www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 9, No: 1, 2022 (241-252)

El-Cezerî Journal of Science and Engineering
Vol: 9, No: 1, 2022 (241-252)
DOI :10.31202/ecjse.961021

ECJSE

Research Paper / Makale

Pipelining Strategies and Design Considerations of Predictive Current Control Method

Ozan Gulbudak[1a], Mustafa Gokdag[1b]

[1]Electrical and Electronics Engineering Department, Karabuk University, Karabuk, TURKIYE
ozangulbudak@karabuk.edu.tr

Received/Geliş: 02.07.2021 **Accepted/Kabul:** 30.09.2021

Abstract: This paper explores the pipelining strategies for the model predictive control methods. The array and vector processing methods are examined to discover their applicability in the model predictive current method. The potential benefits of the pipelining methods are investigated, and their design methodologies are scrutinized. The model predictive control is a nonlinear control technique that predicts the system dynamics. The model predictive control (MPC) provides rapid response to the load variations and guarantees robust operation. However, the lower sampling period is the main design constraint to achieve a reliable system operation. The selection of a low sampling period demands a powerful digital controller due to the increasing computational burden. To handle the high calculation burden, a field-programmable gate array (FPGA) is a powerful solution. A proper pipelining strategy enables the use of the MPC in real-time applications. In this paper, pipelining strategies and practical design considerations of the FPGA-based predictive current method are presented. The nine switch converter (NSC) is selected as an experimental case study. The experimental results are provided to demonstrate the theoretical framework. The experimental results prove the feasibility of the array processing and vector processing methods in MPC applications.

Keywords: Nine switch converter, Pipelining strategies, Model predictive current control, FPGA

Tahmin Akım Kontrol Metodu için Paralel Hesaplama Teknikleri ve Tasarım Kriterleri

Öz: Bu makale, model tahmin kontrol metodları için paralel hesaplama tekniklerini araştırır. Dizi işleme ve vektör işleme metodlarının model tahmin akım kontrol metodu uygulamarında kullanılabilirliği araştırılmıştır. Paralel hesaplama metodlarının potensiyel yararları incelenerek, tasarım metodolojileri mercek altına alınmıştır. Model tahmin kontrol metodu, sistem dinamiklerini tahmin eden doğrusal olmayan bir kontrol yöntemidir. Model tahmin kontrol (MTK) metodu yük varyasyonlarına karşı hızlı bir kapalı-çevrim cevabı sağlar, ayrıca güvenilir bir sistem operasyonunu garanti eder. Ancak, kararlı bir operasyonu elde edebilmek için düşük bir örnekleme zamanı seçilmesi gerekmektedir. Örnekleme zamanının düşük seçilmesi, hesaplama yükünü arttıracağı için yüksek performansa sahip bir gömülü sistem işlemcisine ihtiyaç duyulmaktadır. Bu yüksek hesaplama yükünün tolere edilebilmesi için FPGA cihazları güçlü bir çözüm sunmaktadır. İyi tasarlanmış bir paralel hesaplama mimarisi, MTK metodunun gerçek-zamanlı uygulamalarda kullanılabilmesini sağlar. Bu makalede, FPGA-tabanlı model tahmin akım kontrolü için paralel hesaplama mimarileri ve tasarım kriterleri sunulmuştur. Dokuz anahtarlı konvertör (DAK), deneysel uygulamalar için test senaryosu olarak seçilmiştir. Deneysel sonuçlar çalışmada sunulan teorik konsepti desteklenmektedir. Deneysel sonuçlar dizi işleme ve vektör işleme metodunun MTK kontrol yönteminde kullanılabileceğini ispatlamaktadır.

Anahtar Kelimeler: Dokuz anahtarlı konvertör, Paralel hesaplama metodları, Model tahmin akım kontrolü, FPGA

How to cite this article
Gulbudak, O,, Gokdag, M., "*Pipelining Strategies and Design Considerations of Predictive Current Control Method*" El-Cezerî Journal of Science and Engineering, 2022, 9 (1); 241-252

.

Bu makaleye atıf yapmak için
Gulbudak, O,, Gokdag, M., "Tahmin Akım Kontrol Metodu için Paralel Hesaplama Teknikleri ve Tasarım Kriterleri" El-Cezerî Fen ve Mühendislik Dergisi 2022, 9 (1); 241-252.
ORCID ID: 0000-0001-9517-3630; 0000-0001-5589-2278

1. Introduction

MPC method regulates the control variable by assessing the tailored objective function, and the absolute error between the predicted variable and its reference is introduced in the MPC formulation. MPC can compensate for the errors; nonetheless nonlinearities exist in the monitored plant. Conceptually, a discrete-time plant model is used to the future trend of the control goal. The future error is calculated for each allowable control input, and the control actuation offers a minimum error is recorded as an optimum solution [1]–[3]. The cost function can also contain error terms or more than one objective. Depending on applications, the objective functions are formed by a single term or multiple terms. In multi-objective cases, the weighting factors may be required to tune the closed-loop performance. The weighting factor is not necessary for all multi-objective optimization problems unless the error terms are not in the same nature. Depends on the desired closed-loop operation, the cost function can tailor such that all design specifications are met by the designed feedback system [4]–[7]. Even though the system has some uncertainties and nonlinearities, the MPC provides an acceptable reference tracking capability for a wide range of input and output conditions. On the contrary, the secured converter operation may be infeasible when a linear controller (for example proportional-integral (PI)) is chosen as a feedback strategy. Due to the limited operating range offered by the linear controller, the dynamic perturbations can negatively affect the system performance. In the linear controller design approach, the controller is designed around a linearization point. The deviations from the selected operating point (linearization point) have big influence on the controller performance. On this matter, MPC is a favourable control method since all system constraints and other design limitations can be included in the control law. Thus, load perturbations are effectively regulated while system constraints are satisfied. Despite the benefits of MPC routine, there is an important drawback of the MPC in industrial applications [8], [9]. The MPC requires a high computational speed due to the iterative operation for solving the cost function. All allowable control inputs must be used in the process of cost function evaluation. Hence, the size of the solution set has direct impact on the computation burden. The other critical aspect is that the computational burden also increases with the selected prediction step.

