International Journal of Power and Energy Systems, Vol. 43, No. 10, 2023
An Open Access Paper

RAPID PROTOTYPING OF PREDICTIVE DIRECT CURRENT CONTROL IN A LOW-COST FPGA USING HDL CODER

Deepa Sankar,* S. Lakshmi,** C.A. Babu,* and K. Mathew**

Abstract

This paper discusses a simple and efficient rapid prototyping method for the field programmable gate array (FPGA) implementation of computationally intensive finite state-predictive direct current control (FS-PDCC). The MATLAB/Simulink hardware description language (HDL) coder generates optimised Verilog code from HDL-supported Simulink blocks to fit a low-cost target-Intel Altera MAX®10 FPGA. The proposed work exhibits an efficient resource utilisation of the FPGA with an excellent cost-performance trade-off. In addition, the implemented control algorithm shows superior output quality, validated in Simulink, and verified experimentally in a two-level three-phase voltage source inverter (VSI) prototype. This approach will assist the power electronic designers to implement complex control algorithms rapidly and efficiently without manual coding, reducing the design productivity gap.

Key Words

FPGA, FS-PDCC, model predictive control (MPC), HDL coder, ADC IP core

1. Introduction

Power electronic converters offer efficient control and conversion of electric power and play a significant role in numerous power conversion applications ranging from a few watts to several megawatts. New schemes are being proposed every year to control power converters, with the end goal driven by significant aspects like energy efficiency, reliability, complexity, and cost. However, these targets greatly depend on the computational engines and associated software [1], [2].

In recent years, the application of model predictive control (MPC) for power converters has seen tremendous growth [3], [4]. MPC was developed as an effective time-domain control strategy in the process industry in the 1970s [5], [6]. The technological progress in digital control platforms with powerful processing capability instigated the research on MPCs application for power electronics [7]–[9].

Finite control set-MPC (FCS-MPC) – also known as MPC with reference tracking – utilises power converters' discrete nature and solves the optimisation problem online [10]. It integrates the control and modulation stages into a single computational step without an external modulator. However, one major hindrance to FCS-MPC's widespread application is its computational burden to solve optimisation problems online as power converter applications require rapid control responses in a few microseconds [11]. Hence, more study is needed to persuade the industrial sector to embrace such control systems [12].

The advent of powerful and low-cost microprocessors and digital signal processors (DSPs) allowed the implementation of advanced control techniques that possess concise control intervals for power electronics [2], [13], [14]. However, the computational delay in finding the optimal solution in one sampling period limits its application for complex control techniques like FCS-MPC [15].

The field programmable gate arrays (FPGAs) prove to be an effective alternative for the real-time execution of computationally complex power converter control techniques [15]. Furthermore, FPGA's inherent parallel and pipelined architecture help reduce the complex control algorithm's execution time to a few microseconds [16]. In addition, they offer high flexibility, re-programmability, and reconfigurability.

The most commonly used hardware description languages (HDLs) for electronic design automation (EDA) synthesis in FPGA are very high-speed integrated circuit HDL (VHDL), Verilog HDL, System Verilog, and System C [17]. However, the hardware design expertise required to program complex control strategies using HDL is an impeding factor for its industrial penetration as this may result in higher production costs and a longer time to market [2], [18]. Moreover, rapid progress in the design complexity compared to the productivity of the designers (design productivity gap) also poses challenges as the

* Division of Electrical Engineering, Cochin University of Science and Technology, Kochi 682022, India; e-mail: deepa.eee@adishankara.ac.in; drcababu@gmail.com
** Mar Athanasius College of Engineering, APJ Abdul Kalam Technological University, Kothamangalam 686666, India; e-mail: {laxmi.abhi, kmathewmace}@gmail.com
Corresponding author: Deepa Sankar

(DOI: 10.2316/J.2023.203-0437)

technology is experiencing a great revolution in packaging, integration level, design tools, and methodologies [19].

To address these issues, high-level synthesis tools (HLST) emerged as a means of boosting the performance of reconfigurable computing platforms such as FPGAs [20]. It allows the designer to describe the system design in high-level programming languages, such as C, C++, System C, and OpenCL [17], [21]. However, high area consumption and poor performance are significant downsides for HLST.

Rapid prototyping of the FPGA design can be performed with a Simulink-based HDL Coder from MathWorks. This tool automatically generates portable and synthesisable VHDL and Verilog codes from MATLAB™ functions, Simulink® models, and Stateflow® charts to program FPGA [22]. Thus, it enables the designers to prototype the FPGA designs faster with minimal knowledge in HDL, reducing the design time and cost, and supporting faster *time to market*.

This paper explores the design steps involved in applying the HDL Coder to implement FS-PDCC (also known as FCS-MPC) efficiently in a low-cost FPGA. Two-level three-phase voltage source inverter (Two-level VSI), a widely adopted power converter for industrial applications, is the system under consideration, and the latest and low-cost generation 10 FPGA from Intel-MAX®10-is the target device. HDL Coder offers fast implementation of the control algorithm and minimises the design productivity gap ensuring a low-cost solution with efficient resource utilisation. Section 2 reviews the FS-PDCC algorithm and modelling of Two-level VSI. In Section 3, the rapid prototyping of FS-PDCC using HDL Coder is elaborated, and the performance evaluation of the controller is performed in Section 4. Finally, Section 5 outlines the paper's results.

