

## NPP Condition Diagnosis by Monitoring Critical Safety Functions

Jaemin Yang, Jonghyun Kim\*

Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, 501-709, Republic of Korea

\*Corresponding author: jonghyun.kim@chosun.ac.kr

### 1. Introduction

Diagnosis of the nuclear power plant (NPP) condition is performed under the judgment of operators based on the procedures. However, human errors such as inappropriate judgment or action during the operation may aggravate the safety and integrity of the plant. According to the Operational Performance Information System (OPIS) database in Korea [1], from 2000 to 2016, about 17% of events (47 of 274) were caused by human error.

When anomalies occur in NPPs, the early detection is important for safe and economic operations. In addition, when an emergency occurs in NPPs, the operator should monitor safety functions periodically and repeatedly which are critical for plant integrity. In those situations, the operator needs to identify possible success paths as necessary, and try to stabilize or safely shutdown the plant using emergency operating procedures (EOPs) [2]. In case of abnormal or emergent situation, aside from progressing procedures, other things (e.g., multiple faults, numerous alarms, conflicting data and missing or incomplete information) may bother operators [3].

If an autonomous monitoring and diagnosis algorithm which consider main safety variables is applied to support the operators, it is possible that the proportion of human error can be decreased significantly. Furthermore, the results of monitoring will also be helpful to cope with abnormal or emergent situation with high credibility.

This study attempts to develop an algorithm for monitoring the status of safety parameters in NPP. Critical safety functions (CSFs) which are important to prevent core damage are considered for selection of safety parameters. Then, this study suggests a rule-based algorithm for monitoring safety parameters.

### 2. Selection of major safety parameters

To develop an algorithm for monitoring safety parameters, operating parameters are chosen based on the CSFs considered in three-loop Westinghouse type pressurized water reactor (PWR). Figure 1 shows the hierarchy of safety functions in Westinghouse type PWR. The six CSFs (i.e., subcriticality, core cooling, heat sink, RCS integrity, containment integrity and RCS inventory) based on the potential threat to the three barriers (i.e., fuel cladding, primary coolant system boundary, Reactor Building) are considered. Because each of them has hierarchical structure which comprises system variables affecting the plant safety, CSFs allow the operator to respond to these threats prior to event diagnosis [5].

This study uses the Compact Nuclear Simulator (CNS) as a testbed. The plant model in CNS is three-loop Westinghouse PWR. On the basis of six CSFs, 32 parameters in CNS are selected. The parameters related with CSFs are described in Figure 1. For example, to assess the RCS integrity, RCS pressure and Cold-leg temperature by each loop need to be monitored.