Field-programmable gate arrays (FPGAs) have been receiving attention due to their fast calculation capability. In particular, these type of devices is preferred in MPC implementations to handle the computation burden [10], [11]. The key reason for this preference is that FPGA devices allow a user to work with a flexible computation architecture. The pipelining capability of FPGA be fittings to the discrete character of the MPC strategy [12]–[14]. In the predictive control approach, the prediction process and the cost function evaluation must proceed for each voltage vector. This can be iteratively done by computing devices with single-core and multi-cores. However, for a repetitive calculation code block to take full advantage of the multi-core architecture, the code should be restructured and parallelized. But even so, a fully-pipelined architecture cannot be obtained due to the limited available core in the computing devices. In this sense, FPGA devices are useful to increase the instruction processing throughput. One can easily divide the instruction processing cycle into distinct stages of processing as long as ensuring enough hardware resources to process one instruction in each stage. By doing this, the required MPC calculations can be performed in a shorter period. Thus, a strict design constraint on the sampling period can be satisfied in an application where a noticeable short sampling period is inevitable for obtaining robust converter operation.

This paper presents the design considerations of an MPC-based controller using an FPGA device. Different pipelining strategies are presented, and the computation challenges of the most common objective functions used in MPC approaches are reported. To prove the mathematical concept, a nine-switch converter (NSC) control is selected as a case study. The MPC method of NSC is

explained, and the pipelining methods are explained to implement the MPC. The experimental results are reported to visualize the feasibility of the array processing method and vector processing method. The control of NSC is performed in real-time, and a robust operation is attained. The experimental results demonstrate the superiority of the pipelining strategies. The effects of the cost-function forms are examined, and the steady-state performance analysis is performed under the varying sampling period.

2. System Model and MPC Formulation

In this section, the system model and MPC formulation is explained. The NSC system is selected as a case study; thus the NSC system model is derived to express the mathematical relationship of the electrical quantities. The pipelining strategies are implemented to control the NSC, and the output variables are controlled in a closed-loop fashion. In NSC systems, two switching cases are restricted to ensure the safe transition between phases. The first restricted switching action is avoiding the inductive load current interruption. This is an important case since the inductive load can cause an unpleasant voltage spike. The inductive current flow must be ensured to maintain the energy conversion stability. Another restricted switching action is that the selected switching positions should not cause the dc-bus short circuit. This is also a quite important consideration since the dc-bus short circuit causes immense current flow. The short-circuit current is hazardous and drastically harms the electrical components. These two cases must be considered during the commutation process. On the authority of the switching rules of NSC, 27 different switching states are allowable for reliable operation. These permissible states are the possible solutions to the MPC optimization problem. In MPC control law, the closed-loop control strategy is formulated as an optimal control problem. Thus, the candidate solutions (switching states) are used in exploring the global optimum. Since the solution set is finite, the solver of the MPC optimization problem is based on the exhaustive search technique. In this method, all solutions are tested, and the one that offers a minimum cost value is chosen as the global optimum solution. This generic implementation is a well-known approach and guarantees global optimality. However, the exhaustive search-based solvers suffer from a computational burden if the solution set is large. In this case, a high number of possible solutions are available; thus the use of exhaustive search algorithms is impractical. In this work, the solution set is noticeably low, and an exhaustive-search-based solver is adopted. In NSC, 27 voltage vectors are possible; however, some voltage vectors are redundant since some of them result in same leg voltage. Therefore, the solution is reduced to 15 voltage vectors. The reduction in solution set increases the calculation speed due to the requirement of less iteration in the optimization process [15], [16]. A fewer number of iterations lowers the required control calculation and relaxes the computation burden. Regarding computational complexity, the use of a reduced solution set improves the applicability of the MPC method. The matrices that define the relationship between NSC input and output quantities are given by

$$\mathbf{T_U} = \begin{bmatrix} S_{AU} & S_{BU} & S_{CU} \end{bmatrix} \tag{1}$$

$$\mathbf{T_L} = \begin{bmatrix} (1-S_{AL}) & (1-S_{BL}) & (1-S_{CL}) \end{bmatrix} \tag{2}$$

The matrices defined in (1) and (2) are called transition matrices of the NSC. They define how the input quantities affect the output quantities. One can note that the relationship between input and output directly relies on the switch positions. Each switch can have two discrete values: 1 and 0. When the switch position is 1, the current flows through the active switch. When the switch position is 0, the current is blocked by the active switch. The NSC leg voltages are expressed as

$$\mathbf{V_U} = \begin{bmatrix} V_{aU} & V_{bU} & V_{cU} \end{bmatrix}^T \tag{3}$$

$$\mathbf{V_L} = \begin{bmatrix} V_{aL} & V_{bL} & V_{cL} \end{bmatrix}^T \tag{4}$$

By using the (1)-(2) and NSC leg voltages, the following expressions are obtained.

