p \in P : (p, t) \in F, t = \{p \in P : (t, p) \in F$, while sets of input and output transitions of a place are denoted as: $p = \{t \in T : (t, p) \in F, p = \{t \in T : (t, p) \in F$.

Definition 2: A state of a Petri net $N = (P, T, F, M_0)$ is called a *marking*. A marking is a vector P of nonnegative integer numbers assigning to every place $p \in P$ number of tokens. $M(p)$ corresponds to the number of tokens in place p. A marking can be changed by the firing of a transitions. A transition can fire if every one of its input places is marked. A place is marked if $M(p) > 0$. When a transition is fired tokens move from every one of its input places to every one of its output places. A marking M_n is said to be reachable from marking M_0 if there exists a flow of firings that transforms M_0 to M_n.

Definition 3: A Petri net $N = (P, T, F, M_0)$ is said to be *safe*, if the number of tokens in each place $p \in P$ does not exceed 1 for any reachable marking M_n that is $p \in P : M(p) \leq 1$.

Definition 4: A Petri net $N = (P, T, F, M_0)$ is *live*, if for any reachable marking M_n and any transition $t \in T$ there is marking M_m reachable from M_n, such that t can fire in M_m. A safe and live Petri net is a well-formed net.

Definition 5: A marked graph net (MG-net) is a Petri net $PN = (P, T, F, M_0)$ for which every place p has exactly one input transition and exactly one output transition i.e. $\forall p \in P$: $| \bullet | = |p| = 1$.

Definition 6: A state machine net (SM-net) is a Petri net $PN = (P, T, F, M_0)$ for which every transition t has exactly one input place and exactly one output place i.e. $\forall t \in T$: $| \bullet | = |t| = 1$, and there is exactly one token at the initial marking.

Definition 7: A state machine component (SMC) of a Petri net $PN = (P, T, F, M_0)$ is a Petri net $S = (P^{'}, T^{'}, F^{'}, M_0^{'})$ such that S is an SM-net.

Definition 8: A Petri net $PN = (P, T, F, M_0)$ is *SM-coverable*, if for each place $p \in P$ there is an SMC $S = (P^{'}, T^{'}, F^{'}, M_0^{'})$ of PN such that $p \in P'$.

Definition 9: An incidence matrix of a Petri net $PN = (P, T, F, M_0)$ with $n = |P|$ places and $m = |T|$ transitions is an $A_{m \times n = [a_{ij}]}$ matrix (where m refers to rows, and n refers to columns) of integers, given by:

$$a_{ij} = \begin{cases} -1, & \text{if } p_j \in \bullet_i \text{ and } p_j \notin t_i \bullet \\ 1, & \text{if } p_j \notin \bullet_i \text{ and } p_j \in t_i \bullet \\ 0, & \text{otherwise} \end{cases}$$

Definition 10: A place invariant (p-invariant) of a Petri net $PN = (P, T, F, M_0)$ is a vector $\vec{y}$ of nonnegative integers that solves the equation $\vec{y} \cdot A^T = 0$, where $y \neq 0$ and A^T is a transposed incidence matrix of the net.

IV. THE IDEA OF THE PROPOSED TECHNIQUE

The idea of the proposed method is based on the specification of the cyber-physical system in the form of a live and safe Petri net (Fig. 6). There are 19 places and 11 transitions in the net. It is assumed that tasks executed by the hardware system are associated to the places of the net, as shown in Tab. 3.

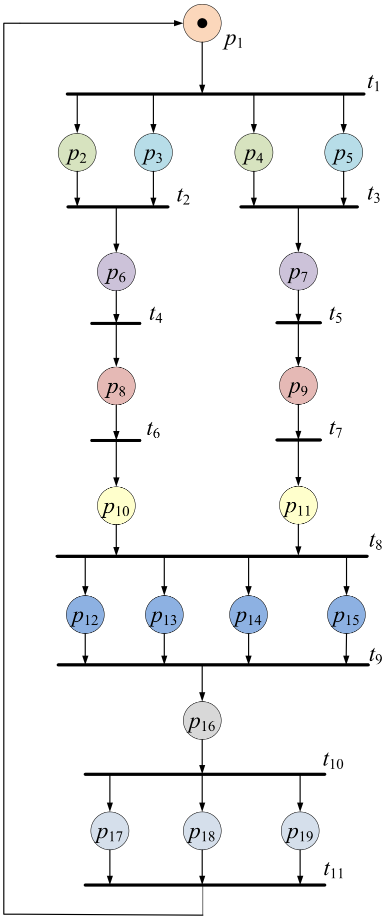

FIGURE 6. Petri net-based specification of the proposed hardware system.

There are 19 main stages (tasks) in the proposed hardware method. They are directly represented by the particular places of the net. Note that most of tasks are executed concurrently, especially computations regarding values for load voltages and source currents (places $p_2, \ldots, p_5$). Furthermore, all the duty cycles are calculated in a simultaneous manner ($p_{12}, \ldots, p_{15}$), as well as commutation between switches (p_{17}, p_{18}, p_{19}).

TABLE 4. Tasks executed by the particular places of the presented Petri net.

Place	Task executed by the system
p_1	Setting of the instantaneous values of the load voltages and source currents.
p_2	Computation of the real part (*Re*) of the space vector for voltages ($\underline{u}_{OL}$).
p_3	Computation of the imaginary part (*Im*) of the space vector for currents ($\underline{u}_{OL}$).
p_4	Computation of the real part (*Re*) of the space vector for currents ($\underline{i}_S$).
p_5	Computation of the imaginary part (*Im*) of the space vector for currents ($\underline{i}_S$).
p_6	Computation of the angle α_O between the real and imaginary parts for voltages.
p_7	Computation of the angle β_i between the real and imaginary parts for currents.
p_8	Computation of the S_O for the voltages space vector.
p_9	Computation of the sector S_i for the currents space vector.
p_{10}	Normalization of the angle α_O.
p_{11}	Normalization of the angle β_i.
p_{12}	Computation of the value of the modulation duty cycle d_I.
p_{13}	Computation of the value of the modulation duty cycle d_{II}.
p_{14}	Computation of the value of the modulation duty cycle d_{III}.
p_{15}	Computation of the value of the modulation duty cycle d_{IV}.
p_{16}	Computation of the PWM signals (switches).
p_{17}	Computation of the commutation signals for *line a*.
p_{18}	Computation of the commutation signals for *line b*.
p_{19}	Computation of the commutation signals for *line c*.