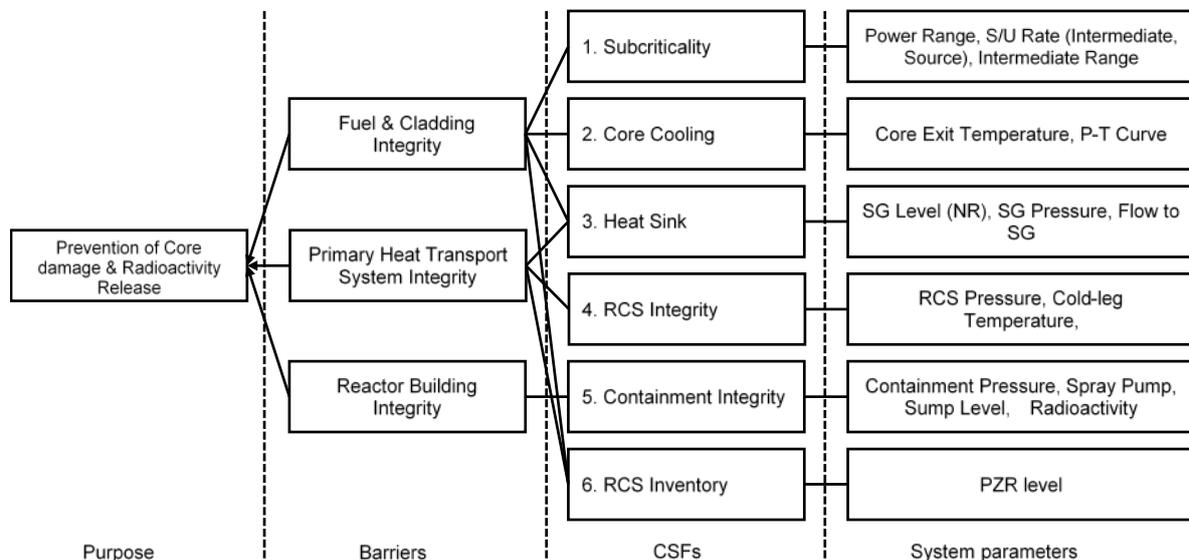


Fig. 1. The hierarchy of safety functions in Westinghouse PWR

### 3. Rule-based expert system for monitoring CSFs

To develop a monitoring algorithm for CSFs, a rule-based expert system which uses rules as the knowledge that mimic the reasoning of human expert encoded in to the system is applied. It is the simplest form of artificial intelligence. It provides a way to code expert knowledge of narrow areas into an automated systems. Simply, rule-based systems can be created by using assertion sets and a set of rules that specify how assertions work such as a set of if-then statements (i.e., IF-THEN rules). [4].

The rule-based expert system is implemented with Python 3.5.3 on the basis of CSF tree procedures applied in three-loop Westinghouse type PWR. Figure 2 shows an example of CSF tree for subcriticality. The determination of subcriticality is based on values of power range and intermediate range detector, start-up rate of intermediate and source range. As a result of determination, the status of CSF is classified into 4 levels. Level 1 states the normal condition of CSF and level 2 indicates the abnormal condition of CSF. Levels 3 and 4 indicate significant and extreme threats of CSF, respectively. An example algorithm is shown in figure 3.

