Development of a Physics Informed Neural Network based Simulation Methodology for Dynamic-PSA

Young Ho Chae[a], Jung Jin Bang[b], Hyeonmin Kim[c], and Poong Hyun Seong[a]

[a] Department of Nuclear And Quantum Engineering, Korea Advanced Institute of Science and Technology, 34141, 291, Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea, cyhproto@kaist.ac.kr, phseong1@kaist.ac.kr

[b] Future and Challenge Tech. Co. Ltd., Heungdeok1ro 13, Yeongdeok-dong, Giheung-gu, Yongin-si, Gyeonggi-do, Republic of Korea,

[c] Korea Atomic Energy Research Institute, 34057, 111, Daedeok-daero, 989 Beon-gil, Yuseong-gu, Daejeon, Republic of Korea

---

Abstract: To evaluate the risk of NPP, PSA analysis is widely conducted. Static PSA is powerful to consider the potential risk of the plant. However, the method is difficult to consider dynamic interactions. For the Dynamic PSA (DPSA) several analysis methods were developed and suggested. One of the widely used analysis methods is discrete dynamic event tree-based DPSA. However, the bottleneck in performing DPSA based on the discrete dynamic event tree method is the physical process model. The conventional physical process model is hard to consider the dynamic interaction in reasonable calculation time. Therefore, we suggested a data-driven simulation method for DPSA. The model consists of a model generator and a solution generator. The model generator dynamically generates governing equations with given data. And solution generator predicts the solution based on the generated equation from the model generator. By using the suggested method, the dynamic interaction considered prediction can be conducted.

---

1. INTRODUCTION

Since nuclear power plant (NPP) is a safety-critical infrastructure, enhancing the safety of NPPs and minimizing risk are always important issues. To evaluate the safety of NPPs, the probabilistic safety assessment method is widely used. However, despite the strengths, PSA has several limitations. The conventional PSA method is hard to consider dynamic interactions.

In the nuclear power plant, dynamic interactions can be classified into two categories. The first is dynamic interactions that have long-time constants. Examples that have a long time constant include plant aging, configuration changes, environmental variations, organizational changes, etc. To deal with dynamic interaction that has long time constants, prognostics research was conducted [1]. The second is dynamic interactions that have short time constants. For instance, operator interactions, thermal-hydraulic processes, interaction within instrumentation and control systems, etc. According to NUREG/CR-6942 [2], dynamic interaction with short time constants is classified into type1 and type2 interaction. For the dynamic probabilistic safety assessment (DPSA), dynamic interaction with short time constants should be modeled and simulated (especially type1 interaction).

For the dynamic PSA analysis, several methods were suggested. Monte Carlo-based DPSA method [3] can simulate the actual process and random behaviors. However, in the case of an actual NPP, the number of possible actions is extensive. Therefore, the method inherently has calculation problems. And also, the Markov modeling-based DPSA methods [4] require intensive computation as the state of the system increases. A continuous event tree-based DPSA method is also suggested. However, the CET method has limitations in that the method requires problem-specific algorithms. A dynamic fault tree-based DPSA methodology [5] using dynamic gates (PAND, SEQ, etc) has been proposed, however, the method requires pre-requirement of physical process responses.

One of the widely used analysis methods for DPSA is the discrete dynamic event tree (DET) method. The dynamic event tree method consists of the physical process model, equipment model, and operator model. However, the bottleneck of DET-based DPSA is the physical process model. In the case of NPP, in Korea, multi-dimensional analysis of reactor safety (MARS) code is widely used which is based on reactor excursion and leak analysis program (RELAP) code. The MARS code accurately simulates the given condition. However, because the analysis is based on a numerical method, the code inherently has a time-accuracy tradeoff. To increase the analysis resolution, the mesh size should be finely divided. Then the finely divided mesh requires more calculations. And also, the analysis code is hard to reflect the change of state due to the operation of the component.

Therefore, we propose a novel AI utilizing Physics Related Information-based Simulation Method (A-PRISM). A-PRISM consists of a solution generator and an equation generator. Both generators are based on physics informed neural network model. The solution generator calculates the simulation results according to the given initial and boundary condition. The equation generator creates the governing equation that best describes the measured data from the plant. The generated equation automatically updates the physics part of the solution generator. As a result, it is possible to calculate the data based simulation result considering the dynamic interactions.