Let us now describe each of the above tasks in more detail. Initially, the instantaneous values of the load voltages u_{ab}, u_{bc}, u_{ca} and source currents i_A, i_B, i_C are set based on the input values. This task is associated with the place p1 in the presented Petri net. Then, the real and imaginary parts for the space vectors $\underline{u}_{OL}$ and $\underline{i}_S$ are simultaneously computed. In particular, the following tasks are executed [9]:

1. Computation of the real part (*Re*) of the space vector for $\underline{u}_{OL}$ (place p_2):

$$Re(\underline{u}_{OL}) = \frac{1}{3}(2u_{ab} - u_{bc} - u_{ca}) \tag{10}$$

2. Computation of the imaginary part (*Im*) of the space vector for $\underline{u}_{OL}$ (place p_3):

$$Im(\underline{u}_{OL}) = \frac{\sqrt{3}}{3}(u_{bc} - u_{ca}) \tag{11}$$

3. Computation of the real part (*Re*) of the space vector for $\underline{i}_S$ (place p_4):

$$Re(\underline{i}_S) = \frac{1}{3}(2i_A - i_B - i_C) \tag{12}$$

4. Computation of the imaginary part (*Im*) of the space vector for $\underline{i}_S$ (place p_5):

$$Im(\underline{i}_S) = \frac{\sqrt{3}}{3}(i_B - i_C) \tag{13}$$

Once the real and imaginary parts of the space vectors are obtained, the calculations regarding angles and sectors are performed. Note that such operations are executed simultaneously for voltages and currents. Indeed, places p_6, p_8, and p_{10} refer to the computation angle $\tilde{\alpha}_O$ for $\underline{u}_{OL}$ (angle α_O computation, the sector S_O computation and angle α_O scaling, respectively), while angle $\tilde{\beta}_i$ for $\underline{i}_S$ (angle β computation, the sector S_i computation and angle $\tilde{\beta}_i$ scaling) is computed by tasks associated to places p_7, p_9, and p_{11}. In particular, computation of the angle α_O between the real and imaginary parts of the space vector for $\underline{u}_{OL}$ is executed according to the equation (place p_6):

$$\alpha_O = atg\frac{Im(\underline{u}_{OL})}{Re(\underline{u}_{OL})} \tag{14}$$

Further computation of the sector S_O for the voltage space vector is based on the angle α_O (place p_8). Finally, normalization (scaling) of the angle α_O from the initial range $-\pi \leq \alpha_O \leq \pi$ to the range $-\frac{\pi}{6} \leq \tilde{\alpha}_O \leq \frac{\pi}{6}$ is executed depending on the sector S_O [9] (place p_{10}).

Computation of the angle β_i between the real and imaginary parts of the space vector for $\underline{i}_S$ is performed in the same manner (place p_7):

$$\beta_i = atg\frac{Im(\underline{i}_S)}{Re(\underline{i}_S)} \tag{15}$$

Similarly, computation of the sector S_i for the voltage space vector is calculated based on the angle β_i (place p_9). Finally, angle β_i is normalized from the initial range $-\pi \leq \beta_i \leq \pi$ to the range $-\frac{\pi}{6} \leq \tilde{\beta}_i \leq \frac{\pi}{6}$ (place p_{11}).

The modulation duty cycles are computed within places $p_{12}, \ldots, p_{15}$. Those values are calculated simultaneously, according to equations (6)–(9). Indeed, the value of the modulation duty cycle d_I is obtained by the execution of place p_{12}, the value of cycle d_{II} is computed by the operations performed within place p_{12}, and so on. Based on those values, the switches (internal PWM signals $S_{aA}, \ldots, S_{cC}$) are generated. This task is executed within place p_{16}.

Finally, the output commutation signals (T_{aA1}, T_{aA2}, T_{aB1}, T_{aB2}, T_{aC1}, T_{aC2}, T_{bA1}, T_{bA2}, T_{bB1}, T_{bB2}, T_{bC1}, T_{bC2}, T_{cA1}, T_{cA2}, T_{cB1}, T_{cB2}, T_{cC1}, T_{aC2}) are computed. They are evaluated concurrently by places p_{17}, p_{18}, p_{19}. Each place represents a task referred to at a particular line. Therefore, place p_{17} represents operations related to the line a, place p_{18} is related to the line b, while place p_{19} refers to the line c. Note that thanks to the applied Petri net-based specification, the tasks for particular lines are executed concurrently.

V. CASE-STUDY EXAMPLE (HARDWARE IMPLEMENTATION OF THE PROPOSED METHOD)

Let us now present the hardware implementation of the Petri-net specification of the system shown in the previous section. Fig. 7 shows the block-diagram of the system. It is assumed that each place of the Petri-net refers to the hardware module of the system. For example, place p_2 corresponds to the component that is responsible for computation of the real part of the space vector for voltages, etc. All but two of the modules have been prepared (written) in a pure Verilog hardware language, the exceptions being two sub-components that are dedicated to the trigonometric operations, which we shall discuss later.

