Exploring the Potential of FPGA in High-Frequency Switching DC-DC Boost Converters Using Model Predictive Control

Qingcheng Sui
Dept. of Electrical Engineering
KU Leuven - EnergyVille
Diepenbeek - Genk, Belgium
qingcheng.sui@kuleuven.be

Bangli Du
Dept. of Electrical Engineering
KU Leuven - EnergyVille
Diepenbeek - Genk, Belgium
bangli.du@kuleuven.be

Yu Zuo
Dept. of Electrical Engineering
KU Leuven - EnergyVille
Diepenbeek - Genk, Belgium
yu.zuo@kuleuven.be

Wilmar Martinez
Dept. of Electrical Engineering
KU Leuven - EnergyVille
Diepenbeek - Genk, Belgium
wilmar.martinez@kuleuven.be

***Abstract*—Model Predictive Control (MPC) has demonstrated significant potential in the control of power electronics; however, the trade-off between the online computational demands and the storage requirements for offline explicit control laws presents a persistent challenge for industrial controller implementation. This work introduces an innovative FPGA-based neural network controller tailored for high-switching frequency DC/DC Boost converters. Multi-Layer Perceptron (MLP) is used to approximate parametrized Explicit MPC controller capable of adapting to fluctuations in the systems variables and parameters. The high-speed implementation of the MLP using High-Level Synthethis (HLS) on the FPGA enables operation at higher switching frequencies and allows for higher power density in power converters design with wide bandgap (WBG) devices such as GaN. This method offers substantial improvements over conventional techniques by delivering low-latency computation. Experimental validation on a 500W, 5MHz GaN-based Boost converter prototype demonstrates that the MLP-based controller closely replicates the performance of the original Explicit MPC (EMPC) controller, achieving superior transient response.**

***Index Terms*—FPGA, HLS, MPC, MLP, Boost Converter**

I. Introduction

Model Predictive Control (MPC) has become a powerful tool in power electronics control due to its ability to handle system constraints and provide fast dynamic responses[1]. By predicting system behavior over a finite horizon and optimizing control actions accordingly, MPC explicitly accounts for constraints, making it more suitable for practical applications than traditional controllers like Linear Quadratic Regulator (LQR) or Linear Quadratic Gaussian (LQG). However, the real-time implementation of online MPC is often hindered by the significant computational and memory demands required to solve quadratic programming (QP) problems, especially in industrial settings that require high-speed solutions[2], [3].

Explicit MPC (EMPC) mitigates some of these computational challenges by pre-storing control laws for various input states, eliminating the need for online computation[4]. While EMPC reduces computation time, it suffers from substantial memory requirements, particularly for systems with long prediction horizons, limiting its implementation on microcontroller units (MCUs)[4][5]. Approximation methods that reduce the number of stored regions can decrease memory usage but may compromise control accuracy.

Recent advancements in machine learning (ML) offer new avenues for approximating MPC control laws, specifically through the use of multilayer perceptron (MLP) neural networks. Learning-based MPC (LB-MPC) leverages data-driven models to approximate optimal control actions, significantly reducing online computation time[6]. MLPs, known for their universal function approximation capabilities, can handle the complexity of MPC control laws without incurring substantial memory overhead on conventional DSP[5].

Field-Programmable Gate Arrays (FPGAs) present an ideal platform for implementing these MLP-approximated MPC controllers. FPGAs offer parallel processing capabilities and hardware-level configurability, enabling them to execute complex algorithms with low latency and high data-throughput[7]. Traditionally, the adoption of FPGAs in industrial control has been limited due to the complexity of hardware design and long time-to-market, with Digital Signal Processors (DSPs) and MCUs being preferred for their maturity and ease of programming in C/C++[8]. However, the introduction of High-Level Synthesis (HLS) tools has lowered the barrier to FPGA programming, allowing developers to design hardware using high-level languages and significantly reducing development time[9], [10].

By combining MLP-approximated MPC with FPGA acceleration, we can implement more complex control algorithms with less programming effort and achieve shorter computation times. This approach not only overcomes the computational limitations of traditional MPC implementations but also leverages the parallelism of FPGAs to attain higher control loop rates. This is particularly crucial for modern power converters employing wide bandgap (WBG) devices like silicon carbide (SiC) and gallium nitride (GaN), which operate at higher switching frequencies[10].