2. FS-PDCC: Operating Principle and System Model

FS-PDCC utilises the discrete nature of the power converters to predict the future behaviour of the controlled variables. Thus, only a few switching states are utilised to solve the optimisation online without an intermediate modulation stage. A cost function (g_c) defines the optimisation problem, which compares the predicted output with a reference value, and the control action with the minimum cost function is applied during each sampling interval [23]. In addition, the control objectives and constraints to be addressed are included in the cost function [13], [24]. The block diagram of FS-PDCC-based current control of a Two-level VSI is shown in Fig. 1.

Three critical elements in the FS-PDCC implementation process are prediction model, optimisation algorithm, and cost function. Euler approximation is utilised for non-linear model discretisation, and the converter and load models are used for future current prediction [11]. A Two-level VSI has eight possible switching vectors, of which six are active, and two are zero vectors. The switching function is given by S_X for $X = a, b, c$. The voltage vectors of the

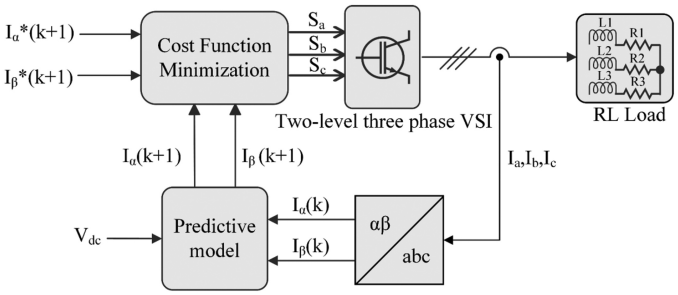


Figure 1. Block diagram FS-PDCC-based current control strategy.

three-phase VSI are provided by the expression [9]:

$$V = \frac{2}{3}\left(v_{an} + v_{bn}e^{j\left(\frac{2\pi}{3}\right)} + v_{cn}e^{j\left(\frac{4\pi}{3}\right)}\right) \quad (1)$$

v_{an}, v_{bn}, and v_{cn} represents the phase to neutral voltages of the VSI.

The circuit topology of a Two-level VSI is given in Fig. 2(a), and its switching states and voltage vectors are provided in Fig. 2(b). The following expression gives the relation between the voltage vector and the switching state:

$$V \;=\; V_{dc}S_j \quad (2)$$

where V_{dc} represents the DC input, and S_j represents the switching state for $j = 0, 1, 2 \ldots 7$.

The predictive current control is usually implemented in an $\alpha\beta$ reference frame, and the load current dynamics of the system in the $\alpha\beta$ reference frame is expressed as [23]:

$$V_{\alpha\beta} = RI_{\alpha\beta} + L\frac{dI_{a\beta}}{dt} \quad (3)$$

where $V_{\alpha\beta}$ and $I_{\alpha\beta}$ represent the voltage and current vectors of the VSI, and the discrete-time model of the load provides the future value of the load current. The load current derivative based on forward Euler approximation for a sampling period T_s is given by [23]:

$$\frac{dI_{\alpha\beta}}{dt} = \frac{I_{.\alpha\beta}(k+1) - I_{\alpha\beta}(k)}{T_s} \quad (4)$$

The load current model of the VSI is given by the expression [23]:

$$I^p_{\alpha\beta}(k+1) = I_{\alpha\beta}(k)\left(1 - \frac{RT_s}{L}\right) + \frac{T_s}{L}\left(V_{\alpha\beta}(k)\right) \quad (5)$$

where p denotes the predicted load current for all the voltage vectors generated by the VSI at an instant $(k+1)$. Another critical element in the control implementation is the cost function which evaluates the optimal switching state with minimum error applied to the inverter. The absolute value of g_c is defined by the expression [9]:

$$g_c = |I^*_\alpha(k+1) - I^p_\alpha(k+1)| + \left|I^*_\beta(k+1) - I^p_\beta(k+1)\right| \quad (6)$$

where $I^*_\alpha(k+1)$ and $I^*_\beta(k+1)$ are one step future reference currents and can be obtained by Lagrange's extrapolation. Assuming a small sampling interval, $I^*_{\alpha\beta}(k+1)$

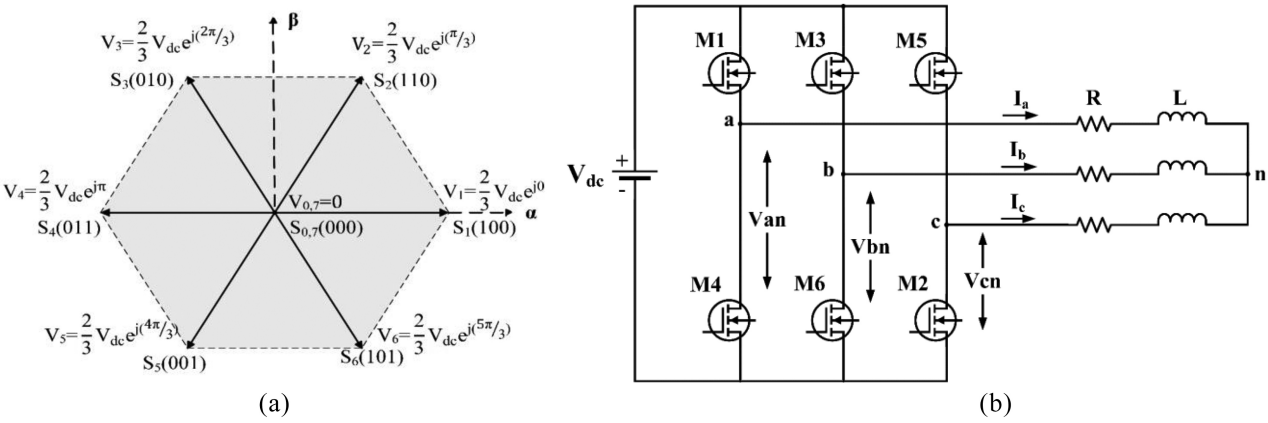


Figure 2. Two-level VSI: (a) circuit topology; (b) switching states and voltage vectors.