2. PHYSICS INFORMED NEURAL NETWORK (PINN)

The physics informed neural network (PINN) was firstly proposed by M Raissi et al. [6]. The major difference between the general artificial neural network and PINN is remarkable in the loss function. Fig.1 shows the schematic diagram of the naive neural network and PINN.

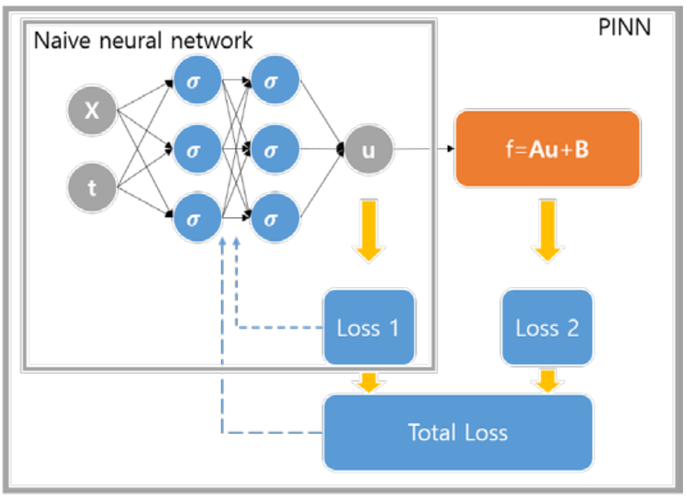

Fig. 1 Schematic diagram of naive neural network and physics informed neural network

The naive neural network utilizes a single loss which is calculated from the difference between the latent vector and target vector. However, PINN not only utilizes a loss from the difference between latent vector and target vector but loss from the form of the equation. The detailed description of the loss function in PINN is shown in Eq. 1.

Total loss: $L(\boldsymbol{\theta},\ \boldsymbol{\Lambda}; D_u, D_c) = L_d(\boldsymbol{\theta}; D_u) + \alpha L_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; D_c) + \beta \|\boldsymbol{\Lambda}\|_0$ Eq.1-1
Data loss: $L_d(\boldsymbol{\theta}; D_u)$ Eq.1-2
Physics loss: $L_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; D_c)$ Eq.1-3

$\boldsymbol{\theta}$ *is a training parameters in neural network*
$\boldsymbol{\Lambda}$ *is a training parameters in physics network*

The total loss can be defined as Eq.1-1 and the first term on the RHS is loss from the data which is also used in a naive neural network. The second term is physics loss. The physics loss is calculated based on the form of the equation. the third term is the regularization term to prevent overfitting.

By using the loss from the equation, the PINN can take several advantages. The first is data efficiency. The provided equation can take a role as a restriction. In conventional neural network trained that restriction with data. However, in PINN the information is eminently provided as a form of the equation. Therefore, data efficiency is improved compared to the conventional neural network. And also, because of the equation loss, the PINN has robust characteristics in extrapolation [6].

3. AI-utilized Physics Related Information-based Simulation Method (A-PRISM)

3.1. Architecture overview

The overall architecture of the A-PRISM model is described in Fig.2.

