Study on the Topological Transformation of Electromagnetic Transient Models in PSCAD and ADPSS

Aobing Li
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources North China Electric Power University
Beijing, China
1164184709@qq.com

Chongru Liu
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources North China Electric Power University
Beijing, China
chongru.liu@ncepu.edu.cn

Chenbo Su
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources North China Electric Power University
Beijing, China
scb2636@foxmail.com

Yubo Wang
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources North China Electric Power University
Beijing, China
1046462292@qq.com

Yizheng Xu
Power System Department China Electric Power Research Institute
Beijing, China
xuyizheng@epri.sgcc.com.cn

Lingran Kong
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources North China Electric Power University
Beijing, China
donald202209@163.com

2024 IEEE 8th Conference on Energy Internet and Energy System Integration (EI2) | 979-8-3315-2352-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/EI264398.2024.10991848

***Abstract*—PSCAD, as an early power system analysis software, has accumulated a large number of simulation cases, while ADPSS is an independently developed all-digital real-time simulation system in China, characterized by high cost-effectiveness and good scalability. With the expanding application range of ADPSS, the demand for modeling is increasing. However, manually converting existing PSCAD models to ADPSS models is inefficient and prone to errors. To address this issue, this paper proposes an automatic topological conversion method for electromagnetic transient models from PSCAD to ADPSS. This method utilizes the topological relationships of PSCAD projects to establish connections between primary and secondary components in ADPSS, employing a depth-first search algorithm to construct electrical islands and resolve the reference voltage of busbars. Finally, the correctness of the proposed method is verified through standard test cases from Cigre.**

Keywords—PSCAD/EMTDC, ADPSS, electromagnetic transient simulation

I. INTRODUCTION

ADPSS is a large-scale all-digital real-time simulation device for power systems developed independently in China. It can achieve real-time simulation of electromechanical transients and hybrid power systems[1]. PSCAD/EMTDC is a widely used electromagnetic transient simulation software[2]. It was developed by Manitoba Hydro in Canada and is internationally recognized for its accuracy[3]. With the maturity of the electromagnetic transient software in ADPSS, the demand for its application is continuously increasing. However, the workload of establishing and converting models between the two software is enormous and prone to errors. Therefore, developing an automatic conversion platform to simplify the model conversion between PSCAD/EMTDC and ADPSS can significantly reduce manual intervention and improve simulation efficiency.

Currently, there are various types of power system simulation and analysis software, each with different application areas. Research on the interfaces and model development between different software is continuously underway. References [4-5] propose a method for converting BPA models to PSCAD/EMTDC models. Reference [4] primarily compares the steady-state and dynamic characteristics of various commonly used electrical components and AC/DC hybrid systems in BPA and PSCAD/EMTDC, and proposes a method for converting component parameters from BPA to PSCAD/EMTDC. Reference [5] focuses on the conversion of BPA engineering model topology to PSCAD/EMTDC engineering model topology based on the node-branch incidence matrix and the depth-first search algorithm from graph theory. The literature [6] proposed a method for converting PSCAD models to ADPSS models. However, it requires generating and parsing an intermediate file in XML format, and it does not achieve the conversion of secondary component topology, nor can it interpret bus reference voltage. The literature [7] achieved the comparison and data conversion between BPA and the

This work was supported by the Science and Technology Project of State Grid Corporation of China under Grant 5100-202455012A-1-1-ZN. (Research and development of an automatic construction platform for electromagnetic transient steady-state operating points in large power grids)

DIgSILENT dynamic model. It utilized a transient data conversion program written in Python, supported by DIgSILENT, to implement data matching. The automatic conversion system from PSCAD/EMTDC to ADPSS is still under development.

To solve this problem, this paper investigates an automatic conversion system for electromagnetic transient models between PSCAD/EMTDC and ADPSS. Based on the current demand for model conversion, it aims to achieve the topology conversion of electromagnetic transient models. First, by analyzing the connection relationships of the components, the positions of the buses in the PSCAD model are identified, resulting in the topology information of the model.Next, a depth-first search algorithm is used to construct the electrical island and determine the reference voltage for each bus.Finally, based on the previous discussions, C++ is used to develop automatic conversion software from PSCAD to ADPSS in Visual Studio 2019. The correctness of the conversion results is tested using standard examples from Cigre. The simulation results indicate that the proposed method for topology conversion from PSCAD to ADPSS is feasible.