$$\mathbf{V_U} = V_{DC}\mathbf{T_U^T} \tag{5}$$

$$\mathbf{V_L} = V_{DC}\mathbf{T_L^T} \tag{6}$$

The NSC model is represented by (1)-(6). The system model derivation should be precisely performed since the MPC uses the explicit model of the system. In case of the model mismatch or model errors, the performance of the MPC controller is negatively influenced. Thus, the system model derivation is a critical process in the MPC routine. To perform the load current prediction, the dynamic model of the load must be derived. In this work, the resistive-inductive load is used; thus the discrete-time model of the load is given by

$$i_o(k+1) = i_o(k)\left[1-\frac{RTs}{L}\right]+\frac{Ts}{L}v_o(k) \tag{7}$$

The discrete-time model is derived by applying numerical methods. Several numerical methods are available in the literature such as zero-order-hold, first-order-hold, Euler method (Forward or Backward), or Tustin. In terms of the precision of converting process (converting continuous data to sampled data), each technique provides a different performance. In this work, the forward Euler method is used to obtain the expression defined in (7). The prediction model is used to predict the load current for each allowable control input. Then, the cost values are computed. The MPC formulation is performed using the state-space approach. The state-space approach is very convenient to express the control variables (state variables). The state-space model is defined as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}x(k)+\mathbf{B}u(k)\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned} \tag{8}$$

where

$$\mathbf{x}(k) = [i_{oaU}(k) \quad i_{obU}(k) \quad i_{ocU}(k) \quad i_{oaL}(k) \quad i_{obL}(k) \quad i_{ocL}(k)]^T \tag{9}$$

$$\mathbf{u}(k) = [S_{AU} \quad S_{BU} \quad S_{CU} \quad S_{AM} \quad S_{BM} \quad S_{CM} \quad S_{AL} \quad S_{BL} \quad S_{CL}]^T \tag{10}$$

$$\mathbf{A}_{6x6} = \left(1-\frac{RT_s}{L}\right)\mathbf{I}_{6x6} \tag{11}$$

$$\mathbf{B} = \begin{bmatrix} Q_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & W_{3x3} \end{bmatrix}_{6x9} \tag{12}$$

$$\mathbf{Q}_{3x3} := \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}_{3x3} \tag{13}$$

$$\mathbf{W}_{3x3} := \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & -\frac{2}{3} \end{bmatrix}_{3x3} \tag{14}$$

In the system model, **A** matrix relies on the load parameter. The matrix **B** determines the gains of the system inputs. As can be seen from (13) and (14), the control inputs are routed to the output with a constant gain. The system input is the status of the active switch. Depending on the switch

position (ON/OFF), the output variables vary around the reference trajectory. In a well-designed closed-loop system, the deviation from the trajectory remains limited.

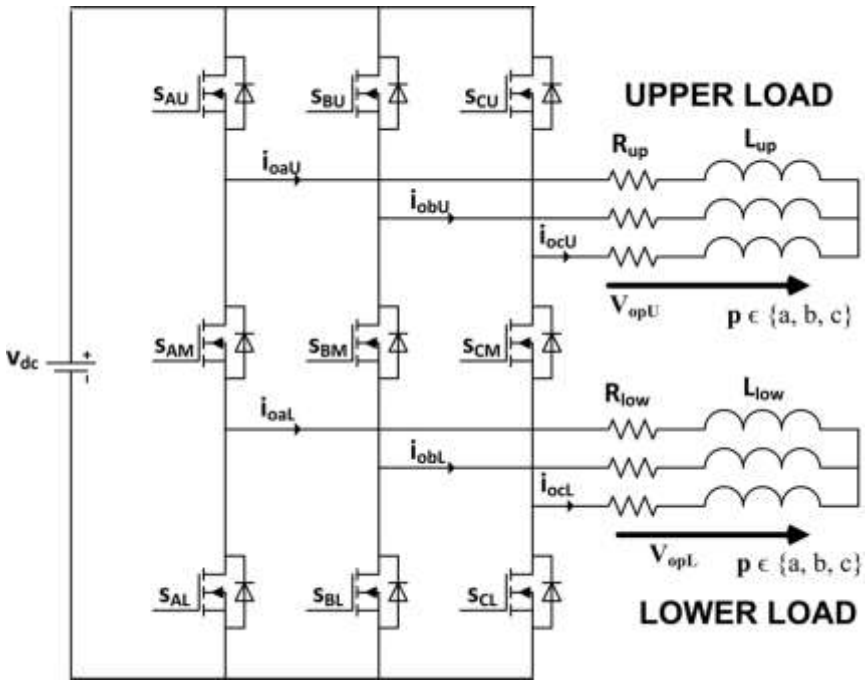


Figure 1. Nine-switch converter circuit diagram.

3. Design Considerations of Model Predictive Control Method

3.1. Pipelining strategy of control tasks

Model predictive control method has the same operations repeated on a large number of different inputs, e.g., voltage vectors of the nine-switch converter. Furthermore, there is no dependency between repeated operations, and the independent instructions can be concurrently performed. In general, the model predictive control method can be divided into four instruction steps:

1. Reading ADC value from the buffer (RADC)\
2. Calculating the future value of the load current (CP)
3. Calculating cost value (CC)
4. Store/Writeback result (WB)

These four steps are repeated for each candidate voltage vector. In a non-ideal pipelining procedure, the instructions flow from the different stages over time. Once the pipeline is filled with all the required instructions, the full pipeline is achieved, see Fig. 2. Different steps do not share the resources. For instance, the calculation of the load current prediction for the voltage vector V_5, and the calculations of the load current prediction for the voltage vector V_4 are completely separate. In the MPC method, pipeline realization is non-ideal because the execution time for different instructions is different from each other.