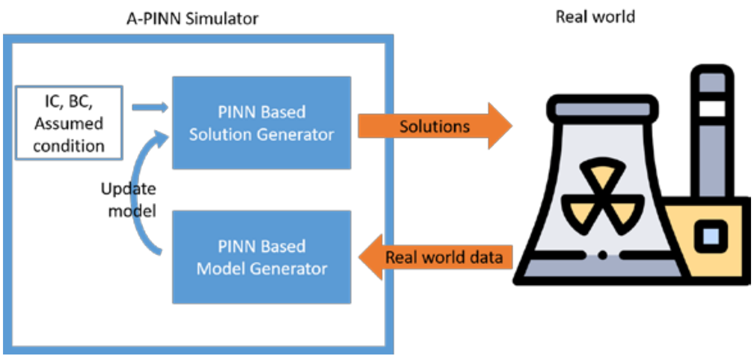

Fig. 2 A-PRISM Model

The model consists of two parts. The first part is the Equation generator, and the second part is the solution generator. Both generators are based on the PINN algorithm. The solution generator calculates the solution with governing equation, initial condition, boundary condition, or arbitrary assumed conditions. The role of the solution generator is similar to conventional simulation codes. For instance, to imitate the MARS code, the solution generator has six kinds of built-in loss function: continuity loss, momentum loss, and energy loss for liquid phase and gaseous phase. However, with only a solution generator alone, it is impossible to consider dynamic interactions in NPP. In actual NPP operation, multiple components interact dynamically. Therefore, the form of the governing equation and initial condition also changes variously. To consider the interaction, the model generator configures a governing equation with given data. As a result of the model generator, the governing equation that describes the given data will be created. And providing the equation to the solution generator, the model estimate and predict parameters including dynamic interactions.

3.2. Model Generator (Equation Generator)

The form of model generator is inspired from Zhao Chen et al. [7]. The model generator has three stages. The first is the conventional deep neural network stage. In this stage, the network calculates the latent vector. The second stage is the AutoDiff stage. In this stage, a possible form of governing equation is created. The third stage is ranking the candidate function stage. In this stage, the suitability ranking is calculated among candidate functions. The schematic diagram of model generator is shown in Fig.3.

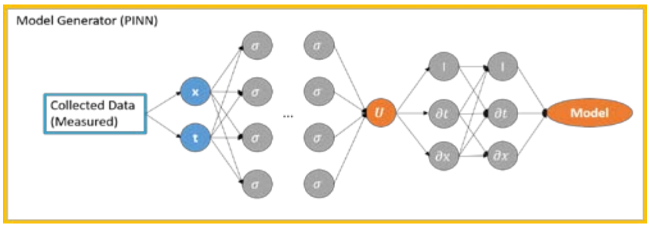

Fig. 3 Schematic diagram of model generator

The detailed architecture of solution generator is listed in below.

- Neural network part
  - Number of layer: 4 layers
  - Neurons in each layer: 20 neurons
  - Activation function: adoptive rectified linear unit
  - Loss function: mean squared error (MSE)
  - Optimization algorithm: Limited memory-BFGS
- Physic network part (AutoDiff)
  - Number of layer: 2 layers
  - Neurons in each layer: 3 (partial derivation to t, partial derivation to z, I)
  - Activation function: adoptive rectified linear unit
  - Loss function: MSE from physics loss

3.3. Solution Generator

The solution generator calculates the result corresponding to the given condition. As an input for the solution generator, IC, BC, and governing equations are provided. And the output from the network is the calculation result (simulation result) of the given condition. The schematic diagram of solution generator is shown in Fig. 4. The detailed architecture of solution generator is listed in below.

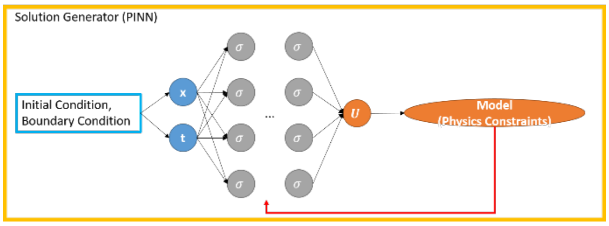

Fig. 4 Schmatic diagram of solution generator

- Neural network part
  - Number of layer: 4 layers
  - Neurons in each layer: 20 neurons
  - Activation function: adoptive rectified linear unit
  - Loss function: mean squared error (MSE)
  - Optimization algorithm: Limited memory-BFGS
- Physic network part
  - Number of layer: 2 layers
  - Neurons in each layer: 3 (partial derivation to t, partial derivation to z, I)
  - Activation function: adoptive rectified linear unit
  - Loss function: MSE from physics loss

4. Experiment

The hardware specifications used in the experiment are as follows.