In this research, we propose a data-driven method to approximate a parameterized MPC controller using an MLP that adapts to changes in both plant state variables and parameters. The MLP is efficiently implemented on an FPGA using HLS tools, supporting high-frequency switching and enhancing power density in power converter designs. Our approach enables the execution of more complex algorithms with less programming effort and shorter computation times, addressing the challenges of real-time MPC implementation in industrial applications.

The main contributions of this research are as follows:

- **Development of a Data-Driven Approach**: We develop a data-driven method to approximate a parameterized MPC controller using an MLP, capable of adapting to variations in plant state variables and parameters.
- **Efficient FPGA Implementation**: We achieve fast implementation of MLP computations on an FPGA using HLS tools, supporting high-frequency switching operations and enhancing power density in power converter designs.

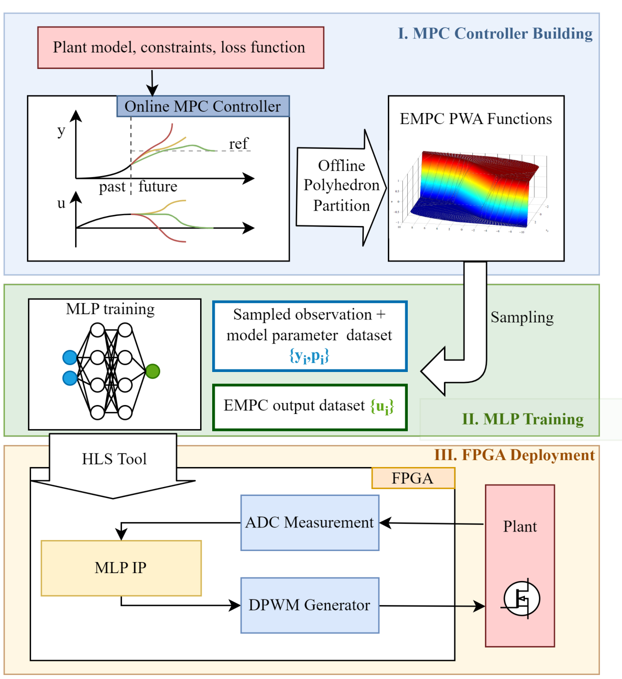


Fig. 1: Overview of proposed approach

II. METHODOLOGY

The design process of our proposed FPGA-based MPC controller for high-switching frequency Boost converter consists of following steps. First, the Boost converter model is built and the MPC problem is formulated. Second, the paired state-control dataset are constructed by sampling input state variables and output control variables of EMPC controller, and quantization-aware training of MLP. Third, MLP are packaged as IP and deployed in the FPGA design with other necessary modules connected.

A. EMPC controller formulation for Boost Converter

Fig.X shows the DC/DC Boost Converter. Under assumption that sampling frequency is sufficiently high, using the Euler approximation, the discretized form of the state-space average equations can be derived as:

$$
\begin{aligned}
V_o(k+1) &= V_o(k) + \frac{1-d}{C} i_L(k) T_S - \frac{1}{RC} V_o(k) T_S \\
i_L(k+1) &= i_L(k) - \frac{1-d}{L} V_o(k) T_S + \frac{1}{L} V_g T_S
\end{aligned}
\tag{1}
$$

where i_L is the current in the inductor of the Boost converter, V_o is the output voltage on the load, V_g is the input source voltage, d denotes the duty cycle, T_s represents the sampling interval, and R, C, L are the load resistance, output capacitance, and inductance, respectively.

