Topology Derivation of Multiport DC–DC Converters Based on Reinforcement Learning

Mi Dong, *Member, IEEE*, Ruijin Liang, Jian Yang, *Member, IEEE*, Chenyao Xu, Dongran Song, and Jianghu Wan

***Abstract*—Multiport dc–dc convertersare attracting wide attention in various applications. However, conventional topology derivation methods for multiport dc–dc converters are usually intricate and time-consuming. In this article, a reinforcement learning (RL)-based topology derivation method is proposed, which can derive topologies of complex converters quickly. To apply the RL framework, the topology derivation process is regarded as a Markov decision process. In each step, the agent selects and connects two components until a complete topology is made. To ensure that the derived topologies are feasible and match given voltage specifications, basic circuit constraints and duty cycle constraints are added as hard constraints. Soft constraints are also added to obtain optimal circuits with low voltage stresses or low current stresses. The testing on topology derivation of multiport dc–dc converters shows the great speed of the proposed method. Simulation and experimental results verify that the derived topologies cannot only satisfy given voltage specifications, but also achieve optimization targets of low voltage stresses or current stresses.**

***Index Terms*—Multiport dc–dc converters, reinforcement learning (RL), topology derivation.**

I. Introduction

WITH the development of renewable energy, multiport dc–dc converters are becoming popular in many applications. Among the various design specifications, the number of ports is fundamental because it greatly determines the applications. For example, three port and four port converters are frequently used in photovoltaics power systems [1], [2], [3], while distributed power systems [4], [5], household power systems [6], [7], or microgrids [8], [9] can require converters with more than five ports. Conventional multiport dc–dc converters are constructed by employing multiple single-input and single-output dc–dc converters [10], [11]. Due to their simple realization, converters with any number of ports can be constructed easily. However, these converters suffer from large volume and high cost.

Aiming to address these issues, integrated multiport dc–dc converters are adopted in many applications. A big challenge when using integrated multiport dc–dc converters is that the construction of this kind of converters is complex, and its complexity is increasing with more ports. So far, many schemes have been proposed to derive topologies of integrated multiport dc–dc converters, such as combination methods [12], [13], duality methods [14], [15], and cell replacement methods [16], [17], [18]. By using these methods, different kinds of topologies of integrated multiport dc–dc converters with several ports are derived. However, most of these methods require manual design and analysis, which are intricate and time consuming, making it difficult to derive converters with more ports.

To derive topologies systematically and automatically, more advanced topology derivation schemes are proposed. In [19], a programmable topology derivation method was proposed, which greatly improved the speed of topology derivation. In [20], the programmable method was combined with graph theory, which further improved the generalizability. However, when the number of ports is large, these methods are computationally unacceptable due to the huge search spaces.

In recent years, reinforcement learning (RL) has been proven promising in solving complex problems that have large search spaces such as Go games [21], chemical molecule synthesis [22], and so on. In [23], an RL-based scheme was proposed to accelerate the floorplanning process of chip design, which perfectly demonstrates the potential of RL in circuit design. In [24], RL was used in topology deduction of nonisolated converters, but mainly focusing on three-port converters.

To further explore the potential of RL in power electronics design, this article focuses on the topology derivation of integrated multiport dc–dc converters, which also has the feature of large search spaces, especially when the number of ports is large. The major contributions of this article are summarized as follows.

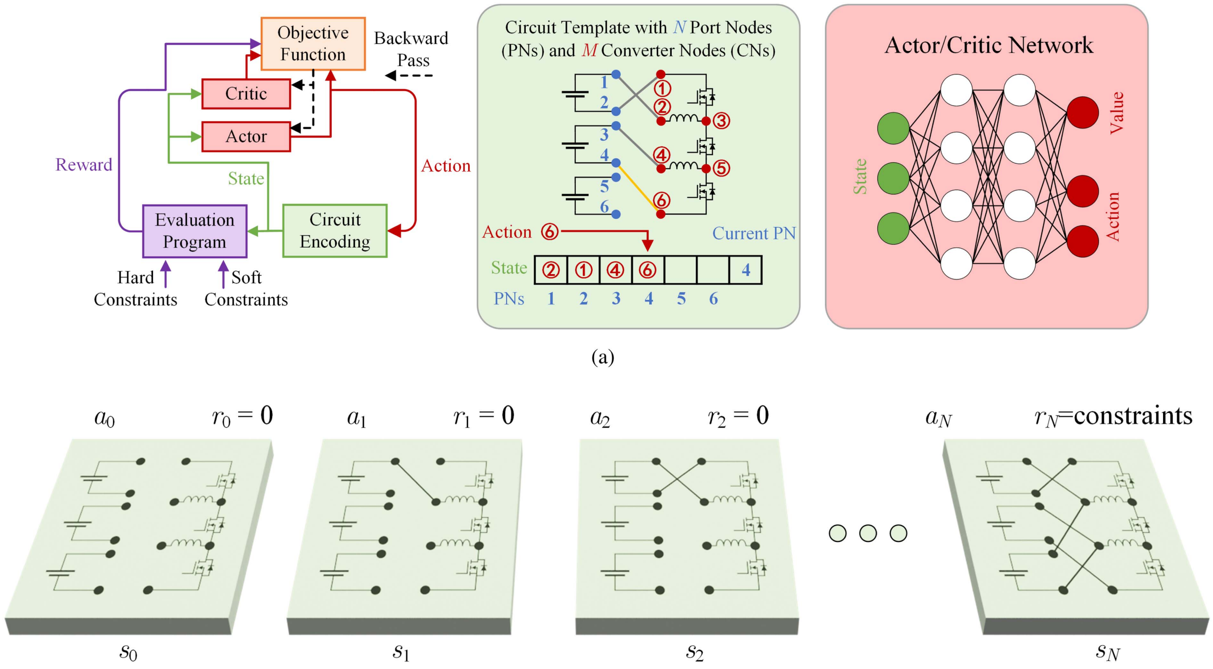
1) The robust and efficient deep reinforcement learning algorithm, proximal policy optimization (PPO), is utilized to train the agent for topology derivation of integrated multiport dc–dc converters. By using this method, the speed of the topology derivation process is greatly improved.
2) Duty cycle constraints are proposed and added to hard constraints to ensure the proposed method can not only derive viable topologies, but also applicable circuits that satisfy given voltage specifications, which can further help engineers accelerate circuit design.

Manuscript received 28 August 2022; revised 15 November 2022; accepted 30 December 2022. Date of publication 9 January 2023; date of current version 14 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 52177204 and Grant 51777217, and in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ4744. Recommended for publication by Associate Editor C. K. Tse. *(Corresponding author: Jian Yang.)*

The authors are with the School of Automation, Central South University, Changsha 410083, China (e-mail: mi.dong@csu.edu.cn; ruijinliang@outlook.com; jian.yang@csu.edu.cn; jsrgxcy1@163.com; humble_szy@163.com; wanjianghu@csu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2023.3235053.