Principally, different instructions are forced to go through the same pipeline stages. However, they do not need the same pipeline stages. This phenomenon is called external fragmentation. Furthermore, the different pipeline stages are not uniformly divided since non-identical stages do not have not the same latency. Hence, some pipeline stages may be faster than the others but still

take the same clock cycle time. Due to this undesirable characteristic, some part of the clock cycle is wasted, and it reduces the performance.

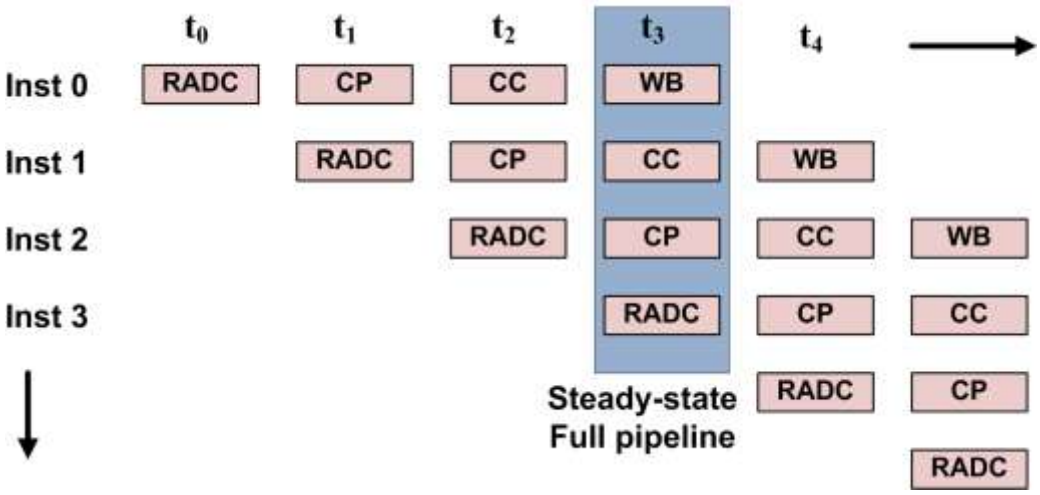


Figure 2. The illustration of the pipeline operation

The resource view of the pipelining process is exemplified in Fig. 3. There are different stages and time steps. Different instructions are performed by the independent FPGA logics, and a full pipeline is achieved.

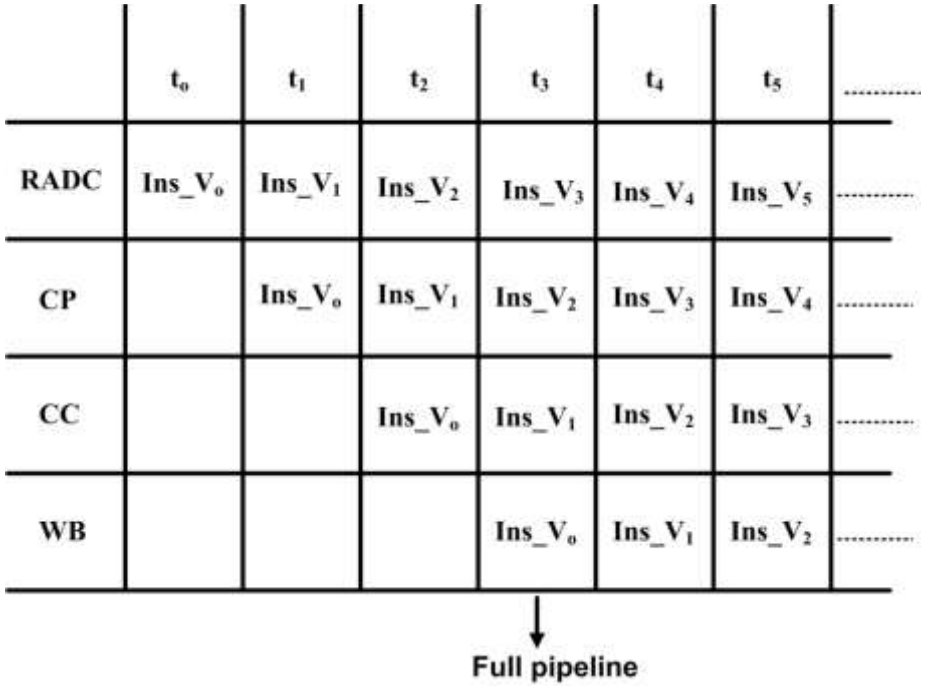


Figure 3. Resource view of the pipelined process

The concurrency arises from performing the same operation on different pieces of data or executing operations in parallel. Besides, the computations can be irregular. Two different pipelining approaches are illustrated in Fig. 4 and Fig. 5. In Fig. 4, the main principle of the array processing approach is exemplified with 15 processing elements. For any given instruction stream, each of the processing elements can execute any type of instruction in the array processing approach. Thus, the same type of instructions for the different voltage vectors is simultaneously executed. This concludes that the same operations are performed at the same time. For any given execution cycle,

all the processing elements execute the same operation. However, different operations are executed in the same FPGA areas.