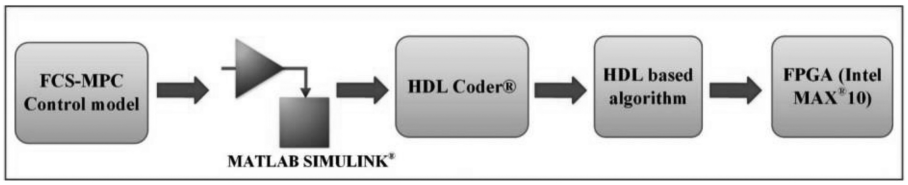


Figure 3. Workflow for FS-PDCC implementation using HDL coder.

$\approx I^*(k)I^*_{\alpha\beta}(k+1) \approx I^*(k)$. $I^p_\alpha(k+1)\, I^p_\alpha(k+1)$ and $I^p_\beta(k+1)$ are the predicted values of load currents in the $\alpha\beta$ domain. The future load current values for all the seven voltage vectors are predicted, and optimum g_c is identified using an exhaustive search algorithm (ESA). The corresponding switching sequence is employed to the inverter during T_s [25].

3. Rapid Prototyping of FS-PDCC using HDL Coder

For achieving better control performance, low-level HDL coding is preferred for implementing FS-PDCC in an FPGA. However, an additional skill set is required to design the complex control strategy in register transfer level (RTL), increasing the production cost and design time. For this reason, HDL code generation using Mathworks HDL Coder™ is adopted as an alternative that accelerates the workflow. The workflow advisor in HDL Coder automates FPGAs' programming, highlighting the critical paths and resource utilisation estimates. It also allows for code verification by providing traceability between the generated HDL code and the Simulink block [26]. However, the direct hardware implementation of the HDL code may pose some limitations in terms of resource utilisation of the target device. Figure 3 shows the workflow using HDL coder.

FS-PDCC algorithm is modelled using HDL compatible Simulink blocks. However, as the model uses floating-point data, it must be converted to fixed-point for HDL code generation as FPGA handles fixed-point data effortlessly. MathWorks has a Fixed-point Designer™ tool that provides data types and optimisation tools for implementing fixed-/floating-point algorithms. However, the result may consume more FPGA resources, and the design will not fit the target FPGA.

Resource sharing can be done effectively with a limited number of multipliers if all the operations use the same fixed-point data type. The sign and fractional length of the data type are also of much importance to avoid overflow. Pipelining the steps enables the design to meet the timing requirements without negative slack. The Simulink model of FS-PDCC and the HDL conversion using Simulink® HDL Workflow Advisor are discussed in the following section.

3.1 FS-PDCC Modelling using Simulink

The subsystem or design under test (DUT) – the predictive current control model – is modelled using HDL-supported Simulink blocks from HDL coder. The DUT will be the top-level module, and the subsystems inside the DUT will be the smaller entities. The Simulink blocks outside the DUT will be a part of the test bench. The floating-point to fixed-point conversion can be done manually using the data type conversion block provided in HDL coder. First, set the signal/parameter attributes in the block parameter menu. For example, the data type fixdt (1,16,0) is modified based on the data flow with '1' representing the signed bit, '16' the total number of bits, and '0' means the fractional length.

As discussed in Section 2, four real-time inputs to the predictive control block are the load currents I_a, I_b, I_c, and DC input voltage, V_{dc}. Moreover, three sinusoidal reference currents are also provided to the predictive control subsystem using the Sine Wave block from the DSP toolbox. Four Dual-Port RAMs are used in the model for each of the inputs. The target device MAX®10 FPGA has an inbuilt 12-bit successive approximation register (SAR)

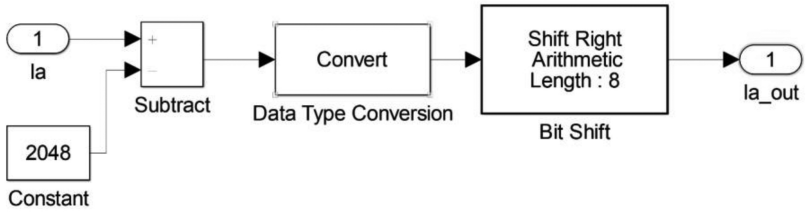


Figure 4. The data type conversion for load current I_a.