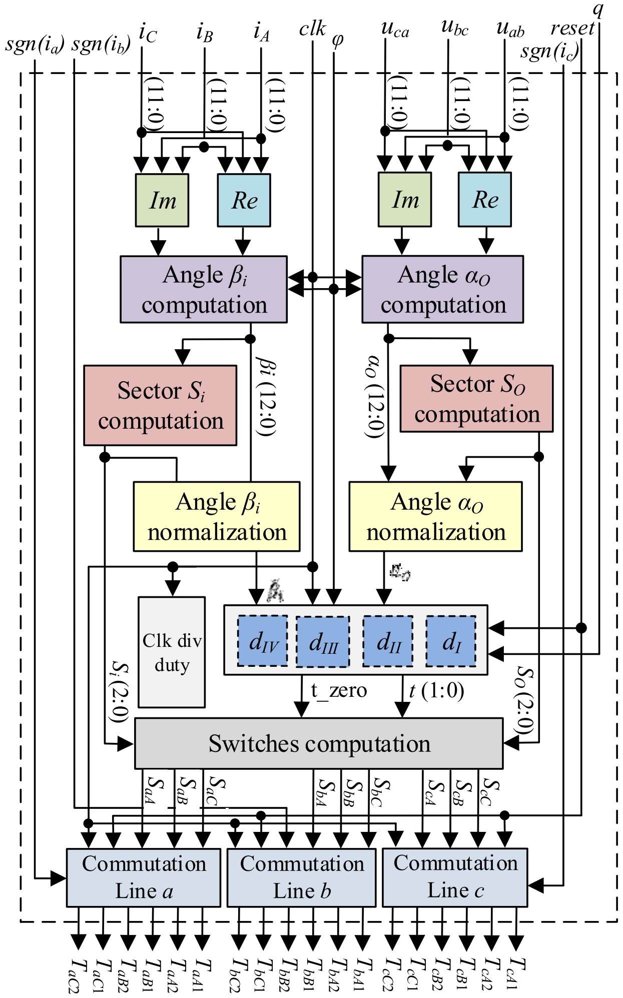

FIGURE 7. Block diagram of the hardware system designed according to the presented Petri net.

All the modules of the system are connected via internal buses. Most of components are synchronous, thus an external clock signal *clk* is delivered (see Fig. 7). Furthermore, seven blocks of the system are zeroed by the reset signal (four modules that compute duty cycles $d_I, \ldots, d_{IV}$, and three commutation blocks). Let us briefly describe all the components of the presented hardware system.

A. COMPUTATION OF THE REAL AND IMAGINARY PARTS OF THE SPACE VECTOR FOR VOLTAGES AND CURRENTS (FOUR MODULES)

There are four blocks that are responsible for calculation of the real or imaginary parts of the space vectors. All those tasks are executed concurrently. The upper two modules (denoted as *Re* and *Im* on the block diagram) refer to places p_2 and p_3 of the Petri net and they are applied in order to compute the proper values of the space vector for voltages. Similarly, two bottom modules *Re* and *Im* are designed to obtain the real and imaginary parts of the space vectors for currents (places p_4 and p_5 of the net). All those components are prototyped as combinational circuits, thus no clock nor reset signal are delivered. The internal computation of the trigonometric function (arctangent) is performed with the use of a COordinate Rotation DIgital Computer (CORDIC) technique. In particular, the dedicated Intellectual Property (IP-core) from Xilinx is applied [58].

B. ANGLE COMPUTATION OF THE SPACE VECTORS (TWO MODULES)

This is a sequential module that computes the proper angle between the real and the imaginary parts of the space vectors. There are two instances of this block. The first one refers to place p_6 of the presented Petri net. This module calculates angle α_O of the load voltages space vector, based on the values generated by the blocks *Re* and *Im* from the previous step. Similarly, the second component computes angle β_i of the source currents space vector.

```
module AngleComputation(angle,ready,X1_raw,X2_raw,X3_raw,clk);
    input [11:0] X1_raw,X2_raw,X3_raw;
    wire [12:0] X1,X2,X3;

    //conversion from RAW Bin to 1QN:
    Convert_Bin_1QN convert1 (X1,X1_raw);
    Convert_Bin_1QN convert2 (X2,X2_raw);
    Convert_Bin_1QN convert3 (X3,X3_raw);
    output ready;
    input clk;
    output [12:0] angle;

    //computation of Re and Im:
    wire [12:0] Re,Im;
    Calculate_Re calcRe (Re,X1,X2,X3);
    Calculate_Im calcIm (Im,X2,X3);

    //computation of angle:
    Atan2 Atan2_sim(angle,ready,Im,Re,clk,1'b1);
endmodule
```

Listing 1. Verilog source code of block Angle computation.

Listing I shows the sample Verilog code of the block Angle computation. The same module is used in order to compute α_O and β_i, however different input values are delivered for particular instances. The main part of the module is for calculation of the real and imaginary parts of the space vector (both components are instantiated within the code). Furthermore, additional conversions between natural binary code and the 1QN system (used by Xilinx CORDIC) is performed. Note that those operations are executed by sub-components prepared in a pure Verilog language.

C. SECTOR COMPUTATION OF THE SPACE VECTORS (TWO MODULES)

There are two instances of this component. The first one (denoted as Sector S_O computation in Fig. 7) computes sector S_O of the space vector for load voltages. This instance corresponds to place p_8 of the Petri net. The result is obtained on the value of the angle α_O. Furthermore, the second instance (*SectorS$_i$computation*, place p_9 of the net) computes sector S_i for source currents, based on the angle β_i.

D. ANGLE NORMALIZATION (TWO MODULES)

Two modules are designed in order to normalize the angles to the range $\left\langle -\frac{\pi}{6}, \frac{\pi}{6} \right\rangle$. The first one, Angle α_O normalization, refers to the place p_{10} of the net and it scales the angle for voltages. The second component, Angle β_i normalization corresponds to place p_{11} and it adjusts the angle for currents. Both modules are prepared as a combinational logic. The resulting (normalized) angle is computed according to the angle α_O (for load voltages) or β_i (for source currents), depending on the current sector of the space vectors (S_O or S_i, respectively).

E. DUTY COMPUTATION AND ORGANIZATION (CONSISTS OF FOUR MODULES)

There are four modules that are in response of the computation of the modulation duty cycles of switches. Those components refer to places $p_{12}, \ldots, p_{15}$ of the Petri net from Fig. 6. Particular values (d_I, d_{II}, d_{III}, or d_{IV}) are computed directly according to Equations (6)–(9). All four components are shown on the diagram (Fig. 7). Since there are several input signals to those modules, they are grouped into the top-module, called Duty computation and organization order, to increase the visibility of the schema. Furthermore, each module is synchronous, and it applies clock signal. The external frequency divisor (denoted as *clk div duty*) is applied in order to adjust the adequate sequence period. Therefore, the switching time can be set depending on the user needs.

Listing II shows the sample Verilog code of the top-module Duty computation and organization. Particular values of duty cycles (denoted as *d1*,..., *d4*) are computed with the use of trigonometric function (namely, *cosine* function is calculated). To obtain the results, the CORDIC module is applied once more. Additional sub-components (Multiply_Three_1QN) are used in order to perform basic mathematical operations (they are written in a pure Verilog code).