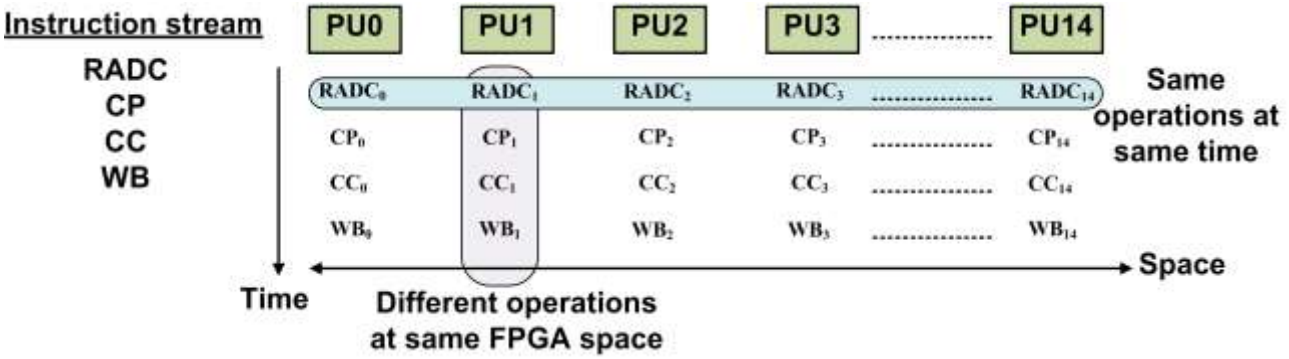


Figure 4. Array processing approach for paralleling the instructions. PU refers to processing unit.

On the other hand, the vector processing approach is exemplified with four specialized processing units. The main difference between the array processing approach and the vector processing approach is that the processing units of the vector processing method are specialized in performing the specific instructions. In the vector processing method, different operations are executed at the same time at any execution cycle. On the contrary, the same operations are executed in the same FPGA areas. Lastly, instructions are performed on multiple data elements at the same time using the different spaces in the array processing approach. In the vector processing approach, instructions are performed on multiple data elements in the following execution steps using the same space. Simply, it can be concluded that these two approaches have different time-space duality characteristics.

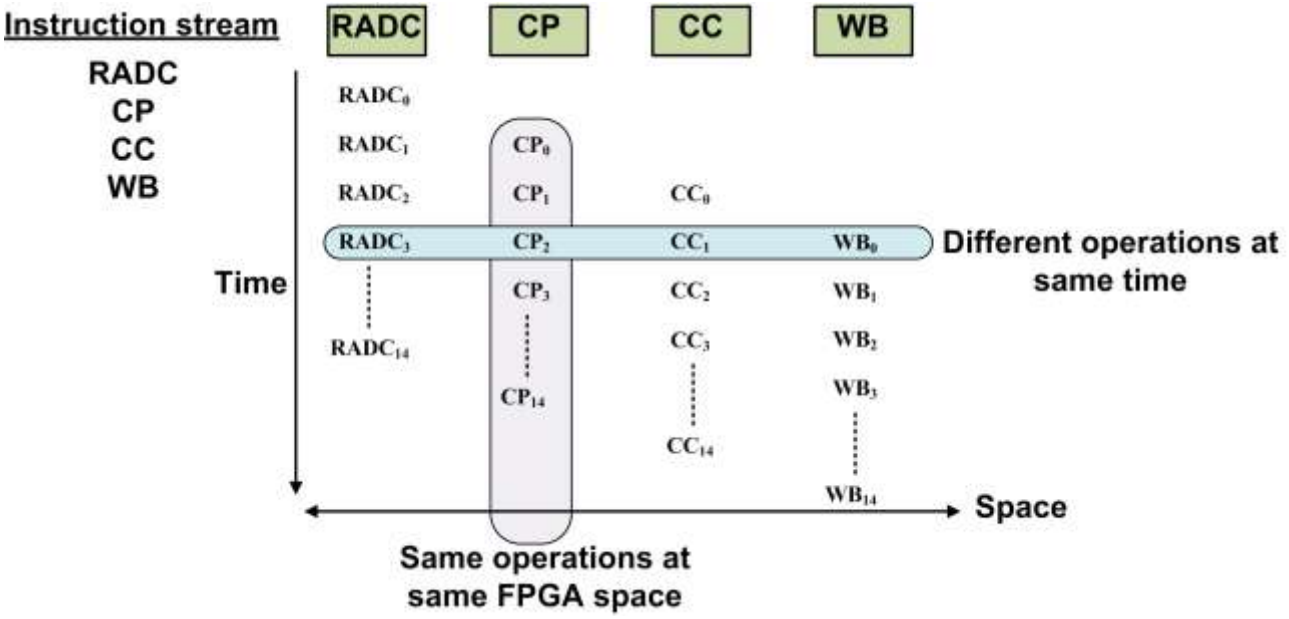


Figure 5. Vector processing approach for paralleling the instructions

3.2. Effects of Cost-Function Choice on the Computational Burden

In the MPC method, the system behavior highly relies on the selected objective since the definition of the cost function characterizes the closed-loop response. In power electronics applications, different types of objective function terms are used to regulate the system. The several types of cost function forms are tabulated in Table I. The most common ℓ_1 norm, the square of the ℓ_1 norm and