We derive the linearized model around the operating point $x_{\text{ref}} = (V_{o,\text{ref}}, i_{L,\text{ref}})$, $u_{\text{ref}} = d_{\text{ref}}$.

$$
\Delta x_{k+1} = A' \Delta x_k + B' \Delta u_k \tag{2}
$$

$$
A' = \begin{bmatrix} 1 - \frac{T_S}{RC} & \frac{(1-d_0)T_S}{C} \\ -\frac{(1-d_0)T_S}{L} & 1 \end{bmatrix}, \quad B' = \begin{bmatrix} -\frac{i_{L0}T_S}{C} \\ \frac{V_{o0}T_S}{L} \end{bmatrix} \tag{3}
$$

The subscript 0 denotes constants under a specific equilibrium and reference operating condition for the MPC tracking problem. The reference values of the output voltage are given as $V_{o,\text{ref}}$, the reference values of the other key variables can be derived as:

$$
d_{\text{ref}} = 1 - \frac{V_{o,\text{ref}}}{V_g}, \quad I_{L,\text{ref}} = \frac{V_{o,\text{ref}}}{R_{\text{load}}} \left(1 - \frac{V_{o,\text{ref}}}{V_g} \right)
$$

.

The discretized linear state-space form of the Boost converter can be derived and applied to the MPC problem formulation. The MPC problem can be established as follows:

$$
\begin{aligned}
u_{\text{mpc}}(\Delta x_k') = u_{\text{ref}} + [1, 0, \ldots, 0] \arg \min_{\Delta u_i', \ldots, \Delta u_{k+N-1}'} \\
\sum_{i=k}^{k+N} \left(\Delta x_i'^{T} Q \Delta x_i' + \Delta u_i'^{T} R \Delta u_i' \right)
\end{aligned}
\tag{4}
$$

subject to:

$$
\Delta x_{k+1}' = A' \Delta x_k' + B' \Delta u_k', \ldots
$$

$$
\Delta x_{k,1}' + V_{o,\text{ref}} \le V_{o,\max}, \quad \Delta x_{k+1,1}' + V_{o,\text{ref}} \le V_{o,\max}, \ldots
$$

$$0 \le u_k \le 1, \quad 0 \le u_{k+1} \le 1, \ldots$$

where Q and R denote the penalty and control regularization coefficient matrices in the cost function, and k denotes the time step. After formulating the MPC problem, we use multiparametric optimization tools, such as MPT3 [11], to obtain an explicit solution.

By identifying the active constraints and collecting primal and dual optimality conditions, the algorithm of EMPC identifies different regions or polyhedrons in state space, which may lead to piecewise affine (PWA) local optimal solutions. The algorithm repeatedly samples states within the predefined state-space limits until whole coverage of the predefined state-space by the PWA control function. Multiparametric optimization tools such as MPT3 [11] can be utilized to obtain an explicit solution. The mapping of two input variables (that is, $V_{\mathrm{o}}, I_{\mathrm{L}}$) to the control output of the MLP is illustrated in Fig. 2.

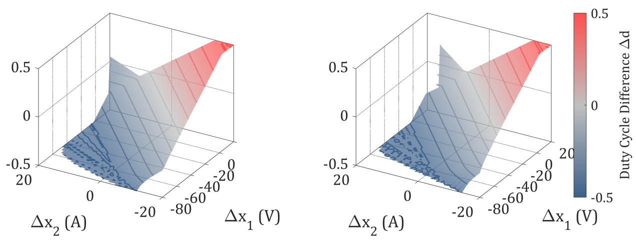


Fig. 2: EMPC state/control mapping visualization under different R_{load} (5Ω and 10Ω)

B. Quantization-aware MLP-based Controller Training

By considering ML algorithms together with heterogeneous computing, we can greatly benefit from developments outside the field of power electronics, while maintaining the excellent control characteristics of advanced control algorithms such as MPC. Compared with online MPC controllers and explicit model predictive control (EMPC), MLP-based MPC controllers can operate in real-time on specialized hardware (e.g., FPGA), thanks to the parallelized and quantized neural network inference computation.

To fully exploit the capabilities of FPGA hardware and achieve efficient real-time control, it is essential to consider quantization during the training of the MLP-based controller. Quantization reduces the numerical precision of the neural network's weights and activations, allowing for faster computations on hardware with limited resources. However, naive quantization after training can lead to significant performance degradation. Therefore, we employ quantization-aware training to mitigate this issue.

In our approach, we focus on accelerating machine learning inference by performing quantization-aware training of the MLP for real-time power electronics controllers. This involves incorporating the quantization effects into the training process so that the neural network learns to compensate for the reduced precision. Specifically, we simulate the finite precision of the FPGA's arithmetic units during training by quantizing the weights and activations to fixed-point representations that match the target hardware.