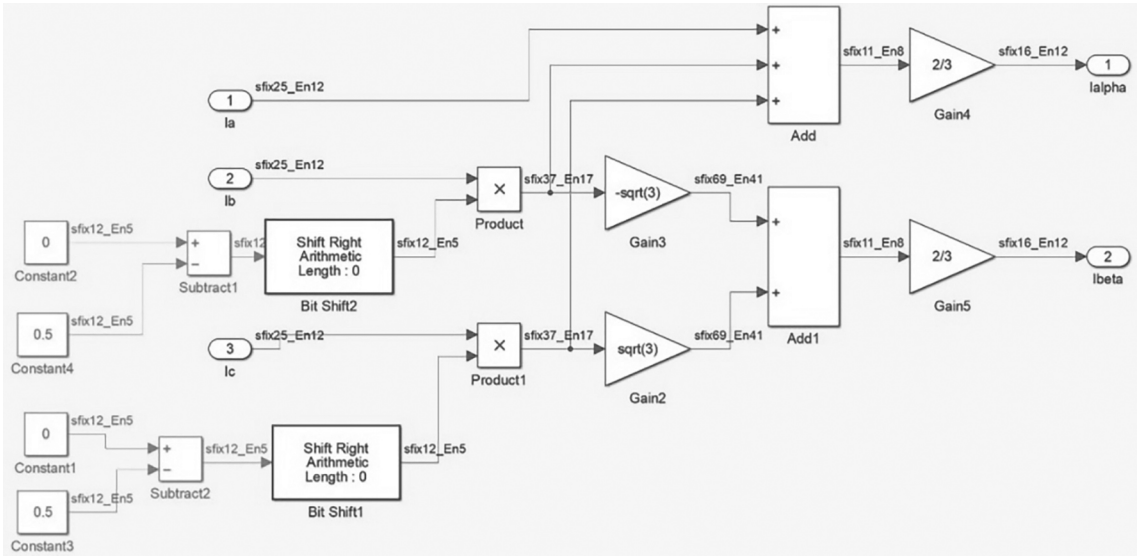


Figure 5. *abc-αβ* Reference frame transformation using Simulink blocks.

type analogue to digital converter (ADC) with a sampling rate of 1MSPS. Modular ADC Core Intel® FPGA IP using Platform Designer tool in Quartus Prime software provides the ADC solution [27].

3.1.1 Data Type Conversion at the Input

The target device possesses 18 × 18 bit multiplier blocks, and the data type for the input parameters are set to fixdt(1,18,11) in the Simulink model as ADC provides a 12-bit accuracy for the values. The load current sensor senses the AC output current and is fed to ADC. V_{in} and V_{ref} represent the analogue input and reference voltage values of ADC. Hence to obtain both positive and negative values from ADC output, the following operation must be performed for load currents I_a, I_b, and I_c represented in Fig. 4.

I_a represents 12-bit ADC output, i.e., 4096 values. Therefore, subtracting 2048 from 4096 will yield positive and negative values and is fed to the arithmetic right shift block to obtain a real analogue value. The voltage sensor senses the DC voltage and is provided to ADC, and the data type conversion is also performed for the DC input.

3.1.2 abc-αβ Reference Frame Transformation

FS-PDCC-based current control is usually implemented in an $\alpha\beta$ reference frame. Using the Clarke's transformation, balanced load current quantities are converted to orthogonal reference quantities using the expression:

$$\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \frac{2}{4}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix}$$

$$= 2/3\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad (7)$$

Figure 5 represents the implementation of the mathematical expression given by (7) in a fixed point. All the blocks are HDL compliant, and word length is shown in the signal path. The reference currents are generated using the Sine Wave block from the HDL-supported DSP system toolbox and converted to the $\alpha\beta$ reference frame. The real and imaginary parts of the complex input signal are segregated using the Complex to Real-Imag block from the HDL Coder.

3.1.3 Implementation of the Predictive Current Model

The modelling of predictive current control is performed in this stage. First, the load parameters are provided in InitFcn* window in the Model Properties. Then, the current prediction for each of the seven voltage vectors at $(k+1)$ instant is modelled using HDL-supported blocks. Finally, all the signal attributes at the input ports are set

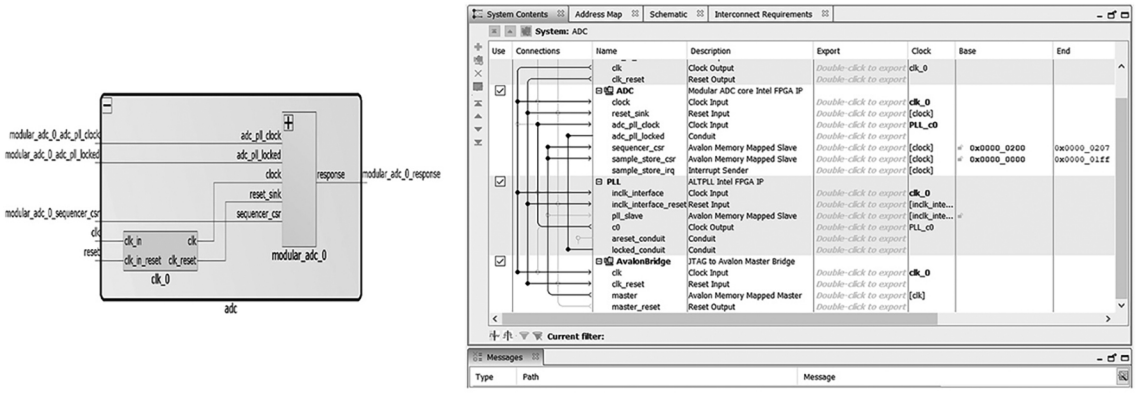


Figure 6. ADC IP core implementation: (a) interconnect in platform designer; (b) ADC schematic.

to Inherit: auto as fixed-point data types are specified in the preceding blocks.

The cost function (g_c) for each vector is evaluated as given in (6). The minimum value of the cost function is evaluated and provided to a *Minimum* HDL-supported block which gives the minimum value of the cost function along with the index. The code generation compatibility of the model is checked using HDL Code Advisor. Finally, HDL code generation, synthesis, and analysis are performed, and the Verilog code is generated. The generated code will be available in the *hdlsrc* folder. These files are added to the Quartus program files and downloaded to the MAX®10 FPGA through the USB Blaster.