Digital Object Identifier 10.1109/TPEL.2023.3235053



(a)

(b)

Fig. 1. Overview of the learning architecture. (a) The RL framework. (b) An episode of the RL framework.

3) Soft constraints are also considered to optimize the topology derivation process, obtaining better circuits with lower voltage stresses and current stresses.

The rest of this article is organized as follows. The proposed learning architecture for topology derivation is introduced in Section II. Analyses on how to adapt the proposed method to topology derivation of multiport dc–dc converters are conducted in Section III. In Section IV, topology derivation results are illustrated, and the derived multiport dc–dc converters are verified by simulation and experimental results. Finally, Section VI concludes this article.

II. Learning Architecture

The proposed learning architecture, depicted in Fig. 1, is based on RL. In this framework, the topology derivation process is regarded as a Markov decision process (MDP). The RL agent changes the circuit topology according to the circuit state, and obtains the reward given by an evaluation program [Fig. 1(a)]. The rewards stay 0 for all steps, except the last step where the reward is composed of hard constraints and soft constraints of the circuit [Fig. 1(b)]. By learning from experiences, the RL agent can maximize the expected cumulative reward, giving favorable topologies of power electronics converters.

The circuit encoding approach depends on the topology derivation scheme. In our architecture, a simple combination method [19] is utilized for demonstration. In this method, the circuit topology varies by changing the connections between N port nodes and M converter nodes.

By the above settings, the action, state, and reward are defined as follows:

1) State: The state is defined as an $N+1$ dimension vector $\boldsymbol{s} = [s_1 \quad s_2 \quad s_3 \quad \cdots \quad s_{N+1}]^{\mathrm{T}}$, where $1 \leq s_i \leq M$ $(1 \leq i \leq N)$ represents the connection between the ith port node and s_ith converter node, and $1 \leq s_{N+1} \leq N$ represents the last processed port node. The state is initialized with a zero vector. When s_{N+1} transits to N, the RL episode is over.

2) Action: The action is defined as an integer scalar a, representing the converter node that the current port node connects to.

3) Reward: The reward is a scalar defined to evaluate the derived topologies. Before an RL episode is over, the reward is always set to be 0, because the topology is incomplete and cannot be evaluated. When the episode is over and a complete topology is derived, an external program checks whether the topology satisfies specific constraints, giving a feasible topology a higher score. The reward is defined as follows:

$$r_t = \begin{cases} 0, & t \neq N \\ d, & t = N \end{cases} \tag{1}$$

where d is the evaluation score, which is composed of hard and soft constraints as follows [25]:

$$d = \begin{cases} c_h + \alpha * c_s + e_0, & \text{if } n_h < N_h \\ c_s + e_1, & \text{otherwise} \end{cases} \tag{2}$$

where c_h and c_s are the score of hard constraints and soft constraints, respectively, n_h and N_h are the number of satisfied hard constraints and the number of total hard constraints, respectively, and α, e_0, and e_1 are hyperparameters.

In the learning process, PPO [26], an actor–critic-based and policy-based RL algorithm, is used to train the RL agent. In policy-based methods, the policy of the agent is explicitly given

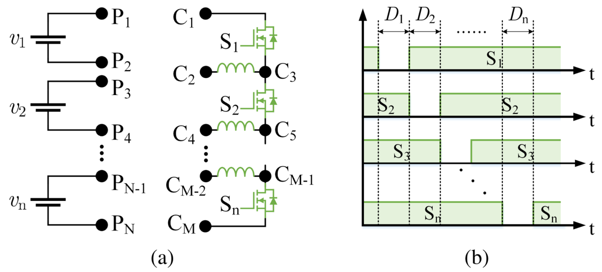


Fig. 2. Circuit configuration. (a) The n-port circuit with N port nodes and M converter nodes. (b) Drive signals.

by a function $\pi(a|s)$. The goal of PPO is to maximize the optimization target J defined as

$$J = \mathbb{E}[J^{\text{act}} + c_1\, J^{\text{cri}}] \tag{3}$$

$$J^{\text{act}} = \mathbb{E}[\min(r_\pi \hat{A}_t, \text{clip}(r_\pi, 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{4}$$

$$J^{\text{cri}} = \mathbb{E}[-(V(s_t) - G_t)^2] \tag{5}$$

where c_1 and ϵ are hyperparameters, $\hat{A}_t$ is the estimated advantage function defined as

$$\hat{A}_t = \sum_{i=0}^{\infty} (\gamma\lambda)^i [r_t + \gamma V_{\text{old}}(s_{t+i+1}) - V_{\text{old}}(s_{t+i})] \tag{6}$$

with hyperparameters γ and λ, r_π is the importance sampling ratio defined as $r_\pi = \frac{\pi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$, and G_t is the accumulated discounted return defined as $G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$.

In practice, the policy $\pi(a|s)$ and the critic $V(s)$ are estimated by neural networks $\pi_\theta(a|s)$ and $V_\theta(s)$, respectively. For simplicity, these networks are both multilayer perception networks with rectified linear units as the activation function of each hidden layers [Fig. 1(a)].

The RL-based topology derivation algorithm is as shown in Algorithm 1. In every epoch, the algorithm can be divided into two stages. In the first stage, the algorithm samples a trajectory by the current policy, and computes corresponding $\hat{A}_t$ and G_t. In the second stage, the algorithm computes the optimization target, and updates the parameters of the networks by ADAM optimizer [27].

III. Topology Derivation of Multiport dc–dc Converters

A. Configuration of Multiport dc–dc Converters

The proposed method is flexible and can derive different kinds of power electronics converters, including dc–ac inverters, dc–dc converters, and so on. In this article, the n-port dc–dc converter with N port nodes and M converter nodes is taken as the example to clarify the proposed topology derivation method. The circuit is as illustrated in Fig. 2(a).

To achieve independent control of each voltage port, the drive signals as illustrated in Fig. 2(b) are employed. By adjusting D_1–D_n, voltages among each port can be regulated.

Algorithm 1: RL-Based Topology Derivation Algorithm.

```
Input: Number of epochs $N_{\text{epo}}$, number of episodes sampled per epoch $N_{\text{epi}}$, number of network updates per epoch $N_{\text{upd}}$, and other hyperparameters of the model and the ADAM optimizer
Output: Feasible circuit set $\mathcal{C}$
Initialize the feasible circuit set $\mathcal{C}$ as a empty set
for $i = 1$ to $N_{\text{epo}}$ do
    Initialize $\mathcal{T}$, $\mathcal{A}$, $\mathcal{G}$ and $\mathcal{P}$ as empty lists
    for $j = 1$ to $N_{\text{epi}}$ do
        Sample a trajectory $\tau = (s_1, a_1, r_1, \ldots, s_N, a_N, r_N)$ by $\pi_\theta$, and for every step $t$ add $\pi_\theta(a_t|s_t)$ to $\mathcal{P}$
        Add the trajectory $\tau$ to $\mathcal{T}$
        Compute $\hat{A}_t$ for every step $t$ and add them to $\mathcal{A}$
        Compute $G_t$ for every step $t$ and add them to $\mathcal{G}$
        if $r_N > 0$ then
            Add the feasible circuit $s_N$ to $\mathcal{C}$
        end if
    end for
    for $j = 1$ to $N_{\text{upd}}$ do
        Compute the optimization target for the actor:
        $J^{\text{act}} = \mathbb{E}_{s_t \sim \mathcal{T}, \hat{A}_t \sim \mathcal{A}, \pi_{\text{old}}(a_t|s_t) \sim \mathcal{P}}[\min(r_\pi \hat{A}_t, \text{clip}(r_\pi, 1-\epsilon, 1+\epsilon)\hat{A}_t)]$
        Compute the optimization target for the critic:
        $J^{\text{cri}} = \mathbb{E}_{s_t \sim \mathcal{T}, G_t \sim \mathcal{G}}[-(V_\theta(s_t) - G_t)^2]$
        Compute the optimization target:
        $J = \mathbb{E}[J^{\text{act}} + c_1\, J^{\text{cri}}]$
        Update the parameters $\theta$ of the actor and critic networks with $\nabla_\theta J$ by ADAM
    end for
end for
```