F. SWITCH COMPUTATION (ONE MODULE)

This module corresponds to the place p_{16} sets the adequate switch configuration by generation of the nine PWM signals ($S_{aA}, \ldots, S_{cC}$). Those values are computed according to Table 2, depending on the modulation duty cycles and positions of voltage and current vectors in sectors S_O and S_i. The configuration can be set in "active" or "zero" mode (cf. Section II). The proper selection is executed by a single conditional assignment. Listing III shows the sample Verilog codes for those operations. The "zero" configuration of switches (upper line of the listing) is implemented as a combinational logic, and as a shift operation combined with modulo operations between the voltage and current sectors. The output values are selected between the "active" and

```
module Duty_Computation_Organization(d1,d2,d3,d4,alfa,beta,clk);
    output [12:0] d1,d2,d3,d4;
    input [12:0] alfa,beta;
    input clk;
    wire signed [12:0] alfa_minus_Pi3,alfa_plus_Pi3,
                       beta_minus_Pi3,beta_plus_Pi3;
    parameter signed Pi_3=13'b0010000110000;
    parameter const_2_sqrt_3_q_cos_fi=13'b0001111111111;    //fi=0
    assign alfa_minus_Pi3=alfa-Pi_3;
    assign alfa_plus_Pi3=alfa+Pi_3;
    assign beta_minus_Pi3=beta-Pi_3;
    assign beta_plus_Pi3=beta+Pi_3;
    wire [12:0] cos1,cos2,cos3,cos4,sin1,sin2,sin3,sin4;
    wire ready1,ready2,ready3,ready4;

    SinCos SinCos1(sin1,cos1,ready1,alfa_minus_Pi3,clk,1'b1);
    SinCos SinCos2(sin2,cos2,ready2,alfa_plus_Pi3,clk,1'b1);
    SinCos SinCos3(sin3,cos3,ready3,beta_minus_Pi3,clk,1'b1);
    SinCos SinCos4(sin4,cos4,ready4,beta_plus_Pi3,clk,1'b1);

    Multiply_Three_1QN m1 (d1,const_2_sqrt_3_q_cos_fi,cos1,cos3);
    Multiply_Three_1QN m2 (d2,const_2_sqrt_3_q_cos_fi,cos1,cos4);
    Multiply_Three_1QN m3 (d3,const_2_sqrt_3_q_cos_fi,cos2,cos3);
    Multiply_Three_1QN m4 (d4,const_2_sqrt_3_q_cos_fi,cos2,cos4);
endmodule
```

Listing 2. Verilog source code of block Duty computation and organization.

```
assign vectorZero=9'b100100100>>(((4-sectorI%3)+sectorU[0])%3);
assign {SaA,SaB,SaC,SbA,SbB,SbC,ScA,ScB,ScC}=(d==3'b100)?vectorZero:switches;
```

Listing 3. Computation of the "zero" configuration and selection between both configurations.

"zero" configurations (bottom line of the listing), depending on the relative switching times. This operation is also performed by a combinational logic (simple multiplexer).

G. FOUR-STEP COMMUTATION (THREE MODULES)

Finally, the output signals of the system are produced by three components that perform four-step commutation. Each module is dedicated for each output line (*a, b, c*) and they refer to the places p_{17}, p_{18}, and p_{19} of the Petri net. The four-step commutation strategy is implemented as a finite-state machine. The switching sequence is executed in four main steps. It is possible to parameterize the commutation time according to the user needs. Moreover, the applied commutation technique assures proper switching in the case of high frequency of input signals. In particular, the operation is immediately terminated if changes to the input signals are detected.

VI. ANALYTICAL, NUMERICAL AND HARDWARE VERIFICATION OF THE SYSTEM

The proposed method has been verified analytically, numerically and experimentally. Initially, the specification of the system was analyzed and verified. Next, the numerical validation of the proposed method was performed. Finally, the system was implemented in an FPGA device. Let us discuss the above procedures in more detail with a brief summarization regarding limitations and scope of the presented technique.

A. VERIFICATION AND ANALYSIS OF THE SPECIFICATION OF THE PROPOSED SYSTEM

The Petri-net based specification of the proposed system has been verified and analyzed with the application of the set of dedicated tools that are available on-line at: *http://gres.uninova.pt* (the IOPT-Tools developed at Universidade Nova de Lisboa [39], [44]) and *http://www.hippo.iee.uz.zgora.pl* (*Hippo* system developed at University of Zielona Góra). In particular, the main properties of the net have been checked such as: liveness, safeness, number of states (markings), classification, number of the minimal SMCs that cover the net, etc. The complete results of the analysis are shown in Table 5. The net has been added to the Hippo system and it is available on-line at: *http://www.hippo.iie.uz.zgora.pl/index.php?id=petri_net&nr=482*

TABLE 5. Results of analysis of the Petri net that specifies the proposed hardware system (from Fig. 3).

Property	Result
Number of places	19
Number of transitions	11
Classification	Marked Graph (MG-net)
Liveness	TRUE
Safeness	TRUE
Number of states (reachable markings)	20
Number of all place invariants in the net	48
Number of all SMCs in the net	48
SM-coverability of the net	TRUE
Number of minimal SMCs that cover the net	4

From the above results, it can be noticed that the Petri net is live and safe. Those two properties are crucial regarding further design of the hardware system. Loosely speaking, the prototyped Petri net is free of any deadlocks and unreachable states (redundancy). It means that from the formal point of view the system is properly specified.

Furthermore, the detailed analysis indicates that the net is classified as a Marked Graph. Such information can be useful in case of further optimizations and additional analyses, since this particular class of Petri nets has unique properties (please refer to [41], [54]–[57], and [61]–[70] for more details). Moreover, there are 20 reachable markings (states) in the net, while the total number of place invariants in the net is equal to 48. All of those invariants correspond to the state machine components of the Petri net, thus there are 48 SMCs in the specified system. Finally, the net is SM-coverable. It means that there is a possibility of further decomposition (splitting) of the system into sequential components. This operation is especially useful in the case of application of sequential devices (such as programmable logic controllers, PLCs) or dynamic partial reconfiguration of the system implemented in FPGA (please refer to [18] for more details).

To summarize the above discussion, it should be pointed out that analysis of the Petri net shows that the net has been properly designed. It is live and safe. Furthermore, it is possible to apply further optimizations of the net (such as decomposition into SM-components) in order to split the system into sequential components, or to apply the idea of dynamic partial reconfiguration [18], [41].