The intellectual property (IP) core, for MAX®10 FPGA, can be generated with HDL Coder™ IP Core Generation Workflow from MATLAB. The target platform is selected from the HDL Workflow Advisor window, and parameters are set for IP Core generation. Alternatively, the IP Cores can be directly added to the Quartus II® environment using the Platform Designer. The proposed work generates the ADC IP core using the Platform Designer as only the Arrow DECA MAX10 evaluation kit is supported in MATLAB. The Quartus II® software consists of a Modular ADC Core Intel® FPGA IP to create, configure, and compile the ADC design. Modular ADC Core Intel® FPGA IP instantiates ADC hard IP blocks, and PLL provides a 10-MHz input clock to the ADC. Standard Sequencer with External Sample Storage configuration is utilised in the proposed system, and various parameters are set in the parameter editor.

The ADC interconnect in Platform Designer is shown in Fig. 6(a) and implemented design is shown in Fig. 6(b).

The generated IP Core is added as a submodule and instantiated in the top-level predictive module. Finally, Quartus II compiles the code and generates the netlist and timing report. A flow summary is generated, and the Programmer loads the *.sof* output file to fit the target through the USB-Blaster.

4. Performance Evaluation of the Controller

The system is first simulated in MATLAB/Simulink, and the parameters used for the simulation and experimental setup are shown in Table 1.

Table 1
System Parameters

Parameter	Value
Input DC voltage (V_{dc})	30 V
Load resistance (R)	5Ω
Load inductance (L)	19 mH
Switching frequency (f_{sw})	20 kHz
Sampling interval (T_s)	20 μs

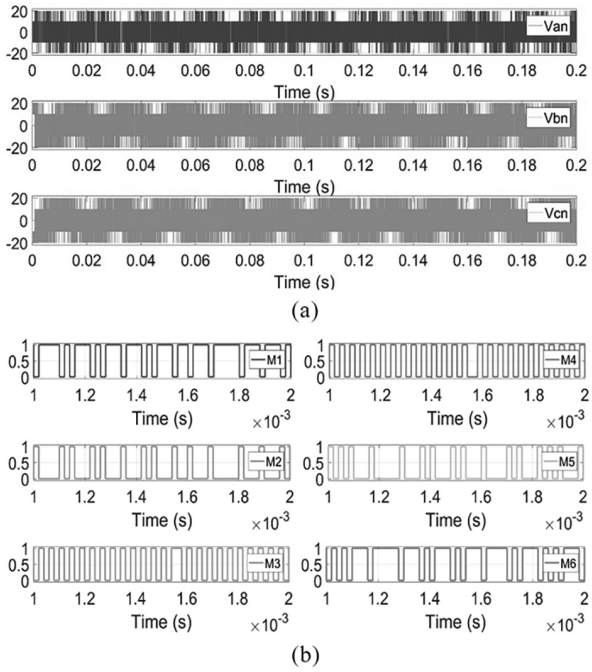


Figure 7. Simulation results of the FS-PDCC-controlled three-phase VSI: (a) switching signals to drive the MOSFETs; (b) the phase-neutral voltage of the predictive controlled VSI.

4.1 Simulation Results

Figure 7(a) shows the switching pulses generated to drive the MOSFET switches (M1–M6) in the three-phase VSI.

Table 2
FPGA Resource Utilisation for Control Implementation

Parameter	Used/Available	Percentage of Utilisation
Total logic elements	3092/8064	38%
Dedicated logic registers	1810/8542	21%
Total memory bits	169664/387072	44%
Embedded multiplier 9-bit elements	16/48	33%
PLL block	1/1	100%
ADC block	1/1	100%

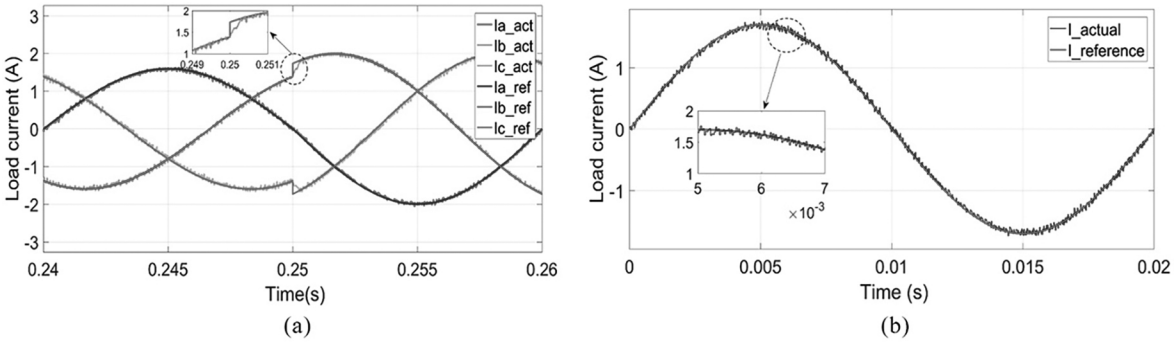


Figure 8. Load current waveform of the FS-PDCC-controlled VSI: (a) load current waveform showing I_actual tracking I_reference for a sampling time of $T_s = 20$ μs; (b) load current response for a step change at $t = 0.25$ s.

