Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 23-24, 2019

Optimization of PI Controller for APR1400

See Chae Jeong [a*], In Ho Song [a], Jae Young Huh[a], Gyu Cheon Lee [a], Nam Kyun Kim [b]
[a] *KEPCO Engineering & Construction Co., 111, Daedeok-daero 989, Yuseong-gu, Daejeon, 34057*
[b] *Korea Hydro. & Nuclear Power Co., 2040 Uljinbuk-ro, Uljin-gun, Geongsangbuk-do, 36304*
[*] *Corresponding author: scjeong@kepco-enc.com*

1. Introduction

The Proportional-Integral (PI) controllers in the control systems of nuclear power plant perform the core control functions of the NSSS (Nuclear Steam Supply System) control systems[2]. The overall system performance is affected by the control equipment because there are some variations of the PI function depending on the suppliers. Up to now, the hardware platform of the NSSS control systems has been supplied from the off-the-shelf equipment with the functionalities proven from the plentiful experiences obtained from many applications.

In one of the APR1400 construction projects, a domestic DCS (Distributed Control System) equipment manufactured by a local supplier has been newly applied to the NSSS control systems by the result of the national wide I&C self-reliance program. It is FOAK (first-of-a-kind) application to the control systems for the APR1400. To get the best optimized PI of the new controller, additional activities have been performed including analysis of the PI control algorithms, function tests, and performance validation tests of the NSSS control systems. The purpose of these activities is to reduce the functional differences between the PI functions of the hardware and the control algorithms used by the system analysis code. In this paper, the analysis of the PI control algorithm and its test method are proposed and the results are discussed.

2. Design and Implementation of PI Controller

The PI controller is theoretically defined as follows:

$$o(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt\right) \qquad (2.1)$$

where, o = output, K = gain, e = setpoint(SV) – process input(PV), and T_i = integral(reset) time. In this section the design activities and the implementation methods of PI controller are described.

2.1 Design Activities

The top tier conceptual design of the NSSS control systems is performed with the system analysis code, KISPAC (KOPEC Integrated Systems Performance Analysis Code). The detailed functions of the NSSS control systems are modeled in the KISPAC with high fidelity. The system behaviors of all modes of power operations related to the NSSS control systems are thoroughly analyzed with the KISPAC to prove the adequacy of control algorithms and to derive well optimized setpoints. Completed this conceptual design, the design specification is prepared for the implementation and it is delivered to the equipment supplier for the manufacturing.

2.2 Implementation Methods

There are two kinds of implementation methods of PI controller which are position and velocity algorithms in digital systems. The PI function may differ from each other depending on the methods of numerical approximation of eq. 2.1.

The formulation of position type algorithms for PI controller is implemented in digital system as follows:

$$\begin{aligned} o_n &= K_p\left(e_n + \frac{\Delta t}{T_i}\sum_{j=1}^{n} e_n\right) \\ &= K_p e_n + \left(I_{n-1} + K_p\frac{\Delta t}{T_i}e_n\right) \qquad (2.2) \\ &= P_n + I_n \end{aligned}$$

where, o_n = output, K_p = gain, e_n = error, T_i = integral time, Δt = scan time, P_n = proportional term, I_n and I_{n-1} = integral terms of current and last scan time. In this method, only the I_{n-1} term needs to be retained in the memory[3].

On the other hand, the formulation of velocity type algorithm for PI controller in digital system is as follows:

$$\begin{aligned} \Delta o_n &= K_p\left(e_n - e_{n-1} + \frac{\Delta t}{T_i}e_n\right) \\ o_n &= o_{n-1} + K_p\left(e_n - e_{n-1} + \frac{\Delta t}{T_i}e_n\right) \qquad (2.3) \end{aligned}$$

Major difference is that the output is calculated based on the increment of each term of the retained data at the last scan. The last error (e_{n-1}) and the last output (o_{n-1}) are required to be saved in the memory[3].

2.3 Types of PI Controllers

For the implementation of eq. 2.1, the new controller is using eq. 2.2 and the KISPAC is using eq. 2.3. It was expected that there may exist functional differences between them so that performance tests and optimization activities have been performed for the new controller to comply with the dynamic behaviors of KIPAC PI.