B. NUMERICAL VERIFICATION OF THE PROPOSED HARDWARE SYSTEM

The correct operation of the control systems of complex power converters before the final verification in the experimental prototype should be checked by numerical experiment. Therefore, the numerical validation of the algorithm was carried out using the Matlab Simulink program. A testing procedure is presented in Fig. 8. The control signals generated in the Active-HDL simulator (Aldec Inc.) were stored in a text file as a sequence of samples with a sampling frequency of 10 MHz. Then the saved results were loaded into the Matlab Simulink program. The numerical verification of the correctness of the algorithm is shown in Fig. 9, where the time waveforms of source and load currents and MC output line-to-line voltage u_{ab} are depicted. The correctness of the implemented SVM algorithm and four-stage commutation has been tested. Because these algorithms are a constant element for various MC applications, their correct verification for a typical load gives information that possible errors in operation will be associated with incorrect implementation of the main program control in a given application.

Active-HDL™
TXT
MATLAB
&SIMULINK

FIGURE 8. The procedure of numerical verification of applied algorithms.

C. HARDWARE VERIFICATION OF THE SYSTEM WITH FPGA DEVICE

Finally, the proposed hardware system has been verified with the application of the real FPGA. In particular, the XC7A100T device from Xilinx has been used (Artix-7 family, *Nexys* 4 *DDR* board). To perform such a task, the following values were set:

- the input clock signal (*clk*) frequency was set to 100 MHz (it applies the internal oscillator of the FPGA),
- the modulation sequence period was set to 100 kHz ($T_{Seq} = 10\mu s$),
- the commutation switch period was set to 0.2 μs.
- the value of the parameter ϕ_i was assumed as a constant ($\phi_i = 0$).

The whole system was described with the Verilog language, as mentioned in Section 5. Furthermore, it was logically synthesized and implemented in order to program the FPGA device (Vivado ver. 2017.2 was applied). The utilization of the resources was as follows:

- the number of utilized Look-Up Tables (Luts): 4830 (which is below 8% of the device),

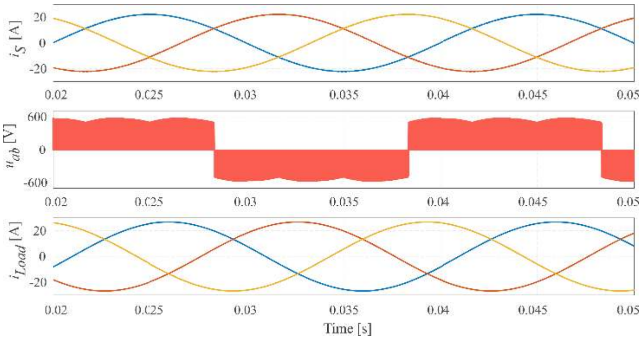

FIGURE 9. The simulation verification of SVM witch four step commutation algorithm for matrix converter.

- the number of utilized Flip-Flop registers: 4304 (below 4% of the device),
- the number of built-in digital signal processing (DSP) blocks: 16 (below 7% of the device).

The above results can be summarized thus, that the utilization of the hardware resources is relatively very low (below 8%). This means that the FPGA can be additionally used for other modules or components, such as analog-to-digital converters, etc.

In order to verify the functionality of the implemented system, the outputs of the FPGA were connected to the oscilloscope (*Tektronix MSO 2024*). Fig. 10 shows the experimental results of control signals for output phase "a". In the presented oscillogram, the control signals of the switches S_{aA}, S_{aB}, S_{aC} after SVM process, as well as the individual transistors T_{aA1}, T_{aA2}, T_{aB1}, T_{aB2}, T_{aC1}, T_{aC2}, after the commutation strategy and the signal of the direction of the output current $sign(i_a)$ are presented. As can be seen, different switching sequences occur for opposite directions of the load current flow. For the negative current direction, the S_{aB} is switch-on and the S_{aA} is switch-off. The transistor switching sequence is as follows: T_{aA1} → OFF, T_{aB2} → ON, T_{aA2} → OFF, T_{aB1} → ON. For the positive current direction, the SaC is in the switch-on position and the S_{aB}, in the switch-off position. The transistor switching sequence is as follows: T_{aB2} → OFF, T_{aC1} → ON, T_{aB1} → OFF, T_{aC2} → ON.

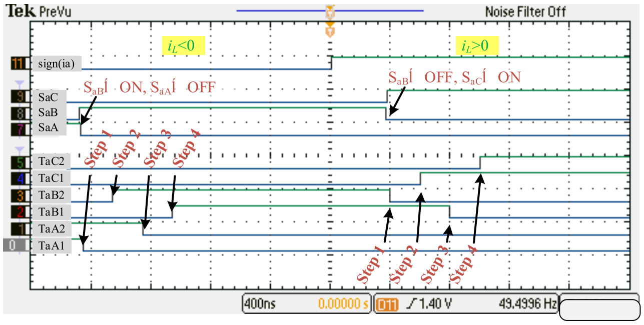

FIGURE 10. The experimental results of control signals for output phase "a".

The switching time of the next transistors for both current directions is 0.2 μs. Both presented switching sequences correspond to the general switching pattern depicted in Fig. 4. The verified program code can be further tested in the real prototype, which will be the subject of further work.

VII. CONCLUSIONS

The novel design method of cyber-physical systems aimed at controlling power converter modulation and commutation processes has been proposed in the paper. The presented technique is based on the Petri net theory, which naturally reflects the concurrency relations in the system. Furthermore, the design can be easily analyzed with the set of existing methods and tools in order to verify whether the system is specified properly. Contrary to similar approaches, the presented solution does not involve any specialized external tools, nor additional conversions. Moreover, the algorithm greatly utilizes the concurrency properties of digital devices (especially FPGA).

On the other side, the proposed solution also has several limitations. First of all, the proposed specification of the cyber-physical system relies on a Petri net theory. This can be considered a great benefit (concurrency, verification and analysis of the specification), but on the other hand such a specification requires specialized knowledge from the designer. Fortunately, there exist computer-aided tools that permit user-friendly specification of the design by a Petri net (for example in a graphical way), and additionally offer an automated (or semi-automated) prototyping process (including verification, analysis, decomposition of the system). Moreover, the proposed hardware implementation of a cyber-physical system is written in a pure Verilog-HDL. This means that any changes to the design require knowledge of this hardware language. On the other side, neither external converters nor advanced tools are required in order to specify, design and implement the cyber-physical system.