### 4. Result

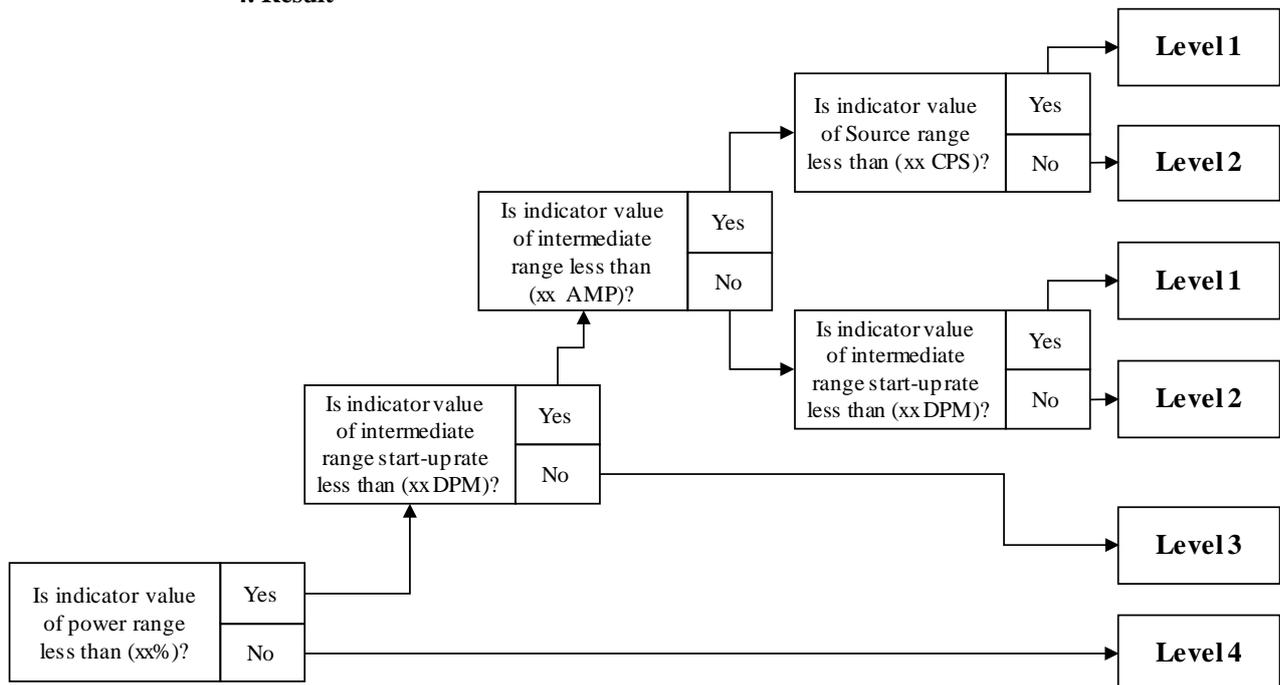


Fig. 2. An example of CSF tree for subcriticality

This study implements six CSFs monitoring algorithm with the rule-based expert system. For demonstration, CNS data for the loss of coolant accident (LOCA) scenario with the size of 200 square centimeters were used. In addition, the malfunction injection time to the CNS simulator is 30 seconds. The end of simulation time is 2532 seconds. There was no control or additional interventions. Figures 3 to 8 show the CSFs monitoring results. Y-axis represents the levels of CSFs. The results indicate that each CSF can be monitored well by the developed algorithm. In addition, the feasibility of the algorithm for seven events, including steam generator tube rupture (SGTR), LOCA with SI valve fail, and main steam line break, has been tested.

### 5. Conclusion

The aim of this study is to develop an algorithm for monitoring the CSFs in NPP for unloading operator's task in abnormal or emergent situation for safety. It is expected that this approach can be applied to not only diagnosis of NPP states but also the performance monitoring.

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

dfRefflat = np.loadtxt('...', delimiter=',', usecols=range(32))

PR_ = dfRefflat[:,0:1]
IR_ = dfRefflat[:,1:2]
IRA_ = dfRefflat[:,3:4]
SR_ = dfRefflat[:,2:3]