B. Circuit Constraints

1) Hard Constraints for Feasible Topologies: To guarantee the derived circuits are feasible, some basic constraints must be satisfied, which are usually related to certain criteria. In [19], two fundamental criteria were proposed. The first was that average voltages of each port must be greater than zero, and they should be independently controlled. The second was that at any switching state, all the ports and their combinations could not be short connected or parallel connected. According to the above criteria, the basic circuit constraints can be formulated as

$$v_{i\_\text{avg}} \neq \sum_{j=1}^{|d|} v_{d_j\_\text{avg}}, \quad i = 1, 2, \ldots, n; \forall d \subseteq D_i \tag{7}$$

$$\sum_{i=1}^{|u|} v_{u_i\_\text{sw}} \neq 0, \quad \forall u \subseteq U \tag{8}$$

$$v_{i\_\text{sw}} \neq \sum_{j=1}^{|d|} v_{d_j\_\text{sw}}, \quad i = 1, 2, \ldots, n; \forall d \subseteq D_i \tag{9}$$

where $U = \{1, 2, \ldots, n\}$, $D_i = \{1, 2, \ldots, n\} \backslash \{i\}$, $v_{i\_\mathrm{avg}} > 0$ is the average port voltage and v_{i_sw} is the port voltage in every switching stage.

At the beginning of the learning process, the circuit is randomly initialized, and most of the constraints are not satisfied. The agent must learn to satisfy more constraints in the learning process, until all the constraints are satisfied, where feasible circuits are derived. To evaluate how many constraints remain to be satisfied, the score of the basic constraints c_b is defined as

$$c_b = -\frac{1}{N_b}\mathrm{count\_zero}(\boldsymbol{b}) \tag{10}$$

where $\boldsymbol{b}$ is an auxiliary vector whose zero components indicate the corresponding constraints are not satisfied, N_b is the number of components of $\boldsymbol{b}$, and $\mathrm{count\_zero}(\boldsymbol{b})$ represents the number of zero components of $\boldsymbol{b}$. To calculate the auxiliary vector, the constraints are reformulated in a matrix form

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{v} \tag{11}$$

where $\boldsymbol{A}$ is the coefficient matrix of the constraints, and $\boldsymbol{v}$ is a vector representing the constrained variables consisting of average port voltages and switching interval port voltages.

The matrix $\boldsymbol{A}$ is only related to the number of ports. In the learning process, the number of ports is fixed, so the matrix $\boldsymbol{A}$ is constant. Taking the reformulation of constraints of two-port converters as an example, according to (7)–(9), constraints of two-port converters are as follows:

$$\begin{cases} v_{1\_\mathrm{avg}} \neq v_{2\_\mathrm{avg}} \\ v_{1\_\mathrm{sw}} \neq 0 \\ v_{2\_\mathrm{sw}} \neq 0 \\ v_{1\_\mathrm{sw}} + v_{2\_\mathrm{sw}} \neq 0 \\ v_{1\_\mathrm{sw}} \neq v_{2\_\mathrm{sw.}} \end{cases} \tag{12}$$

And the auxiliary vector $\boldsymbol{b}$ is calculated by

$$\boldsymbol{b} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} v_{1\_\mathrm{avg}} \\ v_{2\_\mathrm{avg}} \\ v_{1\_\mathrm{sw}} \\ v_{2\_\mathrm{sw}} \end{pmatrix} \tag{13}$$

The vector $\boldsymbol{v}$ is related to the voltages of converter nodes C_1–C_M, which can be represented by

$$\boldsymbol{v} = \begin{bmatrix} v_{1\_\mathrm{avg}} \\ \vdots \\ v_{N\_\mathrm{avg}} \\ v_{1\_\mathrm{sw}} \\ \vdots \\ v_{N\_\mathrm{sw}} \end{bmatrix} = f_{\boldsymbol{s}}\left(\begin{bmatrix} v_{\mathrm{C}_1,\mathrm{avg}} \\ \vdots \\ v_{\mathrm{C}_M,\mathrm{avg}} \\ v_{\mathrm{C}_1,\mathrm{sw}} \\ \vdots \\ v_{\mathrm{C}_M,\mathrm{sw}} \end{bmatrix} \right) \tag{14}$$

where $f_{\boldsymbol{s}}(\cdot)$ represents the connection between port nodes and converter nodes, which is determined by the current state $\boldsymbol{s}$. In the learning process, converter node voltages should be assigned to specific test voltages. The values of these voltages can be chosen arbitrarily, as long as they do not violate some basic principles. In [19], two principles were discussed: The first is that the average voltages across inductors must be zero. And the second is that the average voltages across switches must be larger than zero. In implementation, we find another principle: The converter node voltages should be independent. This is natural because the port voltages are independent, which are linear combinations of converter node voltages. Taking the relationship among three converter nodes as an example, their voltages must satisfy the following inequality:

$$v_{\mathrm{C}_i} \neq v_{\mathrm{C}_j} + v_{\mathrm{C}_k}, \quad \forall i \neq j \neq k. \tag{15}$$

To satisfy this condition, a simple way is to assign the logarithm of distinct prime numbers to them, i.e., let $v_{\mathrm{C}_i} = \ln z$, $v_{\mathrm{C}_j} = \ln p$, $v_{\mathrm{C}_k} = \ln q$, where z, p, and q are distinct prime numbers. It is easy to see that this also obey the principle for more than three converter nodes.

