[UCI]I804:24011-200000264368

2014년 2월

석사학위논문

Development of Smart Measurement Techniques for Early Diagnosis of Safety-critical Systems and Equipment

조 선 대 학 교 대 학 원

원 자 력 공 학 과

박 순 호

Development of Smart Measurement Techniques for Early Diagnosis of Safety-critical Systems and Equipment

안전필수계통, 설비의 상태 조기 진단을 위한
스마트 측정 기술 개발

2014년 2월 25일

조 선 대 학 교 대 학 원

원 자 력 공 학 과

박 순 호

Development of Smart Measurement Techniques for Early Diagnosis of Safety-critical Systems and Equipment

지도교수 나 만 균

이 논문을 공학 석사학위신청 논문으로 제출함

2013년 10월

조 선 대 학 교 대 학 원

원 자 력 공 학 과

박 순 호

박순호의 석사학위 논문을 인준함

위원장 조선대학교 교 수 정 운 관 (인)

위 원 조선대학교 교 수 이 경 진 (인)

위 원 조선대학교 교 수 나 만 균 (인)

2013년 11월

조 선 대 학 교 대 학 원

CONTENTS

List of Figures

List of Tables

초 록

안전필수계통, 설비의 상태 조기 진단을 위한 스마트 측정 기술 개발

박 순 호
지도 교수 : 나 만 균
원자력공학과
조선대학교 대학원

원자력발전소는 설계기준사고가 발생하더라도 원자로를 안전한 상태로 유지 관리될 수 있도록 공학적 안전설비가 설치되어있다. 그러나 TMI, 체르노빌 사고에서 보듯이 사소한 고장 및 오동작 그리고 운전원의 인적 실수가 결합하여 엄청난 사고가 발생하는 경우를 경험한 바 있다.

최근 일본 후쿠시마 원자력발전소 사고에서 보듯이 외부사건에 의한 중대사고 발생 시 원자로 상태를 정확하게 알아내지 못하는 상황에 처하게 되어, 사고 평가 및 시나리오 진행에 대해 전혀 예측하지 못하는 상황에 도달하게 되었다. 이는 필수안전기능을 수행하는 안전필수계통 혹은 설비의 상태를 지시해 주는 계측기가 외부의 영향(쓰나미)으로 인한 침수의 여파로 소 내외 정전을 일으켜 무용지물이 되었기 때문이다. 만일 사고 진행, 시나리오 및 원자로 상태를 정확히 예측할 수 있었다면 상황에 맞는 적절한 대처를 통하여 더 큰 사고로의 진행을 미연에 방지할 수 있을뿐더러 사고 관리를 효율적으로 수행할 수 있었을 것이다. 따라서 정보기술에서 발전되어온 스마트센싱 기술을 활용하여 원자력발전소의 주요 상태 진단에 적용하였다.

안전관련 변수는 원자력발전소의 상태는 확인하는데 매우 중요하다. 특히, 원자로 수위는 원자로 냉각 및 시나리오 진행을 나타내는 중요한 시점(원자로 노심 노출, 원자로 용기 파손 등)에 직접적인 영향을 미친다. 이에 따라, 본 연구에서는 원자로 수위를 측정 할 수 없는 심각한 사고 조건에서 FNN 방법과 GMDH 방법을 이용하여 원자로 수위 예측 모델을 개발하였다. 예측 모델은 학습 데이터를 이용하여 개발 하고 독립적

인 시험 데이터를 이용하여 검증을 수행하였으며, 모델 개발과 검증을 위한 DB는 MAAP4 코드를 이용하여 한국표준형원전(OPR1000)을 대상으로 하는 시뮬레이션을 통해 구축 하였다. 원자로 수위 예측 성능은 매우 만족스러우며, 입력신호에 오류가 존재할 경우 수위 예측에 미치는 영향을 평가하기 위해 인위적인 오류가 포함된 데이터를 사용하였다.

개발된 예측 모델은 센서의 건정성을 보장할 수 없는 심각한 사고 상황에서 원자로 수위를 정확하게 예측할 수 있었으며, 다양한 데이터를 사용하여 최적화한다면 더욱 정확하게 원자로 수위를 예측할 수 있을 것이다. 또한, 안전에 관련된 다양한 변수들에 대한 스마트센싱 시스템이 구축된다면 사소한 고장이 대형사고로 발전하는 것을 방지하고, 여러 초기사건으로부터 중대사고로 전개되어 계측기의 신뢰성이 의심되는 상황에서 원자력발전소의 주요 상태 조기 진단을 통하여 사고 관리를 효과적으로 수행할 수 있을 것으로 기대된다.

Ⅰ. Introduction

After the Fukushima, nuclear power plant accident occurred by the East Japan Great Earthquake, the public's concern and interest in the safety of nuclear power plants has increased considerably. The cause of these concerns and interest is because the operator does not quickly check the status of the plant in an incident or accident situations or respond appropriately to each situation.

The status of a nuclear power plant can be confirmed from the safety-related parameters (reactor vessel water level, neutron flux, pressurizer pressure, pressurizer water level, steam generator pressure, steam generator water level, etc.). In particular, to check the status of a nuclear power plant and take a proper action, it is very important to measure the safety-related parameters for a very short period in the initial event conditions that can lead to a serious accident, such as a loss of coolant accident (LOCA) and steam generator tube rupture (SGTR). In particular, the reactor vessel water level is essential information for confirming the cooling capability of the nuclear reactor core, to prevent the reactor core from melting down and to manage severe accidents effectively. In addition, it cannot be confirmed that the reactor vessel water level is being measured properly in severe accidents, where the reactor core integrity is uncertain. Therefore, it is important to predict the reactor vessel water level to make provisions against severe accidents.

Many artificial intelligence techniques have been applied successfully to nuclear engineering areas, such as signal validation [1]-[3], plant diagnostics [4]-[5], event identification [6]-[9], etc. In this paper, the FNN and GMDH model were proposed to predict the reactor vessel water level, which has a direct impact on the important times (time approaching the core exit temperature exceeding 1200°F, core uncovery time, reactor vessel failure time, etc.). To predict the water level, the break size and other measured signals were used. The break size is not a measured variable. Instead, it is a predicted variable using the trend data for a

short time early in the event proceeding to a severe accident. The LOCA classification algorithm for determining the LOCA position and break size prediction algorithm was explained in previous papers [10]-[12]. Because the break size can be predicted accurately, the break size can be used as an input variable for predicting the reactor vessel water level.

The FNN and GMDH are a data-based model that requires data to develop and verify it. Because the real severe accident data does not exist, it is essential to obtain the data required in the proposed model using numerical simulations. This data was obtained by simulating severe accident scenarios for the Optimized Power Reactor 1000 (OPR1000) using MAAP4 code [13].

Ⅱ. Prediction of the reactor vessel water level

A. FNN(Fuzzy neural network) method

1. FIS(Fuzzy inference system)

In general, the conditional rule, which is described as if/then rule, is used in the FIS, and is composed of a pair of the conditions and conclusions [14]. The fuzzy inference engine, as shown in Fig. 1, uses fuzzy if/then rules to determine the mapping from fuzzy sets in the input universe of discourse to fuzzy sets in the output universe of discourse based on a fuzzy logic principle. A fuzzifier needs to be added to the input because the inputs of the FIS are real-valued variables. The fuzzifier maps the crisp points in to the fuzzy sets in .The membership function in the FIS maps each element of V to a continuous membership value between zero and one. The membership function has no restriction of shape. In general, the Gaussian, triangular, trapezoid and bell-shaped functions are used in the formula of the membership function. In addition, because the reactor vessel water level is a real value, the FIS output should be a real value that requires a defuzzifier to the FIS output. On the other hand, an FNN consists of a FIS and its neuronal training system. To predict the water level in the reactor vessel using FNN, it is important to find the optimal input variables among several variables, and are used to predict the water level in the reactor vessel.

In this study, instead of the Mamdani-type FIS [14], which requires a defuzzifier in the output unit, the Takagi-Sugeno-type FIS [15], which does not require the defuzzifier shown in Fig. 1 because its output value is real, were used. In the FIS, an arbitrary $i-th$ fuzzy rule can be expressed as follows(first-order Takagi-Sugeno-type):

$$If\ x_1(k)\ is\ A_{i1}\ AND \cdots\ AND\ x_m(k) is\ A_{im}, \tag{1}$$

$$then\ y_i(k)\ is\ f_i(x_1(k),\cdots,x_m(k))$$

where

$x_1,\cdots,x_m$: input values of FIS

$A_{i1},\cdots,A_{im}$: membership functions

y_i : output of the $i-th$ fuzzy rule

q_{ij} : weight of the $i-th$ rule and the $j-th$ fuzzy input

r_i : bias of the $i-th$ fuzzy rule

n : number of fuzzy rules

m : number of input values

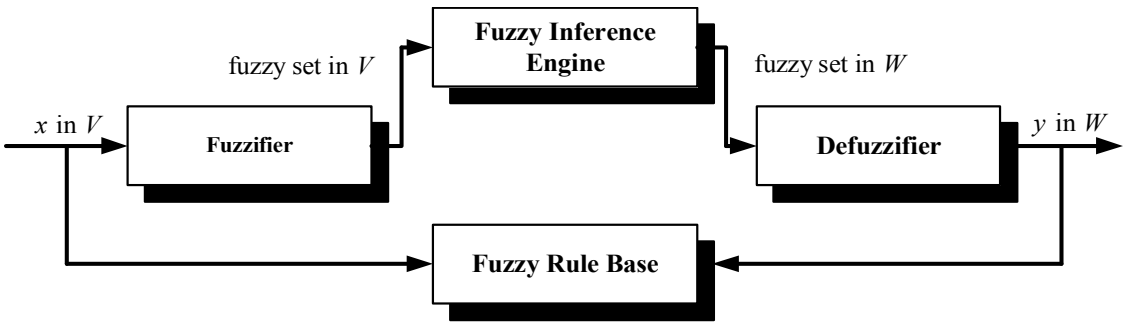


Fig. 1. Fuzzy inference system(Mamdani-type FIS)

The number of N input and output training data of the fuzzy model $z^T(k)=(\mathbf{x}^T(k),y(k))$ (where $\mathbf{x}^T(k)=(x_1(k),x_2(k),\cdots,x_m(k))$ and $k=1,2,\cdots,N$) were assumed to be available and the data point in each dimension was normalized. Generally, there is no special restriction on the shape of the membership functions. In this paper, the symmetric Gaussian membership function was used to reduce the number of the parameters to be optimized.

$$A_{ij}(x_j(k)) = e^{-\frac{(x_j(k)-c_{ij})^2}{2s_{ij}^2}} \tag{2}$$

In Eq. (1), the function, $f_i(x(k))$, is expressed as the first-order polynomial of input variables and the output of each rule is expressed as follows:

$$f_i(\mathbf{x}(k)) = \sum_{j=1}^{m} q_{ij} x_j(k) + r_i \tag{3}$$

The FIS expressed as Eq. (1) is called the first order Takagi-Sugeno-type [15] fuzzy model because the arbitrary $i-th$ rule output, f_i, is a real value and is expressed as the first-order polynomial for the inputs. The output $\hat{y}(k)$ of the FIS is calculated by summing the weighted fuzzy rule outputs y_{wi} as follows:

$$\hat{y}(k) = \sum_{i=1}^{n} y_{wi}(k) \tag{4}$$

where

$$y_{wi}(k) = \overline{w}_i(k) f_i(\mathbf{x}(k)) \tag{5}$$

$$\overline{w}_i(k) = \frac{w_i(x(k))}{\sum_{i=1}^{n} w_i(x(k))} \tag{6}$$

$$w_i(k) = \prod_{j=1}^{m} A_{ij}(x_j(k)) \tag{7}$$

Finally, the output $\hat{y}(k)$ is expressed as the vector product as follows:

$$\hat{y}(k) = \mathbf{w}^T(k)\mathbf{q} \tag{8}$$

where

$\mathbf{q} = [q_{11} \cdots q_{n1} \cdots\cdots q_{1m} \cdots q_{nm} \;\; r_1 \cdots r_n]^T$

$\mathbf{w}(k) = [\overline{w}_1(k)x_1(k) \cdots \overline{w}_n(k)x_1(k) \cdots\cdots \overline{w}_1(k)x_m(k) \cdots \overline{w}_n(k)x_m(k) \;\; \overline{w}_1(k) \cdots \overline{w}_n(k)]^T$

The predicted outputs for a total of N input and output data pairs induced from Eq. (8) can be expressed as follows:

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{q} \tag{9}$$

where

$\hat{\mathbf{y}} = [\hat{y}(1) \; \hat{y}(2) \cdots \hat{y}(N)]^T$

$\mathbf{W} = [\mathbf{w}(1) \; \mathbf{w}(2) \cdots \mathbf{w}(N)]^T$

The vector $\mathbf{q}$ is called a consequent parameter vector, and the matrix $\mathbf{W}$ consists of input data and membership function. The output values of FIS are expressed in a matrix, $\mathbf{W}$, of $N \times (m+1)n$ dimensions and a parameter vector $\mathbf{q}$ of $(m+1)n$ dimensions.

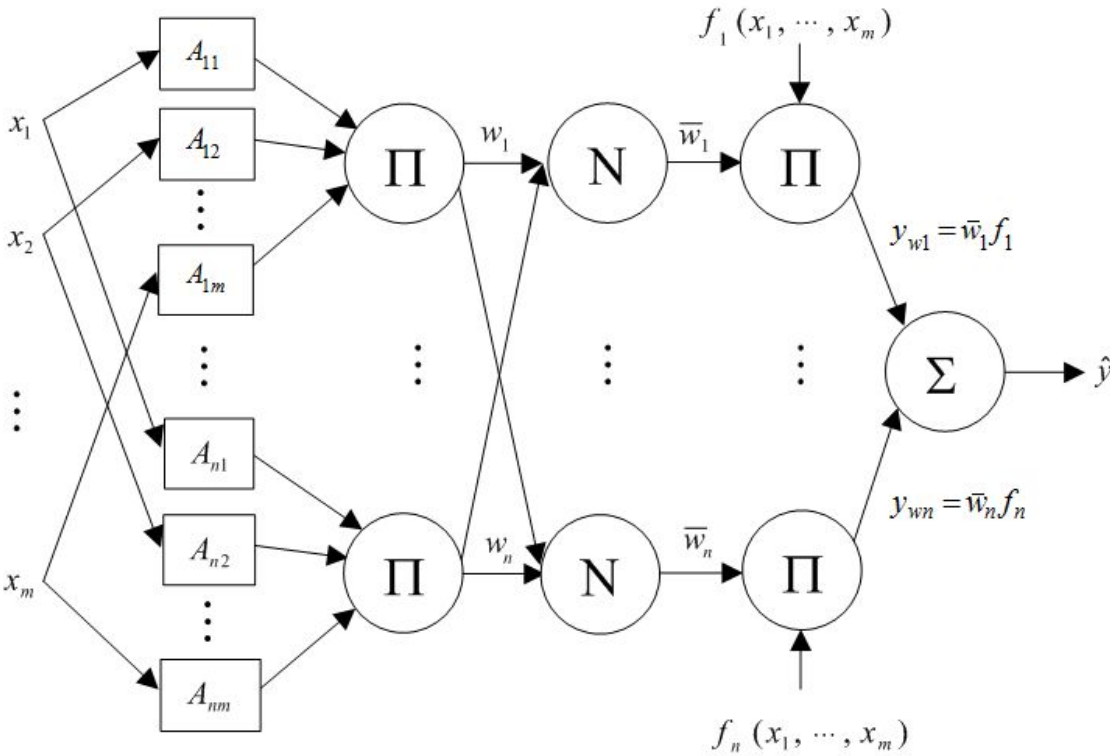


Fig. 2. Fuzzy neural network(FNN)

Fig. 2 describes the calculation structure of the FNN model. The symbols, $\prod$ and N, indicate multiplication and normalization calculations, which are expressed as Eq. (7) and (6), respectively. The second symbol, $\prod$, is expressed as Eq. (5), and the symbol Σ is expressed as Eq. (4), which is the summation of the weighted fuzzy rule outputs.