Out = np.empty_like(PR_)
for i in range(0,len(PR_)):
    PR = PR_[i,0]
    IR = IR_[i,0]
    IRA = IRA_[i,0]
    SR = SR_[i,0]
    if PR <= 0:
        if IR <= 0:
            if SR <= 0:
                Out[i,0]=1
            else:
                Out[i,0]=2
        elif IR <= 1:
            Out[i,0]=1
        else:
            Out[i,0]=2.5
    else:
        Out[i,0]=3
    else:
        Out[i,0]=4

```

Fig. 3. An example algorithm for Subcriticality on Python

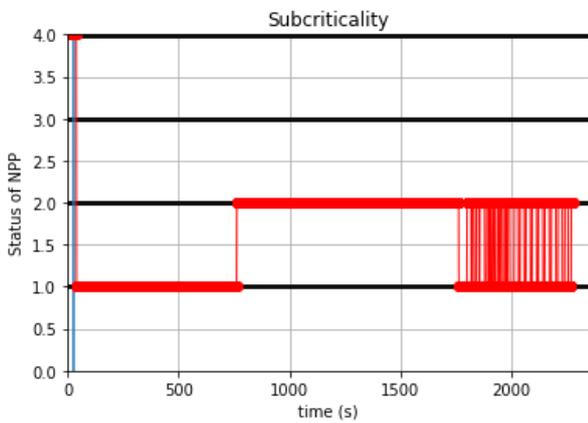


Fig. 4. The subcriticality status by time (LOCA)

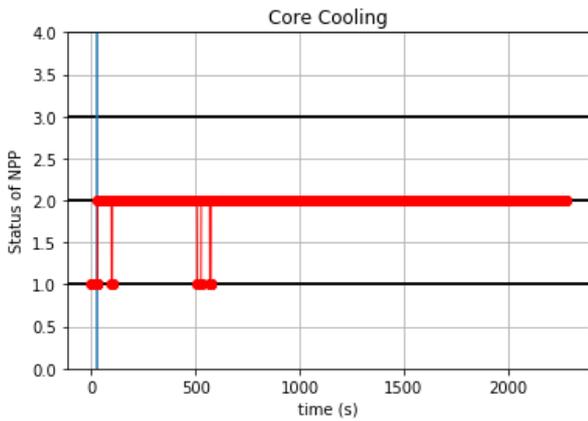


Fig. 5. The core cooling status by time (LOCA)

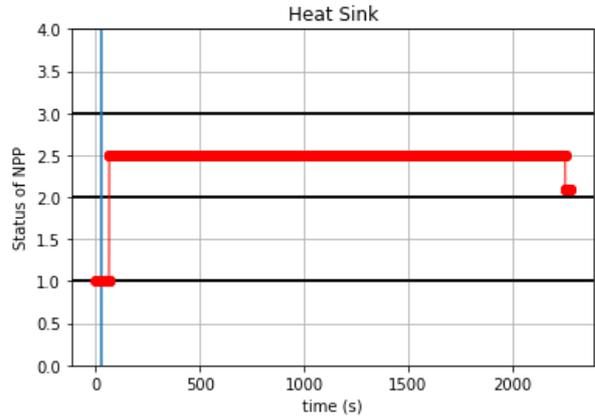


Fig. 6. The heat sink status by time (LOCA)

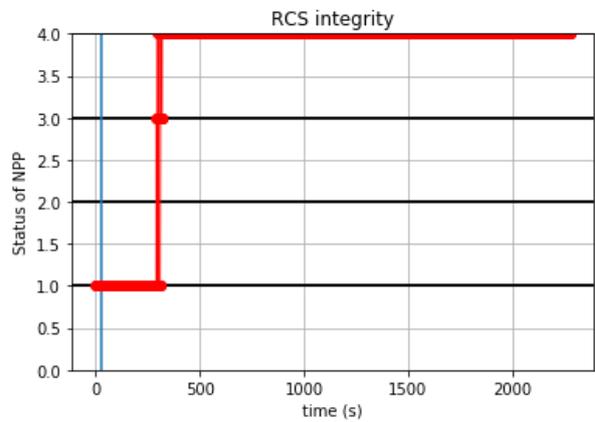


Fig. 7. The RCS integrity status by time (LOCA)

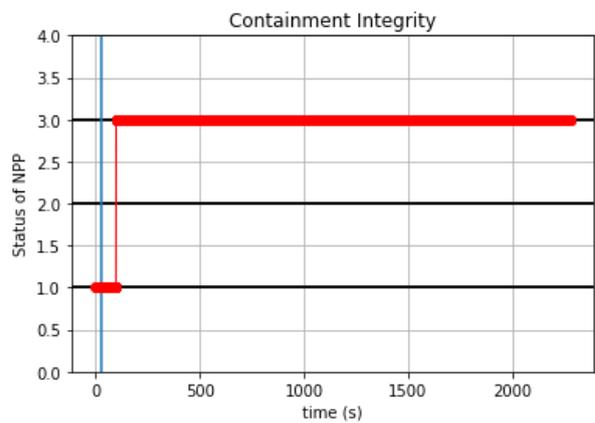


Fig. 8. The containment integrity status by time (LOCA)

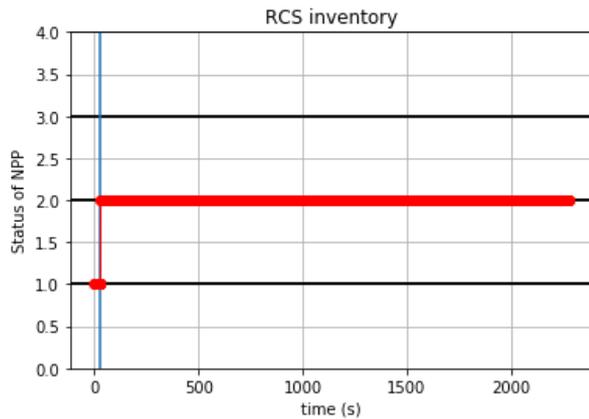


Fig. 9. The RCS inventory status by time (LOCA)

### REFERENCES

- [1] OPIS, Nuclear Event Evaluation Database, Korea Institute of Nuclear Safety, 2016
- [2] O. Maruo et al., HIS for Monitoring the Critical Safety Functions Status Tree of NPP, International Nuclear Atlantic Conference, 2013.
- [3] Dvorak, Daniel L. Expert systems for monitoring and control. Artificial Intelligence Laboratory, University of Texas at Austin, 1987.
- [4] Crina Grosan and Ajith Abraham, Intelligent Systems – A Modern Approach, Springer, 2011
- [5] Y. Huichang, et al., Development of Critical Safety Function Status Trees for Pressurized Heavy Water Reactors, Korean Nuclear Society Autumn Meeting, 2003