To prepare the MLP for quantization-aware training, we first generate a dataset required for training and validation. Within the permitted state-space range, we uniformly sample the system states $x^{(i)} = (v_o, i_L, i_o)$ as input for the neural network model. The corresponding output of EMPC $y^{(i)} = (u_i)$ is also collected to form a paired data set $D = \{(x^{(i)}, y^{(i)}) : 1 \ldots N\}$. The dataset is then divided into three parts for training, validation, and testing separately. A MLP-type neural network model, consisting of more than three fully connected feedforward layers, is built and trained on the dataset. ReLU activation functions are used for the sake of simplicity in the FPGA implementation.

C. FPGA Implementation of MLP-based controller

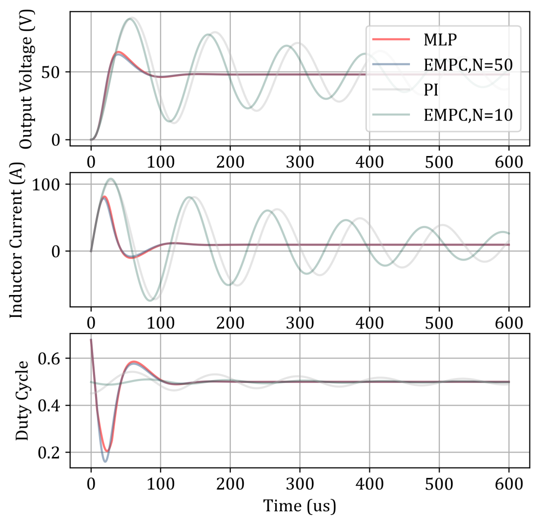
In contrast to conventional DSPs used for power electronics control, FPGAs offer the advantage of custom parallel computation when running, which can greatly reduce computational latency.

However, due to the limited computational resources available for consideration of cost-effectiveness in industrial control, a trade-off between latency and resource utilization is required. In the context of neural network implementations on FPGAs, the most critical resources are often digital signal processing (DSP) slices. In deploying the quantized MLP on the FPGA, we map the fixed-point representations directly onto the hardware's DSP slices and lookup tables (LUTs), optimizing for parallel execution by using tools like HLS. HLS tool (e.g. Vitis HLS) can help to easily find an RTL design that realizes the desired behavior in four simple steps: C-simulation, synthesis, C/RTL co-simulation, IP exporting[8]. The IP block can then be integrated to a block design together with other modules like ADC reading, DPWM control signal generation, DMA data storage, industrial communications, etc.

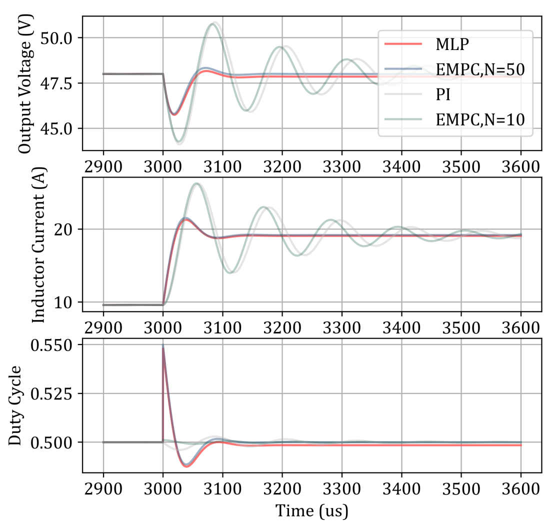
III. Experiments

To demonstrate the effectiveness of the proposed MLP-based MPC controller, a 500W, 5MHz GaN-based Boost converter simulation is built. The system operates with an output voltage of 48V, input voltage of 24V. The load resistance varies between 5Ω and 10Ω. The inductor 4μH, and the capacitor is 20μF. The prediction horizon is 50, the output voltage limit is 100V, and the inductor current is limited to 60A.

The MLP model presented here consists of four fully connected layers with decreasing dimensions, specifically configured as follows: an input layer of 3 units, two hidden layers each containing 10 units, and two subsequent layers with 2 and 1 unit, respectively. Activation is applied using the ReLU function for the initial three layers, while the final output layer employs a clamping function to restrict output values within the [0, 1] range. PI controller is tuned optimally using the same lost function as the MPC controller under the given transient testing cases.