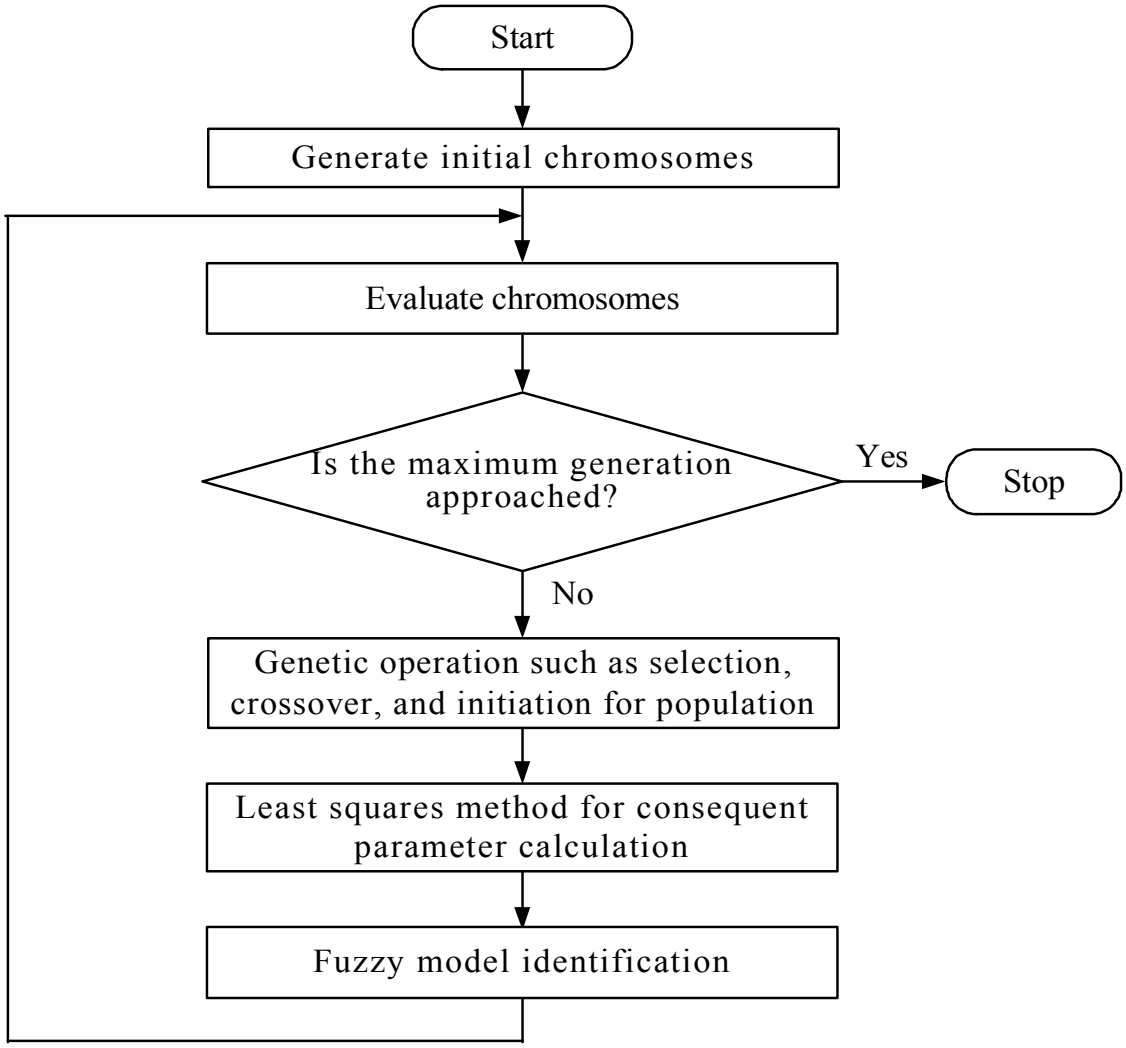


Fig. 3. Optimization procedure of the FNN model

Fig. 3 shows the optimization procedure of a FNN model that is a FIS combined with its neuronal training system. The optimization procedure optimizes each antecedent and consequent parameters using both the genetic algorithm and least square method. In genetic algorithms, the variables to be optimized are encoded within the chromosome, and the superiority regarding each chromosome is judged by the fitness function. If the antecedent parameters are determined using a genetic

algorithm through selection, crossover and mutation, the resulting parameters appear like Eq. (9) as a first-order combination. Therefore, the consequent parameters can be calculated easily using the least squares method.

2. Training of fuzzy inference model

The FNN model, which was developed to predict the reactor vessel water level, was designed by training from the given data. The proposed model should also be optimized to maximize the prediction performance. For this purpose, a genetic algorithm was used in this study. Actually, the genetic algorithm is the most useful method for solving the optimization problem for a range of purposes [16], [17].

The genetic algorithm uses a fitness function that assign each chromosome the score (degree of optimization) in the current population, and solves the optimization problem by the process of the laws of nature, such as selection, crossover and mutation operators. To predict the signals using AI techniques, the prediction error makes a difference depending on how the input signals were selected. In addition, eliminating the unnecessary signals can reduce the time for training because it simplifies the structure of the AI technique. On the other hand, even if the proper input signals were selected, the prediction performance is affected by how the time-step data is utilized. Therefore, in this study, the training data, which contained good information using the Subtractive Clustering (SC) technique, was selected from all acquired data [18].

The data points generally form clusters in high dimensional data space, and the FNN model is trained using the data points, which are located in the center of each cluster that is slightly different from the physical center of the cluster, because the center of each cluster has the most information. The SC technique uses the following function as a measure of the potential of each data, and it can be defined as a function of the Euclidean distance to all other input values [18].

$$P(k) = \sum_{j=1}^{N} e^{-4\|\mathbf{x}(k)-\mathbf{x}(j)\|^2/r_a^2}, \quad k = 1,2,\cdots,N \tag{10}$$

In Eq. (10), r_a is the radius of neighboring parts and it affects the potential significantly. Through this equation, the potential of the data points is high when surrounded by a large volume of neighboring data. The data point with the highest potential was selected as the first cluster center. Let $\mathbf{x}^*(1)$ be the location of the first cluster center and $P^*(1)$ be its potential value. The potential of each data point is revised by the following formula:

$$P(k) := P(k) - P^*(1)e^{-4\|\mathbf{x}(k)-\mathbf{x}^*(1)\|^2/r_b^2}, \quad k = 1,2,\ldots,N, \tag{11}$$

where r_b is also the radius, which is normally greater than r_a in Eq. (10). As shown in Eq. (11), the data points near the first cluster center will have a greatly reduced potential, and are unlikely to be selected as the next cluster center. When the potential of all data points is revised according to Eq. (11), the datum with the highest remaining potential is selected as the second cluster center, $\mathbf{x}^*(2)$. Eq. (11) is repeated by substituting $P^*(1)$ and $\mathbf{x}^*(1)$ with $P^*(i)$ and $\mathbf{x}^*(i)$, respectively, until the inequality $P^*(i) < \varepsilon P^*(1)$ is true or the required number of training data is obtained.

The antecedent parameters of the membership functions were optimized using a genetic algorithm and the input signals used were selected using the correlation coefficient matrix of the input/output signals. In addition, the least squares method was used to calculate the consequent parameters. Many optimization methods use some transition law to determine the next optimal point. This moves from one point in space to the next point. On the other hand, these point-to-point methods can be dangerous because the probability of finding the wrong peak in the search space with many peaks is quite high. By contrast, the genetic algorithm is

ascending many peaks in parallel based on the abundant database of many points. Therefore, the chances of finding a false peak are much lower than with point-to-point methods, and there is no concern of being stuck in a local optimal point [16]-[17].

In this study, the training data was used to calculate the antecedent parameters of the fuzzy rules. The test data was used to check the developed model and is different from the training data set. The fitness function in the following equation was intended to minimize the maximum error and RMS error:

$$F = \exp(-\mu_1 E_1 - \mu_2 E_2) \tag{12}$$

where

$$E_1 = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k))^2}$$

$$E_2 = \max(y(k) - \hat{y}(k))$$

Variable y means the actual measured value, and $\hat{y}$ is its value predicted using the FNN model. If the antecedent parameters are fixed by the genetic algorithm, the results of the proposed model can be explained by the development of some functions. Therefore, the least squares method was used to determine the consequent parameter of fuzzy rules. The consequent parameter, $\mathbf{q}$, was chosen to minimize the objective function. This consists of the square error between the actual value y and its predicted value $\hat{y}$, and it is expressed as follows:

$$J = \sum_{k=1}^{N_t}(y(k) - \hat{y}(k))^2 = \sum_{k=1}^{N_t}(y(k) - w^T(k)q)^2 = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^2 \tag{13}$$

where

$\mathbf{y} = [y(1)\ y(2) \cdots\ y(N_t)]^T$ and $\hat{\mathbf{y}} = [\hat{y}(1)\ \hat{y}(2) \cdots\ \hat{y}(N_t)]^T$

In Eq. (13), N_t is the number of training data. A solution for minimizing the above objective function can be obtained using the following equation:

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{q} \tag{14}$$

To solve the parameter vector, $\mathbf{q}$, the inverse matrix must exists in a matrix, $\mathbf{W}$. On the other hand, there is no inverse matrix generally. Therefore, the pseudo-inverse of the matrix $\mathbf{W}$ was used. The parameter vector, $\mathbf{q}$, is easy to solve from the pseudo-inverse as shown below.

$$\mathbf{q} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{y} \tag{15}$$

The parameter vector, $\mathbf{q}$, can be calculated from a series of input and output data pairs.

B. GMDH(Group method of data handling) method

In order to solve the system problem such as control, monitoring, prediction, diagnosis and so on, a lot of mathematical methods have been studied.

The GMDH method is one of them. The GMDH method which is one of the data-driven models such as ANN (Artificial Neural Network) can be used for the break size prediction in this paper. Data-driven models have many advantages of easy implementation and accuracy, and famous for superior capability in modelling complex systems.

In this paper, the GMDH method has been used to develop a model for the water level prediction in the reactor vessel.

1. Basic GMDH algorithm

The GMDH algorithm is a way of finding a function that well expresses a dependent variable from independent variables. This method can find a correlation in the data automatically to improve the prediction accuracy and select the optimal structure of the model. The GMDH algorithm is similar to multiple regression model, but it uses the data structure. The data set is divided into three subsets. The reason of dividing is to prevent over-fitting and maintain model regularization through cross-validation. Fig. 4-5 show the GMDH data structure and the GMDH structure.

$$
\begin{array}{l}
\boxed{\begin{array}{lcccccc}
\text{training data set} & \left\{\begin{array}{ccccc}
x_{11} & x_{12} & \cdots & x_{1m} & y_1 \\
x_{21} & x_{22} & \cdots & x_{2m} & y_2 \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
x_{l1} & x_{l2} & \cdots & x_{lm} & y_l
\end{array}\right. & & & & \\
\text{verification data set} & \left\{\begin{array}{ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
x_{n1} & x_{n2} & \cdots & x_{nm} & y_n
\end{array}\right. & & & & \rightarrow \text{development data} \\
 & \begin{array}{ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
x_{K1} & x_{K2} & \cdots & x_{Km} & y_K
\end{array} & & & &
\end{array}} \\
\text{test data set} \quad \left\{\begin{array}{ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
x_{N1} & x_{N2} & \cdots & x_{Nm} & y_N
\end{array}\right.
\end{array}
$$

Fig. 4. GMDH data structure

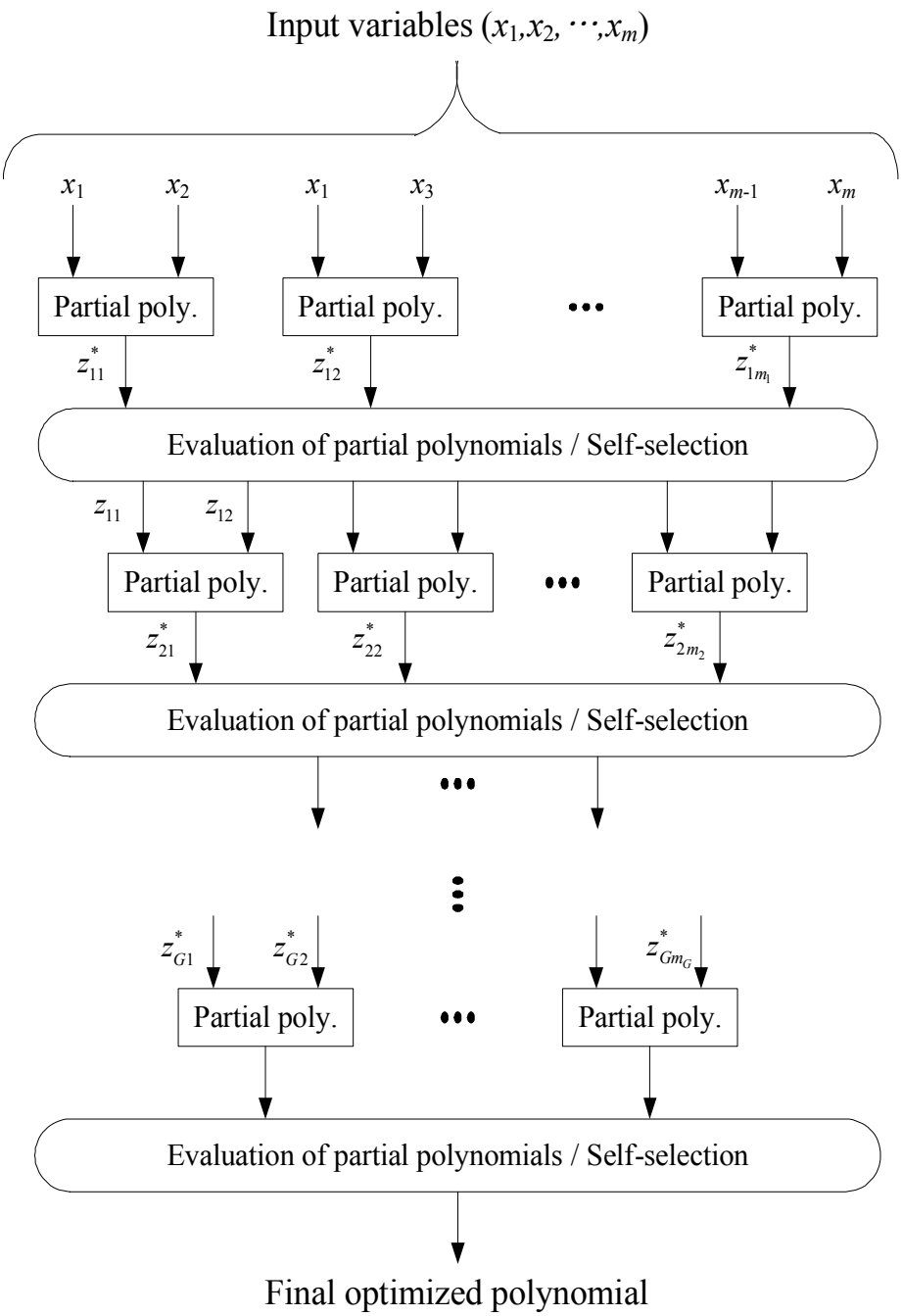


Fig. 5. GMDH structure

The original GMDH method employed the following general form at each level of the successive approximation:

$$y = f(x_i, x_j) = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_i x_j \tag{16}$$

The coefficient parameters of the reference function which is written above such as $A, B, \ldots, F$ can be obtained by using a least square method in an arbitrary pair (x_i, x_j) from independent variables $\mathbf{x} = (x_1, x_2, \cdots, x_m)$. This method takes a form of hierarchical polynomial regression network to model various complex input-output relationships. However, more complicated function forms can be used such as ratio terms (x_i / x_j), trigonometric terms ($\sin(cx), \cos(cx)$), exponential terms ($\exp(-cx)$) and so on in accordance with complexity of the system. The GMDH algorithm uses the Kolmogorov-Gabor form of a high-order polynomial. The Kolmogorov-Gabor form that is called as Ivakhnenko polynomial can be expressed as follows:

$$y = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_{ijk} x_i x_j x_k \ldots \tag{17}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_m)$ is an input vector and $\mathbf{a} = (a_0, a_i, a_{ij} a_{ijk}, , \cdots)$ is a coefficient vector that is a weight vector of Kolmogorov-Gabor polynomial. The GMDH algorithm can determine the structure of the model and also calculate the system output of the most important input simultaneously. This uses the composition of the lower-order polynomials mentioned above, which means that the GMDH algorithm amalgamates lower order polynomials at each generation to reach the subsequent generation. This process continues until the GMDH model begins to show over-fitting in training or exceeds the maximum calculation time. If an evaluation value (R) is greater than a reference value, the regression equation is fallen behind. Otherwise, the regression equation is survived. The survived

regression equation value is used as a training data of the new generation. This process is conducted about all possible pairs of independent variables. The descendant with the smallest evaluation value in the evaluation of this generation is selected as the optimum fit. If the smallest evaluation value of the current generation is smaller than that of the previous generation, the above process is performed repeatedly. When over-fitting of the evaluation value ($R_{\min}^{G}$) is found through alternation of generation, the process is stopped. That is, if the smallest evaluation value of the current generation is larger than that of the previous generation, the process is stopped.