ℓ_2 norm. In particular, the usage of the ℓ_1 norm is widespread due to its simple implementation. The ℓ_1 norm of a vector **x** is given by $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$. The absolute value of the error term is considered, and the cost value increases with the discrepancy between prediction and reference. In FPGA implementation, the evaluation of ℓ_1 norm-based cost function computationally cheap. However, the cost calculation using ℓ_2 norm-based cost function computationally expensive compared to the ℓ_1 norm-based cost calculation. The main reason is that the evaluation of ℓ_2 norm-based objective function requires Coordinate rotation digital computer (CORDIC) implementation. CORDIC method is a very useful approach to calculate trigonometric, square roots, and exponentials. The ℓ_2 norm of a vector **x** is given by $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$. To effectively calculate the ℓ_2 norm-based objective function, the sqrt(.) the function must be implemented using the CORDIC algorithm with a proper resolution. Well-designed function emulation is vital since a poor CORDIC design negatively affects the closed-loop performance. The other two forms Frobenius and Nuclear norms are defined in (15) and (16) for a vector **x**. In terms of computational complexity, the Frobenius and Nuclear norm-based objective function evaluation increases the calculation complexity. These types of objective functions require Look-up-Table (LUT) to store pre-calculated values and an additional CORDIC algorithm implementation. In general, ℓ_1 norm-based cost function has lower computational complexity while provides an acceptable reference tracking performance.

$$\|\mathbf{x}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |x_{ij}|^2} \tag{15}$$

$$\|\mathbf{x}\|_* = trace(\sqrt{\mathbf{X}^*\mathbf{X}}) \tag{16}$$

Table 1. Comparison of cost function forms

Evaluation Metric	$\|\cdot\|_1$	$\|\cdot\|_1^2$	$\|\cdot\|_2$	$\|\cdot\|_F$	$\|\cdot\|_*$
Definition	ℓ_1 norm	$(\ell_1)^2$ norm	ℓ_2 norm	Frobenius norm	Nuclear norm
Computation Cost	Cheap	Cheap	Expensive	Very Expensive	Very Expensive
CORDIC Requirement	No	No	Yes	Yes	Yes
LUT usage	No	No	Yes	Yes	Yes
Complexity	Low	Med.	High	High	High

3.3. Effects of Sampling Period on the Control Performance

The sampling period has a significant influence on closed-loop dynamic response. The control update rate increases with the sampling frequency, and the controller starts to give a quick response to the rapid change. Also, improved steady-state performance can be attained by lowering the sampling period (speed up the update rate). To examine the closed-loop performance (under steady-state conditions) of the MPC method, total harmonic distortions (THD) of the control variables are calculated. In power electronics, the THD is a fair evaluation parameter to assess power quality. The THD of any signal can be computed as

$$\mathrm{THD} = \frac{\sqrt{\sum_{n=2}^{\infty} \mathrm{I}_n^2}}{\mathrm{I}_{\mathrm{fund}}} \tag{17}$$

Since two sets of three-phase ac load are fed by the NSI combined with the MPC method, the average THD of both load stages is considered in evaluating the steady-state performance. The mean value of THD for multiple loads are given by

$$i_{o\_THD} = \frac{i_{up\_THD} + i_{low\_THD}}{2} \tag{18}$$

To calculate the spectral contents, Fast Fourier Transform (FFT) analysis was performed up to 10 kHz. The average total harmonic distortions versus the sampling period are presented in Fig. 6. Based on Fig. 6, THD tends to increase with a higher sampling period. Thus, a lower sampling period is preferable due to a noticeable improvement in load current quality.

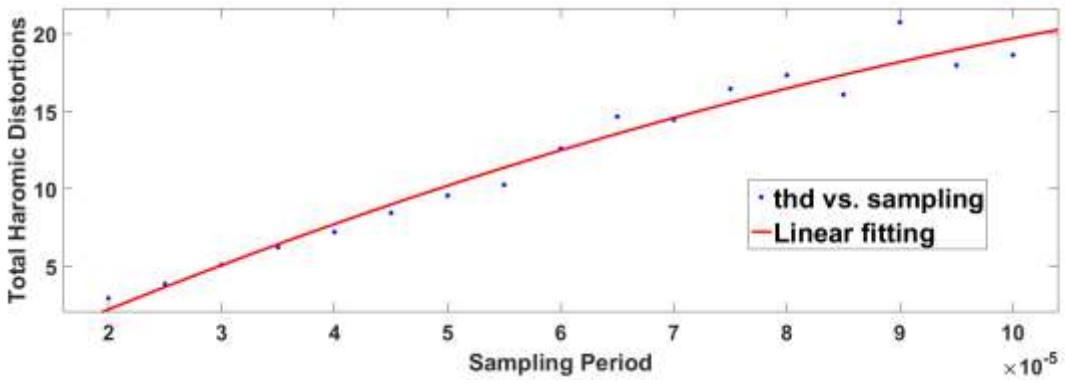


Figure 6. The average total harmonic distortions of load current versus sampling period of the model predictive control method.

To obtain a higher sampling rate, the run-time of the model predictive control algorithm must be decreased. A higher sampling frequency improves the load current THD in steady-state, but the switching losses are negatively affected. Since the converter tends to do more switching transition, the switching loss increases. The working mechanism of the FPGA-based model predictive control method is illustrated in Fig. 7.