Plans for future research include enhancement of the proposed technique. It is planned to optimize the computation tasks in order to increase the switching frequency. Furthermore, other variants of cyber-physical systems specified by Petri nets are going to be designed and verified. Moreover, it is also planned to apply the decomposition methods and to check the possible effects of the dynamic partial reconfiguration of the system. Finally, other forms of analysis, verification and validation techniques (e.g., model checking or methods based on perfect graphs and c-exact hypergraphs) are going to be utilized.

REFERENCES

[1] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed. Cambridge, MA, USA: MIT Press, 2017.

[2] Z. Yu, L. Zhou, Z. Ma, and M. A. El-Meligy, "Trustworthiness modeling and analysis of cyber-physical manufacturing systems," *IEEE Access*, vol. 5, pp. 26076–26085, 2017.

[3] B. Zhang and D. Qiu, *Multi-Terminal High Voltage Converter*. Hoboken, NJ, USA: Wiley, 2018.

[4] L. F. S Alves, P. Lefranc, P.-O. Jeannin, and B. Sarrazin, "Review on SiC-MOSFET devices and associated gate drivers," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Lyon, France, Feb. 2018, pp. 824–829.

[5] A. Fekik *et al.*, "Direct power control of a three-phase PWM-rectifier based on Petri nets for the selection of switching states," in *Proc. 7th Int. Conf. Renew. Energy Res. Appl. (ICRERA)*, Paris, France, Oct. 2018, pp. 1121–1125.

[6] E. A. Jones, F. F. Wang, and D. Costinett, "Review of commercial GaN power devices and GaN-based converter design challenges," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 3, pp. 707–719, Sep. 2016.

[7] E. Ormaetxea *et al.*, "Matrix converter protection and computational capabilities based on a system on chip design with an FPGA," *IEEE Trans. Power Electron.*, vol. 26, no. 1, pp. 272–287, Jan. 2011.

[8] Y. Furukawa *et al.*, "Matrix converter with sinusoidal input-output filter and filter downsizing using SiC devices," in *Proc. IEEE Energy Convers. Congr. Expo. (ECCE)*, Milwaukee, WI, USA, Sep. 2016, pp. 1–5.

[9] P. Szcześniak, *Three-Phase AC-AC Power Converters Based on Matrix Converter Topology: Matrix-Reactance Frequency Converters Concept*. Berlin, Germany: Springer, 2013.

[10] J. Rodríguez, M. Rivera, J. W. Kolar, and P. W. Wheeler, "A review of control and modulation methods for matrix converters," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 58–70, Jan. 2012.

[11] P. Szcześniak and J. Kaniewski, "Power electronics converters without DC energy storage in the future electrical power network," *Electr. Power Syst. Res.*, vol. 129 pp. 194–207, Dec. 2015.

[12] F. Gruson, P. Le Moigne, P. Delarue, A. Videt, X. Cimetiere, and M. Arpilliere, "A simple carrier-based modulation for the SVM of the matrix converter," *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 947–956, May 2013.

[13] J. Wang, Y. Song, W. Li, J. Guo, and A. Monti, "Development of a universal platform for hardware in-the-loop testing of microgrids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2154–2165, Nov. 2014.

[14] A. S. Musleh, S. M. Muyeen, A. Al-Durra, I. Kamwa, M. A. S. Masoum, and S. Islam, "Time-delay analysis of wide-area voltage control considering smart grid contingences in a real-time environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1242–1252, Mar. 2018.

[15] T. Friedli, S. D. Round, D. Hassler, and J. W. Kolar, "Design and performance of a 200-kHz all-SiC JFET current DC-link back-to-back converter," *IEEE Trans. Ind. Appl.*, vol. 45, no. 5, pp. 1868–1878, Sep./Oct. 2009.

[16] S. Safari, A. Castellazzi, and P. Wheeler, "Experimental and analytical performance evaluation of SiC power devices in the matrix converter," *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2584–2596, May 2014.

[17] J. Hormigo and J. Villalba, "HUB floating point for improving FPGA implementations of DSP applications," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 64, no. 3, pp. 319–323, Mar. 2017.

[18] R. Wiśniewski, "Dynamic partial reconfiguration of concurrent control systems specified by Petri nets and implemented in Xilinx FPGA devices," *IEEE Access*, vol. 6, pp. 32376–32391, 2018.

[19] L. Diao, J. Tang, P. C. Loh, S. Yin, L. Wang, and Z. Liu, "An efficient DSP–FPGA-based implementation of hybrid PWM for electric rail traction induction motor control," *IEEE Trans. Power Electron.*, vol. 33, no. 4, pp. 3276–3288, Apr. 2018.

[20] M. Hamouda, H. F. Blanchette, K. Al-Haddad, and F. Fnaiech, "An efficient DSP-FPGA-based real-time implementation method of SVM algorithms for an indirect matrix converter," *IEEE Trans. Ind. Electron.*, vol. 58, no. 11, pp. 5024–5031, Nov. 2011.

[21] O. Gulbudak and E. Santi, "FPGA-based model predictive controller for direct matrix converter," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4560–4570, Jul. 2016.

[22] L. Pilato, L. Fanucci, and S. Saponara, "Real-time and high-accuracy arctangent computation using CORDIC and fast magnitude estimation," *Electronics*, vol. 6, no. 1, p. 22, 2017.

[23] R. Wiśniewski, G. Bazyło, and P. Szcześniak, "Low-cost FPGA hardware implementation of matrix converter switch control," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, to be published. doi: 10.1109/TCSII.2018.2875589.

[24] L. Gomes, A. Costa, J. P. Barros and P. Lima, "From Petri net models to VHDL implementation of digital controllers," in *Proc. 33rd Annu. Conf. IEEE Ind. Electron. Soc. (IES)*, Taipei, Taiwan, Nov. 2007, pp. 94–99. doi: 10.1109/IECON.2007.4460403.

[25] W. Liu, P. Wang, Y. Du, M. Zhou, and C. Yan, "Extended logical Petri nets-based modeling and analysis of business processes," *IEEE Access*, vol. 5, pp. 16829–16839, 2017. doi: 10.1109/ACCESS.2017.2743113.