As shown in Fig. 6, if over-fitting is found, the process of the algorithm is stopped and the optimum fit of the previous generation is selected as the optmized model that predicts the break size.

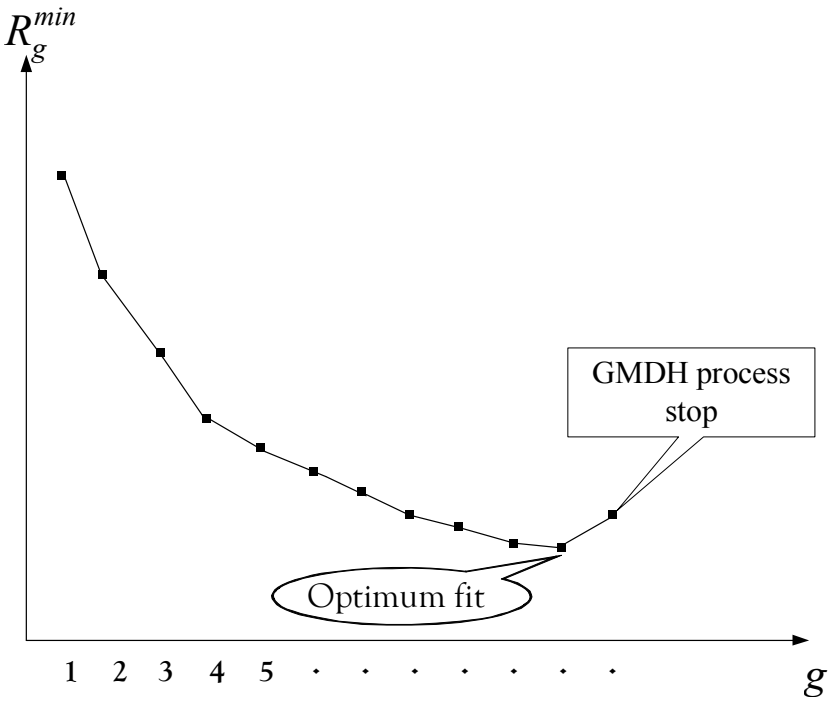


Fig. 6. Evaluation value of each generation

2. Main implementation steps

The GMDH algorithm generates and tests all input-output combinations. Each element in the system that is indicated as a rectangle box in Fig. 5 executes a function of two inputs. The coefficient parameters of Eq. (17) are decided by using a normal least square method, and the variables of the elements are calculated. A threshold value for comparison with the evaluation value in each generation decides whether the outputs of the elements in a generation are acceptable. The output of an element is eliminated in a current generation when the result is larger than the threshold value. Those variables or elements that are useful for predicting the proper output are used at the next generation. The generations are repeated until the satisfactory results are obtained. This process is similar to Darwin's theory. The detailed main implementation steps are given below.

First step, construct each of input and output variable or data of the system. The data structure is modeled and divided into the training and checking data sets, and preprocess the data to normalize them.

Second step, choose the external inputs to the GMDH network. Calculate the regression polynomial parameters for each pair of input variables involved in the training data set using the least square method. Calculate the $m(m-1)/2$ high-order variables in place of the original input variables $x_1, x_2, \cdots, x_m$ in order to predict the output.

Third step, the algorithm designs a group of new variables $(m_g = m_{g-1}(m_{g-1}-1)/2)$ in the previous step. Here, m_g is the number of input variables for generation (g). A criterion is used to evaluate the new variables in the generation g and is related with the error for the checking data, which is defined as follows:

$$r_j^2 = \frac{\sum_{i=l+1}^{n} (y_i - z_{ij})^2}{\sum_{i=l+1}^{n} y_i^2} \quad \text{for } j = 1, 2, \ldots, m_g \tag{18}$$

Last step, when over-fitting is found through checking, the above mentioned process is stopped. If the generation continues, the model will become over-fitted. The polynomial with the minimum error criterion is selected as the final approximate model. Otherwise, the above steps are repeated.

At the end of the GMDH algorithm, regression parameters are stored. The estimated coefficient for the high-order polynomial is determined by tracing back the GMDH structure until it reaches the original variables $x_1, x_2, \cdots, x_m$. As shown in Fig. 7, the tree structure with the optimum fit at the top is called an Ivakhnenko Tree.

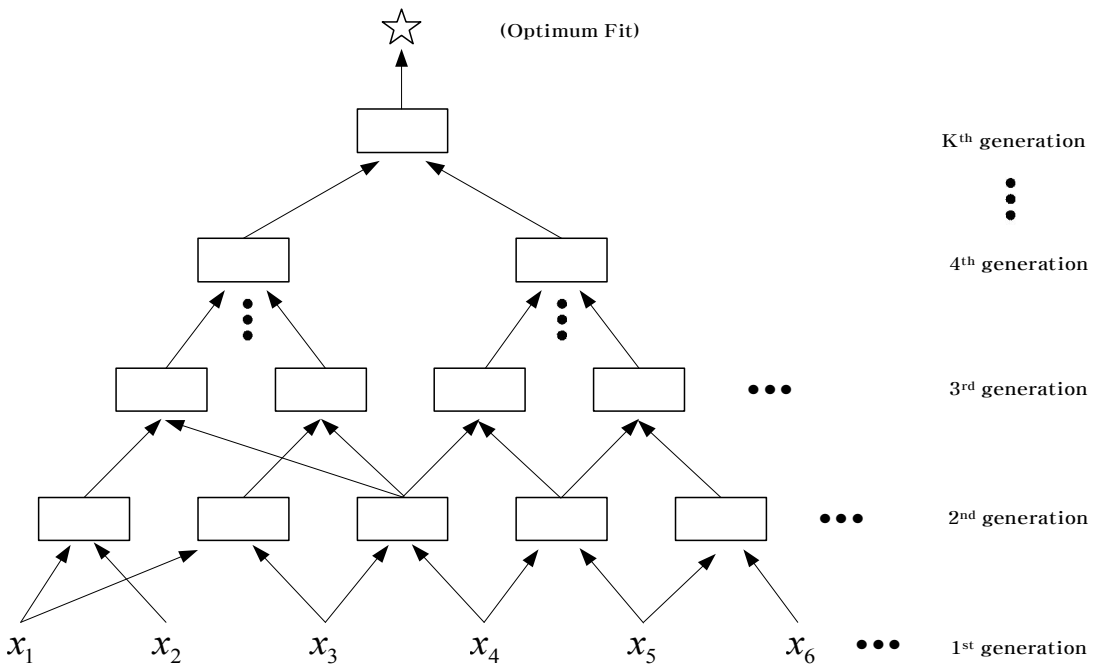


Fig. 7. Ivakhnenko tree

Ⅲ. Accident simulation data

The proposed model was applied to predict the water level in the reactor vessel. To train and independently test a proposed model, it is essential to obtain the data using numerical simulations because there is little real accident data. Therefore, the training and test data of the proposed model was acquired by simulating the severe accident scenarios using the MAAP4 code regarding the OPR1000 nuclear power plant.

The simulation data was divided into the break position and break size of the loss of coolant accident (LOCA). The break position was divided into hot-leg LOCA, cold-leg LOCA and SGTR, and the break size was divided into a total of 270 steps. In addition, the simulations were performed under the conditions that the Safety Injection System (SIS) does not work properly. In accidents concerned with LOCAs, because the LOCA position and size are not detected, they must be identified and predicted. The LOCA position was identified completely and the break size was predicted accurately in previous studies [10]-[12], with an approximately 1% error level. Therefore, the break size signal, which is an input signal to the FNN model, was assumed to be predicted from the algorithms of previous studies.

Through the simulations, a total of 810 cases of severe accident scenarios were obtained. This data was composed of 270 pieces of hot-leg LOCA, 270 pieces of cold-leg LOCA and 270 pieces of SGTR.

Ⅳ. Verification of the proposed model

In severe accident situations, the main concern is whether or not there is sufficient coolant in the reactor core, which is described as the reactor vessel water level. The input variables for predicting the reactor vessel water level are the elapsed time after reactor shutdown, the predicted break size and the pressurizer pressure. These input variables are strongly correlated with the output variable of the reactor vessel water level.

A. FNN model

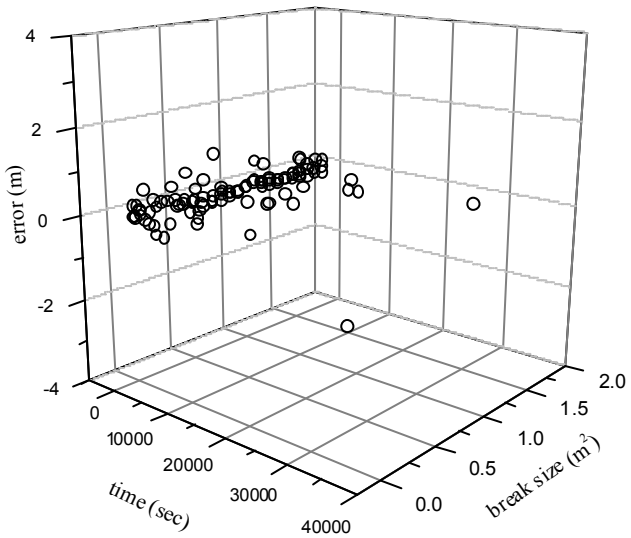
The parameter values used are concerned with the genetic algorithm and the FIS are as follows:

$n = 30$: number of fuzzy rules

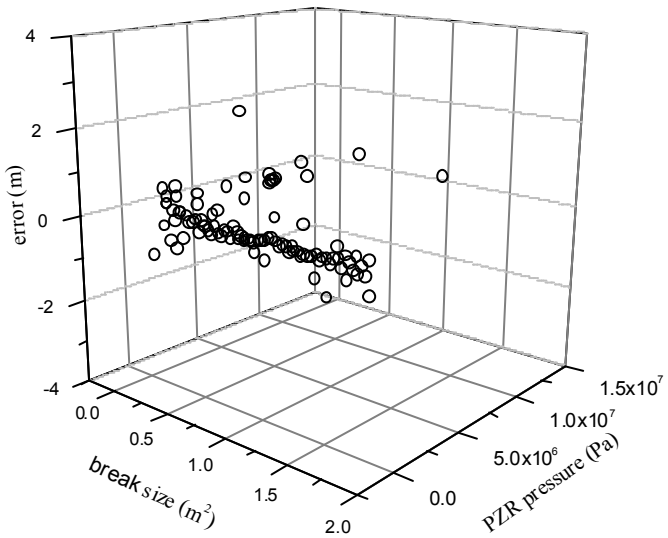
crossover probability = 100%

mutation probability = 0.05%

Fig. 8-10 show the predicted reactor vessel water levels and their errors for the test data in the hot-leg LOCA, cold-leg LOCA, and SGTR situations, respectively. The test data is different from the data used to develop the FNN model and consists of elapsed time, predicted break size, pressurizer pressure, and reactor vessel water level. In this study, 100 data points in each LOCA such as hot-leg LOCA, cold-leg LOCA and SGTR were selected as test data points. As shown in Fig. 8 and Table 1, the prediction errors of the test data for hot-leg LOCA are inside the 1.93m error band and their RMS error is 0.34m.

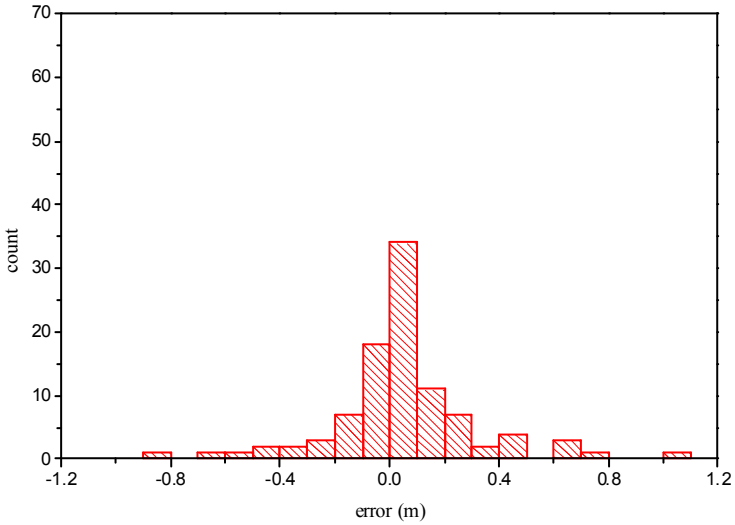


(a) reactor vessel water level error versus time and break size

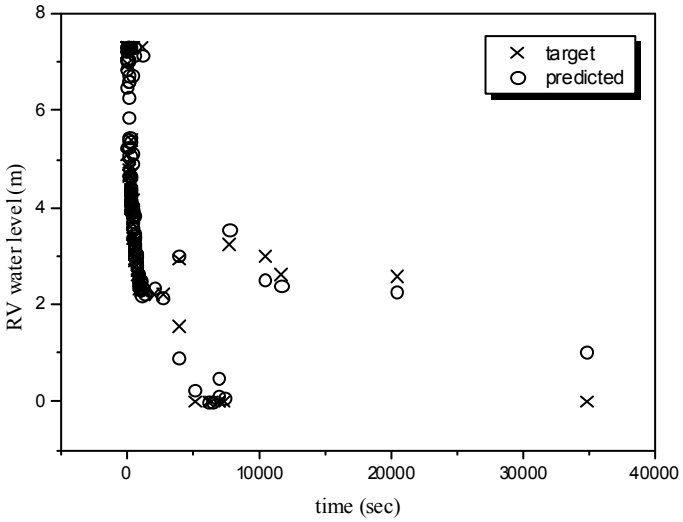


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 8. Prediction performance of the FNN model in hot-leg LOCA