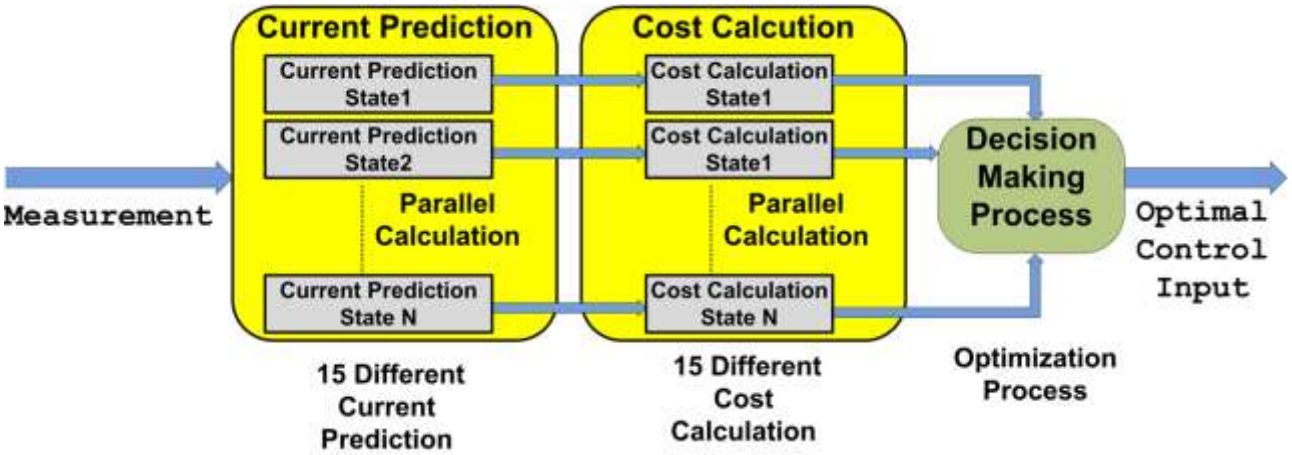


Figure 7. The main idea of the FPGA-based model predictive control approach.

4. Experimental Results

To verify the capability of the pipelining strategy presented in the previous section and to obtain measured results for a nine-switch converter-based multi-load system, an experimental setup was designed and implemented. Fig. 8 shows the realization of the nine-switch converter on a 2-layer printed circuit board. The pipelining architecture is designed based on the array processing approach. The sampling period is 20 µs and the DE0-nano FPGA board from Terasic™ is used for implementing the pipelining methods. Two sets of resistive-inductive loads are used for experimental investigation. The resistive load is 3 Ω, and the inductive load is 3.5 mH.

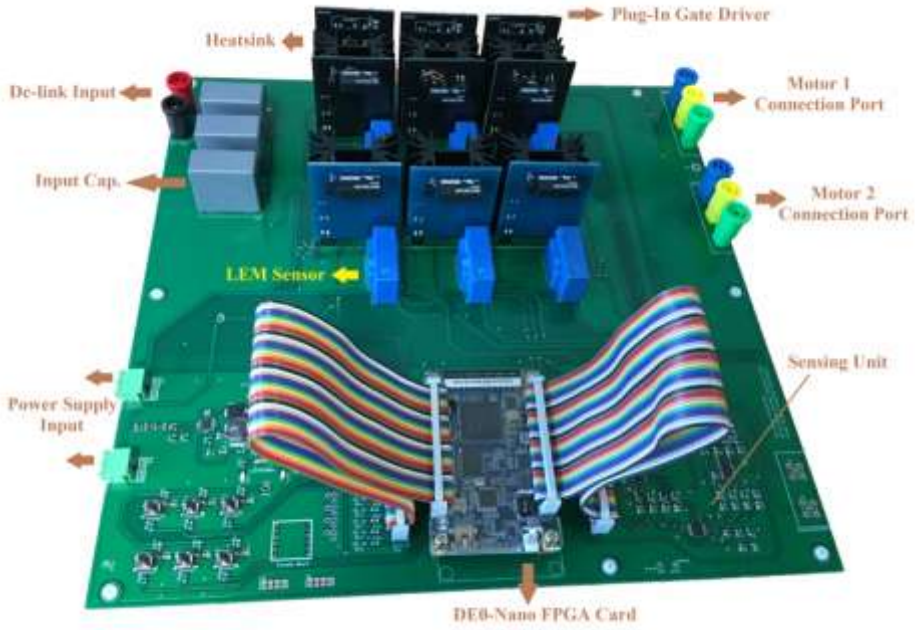
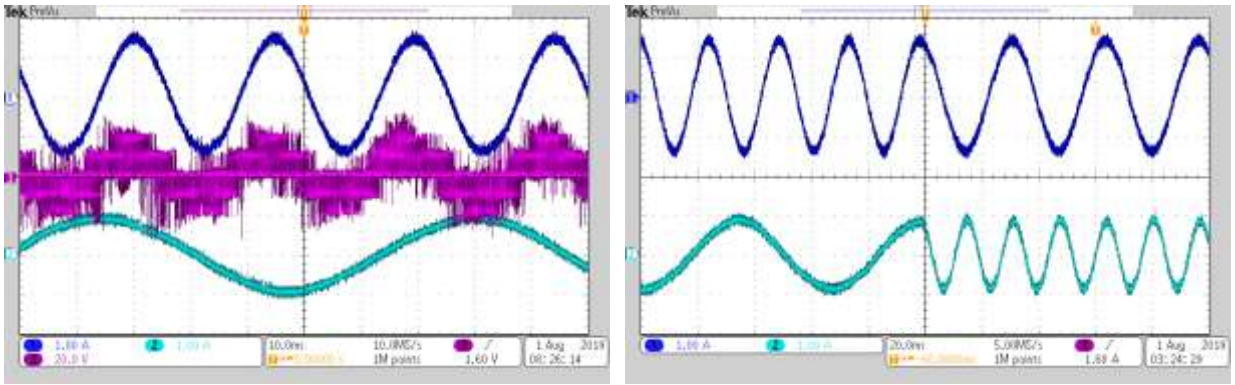


Figure 8. Annotated photography of the nine-switch converter prototype consisting of an FPGA evaluation board.



(a) Steady-state waveform (b) Transient response to frequency step

Figure 9. Experimental results of FPGA-based model predictive controlled nine-switch converter. (a) Steady-state waveforms. (b) Transient waveforms. Scope channel 2: Lower load current. Scope channel 3: Upper load voltage.