2) Hard Constraints for Given Voltage Specifications: The basic constraints only guarantee the derived circuits are feasible. In real applications, the voltages of ports are usually given to be certain values. To check whether a circuit can operate at the given port voltages, a straightforward way is to calculate the port voltage range and compare it with the given voltages. However, this computation is intricate, especially when the circuit is complex. An alternative way is to assume the circuit can operate at the given port voltages and calculate the corresponding duty cycles of each switch. If the circuit can operate at the given voltages, there must exist solutions for duty cycles satisfying

$$0 < D_{si} < 1, \; i = 1, 2, \ldots, n \tag{16}$$

where D_{si} is the duty cycle of the ith switch. When the duty cycles are out of range or the solutions of them do not exist, a penalty term should be added to the score. In this work, the score of the duty cycle constraint is defined as

$$c_d = \begin{cases} L_D, & \text{solutions of } D_{s1}\text{–}D_{sn} \text{ exist} \\ -1, & \text{otherwise} \end{cases} \tag{17}$$

$$L_D = -\frac{1}{n}\sum_{i=1}^{n} l_D(D_{si}) \tag{18}$$

$$l_D(d) = \max(\tanh(|d - 0.5| - 0.5), 0). \tag{19}$$

When the solutions do not exist, the score is set to be –1 directly. When the solutions exist, the score depends on the values of the duty cycles. If the duty cycles are within the range, the score is set to be 0, meaning there is no penalty. Otherwise, the more the duty cycles are out of range, the larger the penalty term is. The hyperbolic tangent function $\tanh(\cdot)$ ensures the score will not less than –1, which also ensures the penalty is the largest when the solutions do not exist.

The duty cycle of each switch can be calculated by the volt-second balance principle, represented by the given port voltages. To use the volt-second balance principle, for each inductor, two different loops containing only one inductor, the inductor itself, must be found. This can be implemented by using the deep-first search on the circuit. Thus, the duty cycles can be calculated by

$$\begin{cases} f_{L_1}(D_{s1}, D_{s2}, \ldots, D_{sn}) = 0 \\ f_{L_2}(D_{s1}, D_{s2}, \ldots, D_{sn}) = 0 \\ \qquad \cdots \\ f_{L_{n-1}}(D_{s1}, D_{s2}, \ldots, D_{sn}) = 0 \\ D_{s1} + D_{s2} + \cdots + D_{sn} = n - 1 \end{cases} \tag{20}$$

where the first $n - 1$ equations are the volt-second balance equations for each inductor, and the last equation is from the drive signals. The n equations are all linear equations with respect to D_{s1}–D_{sn}, so it is relatively easy to solve these equations.

By combining the basic constraints and the duty cycle constraint, the score of hard constraints can be constructed as

$$c_h = w_b * c_b + w_d * c_d \tag{21}$$

where w_b and w_d are weighting factor of the basic constraints and the duty cycle constraint, respectively.

3) Soft Constraints: In real applications, lower voltage stresses and current stresses are preferred due to the higher expenses of switches and lower efficiency under high stresses.

Using the drive signal as illustrated in Fig. 2(b), in every stage, only one switch is OFF, while the other switches are all ON. Thus, the voltage stress of each switch is equal to the voltage across the first converter node and the last converter node, which can be expressed by the voltages of ports as

$$v_{ss} = \sum_j v_{\text{loop},j} \tag{22}$$

where $v_{\text{loop},j}$ is the voltage of the jth port in the loop containing the first converter node and the last converter node.

The current stress of a switch is usually represented by the root-mean-square (RMS) value of the drain-to-source current with the neglect of ripple current of inductors as

$$i_{ss,j} = \sqrt{\sum_k (1 - D_{sk}) \left(\sum_m i_{Lmk} \right)^2} \tag{23}$$

where $i_{ss,j}$ is the current stress of the jth switch, and i_{Lmk} is the average current of the mth inductor which flows through the switch in the kth stage.

To reduce the voltage stresses and current stresses, the following soft constraint is added:

$$c_s = -w_{sv} * v_{ss} - w_{sc} * i_{ss\max} \tag{24}$$

where w_{sv} and w_{sc} are the weighting factors, $i_{ss\max}$ is the maximum current stress among all switches.

IV. Topology Derivation Results

A. Speed Comparison With Conventional Method

To verify the speed of the proposed method, the agent is trained for topology derivation of converters with 4, 5, 6, 7, and 8 ports, respectively, with only basic hard constraints. The training program runs on a computer with AMD Ryzen 7 2700 CPU, Nvidia RTX2080 GPU, and 16 GB of RAM. The configurations of the parameters of the RL algorithm are as listed in Table I.

TABLE I
Parameters of the RL Algorithm

Parameter	Value	Parameter	Value
Batch Size	64	Learning Rate	0.001
Actor hidden layer size	$[64 \quad 64]$	Discount Factor	0.99
Critic hidden layer size	$[64 \quad 64]$		

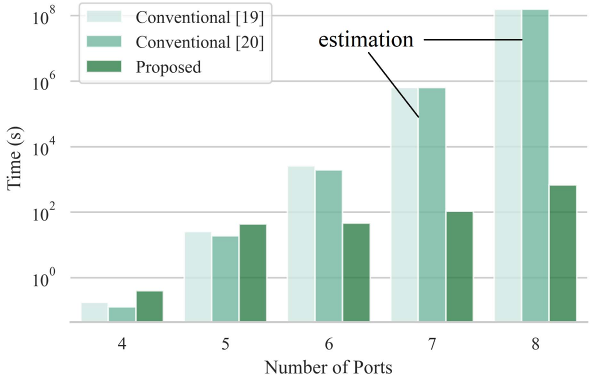


Fig. 3. Time spent on deriving the first 50 feasible topologies.

Due to the complexity of multiport converters, there has not been a quick topology derivation method for a long time. In this article, the proposed method is compared with the conventional programmable topology derivation methods in [19] and [20] by recording the time spent on deriving the first 50 topologies.

Fig. 3 shows the time spent on deriving the first 50 topologies. From the figure, it can be seen that the time consumption of the proposed methods and conventional ones differs in many aspects. When the number of ports is less than 6, all methods take seconds to derive certain number of topologies, but the conventional methods are slightly faster than the proposed method. When the number of ports is up to 6, the situation is reversed. The time consumption of the conventional methods quickly grows to more than 40 min for six-port converters, while the proposed method takes only seconds to derive the same number of them. When the number of ports is larger, the difference is even more significant. Fig. 4 shows some eight-port converter topologies derived by the proposed method. It takes only 700 s to derive the first 50 eight-port converters by the proposed method, while no seven-port or eight-port converter is found by the conventional ones even after 10^4 s.

To understand the differences in time consumption of these methods, the search strategies must be analyzed first. For an n-port converter with $2n$ port nodes and $2n$ converter nodes, there are $(2n)^{2n}$ combinations in total. The proposed method and conventional ones use different strategies to search in the large space.