(a) $V_{\text{o,init}} = 0V, I_{\text{L,init}} = 0A$



(b) $V_{\text{o,init}} = 48V, I_{\text{L,init}} = 9.6A$

Fig. 3: Simulation results of boost DC-DC Boost converters using PI, EMPC controller, proposed MLP-based MPC controller under cold-start and sudden R_{load} changes

Fig. 3 illustrates the system response under rated converter working conditions with various controllers. The results demonstrate that the MLP-based controller closely approximates the performance of the original EMPC controller, with only minor deviations, while both controllers effectively reduce output voltage overshoot. As load resistance varies, which typically necessitates a linear model parameter update in the online MPC scheme for controller adaptability, the offline EMPC requires an additional transformation to an explicit form to accommodate these changes. Without this adjustment, model inaccuracies can lead to significant deviations from the reference values. In contrast, because the MLP-based controller is directly trained offline with load current as an input parameter, it adapts seamlessly to changes without requiring controller updates; the load current serves as an additional input feature for the MLP model, ensuring robustness across continuously varying operating conditions.

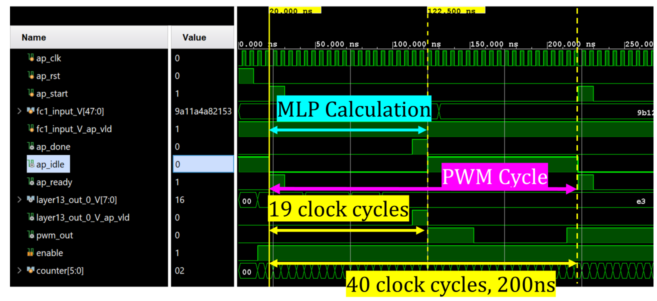


Fig. 4: Waveform Diagram of our RTL implementation on Zynq7020

TABLE I: FPGA Logic and Memory Resource Requirements for MLP IP

FPGA Resources	Req.	Available	Used %
DSP48E	26	220	11.8%
FF	5801	106K	5.5%
BRAM_18K	0	280	0%
LUT	6747	53K	12.7%

Conventional DSPs typically use a sequential execution style, where tasks like signal sensing, conditioning, control algorithm computation, and ePWM register updates must be completed within a tight control time window, often resulting in limited determinism. In contrast, the prototype MLP-based controller was implemented on a Zynq-7020 FPGA (xc7z020clg400-2) with a 200 MHz clock frequency. As shown in Fig. 4, the digital PWM cycle requires 40 clock cycles, while the MLP computation takes only 19 clock cycles, utilizing DSP48E, Flip-Flops (FFs), and LUTs. As shown in Table I, this efficient MLP implementation leaves ample FPGA resources available for additional tasks such as health monitoring and industrial communication. A comparison of various MPC controllers on different platforms (FPGA, DSP, Lab-box) is provided in Table II. Our model demonstrates superior performance with minimal computation delay and an extended prediction horizon.

IV. Conclusion

This work presents a novel FPGA-based neural network controller tailored for high-switching frequency DC/DC Boost converters. By leveraging a dataset to approximate a parameterized Model Predictive Control (MPC) controller, the proposed approach effectively adapts to variations in both state variables and model parameters. The FPGA implementation is resource-efficient, enabling high loop rates and low latency,

TABLE II: Comparison of Different Approaches on NN-Enabled MPC Computation Time

Approach	**Hardware**	**Prediction Horizon**	**Computation Delay**
Ours	Zynq7020	50	19×5ns
MPC-NPI [12]	dSpace DS1202	1	N/A
DualReLU [5]	DSP28335	30	4.3μs
MPC-RL [13]	Altera Cyclone IV-E	15	3×250ns

which are crucial for exploiting the potential of wide bandgap (WBG) devices like GaN in power converters. Experimental validation on a 500W, 5 MHz GaN-based Boost converter prototype demonstrates that the proposed MLP-based controller not only closely mimics the performance of an Explicit MPC (EMPC) controller but also provides superior transient response and steady-state accuracy compared to traditional PI controllers. This solution paves the way for more efficient and robust control strategies in high-performance power electronics systems.