II. Topology Analysis of Electromagnetic Transient Models

In the ADPSS environment, to make the circuit diagram topology clearer, each component must be connected via bus elements. In contrast, in PSCAD, components can be connected directly using wires. Therefore, during the conversion process from PSCAD models to ADPSS models, it is necessary to perform topology analysis on the PSCAD electromagnetic transient model. This allows for determining the topological relationships of each component within the circuit, as well as the positions and orientations of the various buses.

Since the components in PSCAD are interconnected by connection lines or directly, this report first analyzes the coordinates of each component and the endpoints of the connection lines to determine the connectivity between each component and each node. Subsequently, it traverses all nodes to preliminarily assess whether they are buses based on their topological relationships. After the initial screening for buses, all component endpoints are examined: if an endpoint is not connected to a bus, the algorithm searches along the connection lines for a bus based on the node's topological relationships. If no bus is found, a new bus is added between the two components. At this point, all component endpoints are connected to buses. To avoid the situation where both ends of a connection line are connected to a bus, all connection lines are traversed, and if such a case is encountered, the two buses are merged.

A. Primary System Topology Conversion

The specific process for the topology handling of primary elements is shown in Fig 1.

Based on the topological connection relationships analyzed in the previous section, the connection status of each port for each component can be obtained. First, traverse the ports of the components and determine the connection method based on the busid connected to each port, taking the corresponding actions as shown in (1).

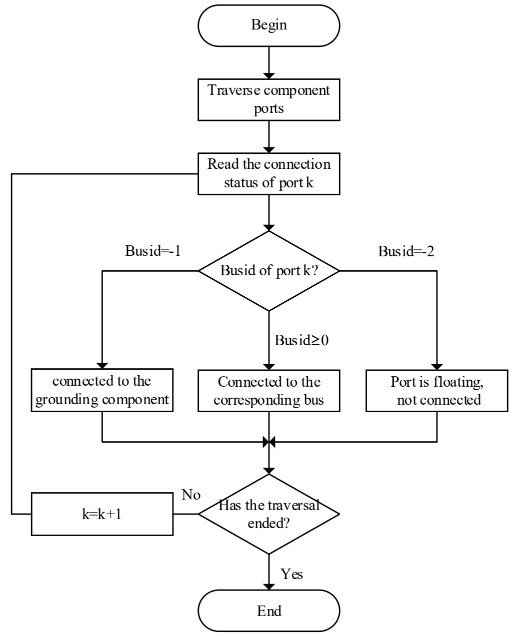
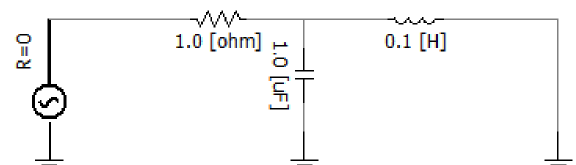


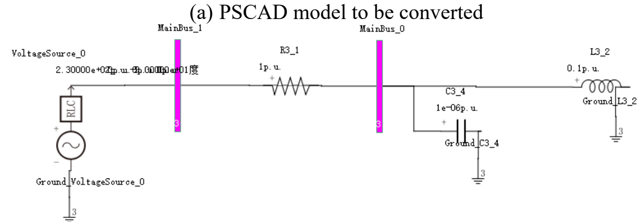
Fig. 1. Flowchart for the topology processing of primary elements

$$\begin{cases} busid = -1, & \textit{The port is not connected to any component.} \\ busid = -2, & \textit{The port is connected to the grounding component.} \\ busid \geqslant 0, & \textit{Directly connected to the bus with the same busid} \end{cases} \quad (1)$$

Fig 2 shows the result of the primary system topology transformation:



(a) PSCAD model to be converted



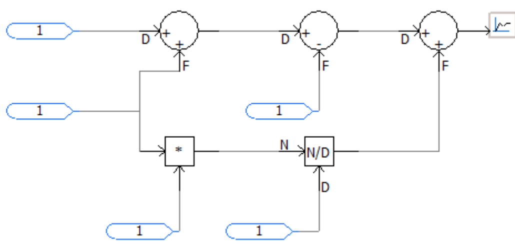
(b)ADPSS topology transformation result

Fig. 2. Comparison diagram of the primary system model topology before and after transformation

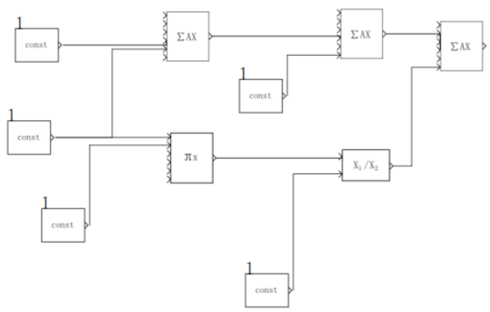
Fig. 2(a) shows the PSCAD project to be converted, while Fig. 2(b) displays the converted ADPSS project. A comparison reveals that the topological relationships in the PSCAD project remain correct before and after conversion, demonstrating the validity of the method proposed in this section.

B. Secondary System Topology Transformation

In ADPSS, secondary components do not need to be connected through a bus; they can be directly connected based on the analyzed connection relationships between the input and output ports of the components. However, in ADPSS, an input port can only be connected to one output port, while an output port can be connected to multiple input ports. Therefore, when multiple input ports are connected to a single output port, it is necessary to find the component port on the bus with the port type "Output" and connect that port to the input ports of other components with the port type "Input."



(a) PSCAD model to be converted



(b) ADPSS topology transformation result

Fig. 3. Comparison diagram of the primary system model topology before and after transformation

Fig. 3(a) shows the PSCAD project to be converted, while Fig. 3(b) displays the converted ADPSS project. A comparison indicates that the topological relationships in the PSCAD project remain correct before and after conversion, demonstrating the validity of the secondary system topology conversion method proposed in this section.

III. Bus Fundamental Voltage Processing Method Based on Depth-First Search Algorithm

Through topology analysis, the bus locations can be determined; by analyzing the phase of the component terminals, the bus types can be identified. However, the reference voltage of the bus has not been obtained. Considering the connection characteristics of components in ADPSS and the fact that only a few components in PSCAD are characterized by reference voltage, this paper employs a depth search algorithm to analyze the bus voltage.

A. Depth-First Search Algorithm

Depth-First Search (DFS) prioritizes depth during traversal, exploring to the endpoints of a tree before backtracking[8].If all vertices in the given graph have not been visited, a random vertex is selected as the starting point. The Depth-First Search algorithm can be defined as follows: First, start from vertex vi and mark it as visited. Second, randomly select an adjacent vertex vj of vi. If vj has not been visited, continue the search with vj as the new starting point. This process constitutes the depth-first search[9]. If we denote the access sequence of DFS as S, then it is represented as shown in (2).

$$S = [v_1, v_2, ..., v_n] \tag{2}$$

Here, v_i represents the nodes accessed in the order of DFS.

It is evident that the Depth-First Search method is implemented through recursion, which aligns well with the principles of computer programming. Because it searches as deeply as possible before backtracking, it is referred to as depth-first search[10]. For example, in the depth-first search method, after visiting vertex vi, the next step will be to search for an unvisited edge (v_i,v_j). If vertex v_j has already been visited, it will return to v_i and choose another unvisited edge from v_i. If vertex v_j has not been visited, it will follow this edge from vi to v_j, visit v_j, and mark it as visited. Then it will start searching from v_j using the same method as with v_i, continuing until all paths starting from v_j have been explored before backtracking to vertex v_i. Finally, it will search for another unvisited edge from v_i and repeat the process until all edges originating from the starting point v_i have been checked. At this point, if v_i is not the initial starting point, it will backtrack to the previously visited vertex before v_i; if v_i is the initial starting point, the entire search process ends.

In an undirected graph, we can denote the connected components accessed by DFS as C:

$$C = \{v \in U | \ DFS(v) is\, reachable\, from\, u\} \tag{3}$$

Here, v is an arbitrary node.