The conventional method in [19] needed to iterate over every possible combination in the search space. To reduce the number of iterations, several strategies are used. First, circuits with two nodes of a voltage port connected to the same converter port are skipped. Second, circuits with port voltages less than 0 are skipped. By using these strategies, the number of iterations

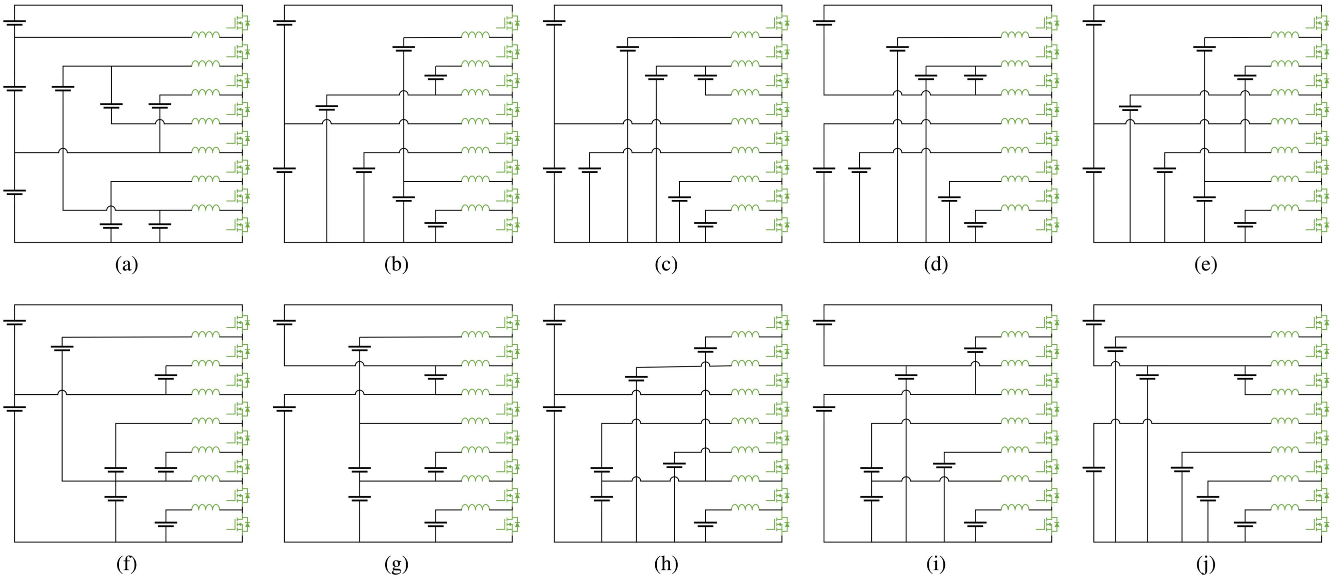


Fig. 4. First 10 of the derived topologies of eight-port dc–dc converters.

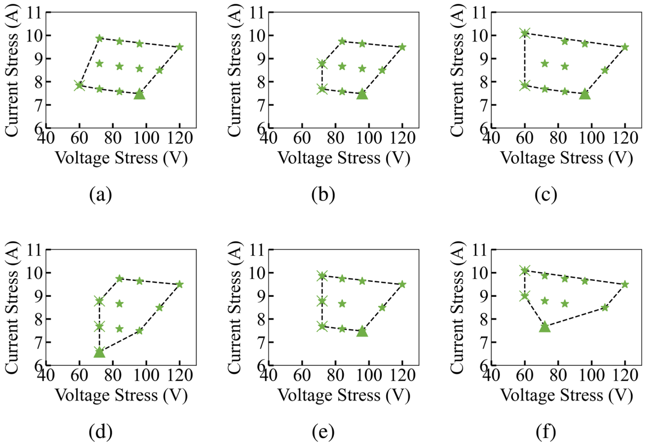


Fig. 5. Voltage stresses and current stresses of the designed four-port converters derived by different w_{sv} and w_{sc}. The stars denote the derived topologies, the crosses denote the circuits with the minimal voltage stresses, and the triangles denote the circuits with the minimal current stresses. (a) $w_{sv} = 0.5, w_{sc} = 0.5$. (b) $w_{sv} = 0.7, w_{sc} = 0.3$. (c) $w_{sv} = 1.0, w_{sc} = 0.0$. (d) $w_{sv} = 0.3$, $w_{sc} = 0.7$. (e) $w_{sv} = 0.0, w_{sc} = 1.0$. (f) $w_{sv} = 0.0, w_{sc} = 0.0$.

becomes $(2n^2 - 2n + 1)^n$. Compared with the whole search space, the number of iterations is greatly reduced. The method in [20] used graph theory to represent the circuits, and simplified the constraints by replacing some circuit constraints with graph constraints, which further improved the speed. However, the time consumption of these methods still grows more than exponentially as the number of ports increases.

On the contrary, the RL-based method does not need to iterate over every possible combination. Instead, it learns to find an optimal path that maximizes the reward in the search space. From Algorithm 1, it can be seen that the time consumption of the RL-based method is composed of two parts: 1) for collecting experience data; and 2) for computing gradients and updating parameters of the neural networks. The second part is directly related to the size of the neural networks, and is fixed when the neural networks are fixed.

When the number of ports is small, the size of the search space is also small, and it takes more time to update neural networks than directly iterate over the search space. This explains why the RL-based method is slower than the conventional method when the number of ports is no more than 5. To accelerate the RL algorithm in this situation, a simple way is to increase the number of sampling and decrease the number of network updates. In practice, this tuning is unnecessary because even without any tuning, it only takes seconds to finish computation when the number of ports is small.

When the number of ports is larger than 5, it becomes computationally unacceptable to directly iterate over the large search space, while the RL-based method can still find shorter paths to maximize the reward, which gives feasible topologies in much shorter time.

B. Topology Derivation With Design Specifications

The proposed method can derive not only feasible topologies, but also applicable topologies that satisfy certain specifications, mainly voltages of each port. To verify this ability, four-port converters with $v_{\text{in}} = 12$ V, $v_{\text{out1}} = 24$ V, $v_{\text{out2}} = 36$ V, $v_{\text{out3}} = 48$ V, and $i_{\text{out1}} = i_{\text{out2}} = i_{\text{out3}} = 1$ A are taken as examples to test the proposed method with design specifications.

Because the soft constraints may influence the topology derivation results, special tests on four-port converters with different w_{sv} and w_{sc} are first carried out. Fig. 5 shows the topology derivation results of four-port converters with different settings of soft constraints. From these results, several discoveries can be concluded. The soft constraints can influence the positions of the boundary points in Fig. 5, indicating the soft constraints can optimize the voltage stresses and current stresses of the

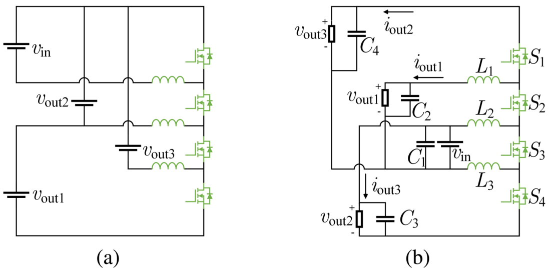


Fig. 6. Derived four-port converters with low stresses. (a) Low voltage stresses. (b) Low current stresses.