- CPU: Intel® Core™ i7-6990K
- GPU: NVIDIA GeForce GTX 1080 Ti *2
- RAM: 128GB

4.1. Pilot Experiment

To figure out the applicability of suggested model, the pilot experiment is conducted. Using the data points generated in Eq.2 as input, the governing equation was inferred, and the prediction based on the inferred governing equation was conducted. The points that satisfy Eq.1 are collected. 10,000 points were collected.

$$\frac{\partial^2 \phi}{\partial x^2} - \phi * 0.5 = \frac{\partial \phi}{\partial t} \qquad \text{Eq.2}$$

After the 37.3345sec, the model generator creates the equation as Eq.3.

$$0.99997\frac{\partial^2 \phi}{\partial x^2} - \phi * 0.50011 = 0.99999\frac{\partial \phi}{\partial t} \qquad \text{Eq.3}$$

And the solution generator calculates solutions as follows (Fig.5):

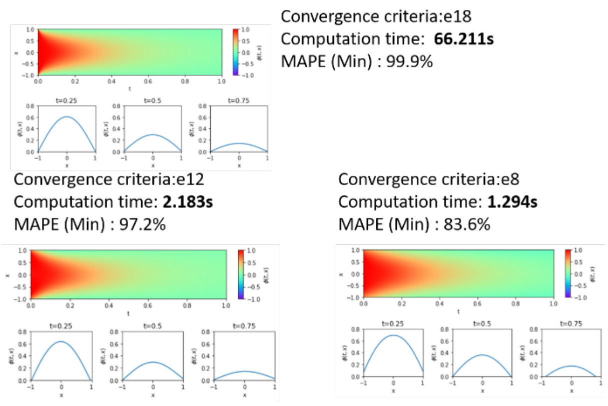

Fig. 3 Calculation results - Pilot experiment

As shown in Eq.3, the model generator successfully modeled the governing equation with data. Also, the solution generator successfully estimated the solutions.

4.2. Experiment – After the Loss of Coolant Accident

To verify that predictions taking into account dynamic interactions in a nuclear power plant are possible, experiments were performed based on data using the Compact Nuclear Simulator (CNS) [8]. Assuming the pressurizer LOCA situation, the axial flux of the core was predicted. The prediction was performed for the next 45 seconds based on the data acquired for 5 seconds at 23 flux measurement points. The Fig. 6,and 7 shows the actual data and generated data. And the Fig. 8 shows error between actual data and generated data. The calculation time of the suggested model (equation generation, solution generation) was 84.2453seconds. The maximum error was 6.9%.