If the search process starting from a source point in a connected undirected graph or a strongly connected directed graph comes to an end, it means that a complete traversal of that component of the graph has been finished. However, in the case of a disconnected undirected graph or a non-strongly connected directed graph, a single traversal starting from a source point can only access one connected component of the graph. Therefore, it is necessary to traverse each connected component in order to completely explore a disconnected graph. The definition of the path length from the starting node s to the target node t in the graph is given by (4).

$$L(s,t) = \sum_{i=1}^{k} d(v_i, v_{i+1}) \tag{4}$$

For the undirected graph, the traversal process of a depth-first search starting from vertex vi is shown in Fig. 4.

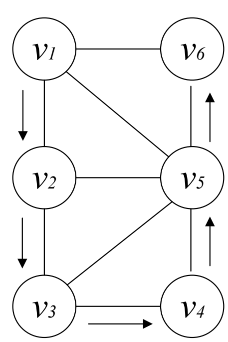


Fig. 4. Comparison diagram of the primary system model topology before and after transformation

When traversing a graph using the depth-first search algorithm, the depth-first search traversal sequence is the sequence obtained by the order in which the vertices are visited. The access sequence of a graph is not unique; it depends not only on the algorithm but also on the data structure used for the vertices and the choice of the starting point.

The complexity of the depth-first search algorithm is mainly divided into time complexity and space complexity. Let the number of nodes be V and the number of edges be E. In the worst-case scenario, each node will be visited once, and each edge will be crossed once. Therefore, the time complexity T(n) is given by (5).

$$T(n) = O(V) + O(E) \quad (5)$$

In terms of space complexity, the depth-first search (DFS) requires additional storage in the form of a stack to keep track of path information. In the worst case, if backtracking to all nodes in the graph is necessary, the space complexity S(n) is given by equation (6).

$$S(n)=O(V) \quad (6)$$

B. *Electrical Island Based on Depth-first Search Algorithm*

In a circuit, there are three types of components that can determine the rated voltage: generators, voltage sources, and transformers. The reference voltage of the bus connected to the generator is the rated voltage of the transformer, while the reference voltage of the bus connected to the voltage source is the rated voltage of the voltage source. The reference voltages of the buses connected to the primary and secondary sides of the transformer are the rated voltages of the primary and secondary sides, respectively. Starting from these three types of components—generators, voltage sources, and transformers—in that order, the reference voltages of the connected buses are analyzed. Buses that have already been analyzed will not be analyzed again. The search for a particular branch stops when encountering a transformer component or a grounding element. An electrical island is defined based on the analysis results, starting from components that can determine the rated voltage, and continues until all branches connected to it are analyzed. All components and buses within these branches together form an electrical island, where the voltage levels of the components in the same electrsical island are the same.

For example, Generator 1 has a rated voltage of 10 kV, and Generator 2 has a rated voltage of 35 kV. The reference voltage of the bus connected to Generator 1 is the same as that of Generator 1. An electrical island A is created, and the bus and components are included in this electrical island. Then, a deep search is initiated from this bus to find other buses connected to it, and all connected buses and components are saved within electrical island A. Buses that have already been analyzed will not be analyzed again, and the search for a particular branch stops when encountering a transformer component or a grounding element. After completing the search starting from Generator 1, the process continues by using Generator 2 as the starting point to analyze the reference voltage of the bus. Again, previously analyzed buses will not be re-evaluated, and the search stops when encountering a transformer component.

C. *Handling Special Cases*

Since the electrical islands in Section 3.2 are limited to single subcircuits, there may be cases where certain subcircuits lack components that can determine the rated voltage. Therefore, after completing the generation of electrical islands for each circuit, it is necessary to handle the components in subcircuits without a determined reference voltage.

The connection between subcircuits and higher-level circuits is achieved through the ports of subcircuit components and external node components (xnode) within the subcircuit. Thus, the external nodes can be utilized to address the connection relationship between the subcircuit and the higher-level circuit, managing the bus reference voltage between the xnode and the island. For electrical islands without a bus reference voltage, a depth-first search algorithm can be used, starting from the external nodes and subcircuit components within the electrical island to analyze the bus reference voltage.