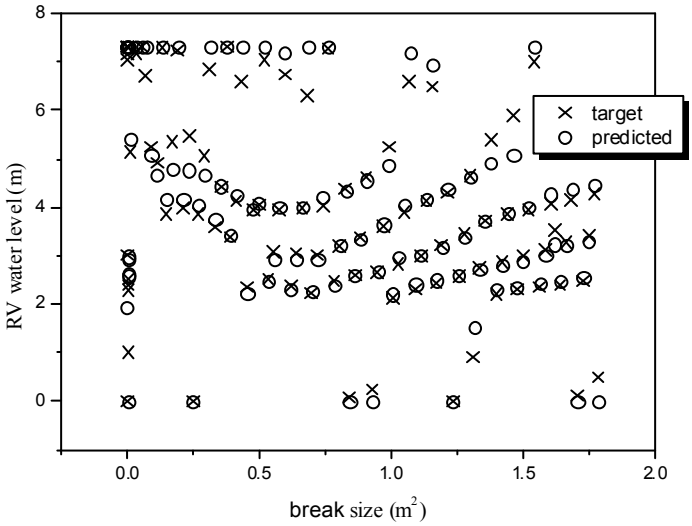


(c) Error histogram

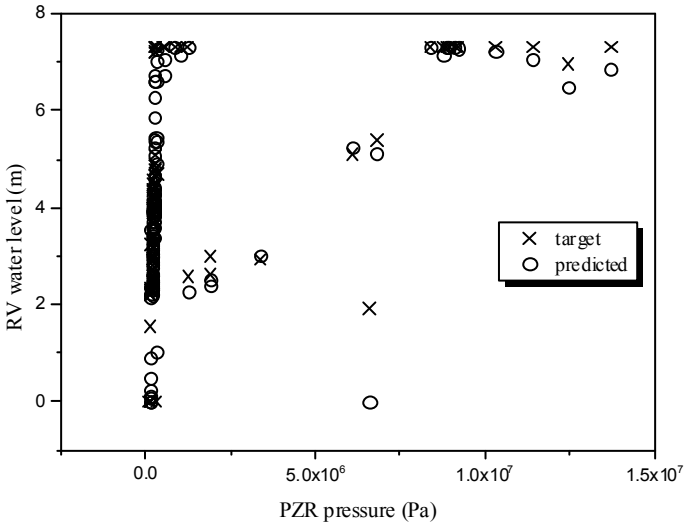


(d) reactor vessel water level versus elapsed time

Fig. 8. Prediction performance of the FNN model in hot-leg LOCA(continued)



(e) reactor vessel water level versus break size



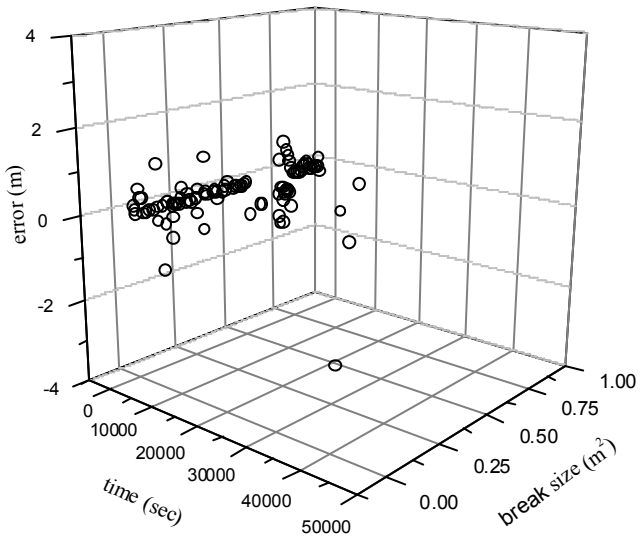
(f) reactor vessel water level versus PZR pressure

Fig. 8. Prediction performance of the FNN model in hot-leg LOCA(continued)

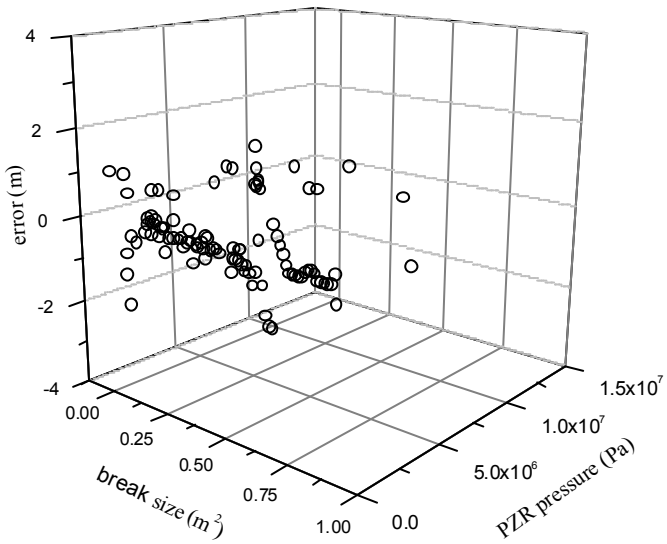
TABLE 1. Performance of the FNN model in hot-leg LOCA

Break position	Training data		Test data	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Hot-leg	4.9418	0.2649	1.9270	0.3447

As shown in Fig. 9 and Table 2, the prediction errors of the test data for the cold-leg LOCA are inside the 2.01m error band and their RMS error is 0.45m.

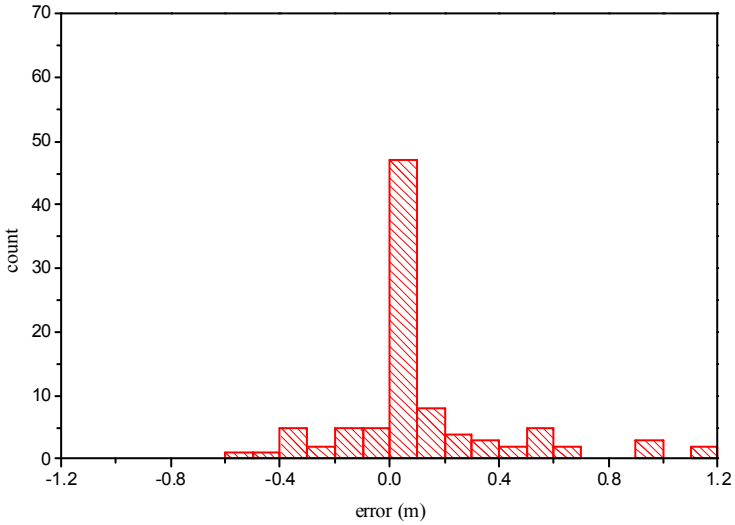


(a) reactor vessel water level error versus time and break size

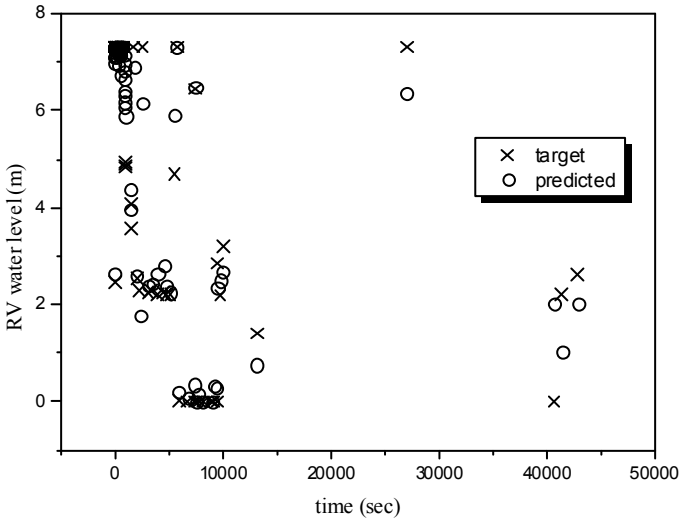


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 9. Prediction performance of the FNN model in cold-leg LOCA

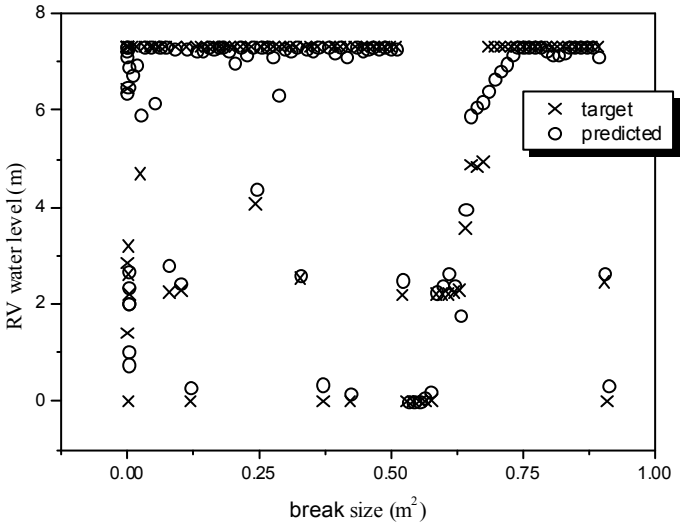


(c) Error histogram

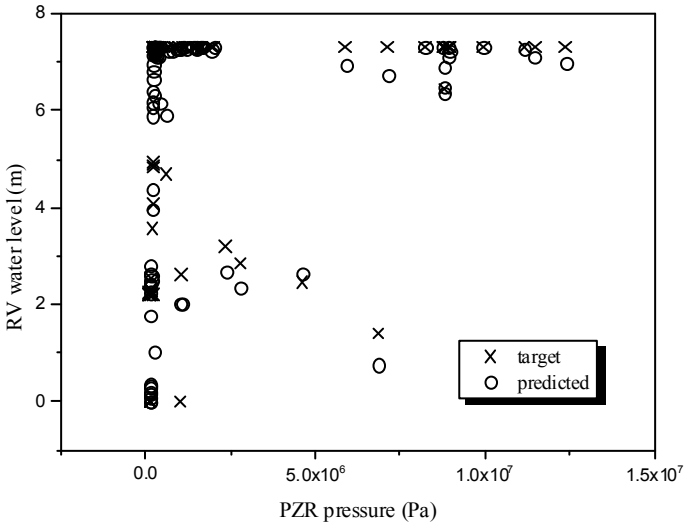


(d) reactor vessel water level versus elapsed time

Fig. 9. Prediction performance of the FNN model in cold-leg LOCA(continued)



(e) reactor vessel water level versus break size



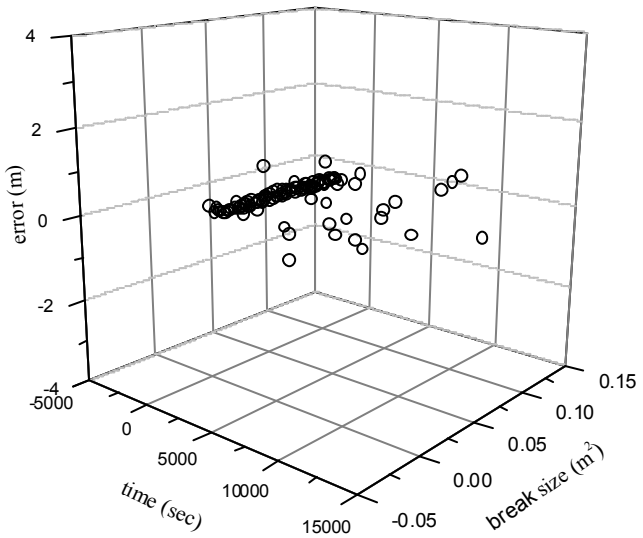
(f) reactor vessel water level versus PZR pressure

Fig. 9. Prediction performance of the FNN model in cold-leg LOCA(continued)

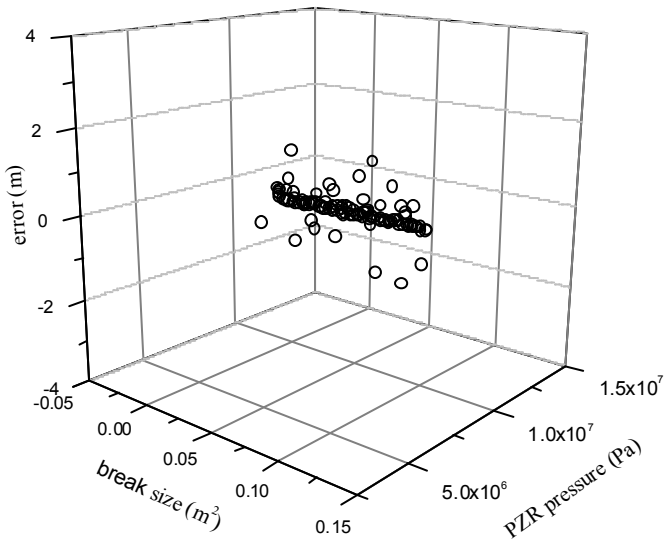
TABLE 2. Performance of the FNN model in cold-leg LOCA

Break position	Training data		Test data	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Cold-leg	3.4524	0.2808	2.0138	0.4468

As shown in Fig. 10 and Table 3, the prediction errors of the test data for the SGTR are inside the 1.31m error band and their RMS error is 0.33m.

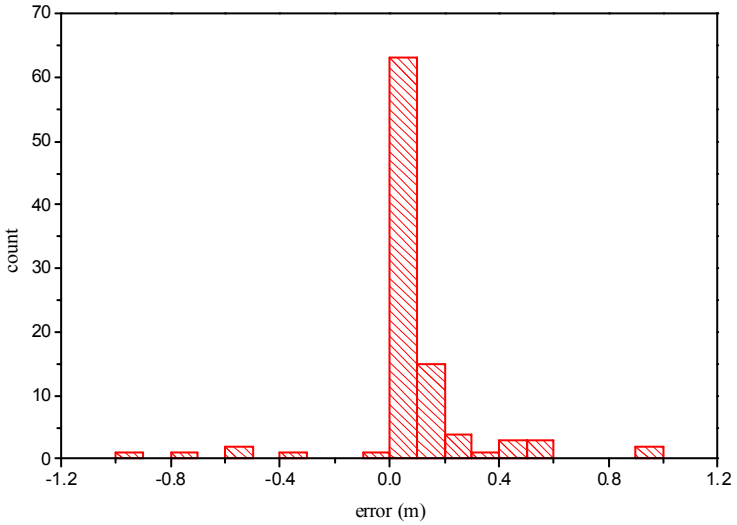


(a) reactor vessel water level error versus time and break size

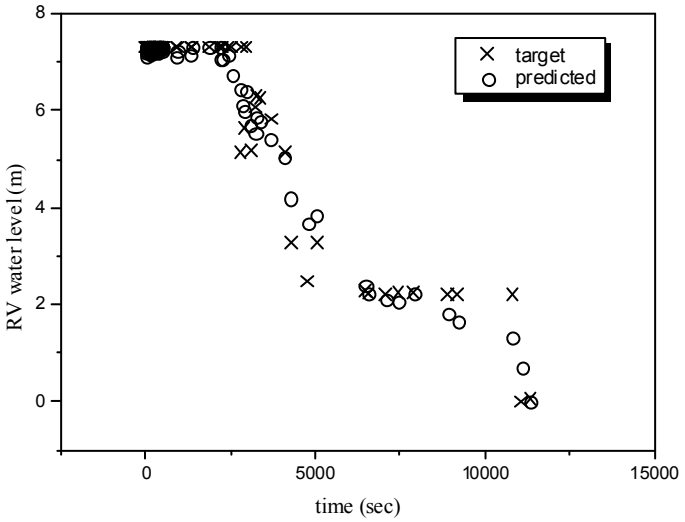


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 10. Prediction performance of the FNN model in SGTR