The proposed system's output phase-to-neutral voltages, v_{an}, v_{bn}, and v_{cn} are represented in Fig. 7(b). In Fig. 8(a), the reference and actual load current waveforms indicate that the actual current (I_actual) precisely tracks the reference current (I_reference) for a sampling time of 20 µs.

The system's dynamic response is analysed in Fig. 8(b). A transient case is considered: A step-change in the reference is introduced at $t = 0.25$ s, changing the reference magnitude from 1.6A to 2A during normal operating conditions. The system responds quickly within a time duration of 273 µs, demonstrating its excellent dynamic response to transient conditions. Another vital performance parameter, the frequency spectrum of the output current, is shown in Fig. 9. The spectrum is spread over a wide range, and the total harmonic distortion (THD) is 1.27% well below the IEEE standard of 5%.

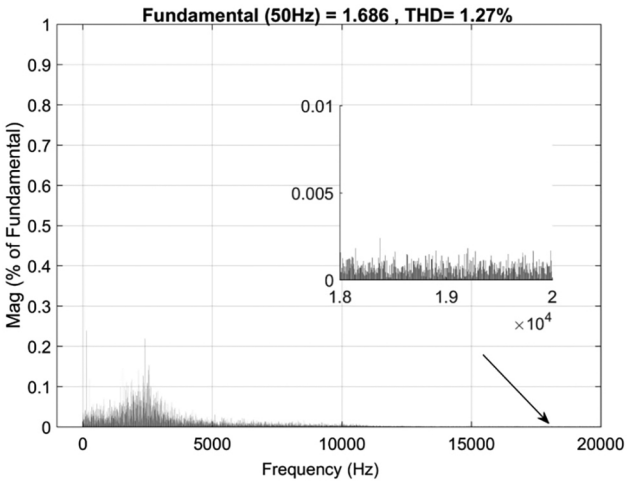


Figure 9. Load current spectrum.

5. Experimental Results

The predictive current control algorithm is implemented in a Two-level VSI shown in Fig. 1. The six MOSFET switches (IRFP150N) are driven by three Si8233BB-C-IS1 isolated high/low side gate drivers. *Allegro*^{TM} ACS722 (ACS722LLCTR-05AB-T^2) galvanically isolated current sensor IC is used for sensing the load current. The magnetic field produced by the applied current is sensed by the integrated Hall IC and a proportional voltage is fed to FPGA ADC. The DC input voltage is measured using BROADCOM® ACPL-C870 optically isolated voltage sensor with an operating range of 0–2V. The differential output from the voltage sensor is converted to a single-ended output signal and is supplied to the ADC. The VSI is provided with a star-connected resistive–inductive (RL) load. The load resistance (R) is 5Ω and the load inductance (L) is 19 mH for an output power of 50 W. The MATLAB generated control algorithm is loaded in the target device Intel MAX®10 FPGA 10M08SAE144C8G. The Quartus II compiles the design and generates a bitstream file loaded into the FPGA using USB Blaster.

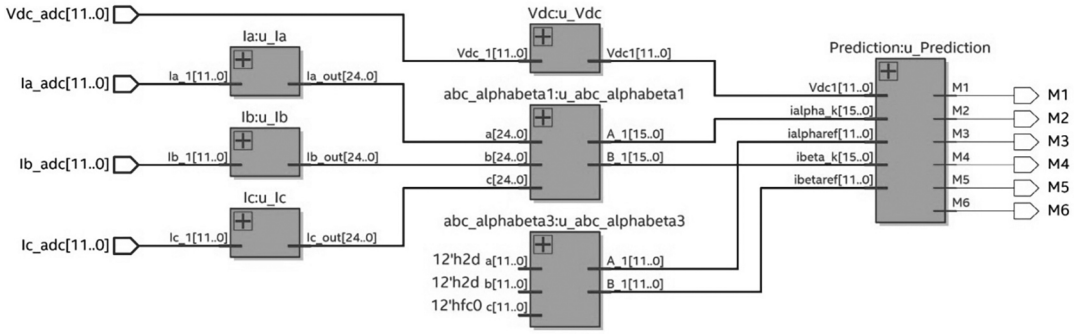
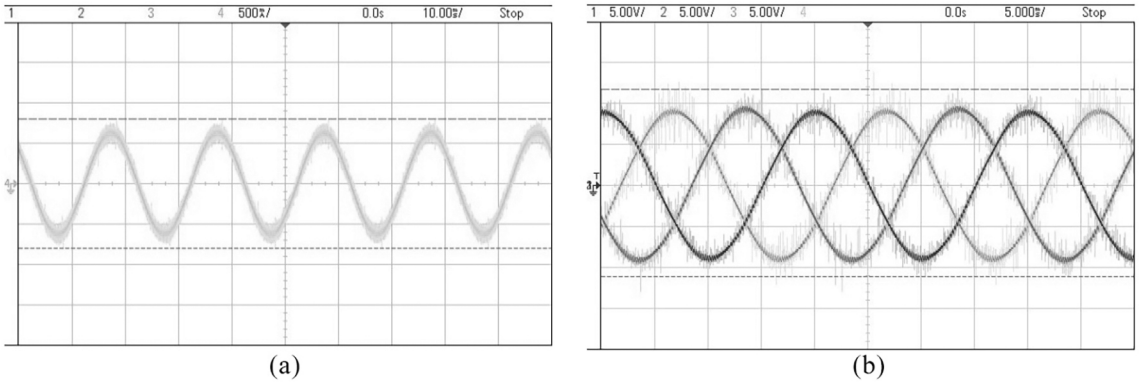
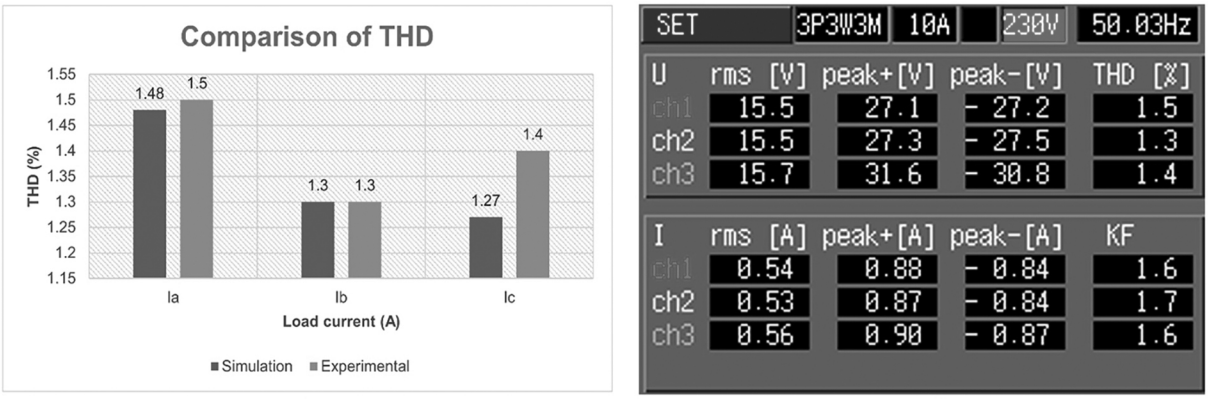