This proof-of-concept work demonstrates the feasibility of integrating machine learning (ML) and heterogeneous computing into power electronics control for high-frequency switching power converters. However, there are several avenues for future research and industrial development to fully mature this technology:

- Enhancing Robustness: A more robust control scheme should be developed to handle uncertainties in parameter estimation, especially for high-switching frequency applications where precise control relies on accurate and robust model parameter estimation.
- Exploration of Model-Free Techniques: The use of model-free RL controllers should be investigated as a potential alternative, offering the ability to adapt dynamically without requiring an accurate system model.
- Hardware Co-Design Optimization: Further optimization of the hardware-software co-design could enable even faster computation and open up opportunities for integrating additional functionalities such as fault diagnosis and predictive maintenance.

References

[1] D. E. Quevedo, R. P. Aguilera, and T. Geyer, "Model predictive control for power electronics applications," *Handbook of Model Predictive Control*, pp. 551–580, 2019.

[2] B. Stellato, T. Geyer, and P. J. Goulart, "High-Speed Finite Control Set Model Predictive Control for Power Electronics," *IEEE Transactions on Power Electronics*, vol. 32, no. 5, pp. 4007–4020, May 2017. DOI: 10.1109/TPEL.2016.2584678. (visited on 01/31/2024).

[3] J. Chen, Y. Chen, L. Tong, L. Peng, and Y. Kang, "A Backpropagation Neural Network-Based Explicit Model Predictive Control for DC–DC Converters With High Switching Frequency," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 8, no. 3, 2020.

[4] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. 2017. (visited on 11/13/2024).

[5] Y. Xiang, H. S.-H. Chung, and H. Lin, "On the Use of DualReLU ANN for Approximating Explicit Model Predictive Control for Buck Converters," in *2024 IEEE Applied Power Electronics Conference and Exposition (APEC)*, Feb. 2024, pp. 2822–2827. DOI: 10.1109/APEC48139.2024.10509403. (visited on 07/19/2024).

[6] J. M. Manzano, D. Limon, D. Muñoz de la Peña, and J.-P. Calliess, "Robust learning-based MPC for nonlinear constrained systems," *Automatica*, vol. 117, p. 108 948, Jul. 2020. DOI: 10.1016/j.automatica.2020.108948. (visited on 11/13/2024).

[7] J. Duarte, P. Harris, S. Hauck, *et al.*, "FPGA-Accelerated Machine Learning Inference as a Service for Particle Physics Computing," *Computing and Software for Big Science*, vol. 3, no. 1, p. 13, Oct. 2019. DOI: 10.1007/s41781-019-0027-2. (visited on 10/23/2024).

[8] S. Sinha and T. Srikanthan, "High-Level Synthesis: Boosting Designer Productivity and Reducing Time to Market," *IEEE Potentials*, vol. 34, no. 4, pp. 31–35, Jul. 2015. DOI: 10.1109/MPOT.2013.2292957. (visited on 10/31/2024).

[9] M. Levental, A. Khan, R. Chard, K. Yoshii, K. Chard, and I. Foster, *OpenHLS: High-Level Synthesis for Low-Latency Deep Neural Networks for Experimental Science*, Mar. 2023. (visited on 07/03/2024).

[10] J. Liu, T. Wei, N. Chen, W. Liu, J. Wu, and P. Xiao, "Data-Mining-Based Hardware-Efficient Neural Network Controller for DC–DC Switching Converters," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 4, pp. 4222–4232, Aug. 2023. DOI: 10.1109/JESTPE.2022.3192354. (visited on 10/29/2024).

[11] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European control conference (ECC)*, 2013, pp. 502–510. (visited on 08/09/2024).

[12] Y. Li, S. Sahoo, T. Dragičević, Y. Zhang, and F. Blaabjerg, "Stability-Oriented Design of Model Predictive Control for DC/DC Boost Converter," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 1, pp. 922–932, Jan. 2024. DOI: 10.1109/TIE.2023.3247785. (visited on 08/09/2024).

[13] D. Alfred, D. Czarkowski, and J. Teng, *Reinforcement Learning-Based Control of a Power Electronic Converter*, Jan. 2024. DOI: 10.3390/math12050671. (visited on 05/27/2024).