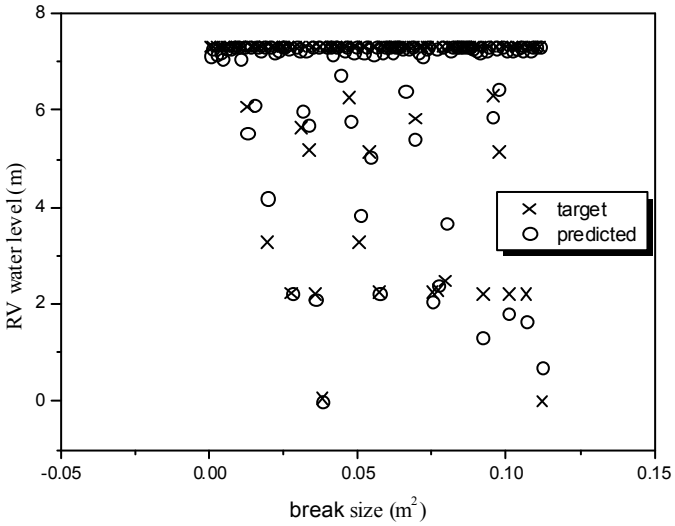


(c) Error histogram

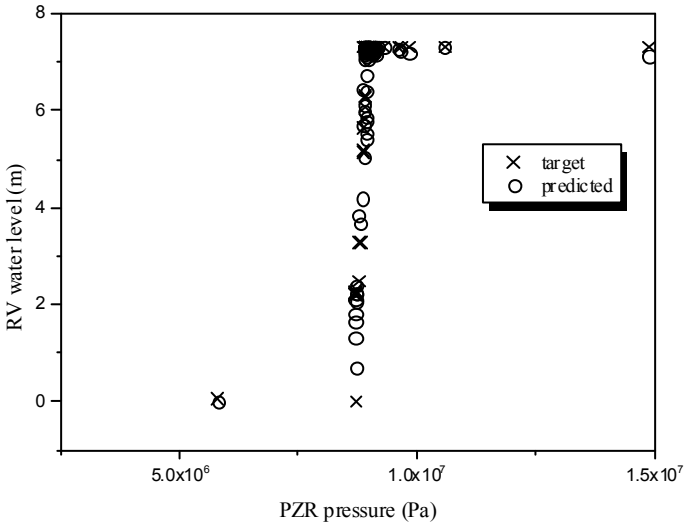


(d) reactor vessel water level versus elapsed time

Fig. 10. Prediction performance of the FNN model in SGTR(continued)



(e) reactor vessel water level versus break size



(f) reactor vessel water level versus PZR pressure

Fig. 10. Prediction performance of the FNN model in SGTR(continued)

TABLE 3. Performance of the FNN model in SGTR

Break position	Training data		Test data	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
SGTR	1.9912	0.2346	1.3081	0.3256

Table 4 summarizes the prediction performance results of the proposed FNN model. This table shows that the RMS errors for the training data are approximately 0.26m, 0.28m and 0.23m for the hot-leg LOCA, cold-leg LOCA, and SGTR, respectively. The RMS errors for the test data are approximately 0.34m, 0.45 and 0.33m for hot-leg LOCA, cold-leg LOCA, and SGTR, respectively. Even if the prediction error is increased a little, the proposed FNN model accurately predicts the reactor water level that is the range of 7.33m (hot-leg height). Sometimes, although the large errors are shown, the RMS error is approximately 0.37m for the test data.

TABLE 4. Performance of the FNN model

Break position	Training data		Test data	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Hot-leg	4.9418	0.2649	1.9270	0.3447
Cold-leg	3.4524	0.2808	2.0138	0.4468
SGTR	1.9912	0.2346	1.3081	0.3256

Until now, it was assumed that the input data have no instrument error. Therefore, the FNN model was tested using the input data with a random error to check the effect of instrument error. The errors were assumed to be inside the 3% band or 5% band. Table 5 shows the effect of the instrument errors.

As shown in Table 5, the FNN models have a little larger error than the case without error. The RMS errors are 0.40m and 0.43m for 3% error band and 5% error band, respectively. The prediction performance was not degraded much due to the measurement uncertainty. We could predict the reactor vessel water level with approximately RMS error of 0.4m even if the input signals have measurement uncertainty.

TABLE 5. Effect of instrument error

Break position	Test data					
	Without input error		With input error (3% error band)		With input error (5% error band)	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Hot-leg	1.9270	0.3447	2.0773	0.4119	2.5948	0.4288
Cold-leg	2.0138	0.4468	2.0266	0.4421	2.0474	0.4776
SGTR	1.3081	0.3256	1.2630	0.3354	1.5735	0.3742

B. GMDH model

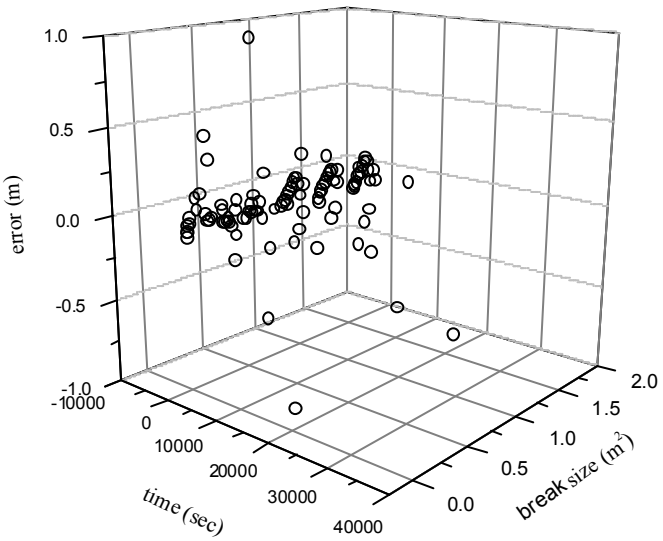
In order to predict the reactor vessel water level, the input signals are selected by considering the correlation between the measured or predicted input signals and the reactor vessel water level. Input signals of the GMDH model were divided into two cases. Input signals of each case are shown in Table 6.

TABLE 6. Input signals of each case

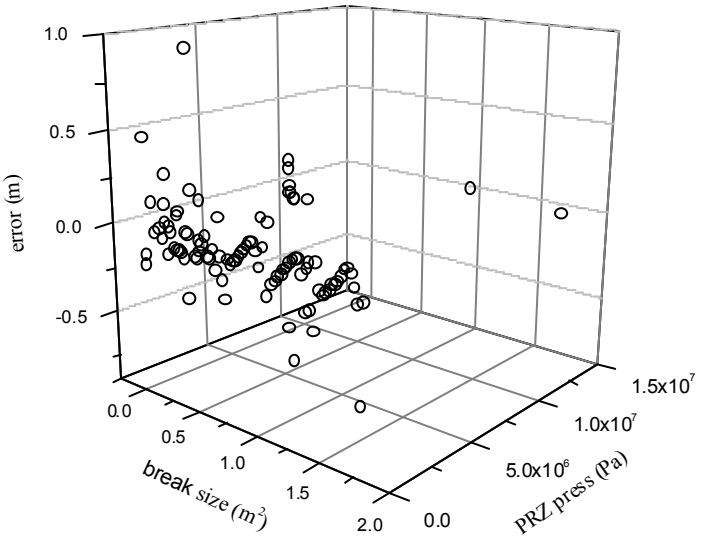
	Input signals
Case Ⅰ	time, predicted break size, PRZ pressure
Case Ⅱ	time, predicted break size, sump water level

Fig. 11-13 show the predicted reactor vessel water levels and their errors for the test data in the hot-leg LOCA, cold-leg LOCA, and SGTR situations, respectively. The test data is different from the data used to develop the GMDH model and consists of elapsed time, predicted break size, pressurizer pressure, sump water level and reactor vessel water level. It was assumed that the input signals have no instrument error. Therefore, the GMDH model was tested using the input data with a random error to check the effect of instrument error.

Table 7. and Fig. 11-13 show the predicted reactor vessel water levels and their errors using case Ⅰ.

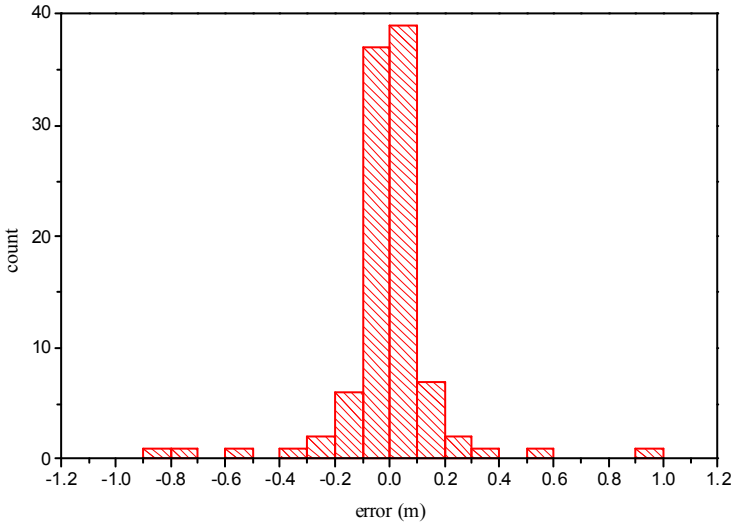


(a) reactor vessel water level error versus time and break size

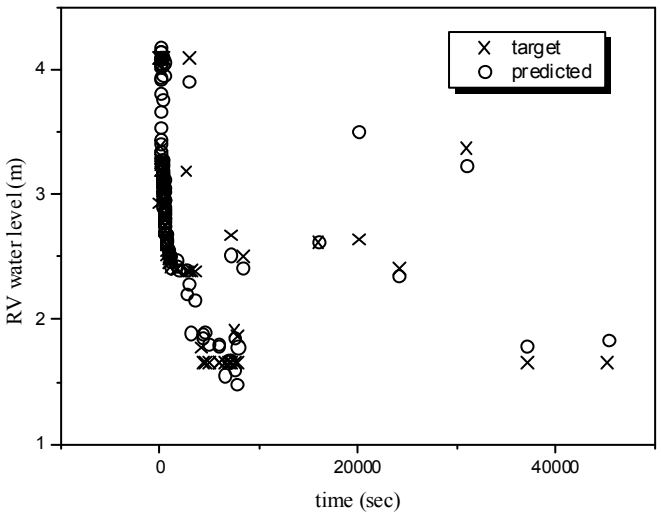


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 11. Prediction performance of the GMDH model in hot-leg LOCA using case Ⅰ

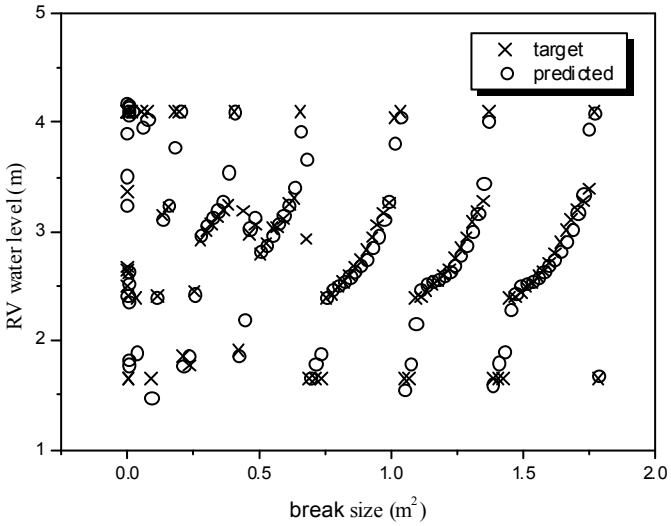


(c) Error histogram

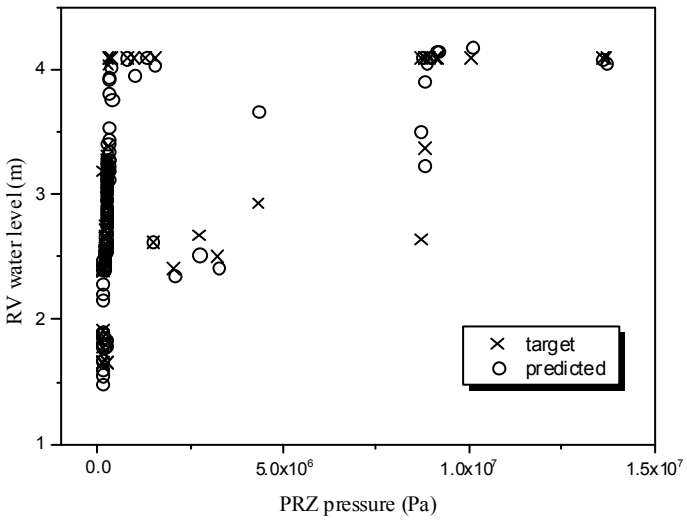


(d) reactor vessel water level versus elapsed time

Fig. 11. Prediction performance of the GMDH model in hot-leg LOCA using case Ⅰ (continued)

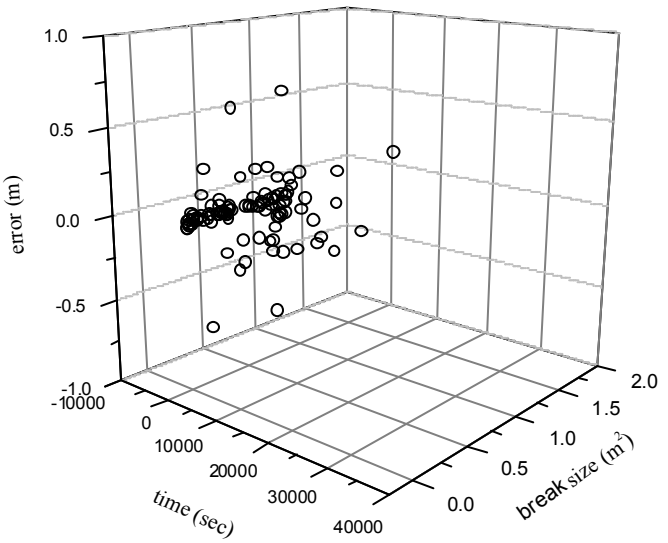


(e) reactor vessel water level versus break size

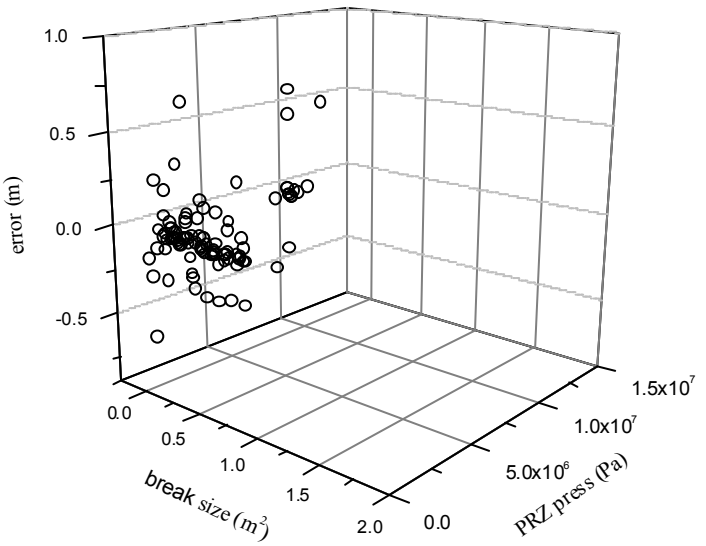


(f) reactor vessel water level versus PZR pressure

Fig. 11. Prediction performance of the GMDH model in hot-leg LOCA using case Ⅰ (continued)