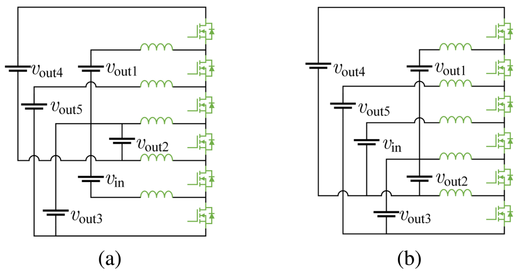


Fig. 7. Derived six-port converters with low stresses. (a) Low voltage stresses. (b) Low current stresses.

newly derived topologies as the learning process goes. The voltage stresses and current stresses seem to be couple with each other, but it is still possible to reduce voltage stresses (current stresses) by increasing w_{sv} (w_{sc}) as shown in Fig. 5(c) [Fig. 5(d)]. Based on these discoveries, four-port converters with low voltage stresses ($v_{s\text{smax}} = 60$ V) or low current stresses ($i_{s\text{smax}} = 6.58$ A) are derived, which are as shown in Fig. 6.

Using the same method, six-port converters with low voltage stresses ($v_{s\text{smax}} = 30$ V) or low current stresses ($i_{s\text{smax}} = 0.76$ A) are derived, which are as shown in Fig. 7.

C. Operational Principle

To illustrate the operational principle of the derived converters, the four-port converter in Fig. 6(a) is taken as an example. For this converter, there are four stages in total. In each stage, only one switch is OFF, and the other switches are ON. Fig. 8 shows the voltages across inductors in each stage. The inductor voltages vary among stages. For inductor L_1, the voltage across it is related to v_{in}, v_{out1}, and v_{out2} in stage 1, but only related to v_{in} in stages 2, 3, and 4. The voltages of other inductors in each stage can be analyzed in the same way. By using the volt-second balance principle, the relationship between port voltages and duty cycles can be formulated as

$$-v_{\text{in}} D_{s1} = (1 - D_{s1})(-v_{\text{out1}} - v_{\text{out2}} + v_{\text{in}}) \tag{25}$$

$$v_{\text{out1}}(2 - D_{s1} - D_{s2}) = (2 - D_{s3} - D_{s4}) v_{\text{out2}} \tag{26}$$

$$(-v_{\text{out3}} + v_{\text{out2}} + v_{\text{out1}}) D_{s4} = (1 - D_{s4}) v_{\text{out3}} \tag{27}$$

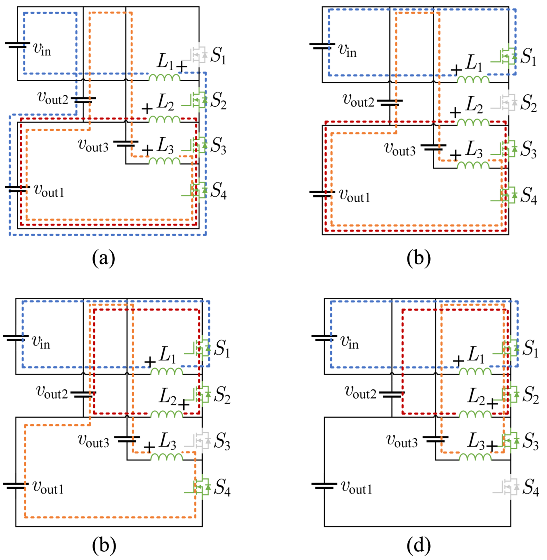


Fig. 8. Voltages across inductors in each stage. (a) Stage 1. (b) Stage 2. (c) Stage 3. (d) Stage 4.

which can further derive the relationship between voltage gain and duty cycles

$$\frac{v_{\text{out1}}}{v_{\text{in}}} = -\frac{D_{s1} + D_{s2} - 1}{D_{s1} - 1} \tag{28}$$

$$\frac{v_{\text{out2}}}{v_{\text{in}}} = \frac{D_{s1} + D_{s2} - 2}{D_{s1} - 1} \tag{29}$$

$$\frac{v_{\text{out3}}}{v_{\text{in}}} = \frac{D_{s1} + D_{s2} + D_{s3} - 3}{D_{s1} - 1}. \tag{30}$$

Using the same way, the voltage relationship of Fig. 6(b) can be derived

$$\frac{v_{\text{out1}}}{v_{\text{in}}} = \frac{D_{s2} + D_{s3} - 2}{D_{s3} - 1} \tag{31}$$

$$\frac{v_{\text{out2}}}{v_{\text{in}}} = -\frac{D_{s1} + D_{s2} - 1}{D_{s3} - 1} \tag{32}$$

$$\frac{v_{\text{out3}}}{v_{\text{in}}} = \frac{D_{s1} + D_{s2} + D_{s3} - 3}{D_{s3} - 1}. \tag{33}$$

From (28)–(33), it can be seen that adjusting one duty cycle affects multiple outputs. To independently control each output port, decoupling controllers can be adopted, which are discussed in Section V.

V. Verification

A. Experimental Verification on Design Specifications

To verify whether the derived converters satisfy the design specifications, the derived four-port converter with low current stresses [Fig. 6(b)] is taken as an example to carry out experiments. The experimental settings are as listed in Table II, and the experimental circuit is as illustrated in Fig. 9. The inductor values and capacity values are chosen to ensure the circuit operates under the continuous current mode according to the drive signals.

TABLE II
PARAMETERS OF THE EXPERIMENTAL CIRCUITS

Parameter	Value	Parameter	Value
L_1–L_3	100µH	S_1–S_4	IPP039N10N5
C_1–C_4	100µF	Control unit	TMS320FDSP28335
Switching period T_s	20µs		

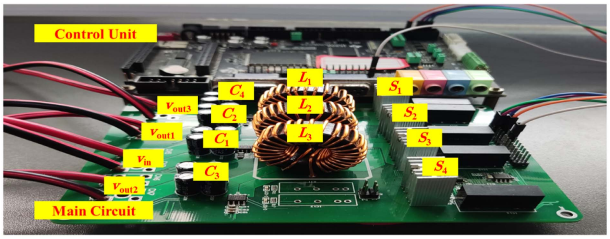
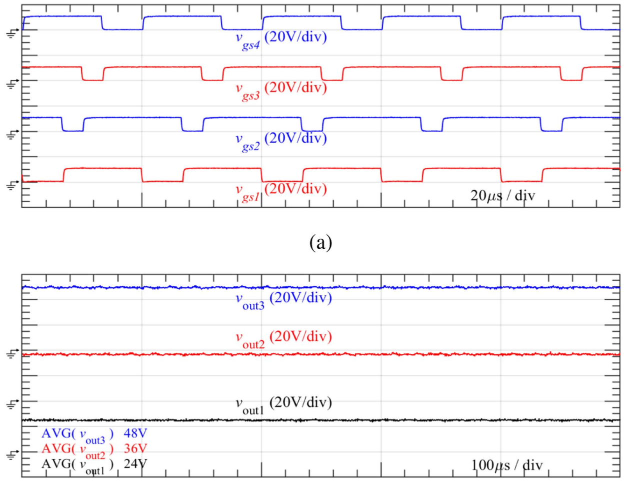


Fig. 9. Experimental four-port converter.