For example, if electrical island B within main circuit A does not contain any components that can determine the rated voltage, but there exists a port D of main circuit component C within electrical island B. Then, a depth-first search is initiated starting from subcircuit component C to find the corresponding xnode component 1 for port D in subcircuit C. If the reference voltage of island E, where component 1 is located, is 35 kV, then the reference voltage of island B will be set to the reference voltage of island E. If island E also lacks a determined reference voltage, the search continues from island E to find either main circuit components or unresolved xnode components until an island with a defined reference voltage is encountered.

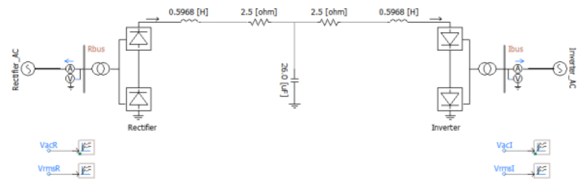
By utilizing a depth-first search algorithm to analyze the connections between subcircuits, external node components, and islands, the reference voltages can be assigned to all electrical islands within the circuit, thus obtaining the reference voltages for all components and nodes. When generating the bus, the reference voltage of the bus can be determined based on the reference voltage of the nodes it corresponds to or the components it connects to.

IV. Analysis of Conversion Results

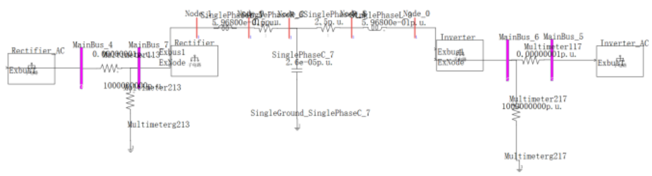
This paper uses C++ language to build an automatic conversion system between PSCAD and ADPSS models based on Visual Studio 2019, and tests the conversion results using Cigre standard test cases.

The case involves a monopole 500 kV, 1000 MW DC transmission system, with both the rectifier and inverter sides utilizing 12-pulse converters, connected to a weak AC system (with a short-circuit ratio of 2.5 at a frequency of 50 Hz). This system is widely used for testing DC systems. In this test case, voltage and current display elements for DC and AC lines were added to the rectifier and inverter sides in PSCAD to verify the correctness of the simulation results before and after model conversion. Figures 7 and 8 show the main circuit diagrams of the Cigre standard test case before and after conversion.

Fig. 5 illustrates the connectivity diagrams of the Cigre-Benchmark model before and after conversion. A comparison of the images shows that the network connectivity remains consistent before and after the conversion.

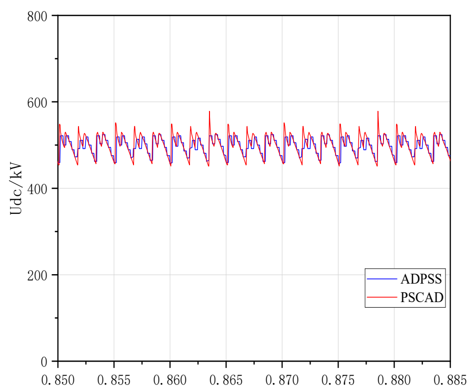


(a) Topology of the Cigre Test Case Circuit in PSCAD

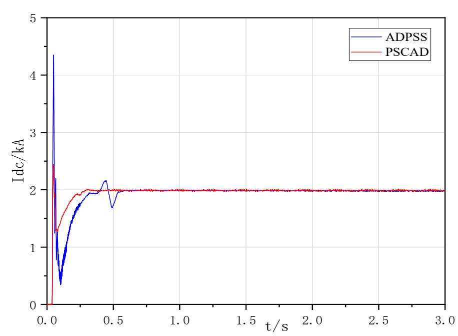


(b) Topology of the Cigre Test Case Circuit in ADPSS

Fig. 5. Topology Comparison Diagram Before and After Cigre Example Conversion.



(a) DC Voltage Image



(b) DC Current Image

Fig. 6. DC Voltage and Current Comparison Diagram Before and After Conversion.