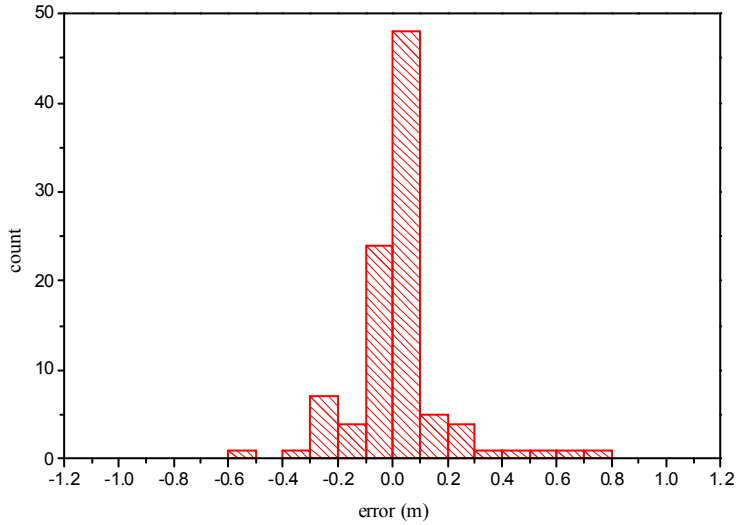


(a) reactor vessel water level error versus time and break size

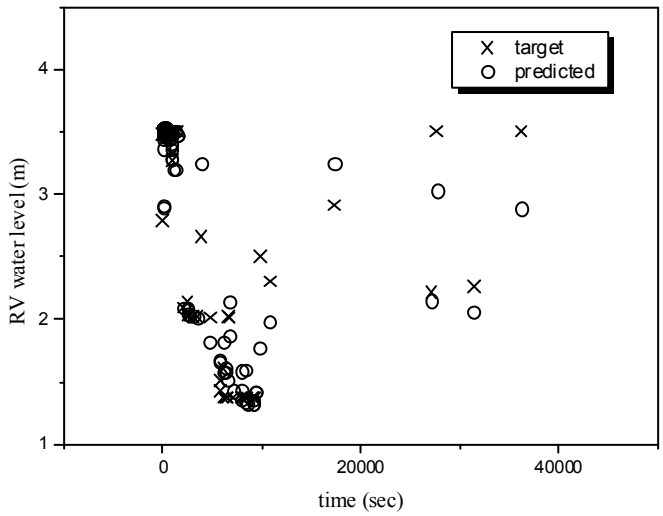


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 12. Prediction performance of the GMDH model in cold-leg LOCA using case Ⅰ

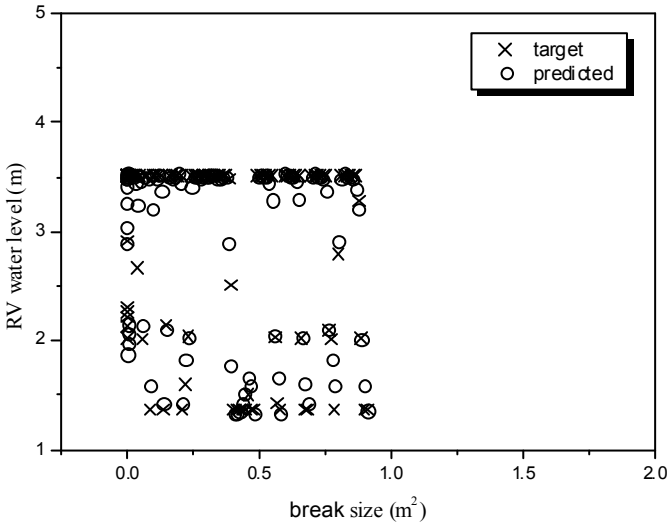


(c) Error histogram

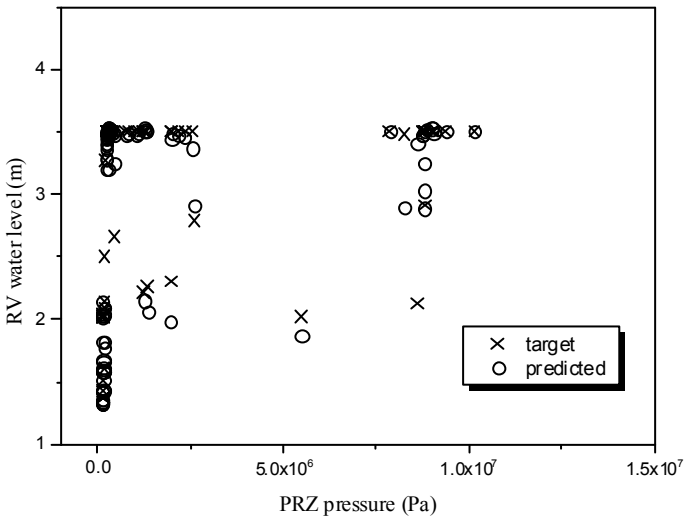


(d) reactor vessel water level versus elapsed time

Fig. 12. Prediction performance of the GMDH model in cold-leg LOCA using case Ⅰ(continued)

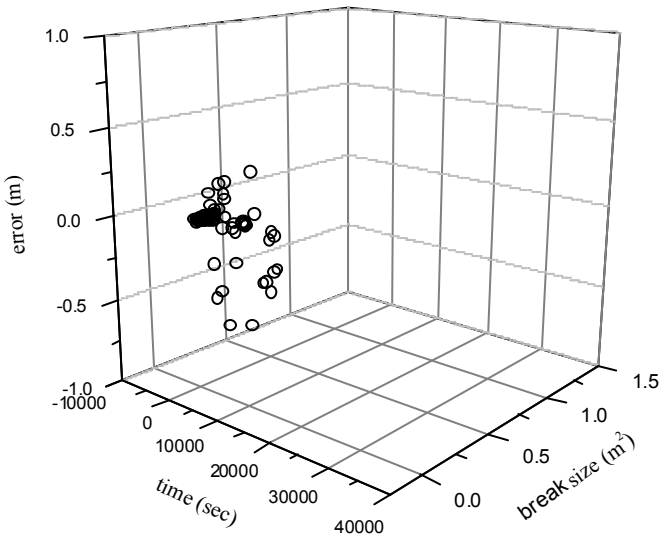


(e) reactor vessel water level versus break size

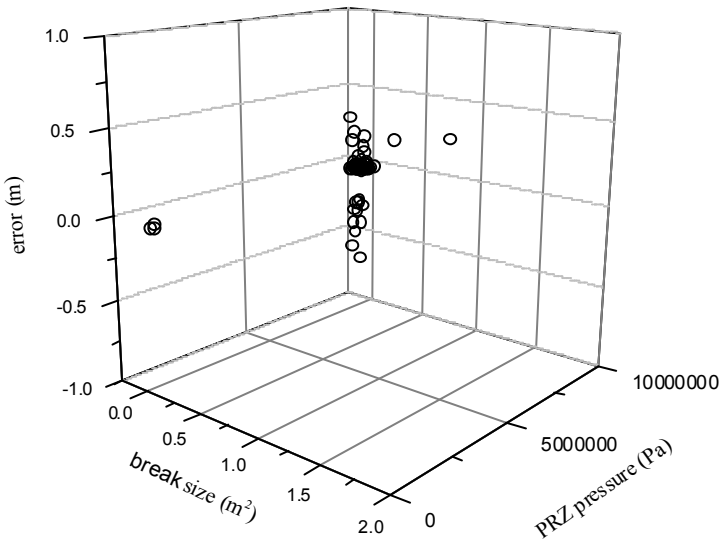


(f) reactor vessel water level versus PZR pressure

Fig. 12. Prediction performance of the GMDH model in cold-leg LOCA using case Ⅰ(continued)

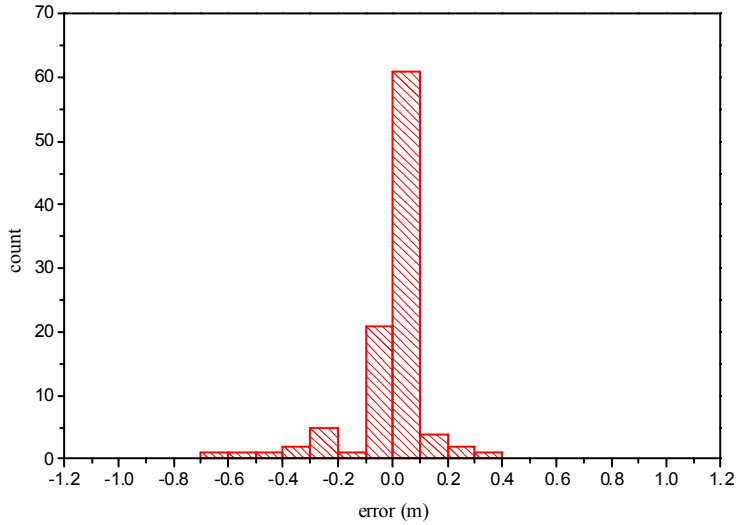


(a) reactor vessel water level error versus time and break size

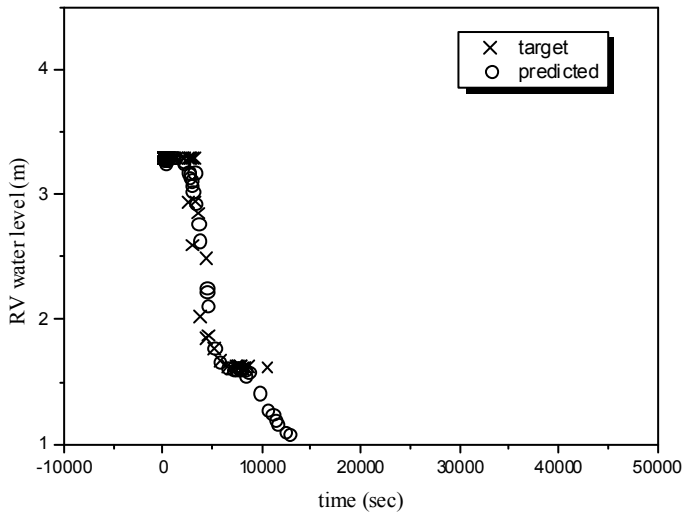


(b) reactor vessel water level error versus break size and PZR pressure

Fig. 13. Prediction performance of the GMDH model in SGTR using case Ⅰ

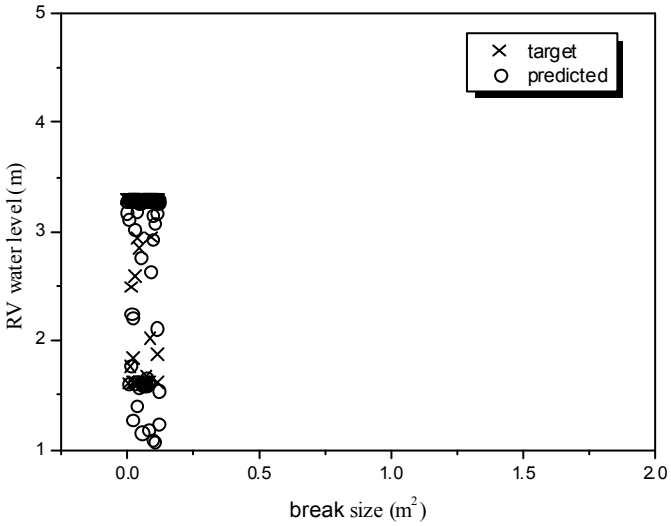


(c) Error histogram

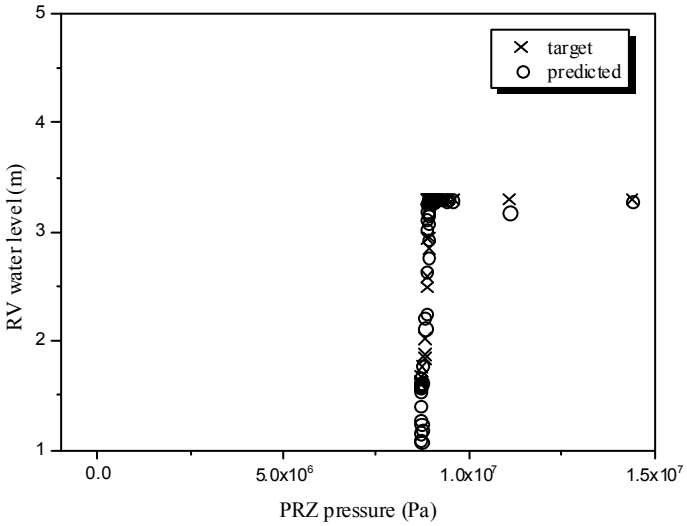


(d) reactor vessel water level versus elapsed time

Fig. 13. Prediction performance of the GMDH model in SGTR using case Ⅰ (continued)



(e) reactor vessel water level versus break size



(f) reactor vessel water level versus PZR pressure

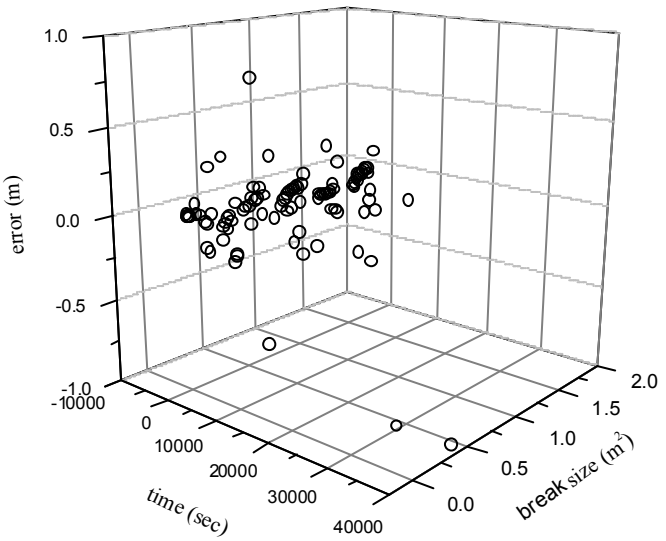
Fig. 13. Prediction performance of the GMDH model in SGTR using case Ⅰ (continued)

TABLE 7. Prediction performance of the GMDH model(case Ⅰ)

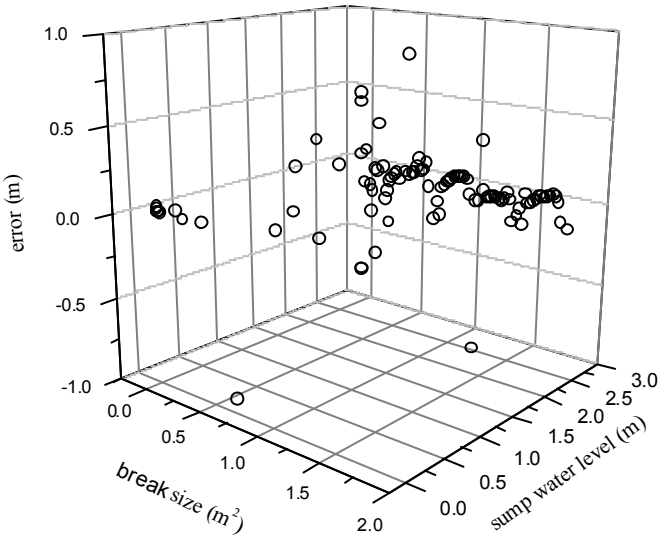
Break position	Training data		Test data (without input error)		Test data (with 5% input error)	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Hot-leg LOCA	1.65	0.17	0.99	0.19	1.02	0.20
Cold-leg LOCA	1.72	0.16	1.29	0.21	1.32	0.22
SGTR	0.83	0.10	0.61	0.13	0.71	0.13

As shown in Table 7, the GMDH model(using case Ⅰ) have a small error than the FNN model. The RMS errors are approximately 0.17m without input error and 0.18m with 5% input error, respectively.

Table 8. and Fig. 14-16 show the predicted reactor vessel water levels and their errors using case Ⅱ.