(a)

(b)

Fig. 10. Experimental verification on design specifications. (a) Drive signals. (b) Output voltages.

Fig. 10 shows the experimental waveforms of the circuit. The drive signals are as illustrated in Fig. 10(a). To get the given port voltages, the duty cycles of S_1–S_4 are set to be 0.67, 0.83, 0.83, and 0.67, respectively, which can be calculated by (28)–(30). To ensure the inductor currents are continuous, only one switch can be turned OFF at any moment. And to avoid short circuits, all of the switches cannot be turned ON simultaneously.

Under the drive signals, the output voltages are as illustrated in Fig. 10(b). The average voltages of the three output ports are 24 V, 36 V, and 48 V, respectively, which are consistent with the given port voltages. The current stresses (RMS) of S_1–S_4 are 3.81 A, 3.96 A, 6.73 A, and 2.98 A, respectively, the maximum of which is 6.73 A. The $i_{s\,s\max}$ is slightly larger than the theoretical analysis due to the ignorance of current ripples in computation.

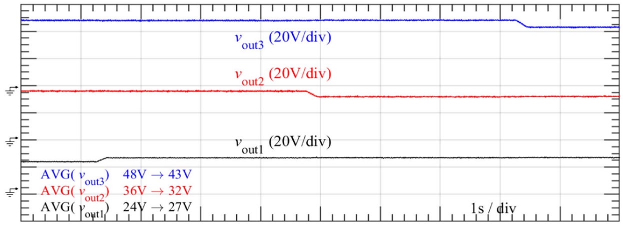


Fig. 11. Experimental verification on decoupling control of a derived four-port converter.

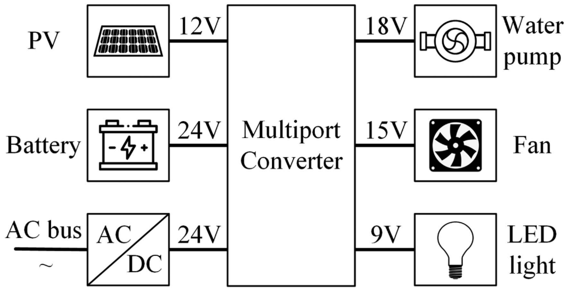


Fig. 12. Household dc power system.

B. Output Voltage Decoupling

Due to the complexity of the structure, each output voltage of the derived converters may depend on multiple drive signals. To independently control each output voltage, decoupling controller can be employed. A simple control strategy is presented to demonstrate the viability of the decoupling control of each output port. Taking the converter in Fig. 6(b) as an example, the desired output voltages are denoted as v^*_{out1}, v^*_{out2}, and v^*_{out3}. Consider a simple control strategy

$$D_{s1} = av^*_{\text{out1}} \tag{34}$$

$$D_{s2} = bv^*_{\text{out2}} \tag{35}$$

$$D_{s3} = cv^*_{\text{out3}}. \tag{36}$$

The object is to let the actual output follow the desired output

$$v_{\text{out1}} = \frac{D_{s2}+D_{s3}-2}{D_{s3}-1}v_{\text{in}} = v^*_{\text{out1}} \tag{37}$$

$$v_{\text{out2}} = -\frac{D_{s1}+D_{s2}-1}{D_{s3}-1}v_{\text{in}} = v^*_{\text{out2}} \tag{38}$$

$$v_{\text{out3}} = \frac{D_{s1}+D_{s2}+D_{s3}-3}{D_{s3}-1}v_{\text{in}} = v^*_{\text{out3}} \tag{39}$$

from which a, b, and c can be solved

$$a = \frac{v^*_{\text{out1}}+v^*_{\text{out2}}-v_{\text{in}}}{v^*_{\text{out1}}(v^*_{\text{out2}}+v^*_{\text{out3}}-v_{\text{in}})} \tag{40}$$

$$b = \frac{v^*_{\text{out2}}-v^*_{\text{out1}}+v^*_{\text{out3}}}{v^*_{\text{out2}}(v^*_{\text{out2}}+v^*_{\text{out3}}-v_{\text{in}})} \tag{41}$$

$$c = \frac{v^*_{\text{out2}}+v^*_{\text{out3}}-2v^*_{\text{out1}}}{v^*_{\text{out2}}v^*_{\text{out3}}-v^*_{\text{out3}}v_{\text{in}}+{v^*_{\text{out3}}}^2}. \tag{42}$$

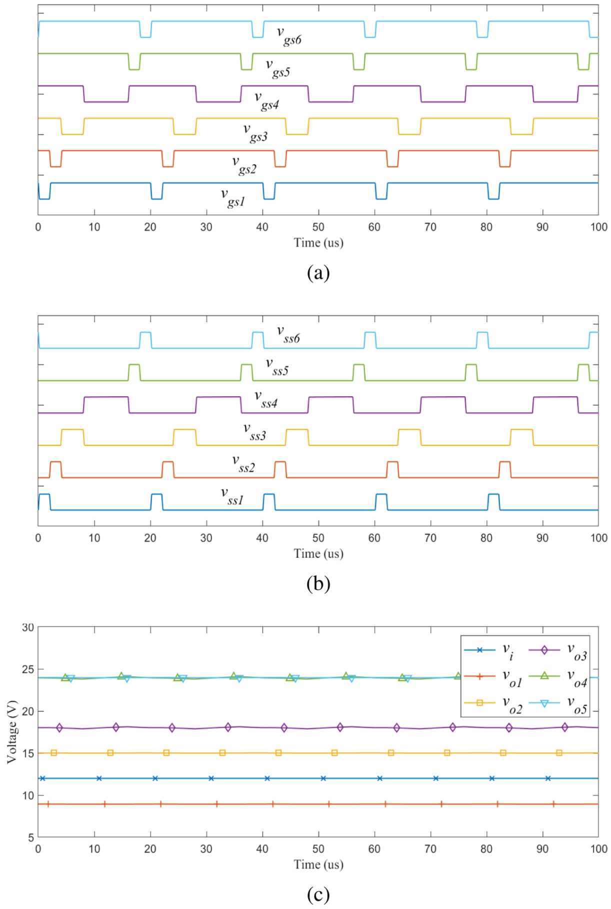


Fig. 13. Simulation results of the six-port converter with low voltage stresses. (a) Drive signals. (b) Voltage stresses. (c) Port voltages.

Fig. 11 shows the experimental verification results on decoupling control of the converter. The desired outputs change at three moments $t_1 = 1.2$ s, $t_2 = 4.8$ s, and $t_3 = 8.2$ s. At t_1, v^*_{out1} changes from 24 to 27 V, along with v^*_{out2} from 36 to 32 V at t_2, and v^*_{out3} from 48 to 43 V at t_3. To illustrate the effects more clearly, a slope is applied for each change. The actual outputs follow the desired outputs. When one output voltage changes, the other output voltages remain unchanged. This result shows that the output voltages can be independently controlled by the desired output voltages.