Figure 10. RTL view of the implemented control.



(a) (b)

Figure 11. Experimental results of the predictive current-controlled three-phase VSI: (a) output voltage with RC filter (Scale: Y-axis:10V/div; X-axis: 5 ms/div); (b) load current waveform of phase 'a' measured using Keysight 1146B current probe (Scale: Y-axis: 500 mA/div; X-axis: 10 ms/div).



(a) (b)

Figure 12. THD measurement: (a) experimental results; (b) comparison of experimental results with simulation output.

Table 2 summarises the resource utilisation of the proposed algorithm in MAX®10 FPGA. The parallel and pipelined architecture of FPGA facilitates the resource-efficient implementation with only 38% of the total logic elements utilised for the entire process. Figure 10 shows a part of the RTL view of the implemented control in FPGA. The experimental results are presented in Fig. 11. In Fig. 11(a), the three-phase output voltage of the inverter with an RC filter is shown with R (10Ω, 10W) and C (22 μF, 65V). The load current waveform measured using Keysight 1146B current probe is given in Fig. 11(b). Figure 12(a) shows the THD values calculated using HIOKI Power Quality Analyzer 3197. Finally, the THD values for simulation and experimental results are presented in Fig. 12(b).

6. Conclusion

A simple and efficient rapid prototyping method for the FPGA implementation of computationally intensive

FS-PDCC is presented here. The control algorithm is modelled using Simulink blocks, and Verilog code in fixed point is generated using the HDL Coder™. It helps to prototype the FPGA design faster with minimal knowledge in HDL, reducing the design time and cost. A hardware prototype of three-phase VSI is developed for the experimental analysis, and low-cost Intel® Altera MAX®10 FPGA is used for the control implementation. The results indicate efficient resource utilisation of FPGA, with only 38% of logic elements utilised for the controller implementation. The experimental results show an excellent output quality with a THD value of 1.27%, in close agreement with the simulation results. The proposed workflow is well suited for the designers to reduce the productivity gap and ensure a low-cost solution with efficient resource utilisation and faster computation.

References

[1] F. Blaabjerg, *Control of power electronic converters and systems*, 1st ed. (London: Academic Press, 2018).

[2] C. Buccella, C. Cecati, and H. Latafat, Digital control of power converters—A survey, *IEEE Transactions on Industrial Informatics*, *8*(3), 2012, 437–447.

[3] S. Kouro, M.A. Perez, J. Rodriguez, A.M. Llor, and H.A. Young, Model predictive control: MPC's role in the evolution of power electronics, *IEEE Industrial Electronics Magazine*, *9*(4), 2015, 8–21.

[4] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, Model predictive control of power electronic systems: Methods, results, and challenges, *IEEE Open Journal of Industry Applications*, *1*, 2020, 95–114.

[5] E.F. Camacho and C. Bordons, *Model predictive control*, 2nd ed. (London: Springer-Verlag, 2007).

[6] M. Morari and J.H. Lee, Model predictive control: Past, present and future, *Computers and Chemical Engineering*, *23*(4–5), 1999, 667–682.

[7] A. Linder and R. Kennel, Model predictive control for electrical drives, *Proc. IEEE Annual Power Electronics Specialists Conf. (PESC Record)*, Dresden, 2005, 1793–1799.

[8] T. Geyer, *Low complexity model predictive control in power electronics and power systems*, Doctoral Dissertation, Swiss Federal Institute of Technology, Zürich, Switzerland, 2005

[9] J. Rodriguez, J. Pontt, C.A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann, Predictive current control of a voltage source inverter, *IEEE Transactions on Industrial Electronics*, *54*(1), 2007, 495–503.