[26] J. Ye, M. Zhou, Z. Li, and A. Al-Ahmari, ‘‘Structural decomposition and decentralized control of Petri nets,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1360–1369, Aug. 2018. doi: 10.1109/TSMC.2017.2703950.

[27] Z. Li, M. Zhou, and N. Wu, ‘‘A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems,’’ *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 173–188, Mar. 2008.

[28] M. Zhou and N. Wu, *System Modeling and Control with Resource-Oriented Petri Nets*, vol. 35. Boca Raton, FL, USA: CRC Press, 2009.

[29] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, ‘‘Online fault diagnosis of discrete event systems. A Petri net-based approach,’’ *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 31–39, Jan. 2007.

[30] N. Ran, H. Su, A. Giua and C. Seatzu, ‘‘Codiagnosability analysis of bounded Petri nets,’’ *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1192–1199, Apr. 2018. doi: 10.1109/TAC.2017.2742659.

[31] W. M. van der Aalst *et al.*, ‘‘Soundness of workflow nets: Classification, decidability, and analysis,’’ *Formal Aspects Comput.*, vol. 23, no. 3, pp. 333–363, 2011.

[32] N. Ran, S. Wang, and W. Wu, ‘‘Event feedback supervision for a class of Petri nets with unobservable transitions,’’ *IEEE Access*, vol. 6, pp. 6920–6926, 2018.

[33] I. Grobelna, ‘‘Model checking of reconfigurable FPGA modules specified by Petri nets,’’ *J. Syst. Archit.*, vol. 89, pp. 1–9, Sep. 2018.

[34] F. Yang, N. Wu, Y. Qiao, and R. Su, ‘‘Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets,’’ *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018. doi: 10.1109/JAS.2017.7510772.

[35] S. M. Vahidipour, M. Esnaashari, A. Rezvanian, and M. R. Meybodi, ‘‘GAPN-LA: A framework for solving graph problems using Petri nets and learning automata,’’ *Eng. Appl. Artif. Intell.*, vol. 77, pp. 255–267, Jan. 2019. doi: 10.1016/j.engappai.2018.10.013.

[36] G. Y. Liu, Z. W. Li, K. Barkaoui, and A. M. Al-Ahmari, ‘‘Robustness of deadlock control for a class of Petri nets with unreliable resources,’’ *Inf. Sci.*, vol. 235, pp. 259–279, Jun. 2013.

[37] M. Gan, S. Wang, Z. Ding, M. Zhou, and W. Wu, ‘‘An improved mixed-integer programming method to compute emptiable minimal siphons in S^3PR nets,’’ *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2135–2140, Nov. 2018. doi: 10.1109/TCST.2017.2754982.

[38] J. Bi, H. Yuan, and W. Tan, ‘‘Deadlock prevention for service orchestration via controlled Petri nets,’’ *J. Parallel Distrib. Comput.*, vol. 124, pp. 92–105, Feb. 2019. doi: 10.1016/j.jpdc.2018.09.010.

[39] F. Pereira, F. Moutinho, and L. Gomes, ‘‘Model-checking framework for embedded systems controllers development using IOPT Petri nets,’’ in *Proc. IEEE Int. Symp. Ind. Electron.*, Hangzhou, China, May 2012, pp. 1399–1404.

[40] R. Wiśniewski , M. Wiśniewska, and M. Jarnut, ‘‘C-exact hypergraphs in concurrency and sequentiality analyses of cyber-physical systems specified by safe Petri nets,’’ *IEEE Access*, vol. 7, pp. 13510–13522, 2019. doi: 10.1109/ACCESS.2019.2893284.

[41] R. Wiśniewski, A. Karatkevich, M. Adamski, A. Costa, and L. Gomes, ‘‘Prototyping of concurrent control systems with application of Petri nets and comparability graphs,’’ *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 2, pp. 575–586, Mar. 2018.

[42] A. G. Karatkevich and R. Wiśniewski, ‘‘A polynomial-time algorithm to obtain state machine cover of live and safe Petri nets,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[43] I. Grobelna, R. Wiśniewski, M. Grobelny, and M. Wiśniewska, ‘‘Design and verification of real-life processes with application of Petri nets,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 11, pp. 2856–2869, Nov. 2017.

[44] R. Wiśniewski, G. Bazydło, L. Gomes, and A. Costa, ‘‘Dynamic partial reconfiguration of concurrent control systems implemented in FPGA devices,’’ *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1734–1741, Aug. 2017.

[45] A. Ramirez-Trevino, I. Rivera-Rangel, and E. Lopez-Mellado, ‘‘Observability of discrete event systems modeled by interpreted Petri nets,’’ *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 557–565, Aug. 2003.

[46] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford, ‘‘Petri net modeling of cyber-physical attacks on smart grid,’’ *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 741–749, Dec. 2011.

[47] J. Luo, Q. Zhang, X. Chen, and M. Zhou, ‘‘Modeling and race detection of ladder diagrams via ordinary Petri nets,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 7, pp. 1166–1176, Jul. 2018. doi: 10.1109/TSMC.2016.2647219.

[48] J. Li, X. Yu, M. Zhou, and X. Dai, ‘‘Lean reachability tree for unbounded Petri nets,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 299–308, Feb. 2018. doi: 10.1109/TSMC.2016.2585348.

[49] D. Chattaraj, M. Sarma, and D. Samanta, ‘‘Stochastic Petri net based modeling for analyzing dependability of big data storage system,’’ in *Emerging Technologies in Data Mining and Information Security* (Advances in Intelligent Systems and Computing), vol. 813. Singapore: Springer, 2019.

[50] M. Leubner, N. Remus, J. Haase, and W. Hofmann, ‘‘Collector-emitter voltage based one-step commutation for direct three-level matrix converter,’’ in *Proc. 20th Eur. Conf. Power Electron. Appl. (EPE ECCE Europe)*, Riga, Latvia, Sep. 2018, pp. P.1–P.9.

[51] M. Leubner, N. Remus, S. Schwarz, and W. Hofmann, ‘‘Voltage based 2/3/4-step commutation for direct three-level matrix converter,’’ in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, San Antonio, TX, USA, Mar. 2018, pp. 2507–2514.