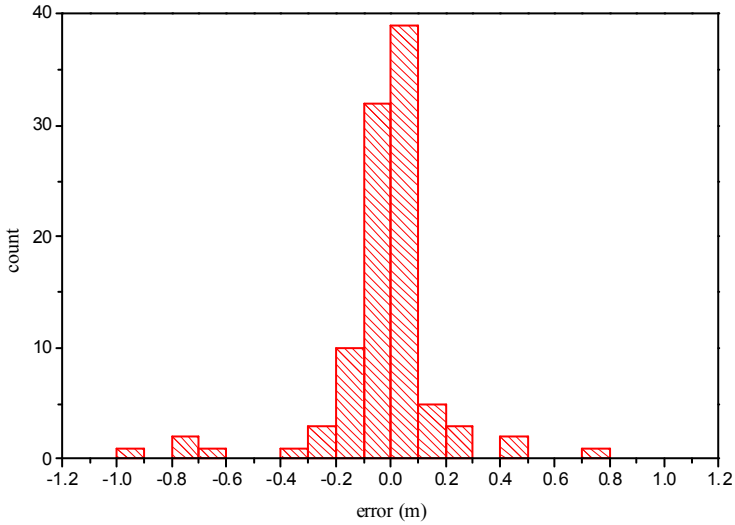


(a) reactor vessel water level error versus time and break size

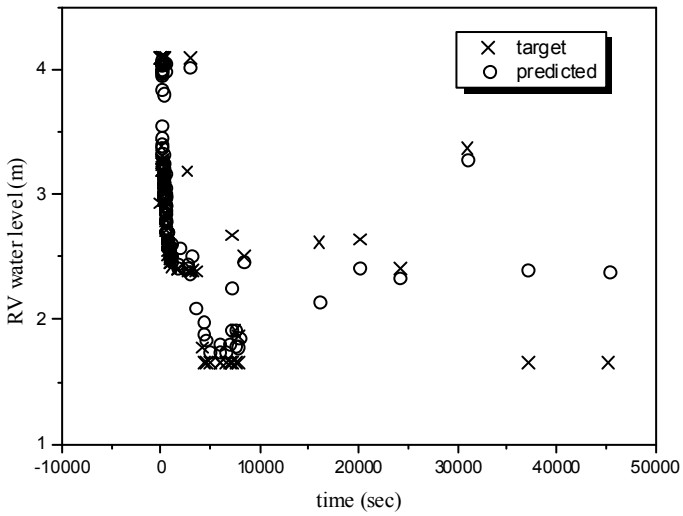


(b) reactor vessel water level error versus break size and sump water level

Fig. 14. Prediction performance of the GMDH model in hot-leg using case Ⅱ

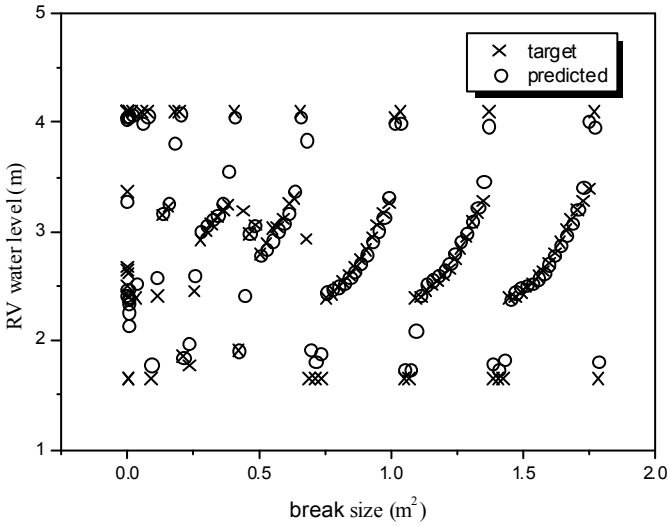


(c) Error histogram

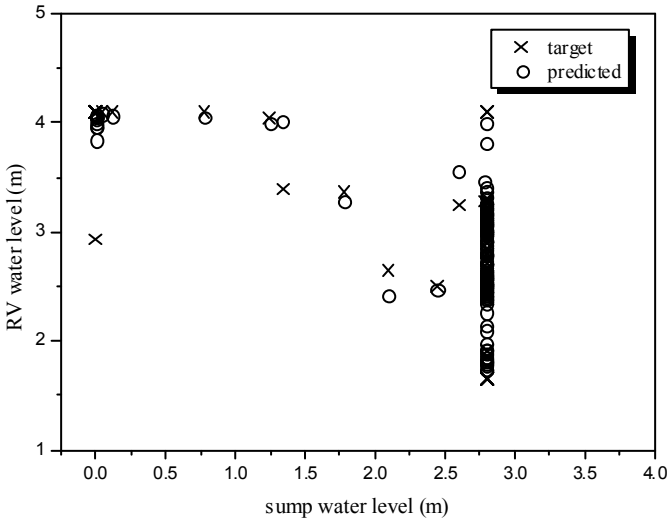


(d) reactor vessel water level versus elapsed time

Fig. 14. Prediction performance of the GMDH model in hot-leg using case Ⅱ(continued)

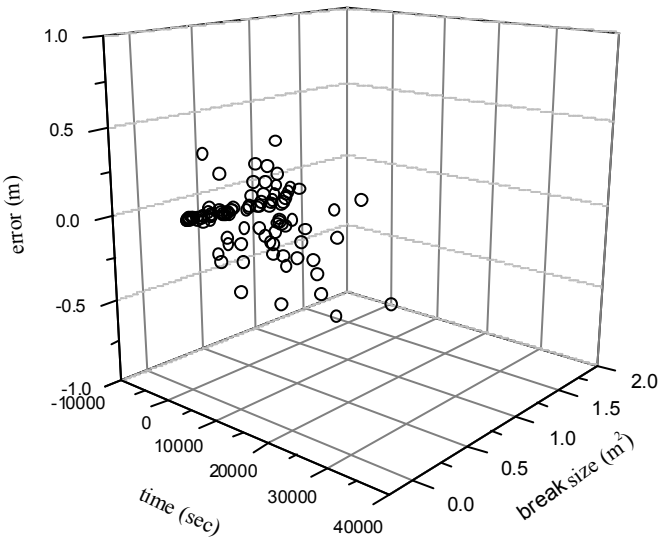


(e) reactor vessel water level versus break size

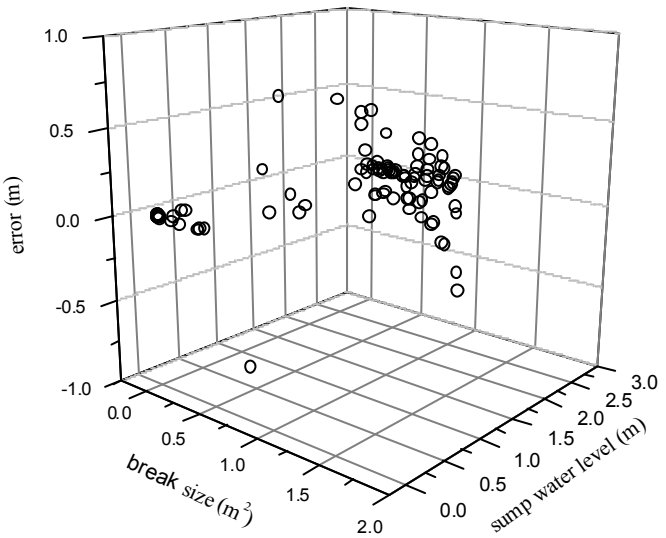


(f) reactor vessel water level versus sump water level

Fig. 14. Prediction performance of the GMDH model in hot-leg using case Ⅱ(continued)

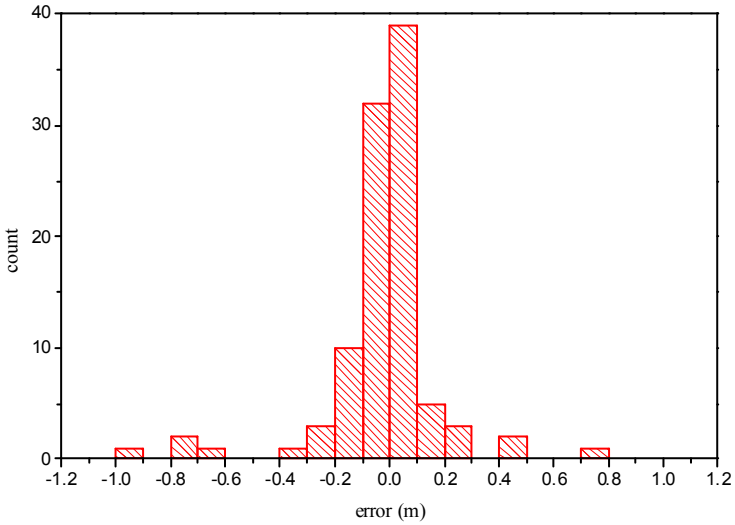


(a) reactor vessel water level error versus time and break size

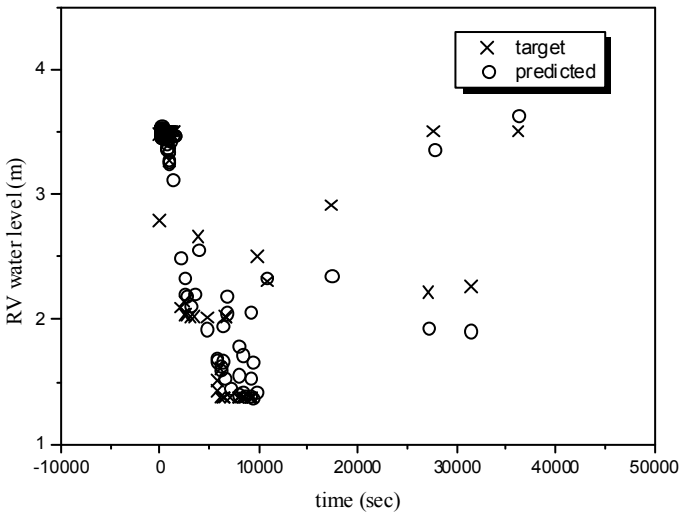


(b) reactor vessel water level error versus break size and sump water level

Fig. 15. Prediction performance of the GMDH model in cold-leg using case Ⅱ

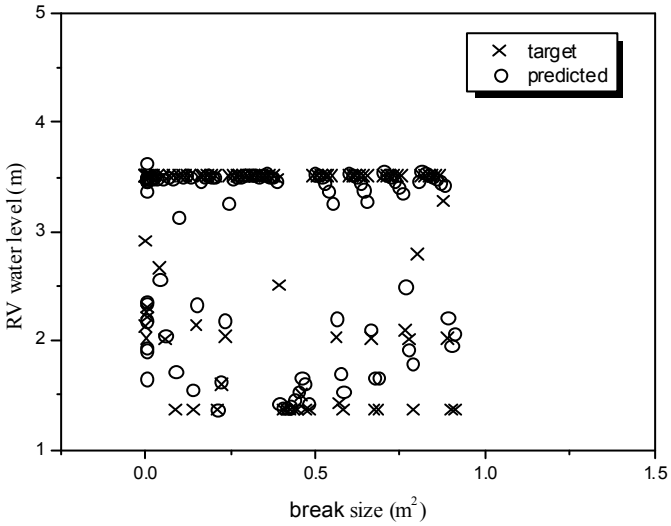


(c) Error histogram

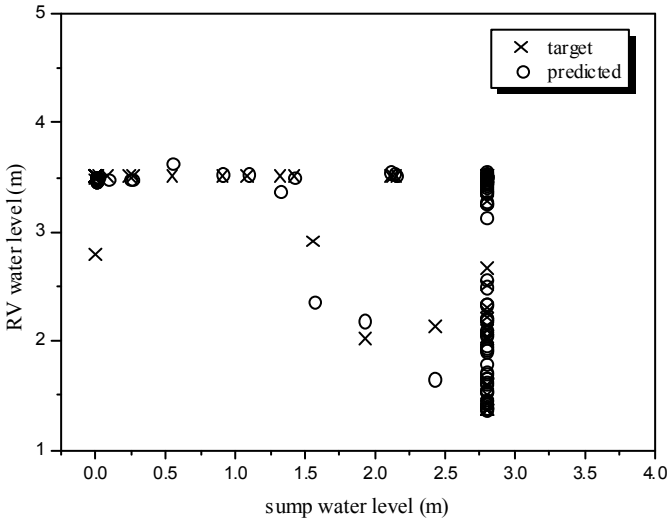


(d) reactor vessel water level versus elapsed time

Fig. 15. Prediction performance of the GMDH model in cold-leg using case Ⅱ(continued)

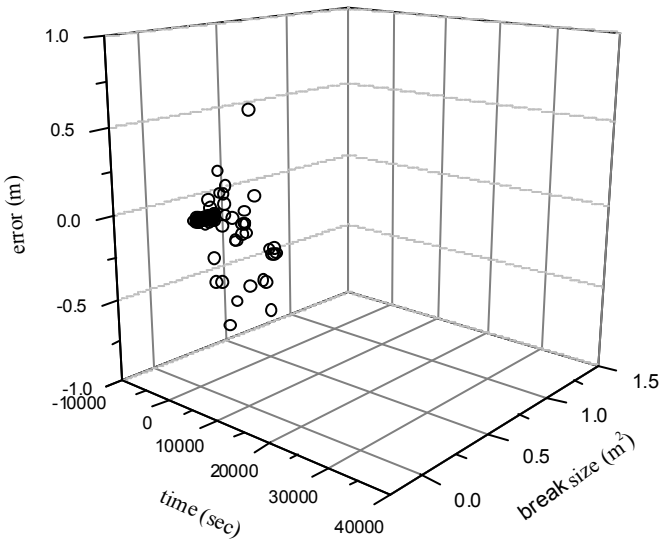


(e) reactor vessel water level versus break size

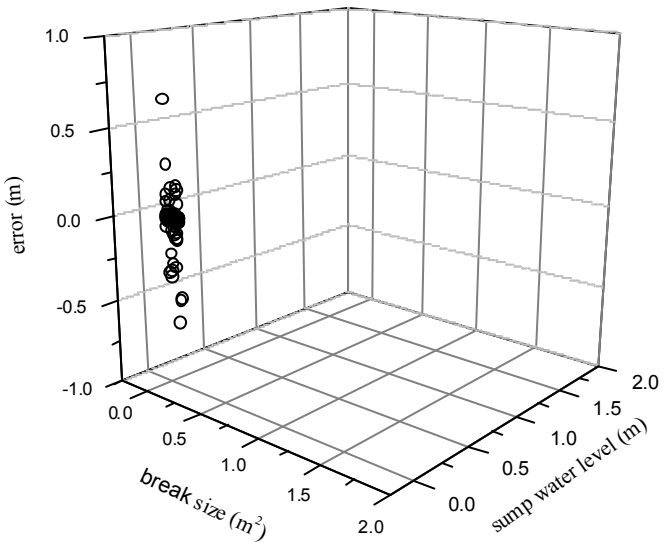


(f) reactor vessel water level versus sump water level

Fig. 15. Prediction performance of the GMDH model in cold-leg using case Ⅱ(continued)