3. Analysis of Functional Difference

The NPA (Nuclear Plant Analyzer) was used to generate test dynamic data of the NSSS control systems of APR1400 including FWCS (Feedwater Control System), SBCS (Steam Bypass Control System), PLCS (Pressurizer Level Control System), PPCS (Pressurizer Pressure Control System), and RRS (Reactor Regulating System)[1]. The NPA has the system performance validation capability by using the digital communications with the new controller. All inputs and outputs of the NSSS control systems are connected to the NPA via digital communications and the dynamic test of the new controller can be performed by the NPA. Because the NPA has same set of control functions with the KISPAC, it can be used as engineering workbench to perform dynamic tests and optimization of the PI controller.

The dynamic test data of the new controller and the NPA for the same set of operational events were used for the analysis to find out the functional differences. The following operational events were selected for the comparison of test data generated by the new controller and the NPA.

- Load Rejection to House Load
- Loss of Main Feed-water Pump
- Reactor Trip
- Turbine Power 10% Step Decrease (100 to 90%)
- Turbine Power 5%/min Ramp Change (100 to 30%)
- Feed-water Valve Transfer (Increasing Direction)
- Feed-water Valve Transfer (Decreasing Direction)

The test results of the above events matches together except the first case (load rejection to house load).

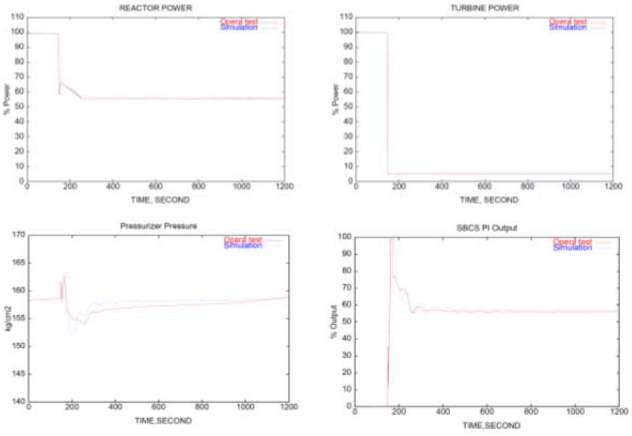

Fig. 1. Test results of the load rejection to house load event to analyze functional difference.

The root cause of the observed difference of fig.1 was analyzed due to the difference of anti-windup functions at the output signal limits. The anti-windup function of the new controller was analyzed as follows:

$$
\begin{aligned}
&If\ (o_n < R_{low}\ )\ o_n = R_{low} \\
&If\ \left(o_n > R_{high}\right)\ o_n = \ R_{high} \\
&If\ (I_n < R_{low})\ \ I_n = R_{low} \\
&If\ \left(I_n > R_{high}\right)\ I_n = R_{high} \qquad (3.1)
\end{aligned}
$$

where, o_n and I_n are the output and integral term of the eq. 2.2, R_{low} is the low limit, and R_{high} is the high limit of the output.

In contrast, the NPA has the anti-windup function like followings.

$$
\begin{aligned}
&If\ (o_{n-1} \leq \ R_{low}\ )\ and\ (e_n \leq 0.0) \\
&\qquad o_n = \ R_{low} \\
&Else\ If\ \ \left(o_{n-1} \geq \ R_{high}\right)\ and\ (e_n \geq 0.0) \\
&\qquad o_n = R_{high} \\
&Else \\
&\qquad o_n = o_{n-1} + K_p\left(e_n - e_{n-1} + \frac{\Delta t}{T_i} e_n\right) \\
&End\ if \qquad\qquad\qquad\qquad\qquad\qquad (3.2)
\end{aligned}
$$

In the event of fig.1, the both of the SBCS PI outputs reached its high limit (100%) right after the event occurrence. Then the PI output of the new controller (red line) drops faster than that of the NPA (dotted blue line) because of the difference of anti-windup functions of eq. 3.1 and 3.2. When the event occurs, the output increases very fast in reaching to the high limit with high gain. However, the integral tem (I_n) of eq. 3.1 slowly reaches to the limit relatively because integral time (T_i) is not high enough to reach to its maximum in this case. By the process feedback of the control output, the process input to the SBCS PI controller reverses its direction before the integral term reaches its high limit and this makes the difference comparing to the NPA.

The performance difference is not acceptable because the new controller function shows less operational margin for the pressurizer pressure in this kind of transients. In addition, considering that a functional difference between the system design and the hardware would result in disadvantages from the viewpoint of the long term technical supports to the site, it is necessary to optimize the new controller to have the same dynamic behaviors as those of the NPA as much as practicable.