[10] P. Cortés, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, and J. Rodriguez, Predictive control in power electronics and drives, *IEEE Transactions on Industrial Electronics*, *55*(12), 2008, 4312–4324.

[11] S. Vazquez, J. Rodriguez, M. Rivera, L.G. Franquelo, and M. Norambuena, Model predictive control for power converters and drives: Advances and trends, *IEEE Transactions on Industrial Electronics*, *64*(2), 2017, 935–947.

[12] G.A. Papafotiou, G.D. Demetriades, and V.G. Agelidis, Technology readiness assessment of model predictive control in medium- and high-voltage power electronics, *IEEE Transactions on Industrial Electronics*, *63*(9), 2016, 5807–5815.

[13] H. Abu-Rub, J. Guziński, Z. Krzeminski, and H.A. Toliyat, Predictive current control of voltage-source inverters, *IEEE Transactions on Industrial Electronics*, *51*(3), 2004, 585–593.

[14] P. Martin Sanchez, O. Machado, E.J. Bueno Pena, F.J. Rodriguez, and F.J. Meca, FPGA-based implementation of a predictive current controller for power converters, *IEEE Transactions on Industrial Informatics*, *9*(3), 2013, 1312–1321.

[15] M. Curkovic, K. Jezernik, and R. Horvat, FPGA-based predictive sliding mode controller of a three-phase inverter, *IEEE Transactions on Industrial Electronics*, *60*(2), 2013, 637–644.

[16] M. Pelcat, C. Bourrasset, L. Maggiani, and F. Berry, Design productivity of a high level synthesis compiler versus HDL, *Proc. Int. Conf. Embedded Computer Systems: Architectures, Modeling and Simulation (IC-SAMOS)*, Agios Konstantinos, 2016, 140–147. https://doi.org/10.1109/SAMOS.2016.7818341.hal-01358210.

[17] J.J. Rodriguez-Andina, M.D. Valdes-Pena, and M.J. Moure, Advanced features and industrial applications of FPGAS-A review, *IEEE Transactions on Industrial Informatics*, *11*(4), 2015, 853–864.

[18] ITRS, 2011. International Technology Roadmap for Semiconductors (http://www.itrs.net/reports.html), available at http://www.itrs.net/reports.html.

[19] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, High-level synthesis for FPGAs: From prototyping to deployment, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *30*(4), 2011, 473–491.

[20] D. Navarro, O. Lucia, L.A. Barragan, I. Urriza, and O. Jimenez, High-level synthesis for accelerating the FPGA implementation of computationally demanding control algorithms for power converters, *IEEE Transactions on Industrial Informatics*, *9*(3), 2013, 1371–1379.

[21] HDL Coder Documentation, https://www.mathworks.com/help/hdlcoder/index.html?s_tid=CRUX_lftnav (accessed Jun. 26, 2021).

[22] Hardware-software co-design—MATLAB & Simulink, https://www.mathworks.com/help/hdlcoder/hardware-software-codesign-1.html (accessed Jun. 26, 2021).

[23] J. Rodriguez and P. Cortes, *Predictive control of power converters and electrical drives*, 1st ed. (Chichester: Wiley, 2012).

[24] P. Cortés, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, and J. Rodriguez, Predictive control in power electronics and drives, *IEEE Transactions on Industrial Electronics*, *55*(12), 2008, 4312–4324.

[25] J. Rodriguez, M.P. Kazmierkowski, J.R. Espinoza, P. Zanchetta, H. Abu-Rub, H.A. Young, and C.A. Rojas, State of the art of finite control set model predictive control in power electronics, *IEEE Transactions on Industrial Electronics*, *9*(2), 2013, 1003–1016.

[26] HDL Coder-MATLAB & Simulink, https://www.mathworks.com/products/hdl-coder.html (accessed Jul. 3, 2021).

[27] Intel. Intel® MAX® 10 Analog to Digital Converter User Guide. N.p., (2021).

Biographies

Deepa Sankar received the B.Tech. degree in electrical and electronics engineering from Pondicherry University in 2004 and the M.Tech. degree in power electronics and drives from Karunya Institute of Technology and Sciences, Coimbatore, in 2006. She is an Associate Professor with Adi Shankara Institute of Engineering and Technology, Kerala, India. Her research interests include power electronic converters for renewable energy, digital control of power electronic converters, and modelling and designing power electronic converters.

S. Lakshmi received the B.E. degree in electrical and electronics engineering from Anna University, Tamil Nadu, India, in 2005 and the M.Tech. degree from Mahatma Gandhi University, Kerala, India, in 2009. She is currently pursuing the Ph.D. degree in power electronics with Mar Athanasius College of Engineering, under APJ Abdul Kalam Technological University, Kerala, India. Her research interests include power electronic converters and its control.

C.A. Babu received the B.Sc. and M.Tech. degrees in electrical engineering from Cochin University of Science and Technology, Cochin, India, and the Ph.D. degree in electrical engineering from the National Institute of Technology, Calicut, India. After acquiring a few years of industrial experience, he worked as a Professor with the Division of Electrical Engineering, School of Engineering, CUSAT. Prof. C A Babu's research interests include energy conservation and management, industrial load management, safety in electrical systems, renewable energy, distributed generation, and smart grids.

K. Mathew received the M.Tech. and Ph.D. degrees from the Indian Institute of Science, Bengaluru, in 2006 and 2013, respectively. Currently, he is serving as a Professor with Mar Athanasius College of Engineering, Kothamangalam, India. His research interests include embedded systems, power electronics, and non-destructive testing.