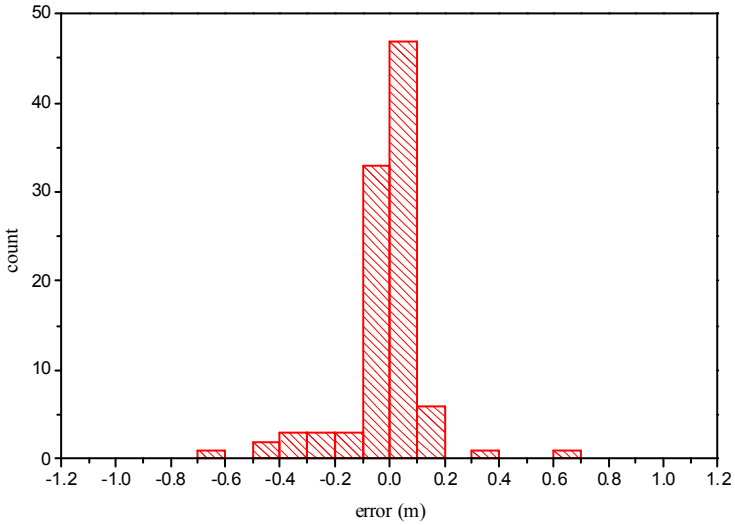


(a) reactor vessel water level error versus time and break size

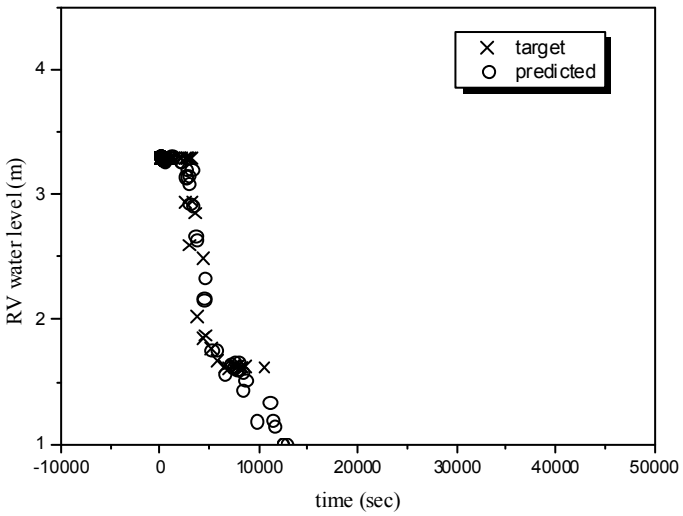


(b) reactor vessel water level error versus break size and sump water level

Fig. 16. Prediction performance of the GMDH model in SGTR using case Ⅱ

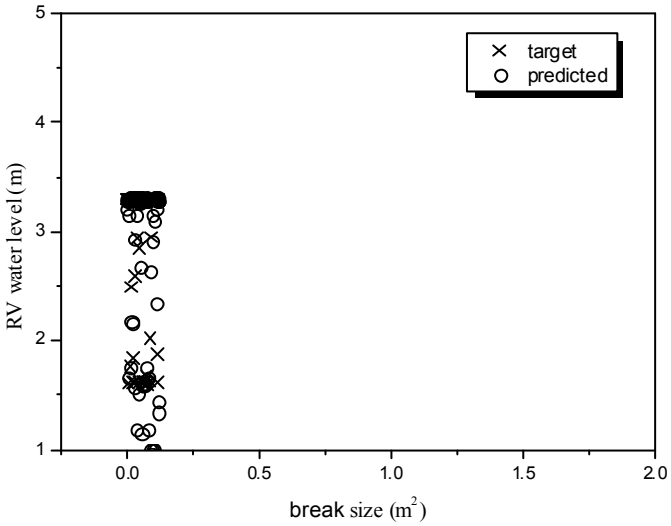


(c) Error histogram

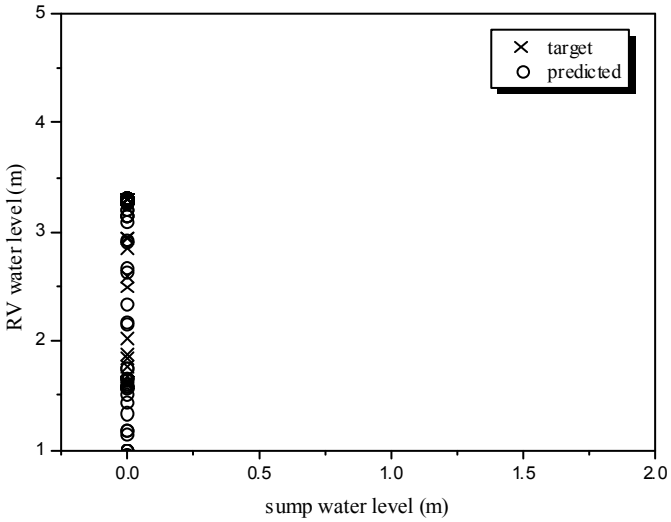


(d) reactor vessel water level versus elapsed time

Fig. 16. Prediction performance of the GMDH model in SGTR using case Ⅱ(continued)



(e) reactor vessel water level versus break size



(f) reactor vessel water level versus sump water level

Fig. 16. Prediction performance of the GMDH model in SGTR using case Ⅱ(continued)

TABLE 8. Prediction performance of the GMDH model(case Ⅱ)

Break position	Training data		Test data (without input error)		Test data (with 5% input error)	
	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)	Maximum error(m)	RMS error(m)
Hot-leg LOCA	1.72	0.18	0.91	0.21	1.47	0.33
Cold-leg LOCA	1.32	0.18	1.09	0.22	1.06	0.22
SGTR	0.91	0.09	0.67	0.14	0.67	0.14

As shown in Table 8, the GMDH model(using case Ⅱ) have a small error than the FNN model. The RMS errors are approximately 0.15m without input error and 0.23m with 5% input error, respectively.

The prediction performance was not degraded much due to the measurement uncertainty. We could predict the reactor vessel water level with RMS error of approximately 0.17m(using case Ⅰ) and 0.23m(using case Ⅱ) even if the input signals have measurement uncertainty.

Ⅴ. Conclusions

In this study, the reactor vessel water level prediction model was developed in severe accident circumstances by using FNN and GMDH method. The training data was selected from all the acquired data using an SC method to train the proposed FNN model with more informative data. The developed FNN and GMDH models predicted the reactor vessel water level using some of the measured or predicted signals except for the reactor vessel water level. The developed model was verified based on the simulation data of OPR1000 using MAAP4 code.

The simulations showed that the performance of the developed FNN and GMDH models were quite satisfactory but a few large errors were observed occasionally. On the other hand, it will be possible to predict the reactor vessel water level precisely if the developed FNN model can be optimized using a variety of data. Also, the prediction result of the GMDH model is slightly superior to the FNN model.

The developed prediction model will be helpful for providing effective information for operators in severe accident situations.

References

[1] J. W. Hines, D. J. Wrest, and R. E. Uhrig, “Signal validation using an adaptive neural fuzzy inference system,” Nucl. Technol., vol. 119, no. 2, pp. 181-193, Aug. 1997.

[2] M. G. Na, “A neuro-fuzzy inference system for sensor failure detection using wavelet denoising, PCA and SPRT,” J. Korean Nucl. Soc., vol. 33, no. 5, pp. 483-497, Oct. 2001.

[3] J. Garvey, D. Garvey, R. Seibert, and J. W. Hines, “Validation of on-line monitoring techniques to nuclear plant data,” Nucl. Eng. Technol., vol. 39, no. 2, pp. 149-158, Apr. 2007.

[4] E. B. Bartlett and R. E. Uhrig, “Nuclear power plant diagnostics using an artificial neural network,” Nucl. Technol., vol. 97, pp. 272-281, March 1992.

[5] M. Marseguerra and E. Zio, “Fault diagnosis via neural networks: The Boltzmann machine,” Nucl. Sci. Eng., vol. 117, no. 3, pp. 194-200, July 1994.

[6] Y. G. No, J. H. Kim, M. G. Na, D. H. Lim, and K. I. Ahn, “Monitoring Severe Accidents Using AI Techniques,” Nucl. Eng. Technol., vol. 44, no. 4, pp. 393-404, May 2012.

[7] M. G. Na, S. H. Shin, S. M. Lee, D. W. Jung, S. P. Kim, J. H. Jeong, and B. C. Lee, “Prediction of major transient scenarios for severe accidents of nuclear power plants,” IEEE Trans. Nucl. Sci., vol. 51, no. 2, pp. 313-321, April 2004.

[8] S. W. Cheon and S. H. Chang, “Application of neural networks to a connectionist expert system for transient identification in nuclear power plants,” Nucl. Technol., vol. 102, no. 2, pp. 177-191, May 1993.

[9] Y. Bartal, J. Lin, and R. E. Uhrig, “Nuclear power plant transient diagnostics using artificial neural networks that allow “don't-know” classifications,” Nucl. Technol., vol. 110, no. 3, pp. 436-449, June 1995.

[10] S. H. Lee, Y. G. No, M. G. Na, K. I. Ahn, and S. Y. Park, “Diagnostics of Loss of Coolant Accidents Using SVC and GMDH Models,” IEEE Trans. Nucl. Sci., vol. 58, no. 1, pp. 267-276, Feb. 2011.

[11] M. G. Na, W. S. Park, and D. H. Lim, “Detection and Diagnostics of Loss of Coolant Accidents Using Support Vector Machines,” IEEE Trans. Nucl. Sci., vol. 55, no. 1, pp. 628-636, Feb. 2008.

[12] M. G. Na, S. H. Shin, D. W. Jung, S. P. Kim, J. H. Jeong, and B. C. Lee, “Estimation of Break Location and Size for Loss of Coolant Accidents Using Neural Networks,” Nucl. Eng. Des., vol. 232, no. 3, pp. 289-300, Aug. 2004.

[13] R. E. Henry, et al., “MAAP4 - Modular Accident Analysis Program for LWR Power Plants, User's Manual,” Fauske and Associates, Inc., vol. 1, 2, 3, and 4, 1990.

[14] E. H. Mamdani and S. Assilian, “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller,” Int. J. Man-Machine Studies, vol. 7, pp. 1-13, 1975.

[15] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and Its Applications to Modeling and Control,” IEEE Trans. Systems, Man, Cybern., vol. SMC-1, no. 1, pp. 116-132, Jan./Feb. 1985.

[16] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” Addison Wesley, Reading, 1989.

[17] M. Mitchell, “An Introduction to Genetic Algorithms,” MIT Press, Cambridge, Massachusetts, 1996.

[18] S. L. Chiu, “Fuzzy model identification based on cluster estimation,” J. Intell. Fuzzy Systems., vol. 2, pp. 267-278, 1994.

감사의 글

시작이 쉽지는 않았지만 ‘간절히 바라고 바라면 이루어질 수 있다’는 마음가짐으로 정신없이 지나온 NICL 생활에 벌써 마침표를 찍어야 할 시간이 왔다는 게 믿기지가 않습니다. 그동안 잊고 지냈던 2010년 어느 날 처음 실험실에 들어왔을 때가 문득 떠오르는 이유는 지난 시간에 대한 아쉬움과 다가올 미래에 대한 설렘 때문이 아닐까 싶습니다. 지금은 집보다 더 편해져 버린 NICL에서 지난날의 추억을 떠올리며 졸업논문의 마지막 장을 작성하고 있는 이 시간이 있을 수 있게 해준 모든 분에게 감사의 마음을 전하고자 합니다.

먼저 지금의 저를 있게 해주신 나만균 교수님께 한없는 감사를 드립니다. 많이 부족하고 실수투성이인 저에게 따뜻한 관심과 세심한 배려로 감싸주시고, 어느 학생과 비교하여도 손색이 없는 수많은 경험과 지식, 그리고 사람됨을 비롯한 모든 부분에서의 가르침을 영원히 간직하고, 앞으로 무슨 일이 있더라도 항상 초심을 잃지 않도록 노력하겠습니다. 또한, 제자들이 잘되기만을 바라시며 하나하나 챙겨주시는 모습과 흐트러짐 없이 학문연구에 매진하시는 모습을 보며 진정한 학자의 모습을 알게 해 주셨습니다. 학문과 더불어 인생에 대한 가르침과 사소한 질문에 대해서도 다시 한 번 생각해 볼 수 있게 방향을 제시해주셨던 김숭평 교수님, 학생들에 대한 애정이 깊으시고 항상 학생들의 진로에 대해 관심을 가지고 조언을 해주셨던 정운관 교수님, 전공뿐만 아니라 다른 분야에 대해 넓은 시야를 가질 수 있게 많은 것들을 가르쳐 주신 이경진 교수님, 교수님이라는 높은 벽을 허물며 학생들이 다가갈 수 있게 만들어 주신 송종순 교수님, 각각의 학생들에게 관심을 보여주시고 자신감을 북돋아 주실 뿐만 아니라 변함없이 연구에 임하시는 김진원 교수님께 진심으로 머리 숙여 감사드립니다. 그리고 원자력발전소 현장에 대한 실무적인 강의를 해주시며 많은 격려를 해주신 박부성 교수님, 바쁘신 와중에도 학생들을 위해 강의를 해주셨던 신원기 교수님, 이기복 교수님, 사회로 나갈 준비를 하는 저에게 인성과 더불어 학생이 아닌 사회 구성원이 지녀야 할 자세와 마음가짐에 대해 많은 조언과 격려를 해주신 박병주 교수님께도 진심으로 감사의 말씀을 전합니다.

처음 저를 NICL에 들어와서 함께 했던 성한이형, 동수형, 영규형, 그리고 동기이자

선배인 심원, 많이 웃기도 하고 많이 혼나기도 했지만 지금 생각해 보면 같이 생활했던 그때가 정말 행복했습니다. 하나하나 표현할 수 없지만, 형들과 심원이의 가르침과 배려가 지금의 저를 있게 해준 원동력이라고 생각합니다. 형들이 하나둘씩 졸업할 때마다 그 허전함은 말로 표현할 수가 없었고, 마지막 인사 때도 말하지 못했지만, 지금이라도 감사하다는 말을 전하고 싶습니다. 그리고 동생이지만 선배이고 때로는 형처럼 항상 힘들 때마다 옆에 있어준 주현, 항상 조언과 부족한 부분에 대한 지원을 아끼지 않았던 재환이형, 다른 과에서 진학해서 힘들지만 항상 쾌활하게 생활하며 웃게 만들어 주는 쾌환, 같이 졸업하지는 못하지만 어떤 상황에서도 넓은 마음으로 이해해주었던 대섭이형에게도 정말 감사하다는 말을 전하고 싶습니다. 이렇게 좋은 사람들과 생활할 수 있었던 건 큰 행운이었습니다. 또한, 실험실을 꽉 차게 해준 후배 동영, 주현, 너희가 있어서 맘이 든든하다. 앞으로 교수님 말씀 잘 따르며 열심히 생활하고 꿈에 한 걸을 나아갈 수 있게 뜻깊은 시간을 보내길 바란다.

항상 찾아가면 반갑게 맞아주던 동위원소실험실의 유선이형, 정민이형, 선동, 인석, 열수력실험실의 용진이형, 핵주기 실험실의 강일이형, 상헌이형, 민영, 영국, 현민, 태빈, 기계재료실험실의 민수형, 사용, 미연, 성재, 그리고 항상 조언과 격려를 아끼지 않고 상담에 응해주신 상준이형, 금주형, 학부생활을 같이한 진현, 해성, 창수 모두 감사합니다. 이외에도 미처 언급하지 못한 선, 후배, 동기들에게도 고마운 마음을 전합니다.

항상 친구라는 이름으로 함께 울고 웃으면 10년이 넘는 시간을 함께해온 두선이형, 기섭, 교범, 준현, 선교, 청기 등등 너무 고맙고 앞으로도 서로 의지하고 잘 지내보자.

마지막으로 지금껏 자신을 희생하시며 한없는 사랑으로 저희 형제를 지켜봐 주신 어머니께 진심으로 감사의 마음을 전합니다. 말로 표현할 순 없지만, 앞으로 살아가면서 자랑스러운 아들이 될 수 있게 노력하겠습니다. 그리고 말없이 버팀목이 되어주는 형에게도 감사의 마음을 전합니다.

지금까지 나의 보금자리였고 앞으로 후배들의 보금자리가 되어줄 NICL에서...

2013년 12월의 어느 날

박순호