4. Modeling of the New Controller

The modeling of the new controller is performed based on the eq. 2.2 and 3.1 by modifying the source code of the NPA. To verify the correctness of modeling, validation test was performed with the event same to fig.1 and the fig.2 shows the test results. As shown in the figure, the dynamic behavior of the modeled PI matches very well with the hardware so that it was successful to get equivalent model of the new controller.

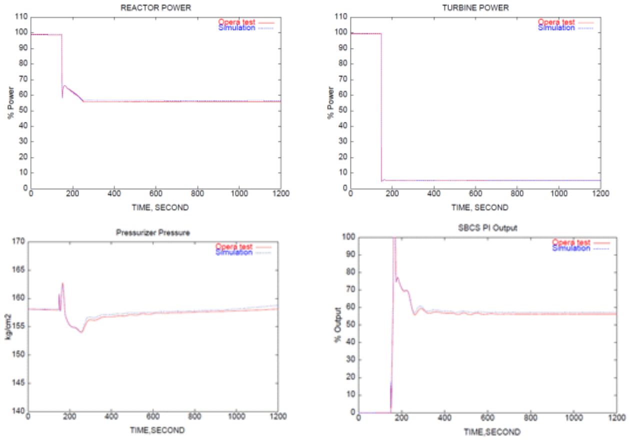

Fig. 2. Test result to verify that new model of the NPA shows equivalent performance to the new controller PI

5. Optimization of PI Controller and Test Results

$$
\begin{array}{l}
If\ (o_{n-1} \le R_{low})\ and\ (e_n \le 0.0) \\
\qquad o_n = R_{low} \\
\qquad I_n = o_n - K_p e_n \\
Else\ If\ \left(o_{n-1} \ge R_{high}\right)\ and\ (e_n \ge 0.0) \\
\qquad o_n = R_{high} \\
\qquad I_n = o_n - K_p e_n \\
Else \\
\qquad I_n = I_{n-1} + K_p \dfrac{\Delta t}{T_i} e_n \\
\qquad o_n = K_p e_n + I_n \\
End\ if \\
\\
If\ (o_n < R_{low})\ o_n = R_{low} \\
If\ \left(o_n > R_{high}\right)\ o_n = R_{high} \qquad (5.1)
\end{array}
$$

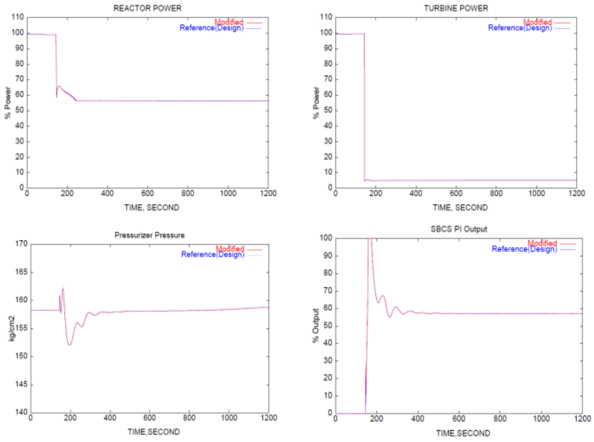

Fig. 3. Test result to prove that eq. 5.1 has equivalent performance comparing to the designed PI function.

By using the PI model of the NPA described in section 4, optimization of the algorithm was performed. After several iterations of functional modifications and validation tests, eq 5.1 was derived as the final control algorithm.

The fig. 3 shows that, if the PI algorithm of the new controller is modified as eq. 5.1, the system performances become very similar to those of the PI functions used by the system designer.

6. Conclusions

Because the implementation method of the PI controller of the newly applied DCS for the APR1400 is different to the system design code, the test result of load rejection to house load event of the new controller did not match with the simulation result of the system design code.

To get equivalent functionality, the PI function of the new controller was modeled with the NPA. The modeling was proven to be successful with the validation test.

By using this model, optimization of the PI algorithm has been performed to get the equivalent functionality of the PI function as intended. After some modifications, eq. 5.1 has been derived and this algorithm shows acceptable performances.

If the control algorithm of eq. 5.1 is applied to the newly applied controller, it is expected that the control system behaviors will closely match with the system design code. As the system designer's point of view, the control systems become more predictable and it would be beneficial to perform technical supports for the plant operations.

REFERENCES

[1] S.C. Jeong, I.H. Song, G. C. Lee, and H.Y. Lim, Development of Performance Validation Tool for the NPCS for APR1400, ANS Annual Meeting, June 11-15, 2017.

[2] M.H. Ahn, Design Specification for NSSS Process Control System (proprietary), 2017.

[3] Karl Johan Astrom, Control System Design, 2002.