Fig. 6 shows the comparison of DC voltage and current waveforms before and after conversion. From the figure, it can be observed that the simulation trends of PSCAD and ADPSS are basically consistent, indicating that the dynamic responses of the two networks are similar. This further validates the consistency of the two cases and demonstrates the feasibility of the proposed model conversion method in engineering applications.

V. Conclusion

This paper proposes a method for converting model topology from PSCAD to ADPSS. By analyzing the differences in component connections between PSCAD and ADPSS, the connections of components in the PSCAD model are resolved, resulting in the extraction of topological information. This method enables the integration of primary and secondary systems. Additionally, a depth-first search algorithm is employed to construct electrical islands and determine the reference bus voltages. Finally, an automatic conversion system from PSCAD to ADPSS is built in Visual Studio, and the feasibility of the proposed method is demonstrated using Cigre standard test cases.

Acknowledgement

This work was supported by the Science and Technology Project of State Grid Corporation of China under Grant 5100-202455012A-1-1-ZN. (Research and development of an automatic construction platform for electromagnetic transient steady-state operating points in large power grids)

References

[1] T. L. Ye, X. W. Wang, J. Gao, and H. Fan, "Applying and debugging of ADPSS in Hebei electric power grid," Power System Protection and Control, vol. 37, no. 13, pp. 104-108+128, 2009.

[2] L. Liang-zhen and Y. Lin, "AN INTRODUCTION TO PSCAD/EMTDC," Power System Technology, vol. 24, no. 1, pp. 65-66, 2000, doi: 10.3321/j.issn:1000-3673.2000.01.020.

[3] N. Eghtedarpour and E. Farjah, "Power Control and Management in a Hybrid AC/DC Microgrid," IEEE Transactions on Smart Grid, vol. 5, no. 3, pp. 1494-1505, 2014, doi: 10.1109/tsg.2013.2294275.

[4] H. Teng, C. Liu, M. Han, and S. Ma, "Synchronous Generator Model Transformation Between BPA And PSCAD," Power and Energy Engineering Conference, vol. null, no. null, pp. 1-4E-ISBN:978-1-4244-4813-5, 2010, doi: 10.1109/appeec.2010.5448637.

[5] H. Tao, Z. Yang, M. Zhang, J.-X. Yang, R.-M. He, and Y. Shi, "A depth-first search algorithm based implementation approach of spanning tree in power system," Dianwang Jishu/Power System Technology, vol. 34, no. 2, pp. 120-124, 2010.

[6] W. Dong, Q. Pan, J. Lei, H. Xie, H. Xin, and H. Wu, "Comparison and data conversion between dynamic models of BPA and DIgSILENT," Dianli Xitong Zidonghua/Automation of Electric Power Systems, vol. 40, no. 8, pp. 140-146, 2016, doi: 10.7500/AEPS20150601013.

[7] X. Guochao, L. Chongru, H. U. Haiyang, W. Chao, and X. U. Dechao, "Research on Electromagnetic Transient Engineering Model Conversion Method Based on XML," Inner Mongolia Electric Power, 2017.

[8] C. Li and M. Ueno, "An extended depth-first search algorithm for optimal triangulation of Bayesian networks," International Journal of Approximate Reasoning, vol. 80, pp. 294-312, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.ijar.2016.09.012.

[9] Y. Du, F. Li, T. Zheng, and J. Li, "Fast Cascading Outage Screening Based on Deep Convolutional Neural Network and Depth-First Search," IEEE Transactions on Power Systems, vol. 35, no. 4, 2020, doi: 10.1109/tpwrs.2020.2969956.

[10] J. Liang, H. Chen, and Y. Liu, "On algorithms for state feedback stabilization of Boolean control networks," Automatica, vol. 84, no. C, pp. 10-16, 2017, doi: 10.1016/j.automatica.2017.06.040

[11] W.-C. Yeh, "A Fast Algorithm for Quickest Path Reliability Evaluations in Multi-State Flow Networks," IEEE Transactions on Reliability, vol. 64, no. 4, pp. 1175-1184, 2015, doi: 10.1109/tr.2015.2452580.