C. Application on a Household dc Power System

The previous example of four-port converters only demonstrates the effectiveness of the proposed method. The real ability of the proposed method is more than that. In this work, six-port dc–dc converters with various port voltages are designed by the proposed method. This type of converters can be used in many applications such as distributed power systems, household dc power systems, microgrids, and so on. Because the number of ports is relatively high, it is difficult to derive feasible circuits by manual analysis or even conventional programmable methods, not to mention derive applicable circuits that satisfy given voltage specifications. However, by using the proposed method, this type of converters can be easily derived.

The topology derivation testing is performed on a household dc power system [6] with the following specifications: the input voltage is 12 V, and the five output voltages are 9 V, 15 V, 18 V, 24 V, and 24 V, respectively, which are as shown in Fig. 12. By using the proposed method, several applicable converters are derived. Fig. 7 shows two of the derived converters with low voltage stresses or low current stresses. All of the derived converters are feasible and satisfy the given voltage specifications.

Taking the converter with low voltage stresses as an example, the simulation results are as shown in Fig. 13. From the simulation results, it can be seen that the output voltages are consistent with the design specifications, and the voltage stress of each switch is about 30 V, which is only slightly larger than the maximum output voltage. This example shows that the proposed method can help engineers design multiport converters for real applications simply and quickly.

VI. Conclusion

In this article, an RL-based topology derivation method for multiport dc–dc converters is proposed, which can derive not only feasible topologies quickly, but also applicable topologies that satisfy design specifications, such as specific output voltages and low switch stresses. By using this method, many new eight-port dc–dc converters are derived. Besides, four-port and six-port dc–dc converters with certain design specifications are derived by the proposed method. Simulation and experimental results show that the topologies derived by the proposed method satisfy the design requirements well. Because the RL framework is flexible, in the future, the proposed method can be used to derive new topologies of other types of complex power electronics converters, such as multilevel inverters.

References

[1] J. Zeng, W. Qiao, and L. Qu, "An isolated multiport bidirectional DC-DC converter for PV-battery-DC microgrid applications," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2014, pp. 4978–4984.

[2] M. Y. A. Khan, H. Liu, and N. U. Rehman, "Design of a multiport bidirectional DC-DC converter for low power PV applications," in *Proc. IEEE Int. Conf. Emerg. Power Technol.*, 2021, pp. 1–6.

[3] Z. Qian, O. Abdel-Rahman, and I. Batarseh, "An integrated four-port DC/DC converter for renewable energy applications," *IEEE Trans. Power Electron.*, vol. 25, no. 7, pp. 1877–1887, Jul. 2010.

[4] M. Salato and U. Ghisla, "Optimal power electronic architectures for dc distribution in datacenters," in *Proc. IEEE 1st Int. Conf. DC Microgrids*, 2015, pp. 245–250.

[5] P. Karlsson and J. Svensson, "DC bus voltage control for a distributed power system," *IEEE Trans. Power Electron.*, vol. 18, no. 6, pp. 1405–1412, Nov. 2003.

[6] K. Wang, X. Liu, L. Zhao, Y. Zhou, and D. Xu, "Research on structure and energy management strategy of household energy router based on hybrid energy storage," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf.*, 2019, pp. 1–5.

[7] X. Jianfang and W. Peng, "Multiple modes control of household dc microgrid with integration of various renewable energy sources," in *Proc. IECON - 39th Annu. Conf. IEEE Ind. Electron. Soc.*, 2013, pp. 1773–1778.

[8] D. J. Becker and B. Sonnenberg, "DC microgrids in buildings and data centers," in *Proc. IEEE 33rd Int. Telecommun. Energy Conf.*, 2011, pp. 1–7.

[9] Y. Ito, Y. Zhongqing, and H. Akagi, "DC microgrid based distribution power generation system," in *Proc. IEEE 4th Int. Power Electron. Motion Control Conf.*, vol. 3, 2004, pp. 1740–1745.

[10] P. J. d. S. Neto, T. A. d. S. Barros, J. P. C. Silveira, E. R. Filho, J. C. Vasquez, and J. M. Guerrero, "Power management strategy based on virtual inertia for DC microgrids," *IEEE Trans. Power Electron.*, vol. 35, no. 11, pp. 12472–12485, Nov. 2020.

[11] O. Cornea, G.-D. Andreescu, N. Muntean, and D. Hulea, "Bidirectional power flow control in a DC microgrid through a switched-capacitor cell hybrid DC–DC converter," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3012–3022, Apr. 2017.

[12] F. Peng, L. Tolbert, and F. Khan, "Power electronics' circuit topology - the basic switching cells," in *Proc. IEEE Workshop Power Electron. Educ.*, 2005, pp. 52–57.

[13] L.-J. Chien, C.-C. Chen, J.-F. Chen, and Y.-P. Hsieh, "Novel three-port converter with high-voltage gain," *IEEE Trans. Power Electron.*, vol. 29, no. 9, pp. 4693–4703, Sep. 2014.

[14] B. W. Williams, "Generation and analysis of canonical switching cell DC-to-DC converters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 329–346, Jan. 2013.

[15] X. L. Li, Z. Dong, C. K. Tse, and D. D.-C. Lu, "Single-inductor multi-input multi-output DC–DC converter with high flexibility and simple control," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13104–13114, Dec. 2020.

[16] Y. Li, X. Ruan, D. Yang, F. Liu, and C. K. Tse, "Synthesis of multiple-input DC/DC converters," *IEEE Trans. Power Electron.*, vol. 25, no. 9, pp. 2372–2385, Sep. 2010.

[17] Q. Tian, G. Zhou, R. Liu, X. Zhang, and M. Leng, "Topology synthesis of a family of integrated three-port converters for renewable energy system applications," *IEEE Trans. Ind. Electron.*, vol. 68, no. 7, pp. 5833–5846, Jul. 2021.

[18] Z. Shan, X. Ding, J. Jatskevich, and C. K. Tse, "Synthesis of multi-input multi-output DC/DC converters without energy buffer stages," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 68, no. 2, pp. 712–716, Feb. 2021.

[19] G. Chen, Z. Jin, Y. Liu, Y. Hu, J. Zhang, and X. Qing, "Programmable topology derivation and analysis of integrated three-port DC–DC converters with reduced switches for low-cost applications," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 6649–6660, Sep. 2019.

[20] L. Mo, G. Chen, J. Huang, X. Qing, Y. Hu, and X. He, "Graph theory-based programmable topology derivation of multiport DC–DC converters with reduced switches," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 5745–5755, Jun. 2022.

[21] D. Silver et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[22] N. De Cao and T. Kipf, "MolGAN: An implicit generative model for small molecular graphs," 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1805.11973

[23] A. Mirhoseini et al., "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207–212, 2021.

[24] Y. Chen, J. Bai, and Y. Kang, "A nonisolated single-inductor multiport DC–DC topology deduction method based on reinforcement learning," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 10, no. 6, pp. 6572–6585, Dec. 2022.

[25] H. Wang, J. Yang, H.-S. Lee, and S. Han, "Learning to design circuits," 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1812.02734

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1707.06347

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.