To investigate the reference tracking capability of the proposed FPGA-based MPC method, two independent load current references are introduced to the different load stages. The steady-state waveforms of the load currents are presented in Fig. 9(a). The upper load current reference is 1.5

A/40 Hz, and the lower load current reference is 15 Hz. Based on the experimental results reported in Fig. 9(a), independent load stages have an individual fundamental frequency, and fully independent control is achieved. This proves that both loads are operated under different conditions. In a well-designed closed-loop implementation, separate load stages should not be affected by the variations of the other loading effects. To verify experimentally this dynamic performance, a frequency step was applied to the upper load from 50 Hz to 30 Hz while the lower load frequency is changing from 15 Hz to 60 Hz. The lower load current is almost insensitive to a change of the upper load current reference. Concurrently, the operation of the upper load is uninfluenced by the applied step to the lower load. In conclusion, an FPGA-based predictive controller properly compensates the independent current error terms and guarantees reliable operation. Both load currents track their reference trajectories, and independent load control is achieved. Employing the pipelining architecture based on the array processing approach, the algorithm run-time of the MPC routine is roughly 1.25 μs.

5. Conclusion

In this paper, the FPGA-based implementation challenges of the model predictive control method have been explained. Two different pipelining procedures have been discussed, and the effects of the cost function selection on computational burden are presented. The algorithm run-time can be significantly reduced by pipelining strategies (Array processing method and vector processing method). The reduction in algorithm execution time relaxes the sampling period constraint and allows us to perform the MPC method in a lower sampling period. The sampling period highly influences the load current quality since THD increases with the sampling period. Thus, the selection of the sampling period is quite vital to obtain a good steady-state and transient performance. The selection of the high sampling frequency improves the power quality and stability of the system. To design the discrete-time MPC with a lower sampling period, FPGA devices combined with proper pipelining strategies are quite effective.

Acknowledgments

This work was supported by the Scientific and Technology Research Council of Turkey (TUBITAK) under Grant 117E769.

Authors' Contributions

OG and MG are responsible for deriving the theoretical concepts, building the hardware platforms, developing the embedded codes and performance analyses. Both authors read and approved the final manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

[1]. I. Gonzalez-Prieto, I. Zoric, M. J. Duran, and E. Levi, "Constrained Model Predictive Control in Nine-Phase Induction Motor Drives," *IEEE Trans. Energy Convers.*, vol. 34, no. 4, pp. 1881–1889, Dec. 2019.

[2]. R. E. Perez-Guzman, M. Rivera, and P. W. Wheeler, "Recent Advances of Predictive Control in Power Converters," in *2020 IEEE International Conference on Industrial Technology*

(ICIT), 2020, pp. 1100–1105.
[3]. G. Pei, L. Li, X. Gao, J. Liu, and R. Kennel, "Predictive Current Trajectory Control for PMSM at Voltage Limit," *IEEE Access*, vol. 8, pp. 1670–1679, 2020.
[4]. M. Gokdag and O. Gulbudak, "Model predictive control of AC-DC matrix converter with unity input power factor," in *2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)*, 2018, pp. 1–5.
[5]. M. Gokdag and O. Gulbudak, "Model Predictive Control for Battery Charger Applications with Active Damping," in *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, 2019, pp. 140–145.
[6]. S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model Predictive Control for Power Converters and Drives: Advances and Trends," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Feb. 2017.
[7]. S. Vazquez *et al.*, "Model Predictive Control: A Review of Its Applications in Power Electronics," *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, Mar. 2014.
[8]. M. Siami, D. A. Khaburi, M. Rivera, and J. Rodriguez, "A Computationally Efficient Lookup Table Based FCS-MPC for PMSM Drives Fed by Matrix Converters," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 7645–7654, Oct. 2017.
[9]. P. Karamanakos, T. Geyer, and R. P. Aguilera, "Long-Horizon Direct Model Predictive Control: Modified Sphere Decoding for Transient Operation," *IEEE Trans. Ind. Appl.*, vol. 54, no. 6, pp. 6060–6070, Nov. 2018.
[10]. O. Gulbudak and M. Gokdag, "Asymmetrical Multi-Step Direct Model Predictive Control of Nine-Switch Inverter for Dual-Output Mode Operation," *IEEE Access*, vol. 7, pp. 164720–164733, 2019.
[11]. 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.
[12]. T. J. Vyncke, S. Thielemans, and J. A. Melkebeek, "Finite-Set Model-Based Predictive Control for Flying-Capacitor Converters: Cost Function Design and Efficient FPGA Implementation," *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 1113–1121, May 2013.
[13]. Z. Zhang, F. Wang, T. Sun, J. Rodriguez, and R. Kennel, "FPGA-Based Experimental Investigation of a Quasi-Centralized Model Predictive Control for Back-to-Back Converters," *IEEE Trans. Power Electron.*, vol. 31, no. 1, pp. 662–674, Jan. 2016.
[14]. F. Wang, L. He, and J. Rodriguez, "FPGA-Based Continuous Control Set Model Predictive Current Control for PMSM System Using Multistep Error Tracking Technique," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13455–13464, Dec. 2020.
[15]. O. Gulbudak and E. Santi, "Model predictive control of dual-output nine-switch inverter with output filter," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2015, pp. 1582–1589.
[16]. O. Gulbudak and M. Gokdag, "Predictive dual-induction machine control using nine-switch inverter for multi-drive systems," in *2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)*, 2018, pp. 1–6.