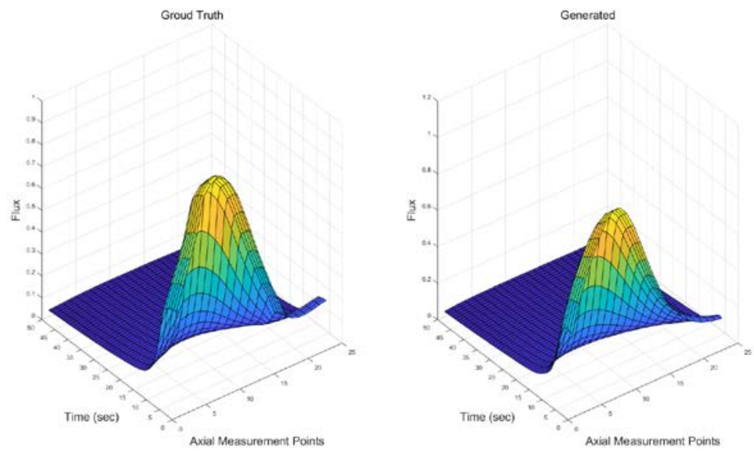

Fig. 4 Prediction results 1 - CNS experiment

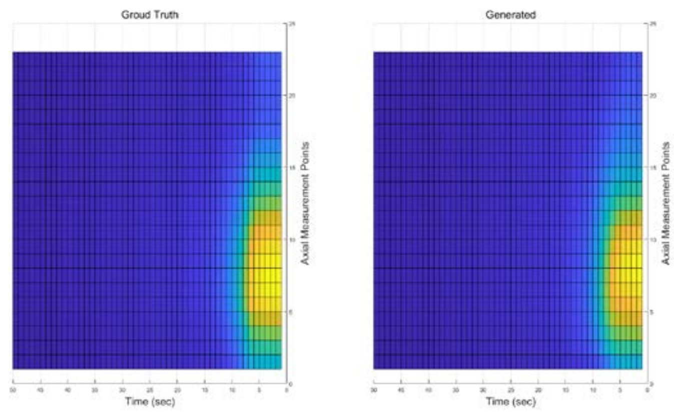

Fig. 5 Prediction results 2 (Top view) - CNS experiment

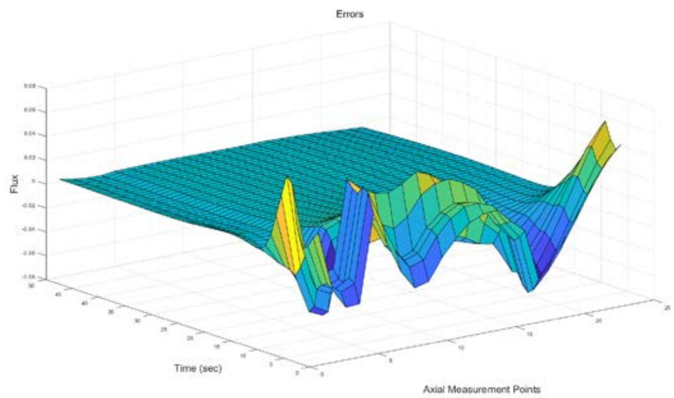

Fig. 6 Errors (Ground truth - prediction)

5. CONCLUSION

The bottleneck in performing DPSA based on the dynamic event tree method is the physical process model. The calculation speed of the physical process model is too slow and it is difficult to consider the interaction of components in real-time. Therefore, in this study, a data-driven simulation methodology was proposed to solve the problem. The suggested simulation methodology consists of a model generator and a solution generator. The conventional simulation is possible with only the solution generator itself, but a model generator is added to perform real-time changing calculations in consideration of dynamic interaction.

The model generator receives real-time data and generates the governing equation most suitable for the current state. The solution generator provides prediction results based on the governing equation generated by the model generator. Therefore, using the proposed methodology, both data-based simulation and prediction can be performed.

In the case of 1d prediction, the calculation was completed in less than a minute, but in the case of 2d prediction (23 by 50), the calculation takes more than a minute, so it seems that the optimization of the calculation process is still required.

Using the proposed model, conventional simulation can be performed using only the solution generator, and prediction considering dynamic interaction can also be performed by combining the model generator and solution generator.

Acknowledgements

This research was supported by the National R&D Program through the National Research Foundation of Korea (NRF) funded by the Korean Government. (MSIP: Ministry of Science, ICT and Future Planning) (No. NRF-2016R1A5A1013919)

References

[1] Baraldi, Piero, Francesca Mangili, and Enrico Zio. "A prognostics approach to nuclear component degradation modeling based on Gaussian Process Regression." Progress in Nuclear Energy 78 (2015): 141-154.
[2] NUREG/CR-6942 Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments [R]. U.S. Nuclear Regulatory Commission, 2007
[3] Marseguerra, Marzio, and Enrico Zio. "Monte Carlo approach to PSA for dynamic process systems." Reliability Engineering & System Safety 52.3 (1996): 227-241.
[4] Faghih-Roohi, Shahrzad, Min Xie, and Kien Ming Ng. "Accident risk assessment in marine transportation via Markov modelling and Markov Chain Monte Carlo simulation." Ocean engineering 91 (2014): 363-370.
[5] Rao, K. Durga, et al. "Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment." Reliability Engineering & System Safety 94.4 (2009): 872-883.
[6] Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational physics 378 (2019): 686-707.
[7] Chen, Zhao, Yang Liu, and Hao Sun. "Physics-informed learning of governing equations from scarce data." Nature communications 12.1 (2021): 1-13.
[8] Korea Atomic Energy Research Institute, Advanced Compact Nuclear Simulator Textbook, Nuclear Training Center in Korea Atomic Energy Research Institute.A. B. Author and C. D. Author. "*Title of the articl*e", The Journal, volume, pp. 110-120, (2020).