[52] K. Zhou *et al.*, ‘‘Characterization and performance evaluation of the super-junction RB-IGBT in matrix converter,’’ *IEEE Trans. Power Electron.*, vol. 33, no. 4, pp. 3289–3301, Apr. 2018.

[53] J. Martínez and M. Silva, ‘‘A simple and fast algorithm to obtain all invariants of a generalised Petri net,’’ in *Proc. Eur. Workshop Appl. Theory Petri Nets*. London, U.K.: Springer, 1982, pp. 301–310.

[54] C. Girault and R. Valk, *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*. Berlin, Germany: Springer, 2013.

[55] A. Karatkevich, *Dynamic Analysis of Petri Net-Based Discrete Systems*. Berlin, Germany: Springer, 2007, p. 356.

[56] T. Murata, ‘‘Petri nets: Properties, analysis and applications,’’ *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[57] W. Reisig, *Petri Nets: An Introduction*. Berlin, Germany: Springer, 2012, p. 4.

[58] Xilinx. *CORDIC*. Accessed: Feb. 27, 2018. [Online]. Available: https://www.xilinx.com/products/intellectual-property/cordic.html

[59] L. Gomes, F. Moutinho, and F. Pereira, ‘‘IOPT-tools—A Web based tool framework for embedded systems controller development using Petri nets,’’ in *Proc. 23rd Int. Conf. Field Program. Log. Appl.*, Porto, Portugal, Sep. 2013, p. 1.

[60] E. Best, R. Devillers, and M. Koutny, *Petri Net Algebra*. Berlin, Germany: Springer, 2013.

[61] D. A. Zaitsev, ‘‘Sleptsov nets run fast,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 5, pp. 682–693, May 2016. doi: 10.1109/TSMC.2015.2444414.

[62] Z. Jiang, Z. Li, N. Wu, and M. Zhou, ‘‘A Petri net approach to fault diagnosis and restoration for power transmission systems to avoid the output interruption of substations,’’ *IEEE Syst. J.*, vol. 12, no. 3, pp. 2566–2576, Sep. 2018. doi: 10.1109/JSYST.2017.2682185.

[63] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, ‘‘Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018. doi: 10.1109/TSMC.2016.2598303.

[64] B. Huang, Y. Pei, Y. Yang, M. Zhou, and J. Li, ‘‘Near-optimal and minimal PN supervisors of FMS with uncontrollability and unobservability,’’ in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Banff, AB, Canada, Oct. 2017, pp. 3715–3720. doi: 10.1109/SMC.2017.8123211.

[65] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou, ‘‘Analyzing E-commerce business process nets via incidence matrix and reduction,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 130–141, Jan. 2018. doi: 10.1109/TSMC.2016.2598287.

[66] Z. Ding, Y. Zhou, and M. Zhou, ‘‘Modeling self-adaptive software systems by fuzzy rules and Petri nets,’’ *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 967–984, Apr. 2018. doi: 10.1109/TFUZZ.2017.2700286.

[67] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, ‘‘Fault identification of discrete event systems modeled by Petri nets with unobservable transitions,’’ *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 2, pp. 333–345, Feb. 2019. doi: 10.1109/TSMC.2017.2762823.

[68] Y. Teng, Y. Du, L. Qi, and W. Luan, ‘‘A logic Petri net-based method for repairing process models with concurrent blocks,’’ *IEEE Access*, vol. 7, pp. 8266–8282, 2018. doi: 10.1109/ACCESS.2018.2890070.

[69] D. Fendri and M. Chaabene, ‘‘Hybrid Petri net scheduling model of household appliances for optimal renewable energy dispatching,’’ *Sustain. Cities Soc.*, vol. 45, pp. 151–158, Feb. 2019. doi: 10.1016/j.scs.2018.11.032.

[70] M. Naybour, R. Remenyte-Prescott, and M. J. Boyd, "Reliability and efficiency evaluation of a community pharmacy dispensing process using a coloured Petri-net approach," *Rel. Eng. Syst. Saf.*, vol. 182, pp. 258–268, Feb. 2019. doi: 10.1016/j.ress.2018.09.022.

REMIGIUSZ WI NIEWSKI (M'17) received the D.Sc. degree in computer science from the Silesian University of Technology, in 2018, and the Ph.D. degree in computer science from the University of Zielona Góra, in 2008, where he is currently the Head of the Division of Applied Informatics, Institute of Electrical Engineering.

From 2000 to 2003, he was with Aldec, Inc., Henderson, NV, USA, where he conducted specialized training for companies, such as Xilinx, San Jose, CA, USA, and Intel, Austin, TX, USA. He is a Co-Founder and a Coordinator of the research project Hippo. He has authored over 100 peer-reviewed research papers and books. His current research interests include the design of control systems, Petri nets, programmable devices, field-programmable gate arrays (FPGAs), partial reconfiguration of FPGAs, perfect graph and hypergraph theories, and cryptology.

Dr. WIŚNIEWSKI is a member of IES and SMC societies.

GRZEGORZ BAZYDŁO received the M.Sc. and Ph.D. degrees in computer science from the University of Zielona Góra, Poland, in 2003 and 2010, respectively. Since 2012, he has been an Assistant Professor with the Faculty of Computer Science, Electrical Engineering and Automatics, University of Zielona Góra, where he is currently with the Institute of Electrical Engineering. His current research interests include graphical methods (especially UML) in design, synthesis, and verification of embedded systems implemented in field-programmable gate array devices.

PAWEŁ SZCZE NIAK (M'12–SM'18) received the D.Sc. degree in electrical engineering from the University of Zielona Góra, in 2018, and the Ph.D. degree in electrical engineering from the University of Zielona Góra, in 2010. He is currently the Head of the Division of Power Electronics, Institute of Electrical Engineering, University of Zielona Góra. He has authored over 90 peer-reviewed research papers and one book. He has been involved in the realization of many national research projects. His current research interests mainly include power electronics systems in power networks, particularly the analysis, modeling, and properties' study of ac/ac converters without dc energy storage.

MARCIN WOJNAKOWSKI received the B.Sc. and M.Sc. degrees in computer science from the University of Zielona Góra, in 2017 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Institute of Electrical Engineering, Faculty of Computer Science, Electrical Engineering and Automatics, University of Zielona Góra.

He is a member of the research project Hippo. His research interests include Petri nets and modeling, analysis, and decomposition of concurrent systems.